Improving Search Efficiency from LinkedIn Job Postings by a Factor of Four

Created by: Marin Stoytchev
Date: April 14, 2021

Link to project data and notebooks:
https://github.com/marin-stoytchev/Springboard-projects/tree/master/CapstoneProject_3

Executive Summary

Problem Statement

- Using data from LinkedIn job postings for the past week, can we improve the job search efficiency by 100% and increase the accuracy of predicting which daily job postings of interest to us will have the smallest number of applicants over the next five to seven days?

General Job Search Findings

- Currently, LinkedIn (and other sites) job searches are rather inefficient in producing highly relevant results. More than 50% of the job search results have been found to not be related to the position of interest.
- In addition, a large portion of the information contained in the job postings is very ambiguous and does little to differentiate different types of positions.

Objectives Achieved

- The first issue outlined above has been solved by implementing our Intelligent Filtering Algorithm (IFA©) to increase the job search efficiency by a factor of 2.1 (better than 100% improvement).
- In regards with the second issue, we have created a model which improves the accuracy of predicting job postings with the least number of applicants by a factor of 1.9.
- When accounting for both, the overall search efficiency for our specific purposes is increased by a factor of four.

I. Process Outline

Six-Step Process

1. Problem Definition
2. Data Collection
3. Data Cleaning
4. Data Exploration and Visualization
5. Data Modeling and Model Optimization
6. Model Predictions and Prediction Results Analysis

II. Problem Definition

Problem Definition

Problem

- Job searching has always been an integral part of one's career. When conducted in an efficient and meaningful way, it can lead to numerous valuable opportunities. In the current economic environment, the value of an effective job search has increased even further. We are a consulting company helping our clients in identifying highly relevant opportunities with limited competition, thus increasing their success rate in finding the right opportunity for a successful career.

Question

- Can we improve the efficiency of job searches by 100%? In addition, can we increase the accuracy of predicting which daily job postings will have the smallest number of applicants over the next five to seven days?

Risk

- It is possible that the information in a large number of job postings is either missing, or inaccurate, or there is little to no differentiating information. This would likely lead to inaccurate predictions from our model.

Tasks

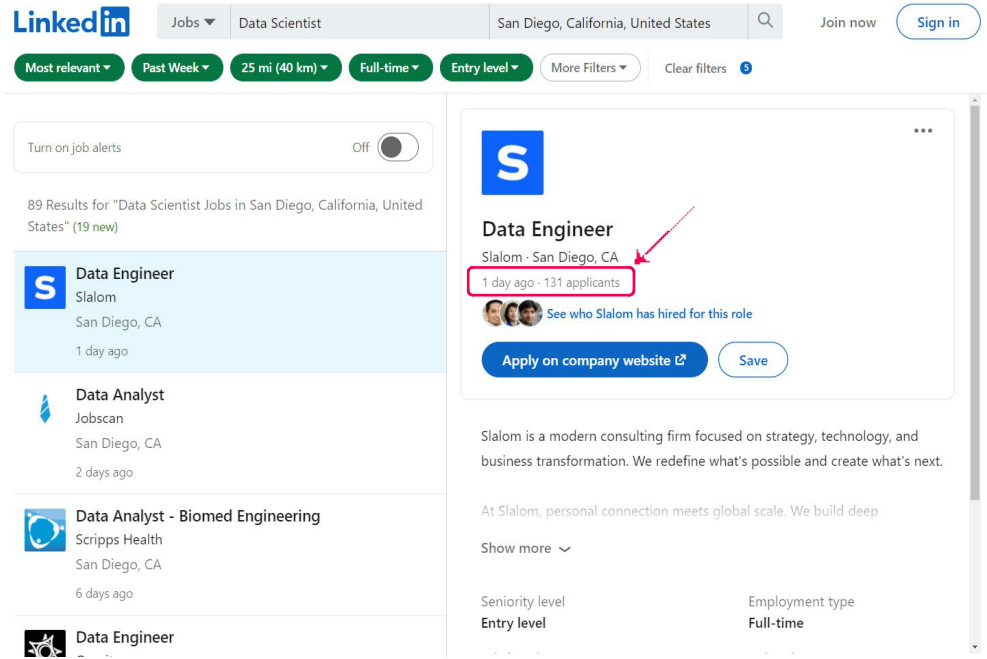
- Collect, process and analyze LinkedIn job postings over several weeks. Using this data, create a model that will predict with high confidence level which daily job postings of interest to us will have the smallest number of applicants over the next seven days. These tasks should be performed within six to eight weeks and have the data and tools available for use by our customers.

III. Data Collection

Collecting Data by Scraping LinkedIn Job Postings

- Information from LinkedIn job postings has been collected by using in-house developed custom internet scraping tool which uses Selenium with Chrome driver
- The decision for using LinkedIn job postings is made based on the availability of information critical to us which is not available from other job search sites, namely:
  - **Number of days since posting**
  - **Number of applicants for the position**
- Search is conducted for Data Scientist full-time positions, posted in the past week for all seniority levels – Entry, Associate, Senior – and for 16 large metropolitan areas in the United States
- The search is **anonymous** (not logged-in into a LinkedIn account) to avoid violating LinkedIn membership policies
- The data used in the analysis and modeling has been collected over five weeks starting on Feb. 5, 2021

Example of LinkedIn Job Search Results



Information Extracted from LinkedIn Job Postings

- The different data features extracted by our scraping tool are shown in the example provided below.
- Special mention deserves the Data Science Terms Count which is determined using Python Regular Expressions (regex) package by counting the number of occurrences of the following terms in the job description: 'data science', 'data scientist', 'machine learning', 'deep learning', 'unsupervised learning', 'artificial intelligence', 'data model', 'predictive model', 'data visualization', 'classification', 'regression', 'clustering'.

