Machine Learning Design

Duration: 20 Minutes
Difficulty: Medium
Domains: Product

Problem

How would you estimate the salary of a job occupation on LinkedIn?

Solution

[Candidate] I'd like to first scope the problem as it does not follow the traditional regression or classification problem. Can I assume that I already have access to salary data from the platform?

[Interviewer] In this case, we are designing the system from scratch. So, you have to figure out how to collect the data and frame it.

[Candidate] I see. I need to acquire the salary information somewhere. One possible source is the government economic data. I am sure that the Bureau of Labor Statistics (BLS) would contain such salary information that could be provided to the users.

[Interviewer] Okay, what if the occupation isn't in the list, how would you derive the estimation then?

[Candidate] I believe this is when a survey could be used. It's pretty typical for such information to be collected directly from users when it cannot be inferred indirectly. But, usually surveys are ignored by users. So, an incentive is required for them to response to surveys about their occupation and salary information.

[Interviewer] Gotcha. How would you provide the salary estimation of an occupation at the geography level?

[Candidate] I'd say that this can either be inferred based on the location of the company that the responder works for or a text field in the survey.

[Interviewer] Suppose that the data scientist salary is known in NYC, but not in London, how would you perform the estimation then?

[Candidate] Hmm… I'd say this is a situation when a regression model is required. Basically predict the unknown salary distribution given the known distributions. I could use features sourced from the regional economic data (i.e. cost of living, population, average salary), occupation data (i.e. salary estimations of the knowns), and company data (i.e. industry, sector, economic indicators). The input data can be trained in a regression model, either a linear model or random forest, to predict the unknown.

[Interviewer] Okay, how would you handle outliers in the survey response?

[Candidate] A simple approach to handling outliers should suffice. I'd use the IQR method to remove outliers that are greater than 1.5 or 2 times the 75th percentile. This should remove unusual salary estimation information such as $1M for an entry-level data scientist position.

[Interviewer] Okay great. So how would you now provide the estimates to the user?

[Candidate] I believe simple statistics such as the median, 10th percentile and 90th percentile should suffice.

> **Interviewer Solution**
>
> The problem seems straightforward at first until you try to flesh it out. In ML design problems, you need to scope what to include and exclude in your solution. Hence, in the beginning, before diving into the solution, it's vital to explore ideas and validate assumptions with the interviewer.
>
> First, scope the problem in terms of specification, data collection and potential issues. Let's talk about specification first.
>
> **Specification - Estimation**
>
> When the interviewer is asking you to "estimate" the salary of an occupation, what does he mean? Retrieve the mean, median or distribution? You have to scope this out with the interviewer. Usually when the task involves estimation, there's uncertainty involved so just returning the mean or median is not suffice. So, estimate the median, 10th percentile and 90th percentile of the salary range.
>
> **Specification - Granularity**
>
> When you go on salary estimation sites such as LinkedIn, PayScale, or Glassdoor, they contain salary estimation not just at the occupation level, but also geo, company, and years-of-experience (YOE) levels. You need to frame this upfront with the interviewer. Otherwise, there's a chance that the interviewer may ask: "Okay, so I see that you build the estimation model at the occupation level. Now, what if it's occupation x geo?"
>
> **Data Collection**
>
> This is the key step to define. Most candidates will just presume that the data is readily available. The reality in many data science problem is that the data science project is not like Kaggle when the cleaned data is already given to you. You have to design what the dataset looks like and how to acquire the labels.
>
> Let's start with one data source that may contain salary data. The Bureau of Labor Statistics contain salary information (median and percentiles) of thousands of occupations by geo. This is one way to leverage your information.
>
> However, here's one problem, the occupation list is not comprehensive. What if the you'd want the estimation at the company level? YOE? The BLS information, although could be utilized is not comprehensive. Hence, there has to be another way to extract the information. The primary data collection method would come from users on LinkedIn.

There's a incentive-based survey technique called "give-to-get." Incentivize users to respond to your survey with an offer. You could dispatch email surveys to ask for salary details. In exchange for the response, the user could receive some kind of reward. Perhaps, 1 month of free premium subscription or the aggregate salary information when it's ready. In fact, this incentive model is not to different from how PayScale and Glassdoor perform their salary estimation of occupations.

Potential Problems

Now, this is one area that differentiates a quality data scientist from the herd. Can you pre-emptively foresee issues and prepare strategies to mitigate them? Here are a couple list that could arise from this salary estimation problem:

1. Sparsity - The job cohort may have insufficient data given the lack of survey response.
2. Outlier - The presence of outlier may throw off the estimation.

Let's talk about the first one. Suppose you only have 10 data points per cohort (occupation x geo), do you have enough information to estimate the distribution of salary? You'd have to apply a rule that there has to be sufficient data-points for the estimation to kick-in, let's just say 100 data-points.

If there's sufficient data-points (let's say 100 data-points) in the cohort, return the salary estimation. Otherwise, wait for enough information or build a prediction model for the missing details. How?

Think about estimating the missing cohort of a salary information as a regression exercise. You could use the existing salary information of a cohort to estimate the unknown. You could employ various features (i.e. regional economic data, occupation data, company information and e.t.c.). You could evaluate your model using the BLS and/or cohort information you already have.

Now, let's talk about the outlier issue. Suppose the user provided a false entry, let's say $1 million / year on an entry-DS role. That salary is highly unlikely for an entry role. You could propose a simple technique such as the IQR method to remove outliers before applying estimations.

Interviewer Assessment

In the statistics section, a candidate is assessed based on correctness and soundness of statistical methodology, product sense and communication. For each dimension the candidate is rated in the following scale: (5) superior, (4) good, (3) adequate, (2) marginal, (1) not competent.

Assessments	Rating	Comments
ML Methodology	4	The candidate understands the framework of solving the salary estimation problem. He understands how to get the data to train a model (i.e. BLS and survey), how to incentivize users to get survey responses, build an estimation model and handle outliers. One area he should explore more is how he would handle sparsity in the survey responses or cold-start situations when a new job is created and there is no survey data.
Product Sense	5	The candidate understands which external data source to tap into for the salary estimation. He also understands that the nature of the product is largely driven by survey, and demonstrated that he understands that users want incentives to provide responses.
Communication	5	The structure of his conversion was well-framed. In the beginning, he started with questions and assumptions to clarify. This helps him gather information to provide responses. He didn't just jump into how he would estimate the salary using a model. He followed a chronological order in model development which involve, getting the data, preprocessing, estimating and reporting. In addition, his responses were clear and succinct.