Predicting Compensation for Job Seekers

Stanford University

CS 229 Final Project (Autumn 2016)
December 13, 2016

Darren Baker
Megan Fazio
Jen Kilpatrick
{drbaker, mefazio, jdki}
@stanford.edu

Problem Background

- Big question on the mind of most job seekers: **"how much can I expect to be paid?"**
- This question is surprisingly hard to answer!
  - Employers rarely share data about their compensation levels
  - Wages/salaries can vary widely across geographies, industries, employee experience levels, and more
- **Our goal:** based on some combination of input values describing a possible job, **can we produce a reliable prediction of base salary for that job?**

Data Overview

- We obtained a dataset with over 2 million user-reported salary datapoints from leading careers site Glassdoor
- Data is all from the United States over the years 2014 - 2016, across many industries, jobs, and employers
- Examples of key features:
  - Job title (e.g. "sales manager")
  - Employee attributes (experience, etc.)
  - Employer name and attributes (total employee count, industry, etc.)
  - Employer type (private, gov't, etc.)
  - Metro area
  - Proprietary Glassdoor categories

Modeling & Prediction Approaches

Data cleansing & feature engineering

Several of the most critical features in the Glassdoor data were categorical, with huge numbers of possible values; others were noisy due to UGC or data limitations. We took steps to address sparsity and noise by:

- **Normalizing** text & **filtering** out invalid salary values
- **Extracting** prefix ("senior"), suffix ("trainee"), and other role descriptors ("manager", "engineer") from job titles
- **Consolidating** small metro areas into single values
- **Bucketing** employee counts into discrete groupings

Basic linear regression

We began with a standard **squared-loss linear regression** with **6 different models** (reflecting progressively larger subsets of the original feature set).

Adding regularization & log salaries

Our data remained highly sparse and somewhat noisy, so we experimented with **3 types of regularization:** LASSO (L1), Ridge (L2), and Elastic Net (linear combination of L1 and L2).

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1$$

Elastic net minimization formula

Since the distribution of salaries is also **right-skewed**, we tried predicting the **log of salary values** to compensate.

Exploring alternate models

Finally, we experimented with **tree-based regression models** to explore whether they might improve accuracy, but ultimately these models proved **computationally infeasible** due to our huge number of features / values.

Results

- We achieved accuracy **close to Glassdoor's benchmark values** (using proprietary categories): **15-19% median test error**.
- **Regularization** delivered **small improvements**, but predicting **log values** gave a **bigger boost**.

	Original salary predictions			
	Regularization Type	Training MSE	Test MSE	Median % Error
Model 1 Best Results	Lasso (lambda = 1)	23854	24370	18.58
	Ridge (lambda = 1000)	23869	24380	18.58
	Elastic Net (lambda = 10)	23868	24366	18.57
Model 6 Best Results	Lasso (lambda = 1)	19391	22527	16.87
	Ridge (lambda = 1000)	19409	22575	16.90
	Elastic Net (lambda = 10)	19480	22388	16.73

	Log salary predictions			
	Regularization Type	Training MSE	Test MSE	Median % Error
Model 1 Best Results	Lasso (lambda = 1e-5)	24154	24804	17.48
	Ridge (lambda = 0)	24151	24806	17.49
	Elastic Net (lambda = 1e-4)	24174	24807	17.48
Model 6 Best Results	Lasso (lambda = 1e-4)	19610	22149	15.16
	Ridge (lambda = 0)	19325	22301	15.27
	Elastic Net (lambda = 1e-4)	19431	22190	15.14

Future Work

- **Inflation adjustment**:
  - We did not adjust salary values for inflation or wage growth over past years.
  - We expect that adjustments may improve accuracy by several percentage points.
- **Interaction terms:**
  - Our feature engineering does give up some "implicit" interaction when categorical variables are binarized.
  - We would experiment with re-adding explicit interaction terms to capture non-additive relationships.