Example of Data Collected by Scraping

	Job Title	Company Name	Location	Metro Area	Time Posted	Number of Applicants	Industry Info	Education-Bachelor	Education-Master	Education-Doctor	Experience	Data Science Terms Count	Seniority Level
0	Data Scientist	Honeywell	Atlanta, GA	ATL	2 days ago	40 applicants	['Seniority level', 'Entry level', 'Employment...	0	1	1	['2 years']	7	entry
1	ENTRY LEVEL (Data Analyst/Data Scientist)	SynergisticIT	Atlanta, GA	ATL	4 hours ago	Be among the first 25 applicants	['Seniority level', 'Entry level', 'Employment...	1	1	0	['10 years']	2	entry
2	Data Scientist	Inspire Brands	Atlanta, GA	ATL	7 hours ago	25 applicants	['Seniority level', 'Entry level', 'Employment...	0	0	0	['1-3 years']	4	entry
3	Associate Data Scientist	The Home Depot	Atlanta, GA	ATL	22 hours ago	29 applicants	['Seniority level', 'Entry level', 'Employment...	0	1	0	['2+ years']	3	entry
4	Associate Data Scientist	Edelman Data & Intelligence (DxI)	Atlanta, GA	ATL	1 hour ago	Be among the first 25 applicants	['Seniority level', 'Entry level', 'Employment...	0	0	0	[]	3	entry

III. Data Cleaning

Data Cleaning

- After scraping, the data features ‘Job Title’, ‘Time Posted’, ‘Number of Applicants’, ‘Industry Info’, and ‘Experience’ contain mixed and ambiguous information

- That’s why a significant time was spent to process and clean these features, so that the information contained there can be used in data analysis and modeling

Scraped Data

	Job Title	Company Name	Location	Metro Area	Time Posted	Number of Applicants	Industry Info	Education-Bachelor	Education-Master	Education-Doctor	Experience	Data Science Terms Count	Seniority Level
0	Data Scientist	Honeywell	Atlanta, GA	ATL	2 days ago	40 applicants	['Seniority level', 'Entry level', 'Employment...	0	1	1	['2 years']	7	entry
1	ENTRY LEVEL (Data Analyst/Data Scientist)	SynergisticIT	Atlanta, GA	ATL	4 hours ago	Be among the first 25 applicants	['Seniority level', 'Entry level', 'Employment...	1	1	0	['10 years']	2	entry
2	Data Scientist	Inspire Brands	Atlanta, GA	ATL	7 hours ago	25 applicants	['Seniority level', 'Entry level', 'Employment...	0	0	0	['1-3 years']	4	entry
3	Associate Data Scientist	The Home Depot	Atlanta, GA	ATL	22 hours ago	29 applicants	['Seniority level', 'Entry level', 'Employment...	0	1	0	['2+ years']	3	entry
4	Associate Data Scientist	Edelman Data & Intelligence (DxI)	Atlanta, GA	ATL	1 hour ago	Be among the first 25 applicants	['Seniority level', 'Entry level', 'Employment...	0	0	0	[]	3	entry

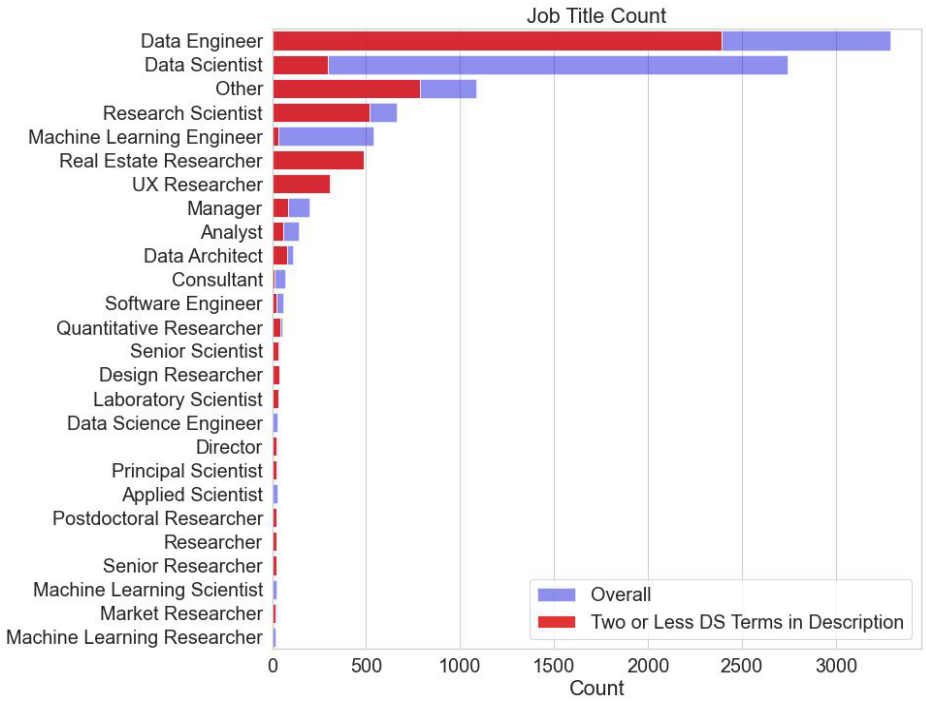
Cleaned Data

	Job Title	Company Name	Industry	Job Function	Metro Area	Education-Bachelor	Education-Master	Education-Doctor	Minimum Experience	Seniority Level	Data Science Terms Count	Time Posted	Number of Applicants
0	Data Scientist	Honeywell	Staffing and Recruiting	Engineering	ATL	0	1	1	2	entry	7	2	40
1	Data Scientist	SynergisticIT	Staffing and Recruiting	Information Technology	ATL	1	1	0	10	entry	2	1	0
2	Data Scientist	Inspire Brands	Financial Services	Engineering	ATL	0	0	0	1	entry	4	1	25
3	Data Scientist	The Home Depot	Financial Services	Engineering	ATL	0	1	0	2	entry	3	1	29
4	Data Scientist	Edelman Data & Intelligence (DxI)	Research	Analyst	ATL	0	0	0	0	entry	3	1	0

