Glassdoor

Laura Borton

Project Design:

Glassdoor is a website where employees and former employees anonymously review companies and their management. By scraping information off their website, I hoped to determine which features contributed most to a company's rating. These features were: Revenue, Size (number of employees), CEO Approval Percentage, Headquarters Location, Company Status (Private, Public, Government, etc.), Benefits Rating, and Year Founded. The overall company rating was also downloaded as it was used to train the model. The companies evaluated were those hiring data analysts in Seattle. This model could be helpful for both employers and potential employees. Employers can use the model to predict how their employees' sentiments may change if the company changes one of the feature levers. In addition, new companies not on Glassdoor could predict how its future employees might like their company. On the other side, prospective employees can use the model to predict a company rating for companies not already on Glassdoor.

Tools and Data:

By searching Glassdoor by "Data Analytics" and "Seattle", I found a list of more than 2200 job postings. I wanted to find the companies that posted these jobs so I used Selenium to web-scrape the list. Python's BeautifulSoup was rejected by Glassdoor -- it gave a "No bot" message. There were companies that had multiple job postings. After removing the duplicate companies, the number of unique companies was reduced to 300. I saw that there were still some duplicates because the same company was called different names, such as Fred Hutchinson being called Fred Hutch, Fred Hutchinson Cancer Research, etc. By looking at all features and checking to see which companies were exact matches, I was able to remove the duplicate companies based on the Year Founded, Headquarters, etc. Lastly, many companies did not report Revenue. After these rows were removed, the data set was incredibly small--only about 135 rows.

Modeling:

For the initial exploratory analysis, I looked at a heatmap and a pair-plot. Neither of these plots showed a very high correlation among features showing there was not much multi-colinearity. The features that seemed to correlate most with the company reviews were: CEO Rating, Year Founded, and Benefits. Looking at the values of both Size and Revenue, which both incremented by factors of 10, I took the logarithm of both. After doing this, their correlations with the company reviews increased. After running a few linear regression models with a combination of features, the five features that produced the highest Adjusted R-squared were: CEO Rating, Year Founded, Benefits, log Revenue, and log Size. The residuals were plotted and they were random as desired.

In order to test how good a model is, the output values need to be compared to something. Thus, the data set was split 70/30 into a training and testing set, respectively. Plotting the Mean Squared Error (MSE) vs Polynomial Degree showed a polynomial degree "2" might give better results than the ordinary degree "1" model. After running the training set through the linear regression models with polynomial degrees "1" and "2", the process was repeated with a Lasso and Ridge Regression. The best model was the Lasso regression (polynomial degree "2" and

lambda value of 1e-30) using CEO Rating, Benefits, Year Founded, log Revenue, and log Size as predictors. It gave an Adjusted R-squared of 0.34, an MSE of 0.102, and a mean absolute error of 0.25 This means that the model explains 34% of the variance in company rating. Although this is not a stellar value, it beat the baseline model which predicted the average company review of 3.76 with an MSE of 0.193 and an absolute error of 0.369.

Selenium takes a long time to scrape the data, but I think it is necessary to include more data. I could include companies across the US that hire data analysts. Or I could use all companies in Seattle.