Springboard---DSC
Capstone Three
Predicting Data Science Salary

By Grace Riccadonna
June 20, 2024

1 Introduction

Negotiating salaries can be challenging for data scientists due to the lack of clear benchmarks. This project aims to solve this problem by creating a tool that estimates data science salaries with a Mean Absolute Error (MAE) of approximately $11,000. This tool helps data scientists negotiate their income more effectively when they receive job offers.

The primary stakeholders are data scientists and job seekers in the data science field, along with HR professionals and recruiters. By providing reliable salary estimates, this tool empowers data scientists to make informed decisions during negotiations and assists companies in setting competitive compensation packages.

The project involved scraping over 1,000 job descriptions from Glassdoor using Python and Selenium. Features were engineered from the text to quantify the value companies place on skills like Python, Excel, AWS, and Spark. The model was optimized using GridSearchCV on Linear, Lasso, and Random Forest Regressors to find the best-performing model.

2 Data Acquisition and Wrangling

To predict data science salaries, we utilized a dataset scraped from Glassdoor.com, containing over 1,000 job postings. This comprehensive dataset includes various features related to job listings, company information, and job requirements. The key features of the dataset include:

- **Job Information:** Job Title, Salary Estimate, Job Description, Rating
- **Company Information:** Company Name, Location, Company Headquarters, Company Size, Company Founded Date, Type of Ownership, Industry, Sector, Revenue, Competitors

2.1 Data Acquisition

The dataset was acquired through web scraping using Python and Selenium. The following steps outline the data acquisition process:

- **Data Collection:** We utilized a web scraper to extract job postings from Glassdoor.com. The scraper was configured to collect detailed information for each job, including the job title, salary estimate, job description, company rating, and various company attributes.
- **Data Loading:** The scraped data was stored in a CSV file and loaded into a pandas DataFrame. This format is convenient for further processing and analysis using Python's data science libraries.

2.2 Data Wrangling

After acquiring the data, several steps were taken to clean and prepare it for analysis:

- **Parsing Numeric Data:** Extracted numeric values from salary estimates to create a usable salary column.
- **Employer-Provided Salary and Hourly Wages:** Created separate columns for employer-provided salaries and hourly wages.
- **Removing Incomplete Rows:** Removed rows that did not contain salary information to ensure a complete dataset.
- **Parsing Company Ratings:** Extracted the company rating from the text.
- **Company State:** Added a column for the state in which the company is located.
- **Headquarters Indicator:** Added a column indicating whether the job was located at the company's headquarters.
- **Company Age:** Transformed the company's founded date into a numerical column representing the age of the company.
- **Skills Columns:** Created columns indicating whether specific skills (Python, R, Excel, AWS, Spark) were mentioned in the job description.

- **Job Title and Seniority:** Added columns for simplified job titles and seniority levels.
- **Description Length:** Added a column for the length of the job description.

These steps ensured the dataset was clean and feature-rich, making it suitable for building and training the predictive model.

3 Exploratory Data Analysis

3.1 Overview of the Data and Standardizing the features

After display the dataset and get understanding of data structure, I used different function to standardizing and simplifying the features, for example:

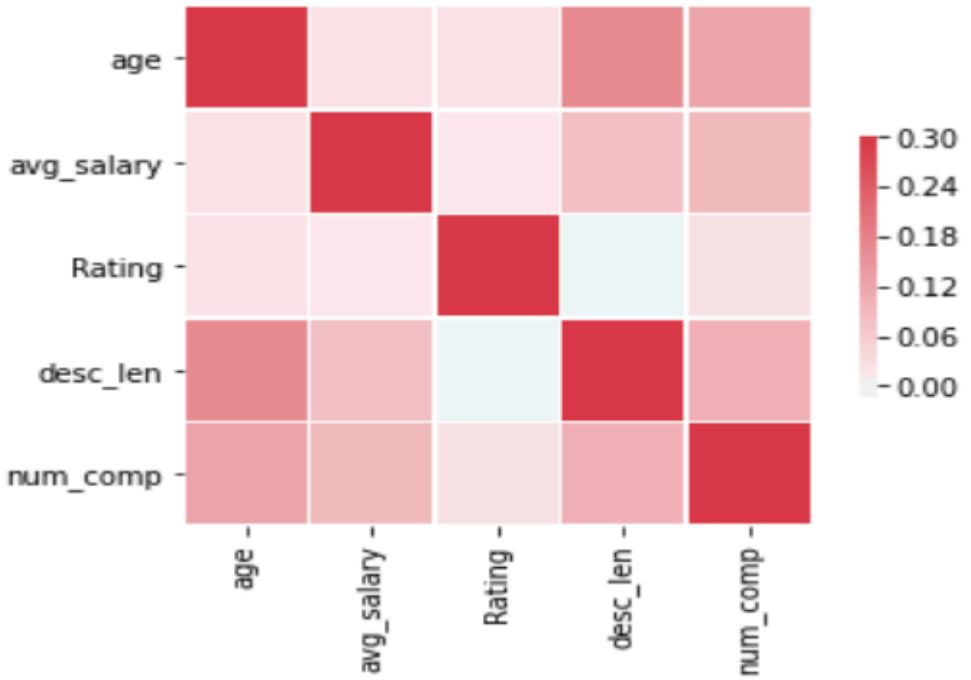
- Simplify job titles into a set of predefined categories. This helps in standardizing job titles for better analysis.
- Categorize "job title" into seniority levels, this helps in understanding the level of experience or responsibility associated with the job titles.
- Convert hourly wages to annual salaries, it ensures that salary comparisons are on an annual basis, making the data consistent for further analysis.

3.2 Correlation Analysis

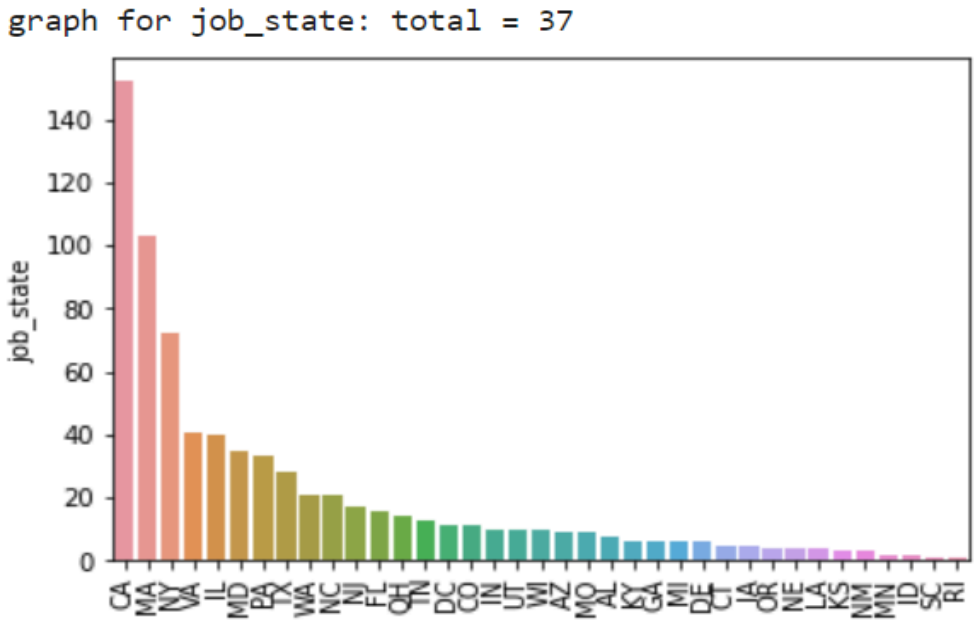
(1) 'age','avg_salary','Rating','desc_len' correlation?

- Using **'corr()'** to summarize the linear relationships between these variables and heatmap to visualize the correlation matrix.
- **Weak or No Linear Relationships:** There are weak or no significant linear relationships between most pairs of variables.
- **Slight Positive Relationship:** The only notable correlation is the weak positive relationship between **'age'** and **'desc_len'**.