IV. Data Processing and Exploration

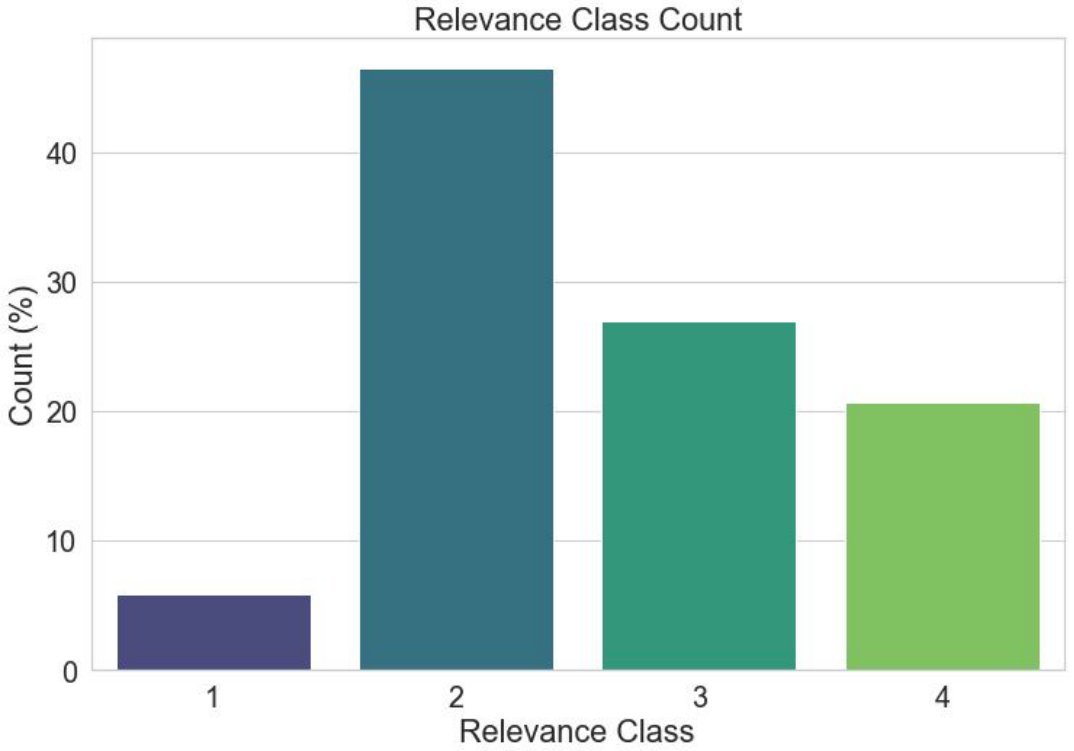
Filtering Data by Relevance: Data Science Terms Count

- In the first data filtering stage we use the 'Data Science Terms Count' feature which determines the postings relevance.
- We define a minimum of three data science terms in the job description as the relevance threshold.
  - The reason behind is that there are many postings for positions different from data scientist with a description which reads: *"The data engineer will be working closely with the data science team to support various data science projects ...."*
- The plot of the overall count of postings with different job titles and those below the relevance threshold reveals two big flaws (these flaws are pertinent to all job search sites we have explored and not only to LinkedIn):
  - **1) Although the search is for Data Scientist positions, only about 27% of the results actually have the title Data Scientist. The largest number of records is for postings with the title of Data Engineer.**
  - **2) A large majority of the results do not pass the relevance threshold regardless of the job title.**
- That's why all results below the relevance threshold, except the postings with 'Data Scientist' title, are eliminated from the data for future analysis and modeling – 5076 out of 10165 data points are left after filtering.



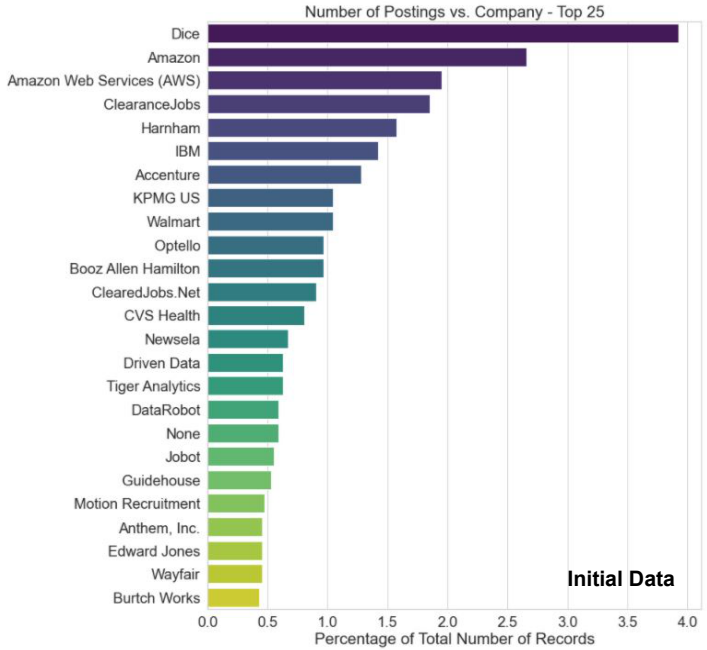
Filtering Data by Relevance: Creating Relevance Feature

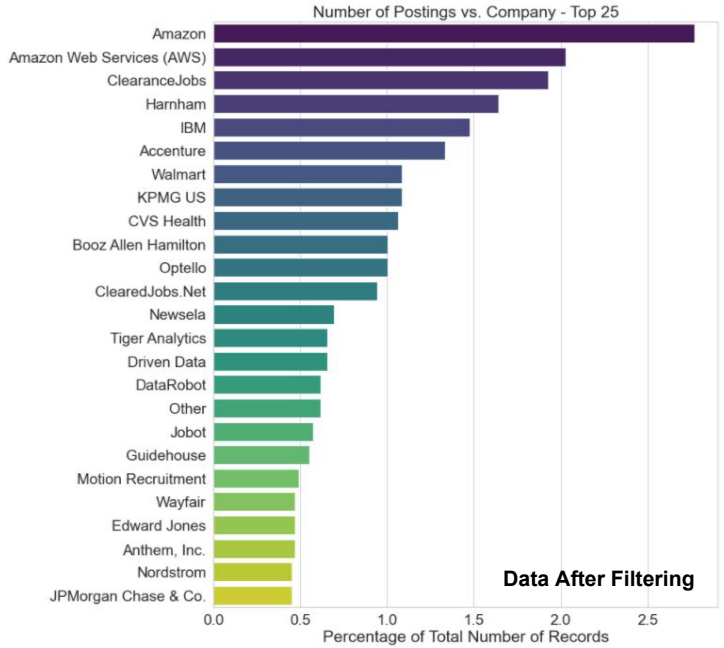
- After the data filtering performed by relevance, we use the data science terms count values to create a new categorical feature ‘Relevance’ following the scheme below:
  - Class 1: Two or less data science terms
  - Class 2: Three to six data science terms
  - Class 3: Seven to ten data science terms
  - Class 4: More than ten data science terms
- The representation of each of the Relevance classes as percentage of all data is shown on the plot to the right:
  - Class 2 occurs most often, followed by Classes 3 and 4
  - Class 1 is least represented and consists of the Data Scientist postings which have less than three data science terms in the description



