Project 4: Web Scraping & Logistic Regression

Description

This week, we learned about web scraping and logistic regression. Now, we're going to put both of these skills to the test!

You're working as a data scientist for a contracting firm that's rapidly expanding. Now that they have their most valuable employee (you!), they need to leverage data to win more contracts. Your firm offers technology and scientific solutions and wants to be competitive in the hiring market. Your principal thinks the best way to gauge salary amounts is to take a look at what industry factors influence the pay scale for these professionals.

Aggregators like Indeed.com regularly pool job postings from a variety of markets and industries. Your job is to understand what factors most directly impact data science salaries and effectively, accurately find appropriate data science related jobs in your metro region.

Project Summary

In this project, we will practice two major skills. Collecting data by scraping a website and then building a binary predictor with Logistic Regression.

We are going to collect salary information on data science jobs in a variety of markets. Then using the location, title, and summary of the job, we will attempt to predict a corresponding salary for that job. While most listings DO NOT come with salary information (as you will see in this exercise), being to able extrapolate or predict the expected salaries for other listings will be extremely useful for negotiations :)

Normally we could use regression for this task; however, instead we will convert this into a classification problem and use Logistic Regression.

- **Question**: Why would we want this to be a classification problem?
- **Answer**: While more precision may be better, there is a fair amount of natural variance in job salaries; therefore, predicting a range be may be useful.

The first part of assignment will be focused on scraping Indeed.com and the second will be focused on using the listings with salary information to build a model and predict salaries.

Your job is to:

1. Collect data from Indeed.com on data science salary trends for your analysis.
    - Select and parse data from at least 1000 postings for jobs, potentially from multiple location searches.
2. Find out what factors most directly impact salaries (Title, location, department, etc.). In this case, we do not want to predict mean salary as would be done in a regression. Your boss believes that salary is better represented in categories than continuously
    - Test, validate, and describe your models. What factors predict salary category? How do your models perform?
3. Author a report to your Principal detailing your analysis.

BONUS PROBLEMS: 1. Your boss would rather tell a client incorrectly that they would get a lower salary job than tell a client incorrectly that they would get a high salary job. Adjust one of your logistic regression models to ease his mind, and explain what it is doing and any tradeoffs. Plot the ROC curve. 2. Text variables and regularization:

- **Part 1**: Job descriptions contain more potentially useful information you could leverage. Use the job summary to find words you think would be important and add them as predictors to a model.
- **Part 2**: Gridsearch parameters for Ridge and Lasso for this model and report the best model.

Goal: Scrape & clean data, run logistic regression, derive insights, present findings.

Requirements

- Scrape and prepare your data using BeautifulSoup.
- A Jupyter Notebook with your regression analysis for a peer audience of data scientists.
- A written report directed to your (non-technical!) Principal

  Pro Tip: You can find a good example report here.

Necessary Deliverables / Submission

- Materials must be in a clearly labeled Jupyter notebook.
- Materials must be pushed to student's github master branch.
- Materials must be submitted by the end of Week 5.

Dataset

1. We'll be utilizing a dataset derived from live web data: Indeed.com
2. To get the data, we will use the requests library and BeautifulSoup to scrape the webpage.

Suggested Ways to Get Started

- Read the docs for whatever technologies you use. Most of the time, there is a tutorial that you can follow, but not always, and learning to read documentation is crucial to your success!
- Document **everything**.
- Look up sample executive summaries online.

Additional Resources

- Advice on How to Write for a Non-Technical Audience
- Documentation for BeautifulSoup can be found here.

Project Feedback + Evaluation

Attached here is a complete rubric for this project.

Your instructors will score each of your technical requirements using the scale below:

Score	Expectations
0	*Incomplete.*
1	*Does not meet expectations.*
2	*Meets expectations, good job!*
3	*Exceeds expectations, you wonderful creature, you!*

This will serve as a helpful overall gauge of whether you met the project goals, but **the more important scores are the individual ones** above, which can help you identify where to focus your efforts for the next project!