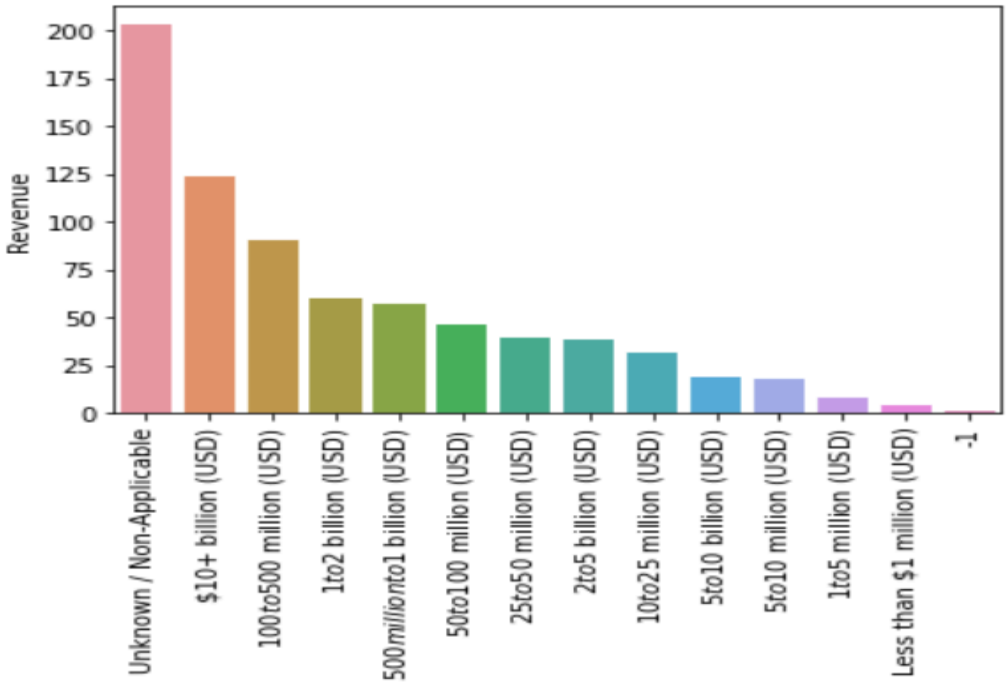
	age	avg_salary	Rating	desc_len
age	1.000000	0.019655	0.021655	0.163911
avg_salary	0.019655	1.000000	0.013492	0.078808
Rating	0.021655	0.013492	1.000000	-0.012281
desc_len	0.163911	0.078808	-0.012281	1.000000