Filtering Data by Company Name: Eliminating Dice Postings

- There are 1534 distinct company names in the data and 4% of the postings are from Dice
- This reveals another major issue valid for many job search sites:
  - Dice is another large job posting and job search site. However, it is not a direct employer or a job placement company. Therefore, postings from Dice replicate the postings from direct employers or job placement companies. Because of this redundancy, all postings from Dice are removed from the data
- This completes the second data filtering step.
  - After the two data filtering steps there are 4877 out 10165 of data points left – achieves search efficiency improvement by a factor of 2.1





Data Separation: Known and Unknown Number of Applicants

- The two most critical features for our project are 'Time Posted' and 'Number of Applicants'
- 'Number of Applicants' reveals a major issue specific for **anonymous** LinkedIn job search:
  - A large majority of the collected data – 7776 out of 10167 records (~76%) – has no definite number of applicants and is instead denoted as 'Be among the first 25 applicants'
  - Even after the data filtering performed in the previous data processing stage, still 69% of the records have unknown number of applicants
- In order to conduct adequate analysis and data modeling, we separate the current filtered data into two different datasets
  - **Data A** with known number of applicants, which will be used for modeling (1513 data points)
  - **Data B** with unknown, but capped, number of applicants, which will be used in predictions (3364 data points)

Unprocessed 'Number of Applicants' Information

```
Be among the first 25 applicants    7776
Over 200 applicants                  385
26 applicants                         62
33 applicants                         51
35 applicants                         49
                                     ...
179 applicants                         1
147 applicants                         1
191 applicants                         1
175 applicants                         1
136 applicants                         1
Name: Number of Applicants, Length: 177, dtype: int64
```

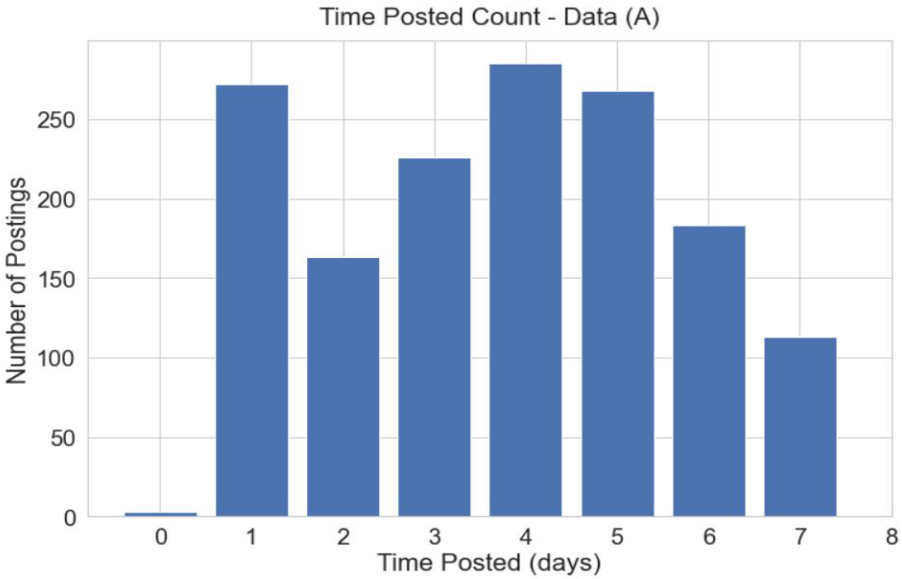
Data A

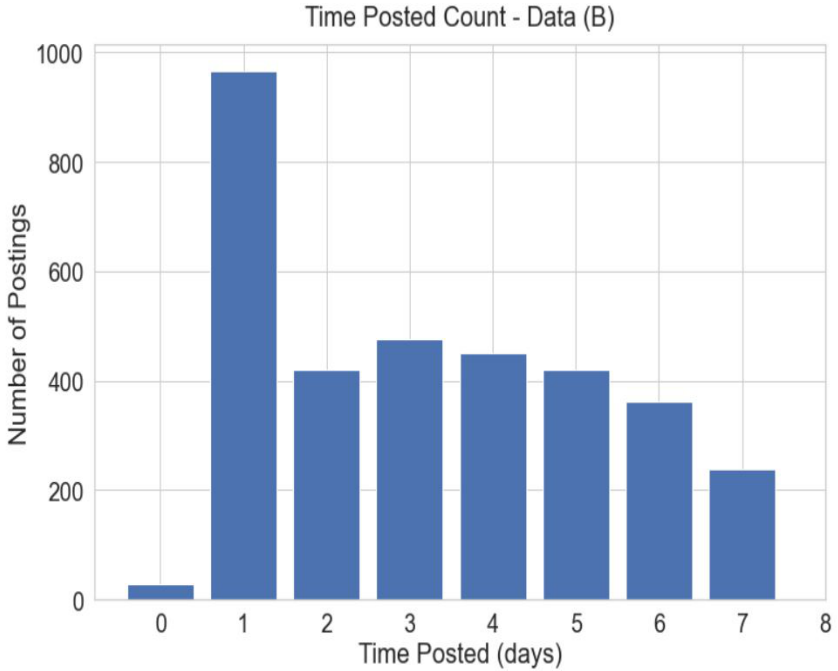
	Job Title	Company Name	Industry	Job Function	Metro Area	Education-Bachelor	Education-Master	Education-Doctor	Minimum Experience	Seniority Level	Data Science Terms Count	Time Posted	Number of Applicants	Relevance
0	Data Scientist	Honeywell	Staffing and Recruiting	Engineering	ATL	0	1	1	2	entry	7	2	40	3
1	Data Scientist	Inspire Brands	Financial Services	Engineering	ATL	0	0	0	1	entry	4	1	25	2
2	Data Scientist	The Home Depot	Financial Services	Engineering	ATL	0	1	0	2	entry	3	1	29	2
3	Data Scientist	Georgia-Pacific LLC	Construction	Engineering	ATL	0	1	0	2	entry	7	2	27	3
4	Data Scientist	Mailchimp	Internet	Engineering	ATL	0	0	0	2	entry	6	5	192	2

