Benjy Katz

Week 4

Sep 27, 2023

**Problem Framing:**

* 1. What business problem are you trying to solve?
  2. What metrics will you track?
  3. What is the deliverable to the recruiting team/customer?
  4. How will you measure success?
  5. Think about and discuss what trade-offs you might make (and why) if you optimize for precision. How about recall? Think about it from the stakeholder (i.e. business unit) perspective. Hint: what if you only need one or two candidates and don’t want to waste time talking to a lot of people? Or if you want a lot of candidates and are okay with some people who decide they don’t want to leave their job?
  6. What else can you think of that’s relevant to framing the problem?

1. We want to write a program to assist recruiters in optimizing their time and only reach out to candidates that are likely to take the job. It is a waste of time for a recruiter for a fast food franchise to reach out to high level executives to try and convince them to leave a lucrative job to become a fry cook.
2. Metrics such as precision, recall and accuracy will be important metrics to track. Precision is important because with a high precision, the recruiter is not wasting time. High precision means that many of the recruiters' calls are turning into employees. Recall is also important because it is important to not leave talent on the table. Recall is how much of the total possible hirable candidates were reached out to. Accuracy is not as important but it is a good metric to include for verbosity.
3. The deliverable to the team will be a python python shell script that recruiters can feed a csv file with a few potential candidates and the program will evaluate each candidate and return a score of how hirable the candidate is.
4. It is impossible to measure recall because the team will not pursue all the candidates with the low score to determine a false negative so only precision can be measured once the system is used.
5. Much of these tradeoffs were discussed in section b but this all depends on business needs. In the event that the team needs to hire a lot of people a high Recall is important to not leave talent on the table, if only a small number of candidates need to be hired and the recruiters are very busy, it is important to optimize for precision.
6. It is important to recognize that this does not measure the qualifications of the candidate and how good they will be at their job. This just tracks hireability. It seems likely that these two quantities are inversely related. If one offers a very good job to a candidate that does not have many qualifications, he is very likely to take the job but not likely to do the job well.

**Part one:**

<https://colab.research.google.com/drive/1OSUn5IDrqAzDTUcVfhUGaxyMyvHd77CP?usp=sharing>

**Model comparison**

My first model is a simple heuristic based on data exploration. A huge factor in hireability is how good of a city they live in. The worse the city, the more hirable. Another major factor is experience, the less experience, the more hireable. The model assigns points to a candidate using this basic heuristic.

We will extremly likely to hear back from people who live in low developed cities.

We are fairly likely to hear back from someone with no relevant experience

We are very likely to hear back from someone in a full time courses

We are likely to hear from a graduate

We are likely to hear from stem

This model works very well with the following metrics:

Accuracy: 0.77

Precision: 0.52

Recall: 0.50

The second model is a DNN which learns based on all the metrics and has the following metrics:

Accuracy: 0.76

Precision: 0.5

Recall: 0.01

To demonstrate this difference we can look at candidate 29452 who seems like a very hirable candidate, but the DNN still rejected him under the assumption that ‘most people are not hirable’.

Because the model was optimizing for accuracy it essentially just told the system to hire no one at all. This is obviously not a good model and more can be done to develop a better DNN to have better precision and recall but I ultimately decided to opt with my heuristic model.

**Part Two:**

<https://github.com/Yeshiva-University-CS/Katz_Benjy_800564547/tree/f137ca8a6740cce920e7c1b59828a14ed160c029/ML_applied/hiring%20rec>

**Threshold selection:**

When deciding on a hiring threshold for the model score I chose something that balances precision and recall. However, I left the score in for the recruiters to make there own determinations so they can simply sort by hireability score and work there way down the list

**Demo:**

<https://youtu.be/bD4Uynfm3tQ>