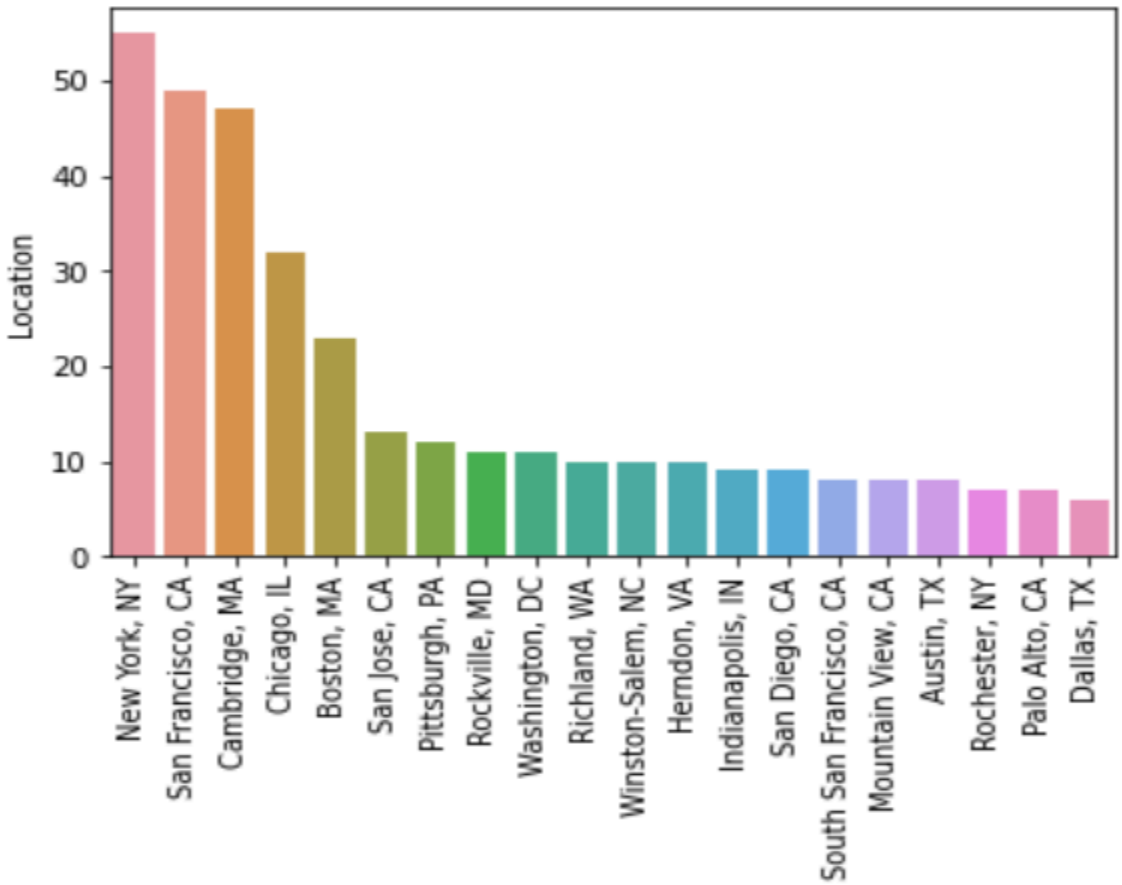


(2) Each categorical variable correlation?



graph for Revenue: total = 14





3 Modeling

3.1 Model Building

To construct robust predictive models, I followed a systematic approach involving data preprocessing, splitting the data, and evaluating multiple models using appropriate metrics.

- Categorical Variable Transformation: Transformed the categorical variables into dummy variables using one-hot encoding. This technique creates binary columns for each category, allowing the models to process categorical data effectively.
- Train-Test Split: Split the dataset into training and testing sets, allocating 80% of the data for training and 20% for testing. This split ensures that the model can be evaluated on unseen data, providing a realistic measure of its performance.

3.2 Model Selection

I employed three different machine learning algorithms to build the models and evaluated their performance using Mean Absolute Error (MAE). MAE was chosen as the evaluation metric because it is straightforward to interpret and provides a clear measure of prediction accuracy, without giving undue weight to outliers.

- **Multiple Linear Regression:** This served as the baseline model.
- **Lasso Regression:** Given the high dimensionality and sparsity of the dataset due to many categorical variables.
- **Random Forest Regressor:** To further address the sparsity and capture non-linear relationships in the data

4 Conclusions

The Random Forest model demonstrated superior performance compared to the other approaches on both the test and validation sets. Here are the Mean Absolute Error (MAE) values for each model:

- **Random Forest Regressor:** MAE = 11.22
- **Linear Regression:** MAE = 18.86
- **Ridge Regression:** MAE = 19.67

The Random Forest model's significantly lower MAE indicates its ability to better capture the underlying patterns in the data, making it the most effective model among the ones tested. This superior performance can be attributed to the model's capability to handle high-dimensional data and capture non-linear relationships, which is crucial given the sparsity and complexity of the dataset.

For job seekers, HR professionals, or hiring managers, using this model to inform salary negotiations, an MAE of $11,220 signifies the typical error margin. This helps in setting realistic salary expectations and negotiating offers. For example, if the model predicts a salary of $100,000, the actual salary could typically be expected to fall within the range of $88,780 to $111,220.