Data B

	Job Title	Company Name	Industry	Job Function	Metro Area	Education-Bachelor	Education-Master	Education-Doctor	Minimum Experience	Seniority Level	Data Science Terms Count	Time Posted	Number of Applicants	Relevance
0	Data Scientist	SynergisticIT	Staffing and Recruiting	Information Technology	ATL	1	1	0	10	entry	2	1	-10	1
1	Data Scientist	Edelman Data & Intelligence (DxI)	Research	Analyst	ATL	0	0	0	3	entry	3	1	-10	2
2	Data Scientist	Leidos	Defense	Engineering	ATL	1	0	0	1	entry	10	1	-10	3
3	Data Scientist	Vanderlande	Staffing and Recruiting	Information Technology	ATL	1	0	0	4	entry	4	1	-10	2
4	Data Scientist	Search Discovery	Marketing and Advertising	Engineering	ATL	0	0	0	2	entry	7	1	-10	3

Data A and Data B Comparison: 'Time Posted'

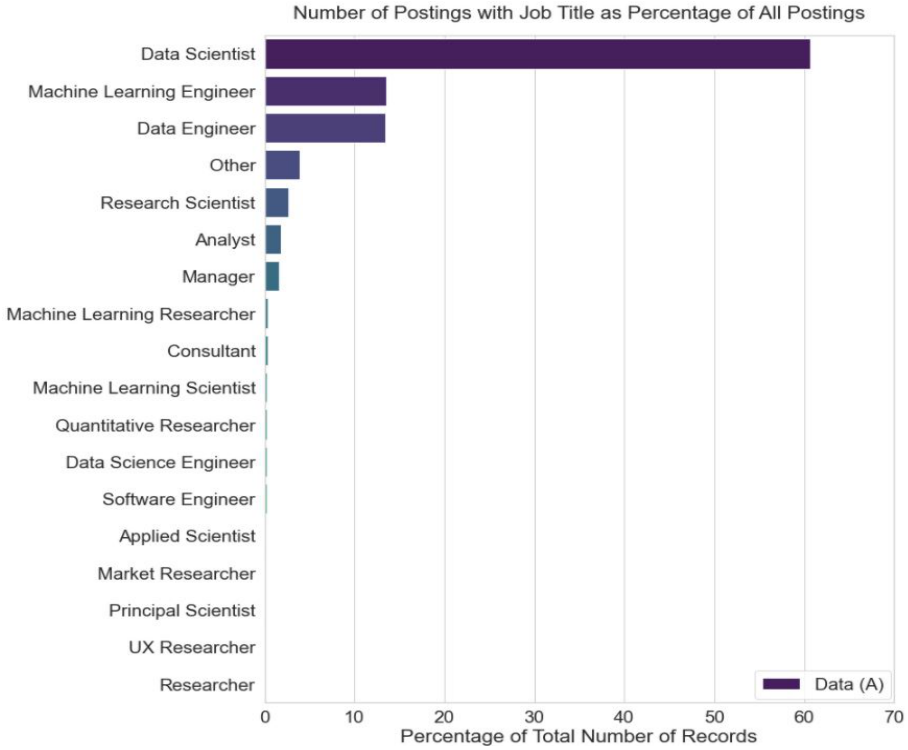


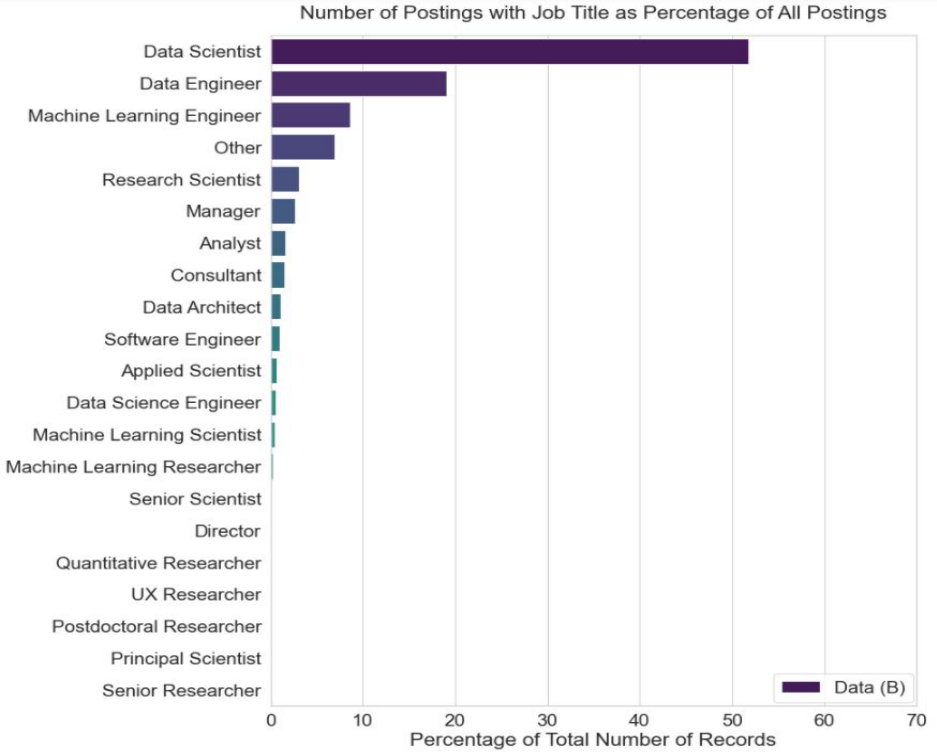


- The number of postings with different 'Time Posted' values vary between approximately 110 and 290 (excluding 0 which corresponds to time posted in minutes)
- The highest count belongs to 4, 1 and 5 days
- There is not one dominant 'Time Posted' value

- 'Time Posted' values of 2, 3, 4, 5, 6, and 7 days show similar variations as observed in Data A
- Here, however, time posted of 1 day dwarfs the second highest by a factor of two

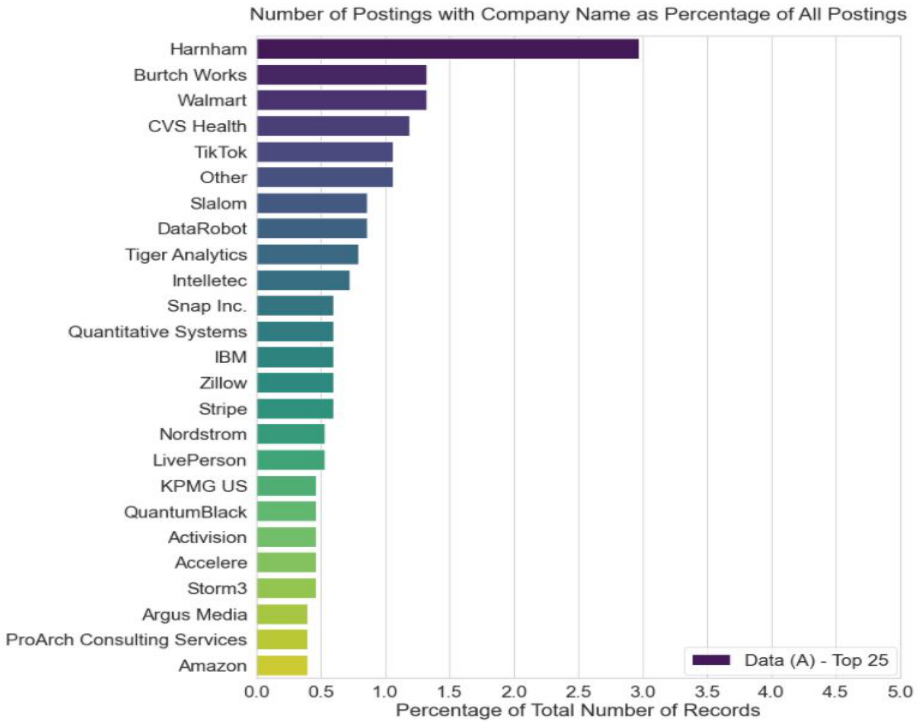
Data A and Data B Comparison: 'Job Title'

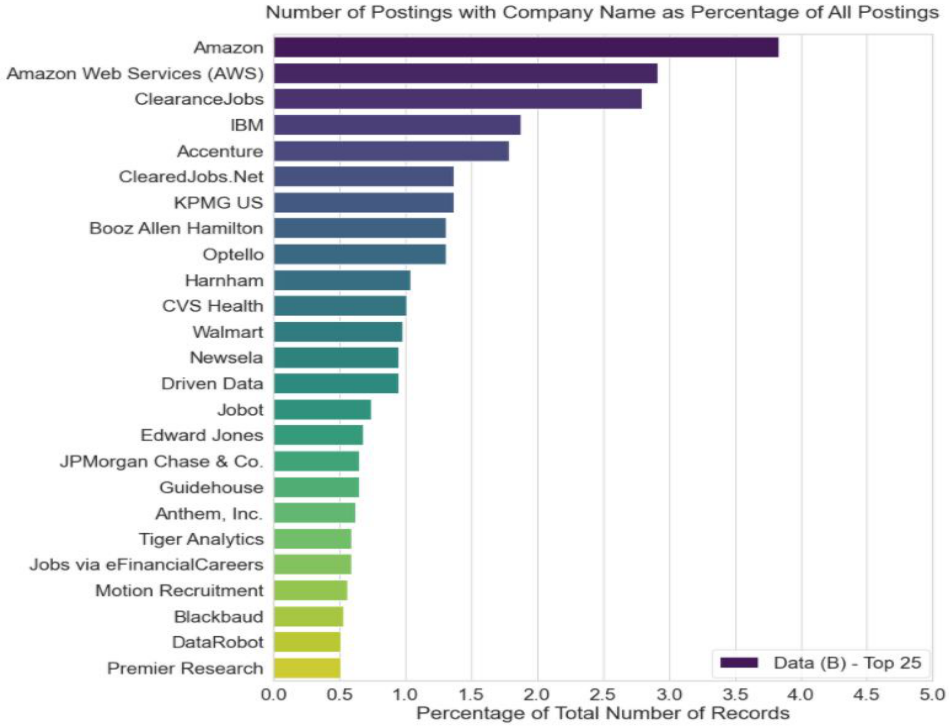




- In both datasets Data Scientist has the highest count by a large margin
- Among the high count job titles, the most notable difference is that Data Engineer is placed second in Data B (unknown number of applicants) with a significantly larger count compared to Machine Learning engineer, while these two are very close in Data A.

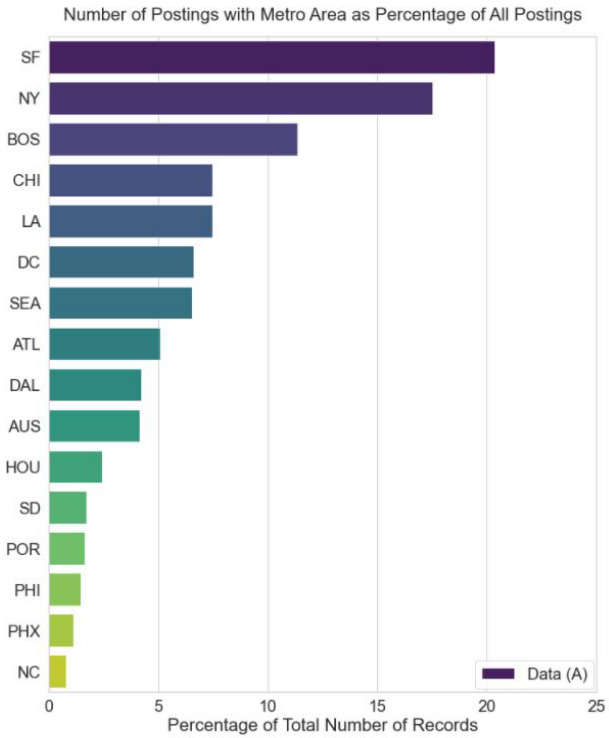
Data A and Data B Comparison: 'Company Name'

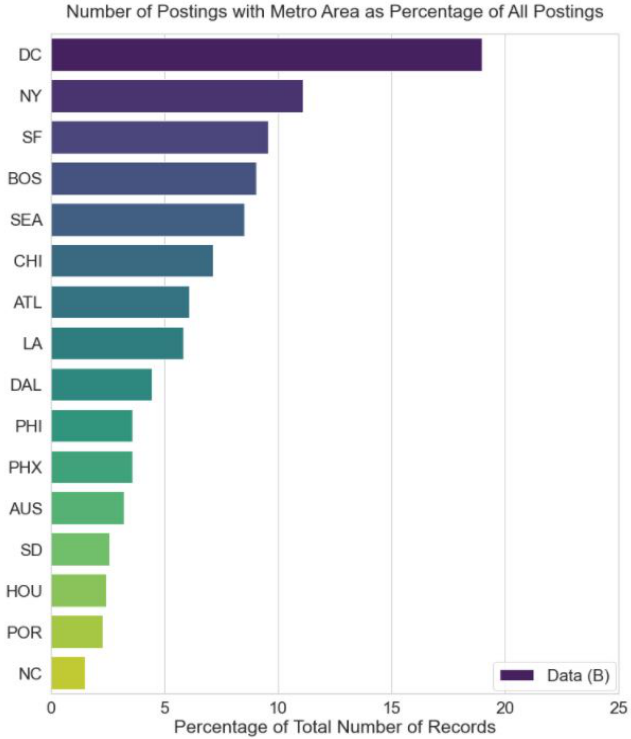




- In the case of 'Company Name', the differences between the two datasets are much more significant.
- The top places in Data A belong to two of the largest recruitment and placement companies, followed by Walmart and CVS.
- The top place in Data B is occupied by Amazon. Another notable company at the top here is IBM.

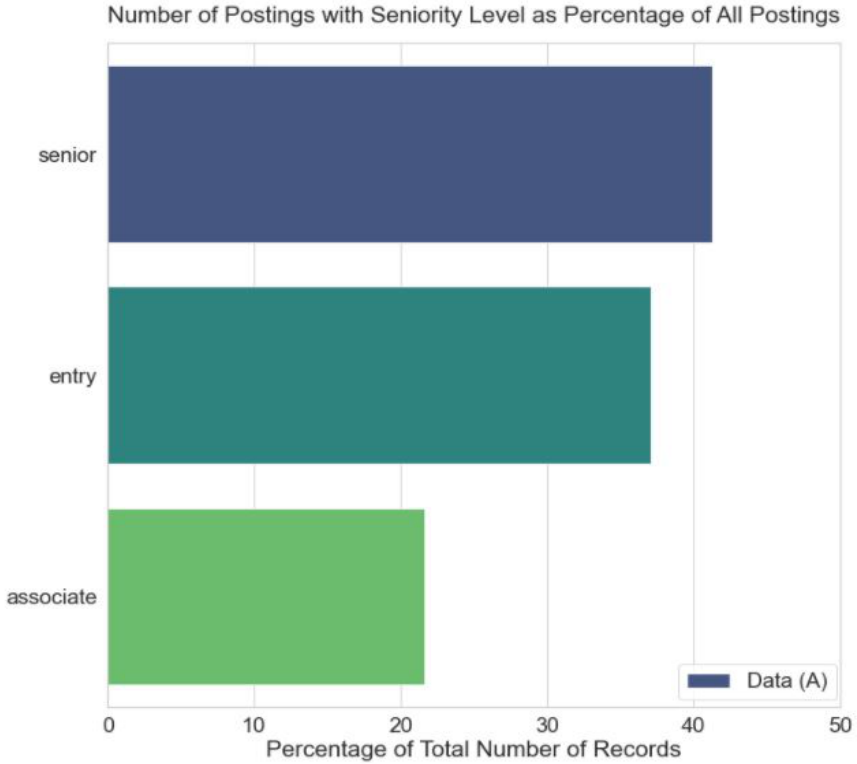
Data A and Data B Comparison: 'Metro Area'

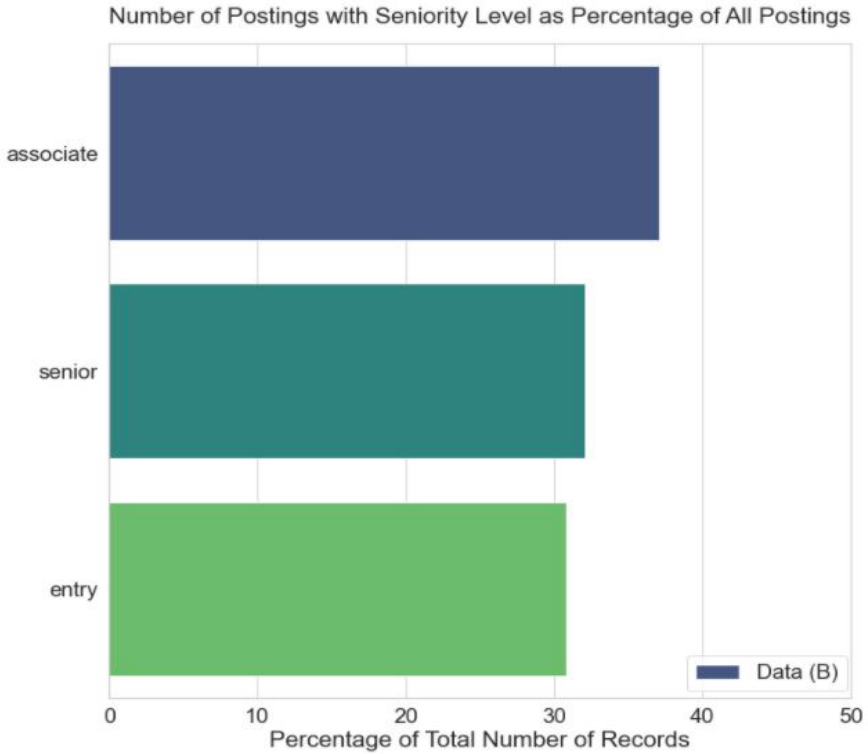




- The biggest difference here is that the Washington D.C. metro area has the highest percentage of postings in Data B, while being quite low in Data A.
- San Francisco, New York, and Boston are at the top in both Data A and Data B.

Data A and Data B Comparison: 'Seniority Level'



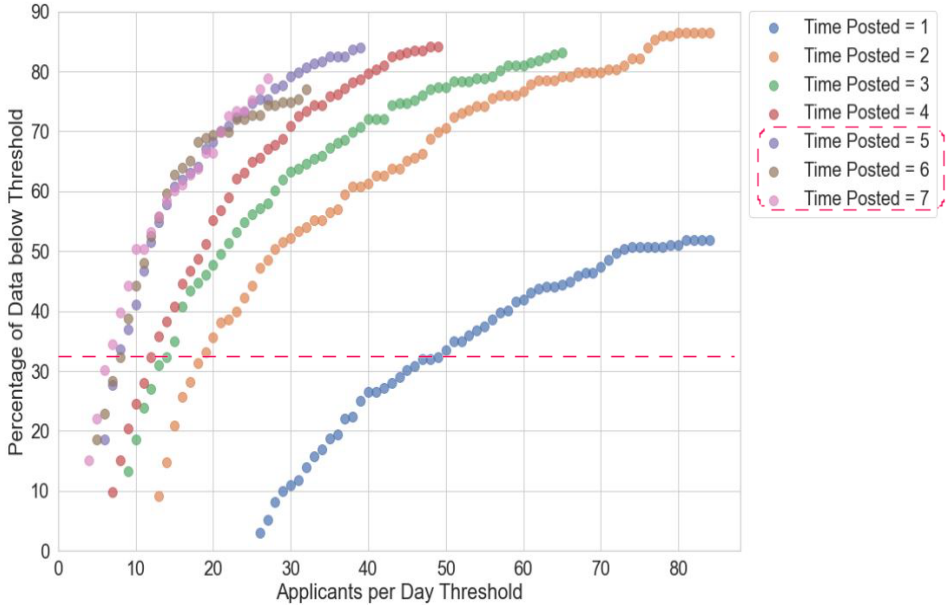


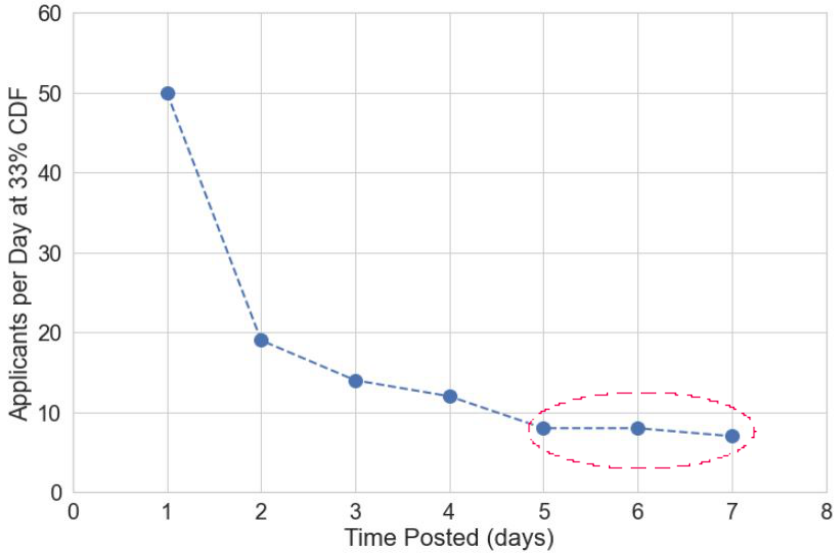
- Although relatively close, it is worth noting that Associate has the highest number of postings in Data B and the smallest percentage in Data A.

V. Data Modeling and Model Optimization

Data Modeling: Creating New Feature, ‘Applicants per Day’

- For modeling we use Data A in which both ‘Time Posted’ and ‘Number of Applicants’ are known. This allows us to create a new feature, ‘Applicants per Day’ as the ratio of the above two features providing the *average* number of applicants per day for each posting.
- Despite being an average metric, it allows to quantitatively measure the changes in the number of applicants for different times posted. This is done by creating the CDFs of ‘Applicants per Day’ for the different ‘Time Posted’ values and by tracking the number of applicants per day at a specific CDF value (for our goals we have selected the 33% mark).
- As shown below, there is a clear decrease in the average number of applicants per day with the increase in time posted. More importantly, the results show that data for time posted of 5, 6, and 7 days has the same behavior. Therefore, these records comprise a uniform subset which can be used for consistent modeling.





Data Modeling: Creating Target Feature

- As discussed in the previous slide, the Data A subset with Time Posted of 5, 6, and 7 days is selected to be used for modeling.
- However, 'Applicants per Day' is not suitable for our problem. We are not interested in predicting whether a certain posting will have 18 or 20 applicants per day, but rather to find out if the number of applicants will be under a certain threshold. ***That's why we use the data to create a new categorical target feature 'Position Demand' which has two classes***:
  - **Class 1**: Postings with 8 or less applicants per day. Class 1 is the main focus of our predictions which translates into determining which postings will have no more than 40 to 60 total number of applicants five to seven days after the date of posting, respectively.
  - **Class 2**: Postings with more than 8 applicants per day
- The breakdown of the percentage of data points in each class is shown in the plot to the right. Class 1 accounts for 34% and Class 2 accounts for 66% as expected from the CDF plot.

Data Modeling: Modeling Process and Model Validation

- For the model, we use the machine learning algorithm, *XGBClassifier*, which is best suited to tackle this problem.
- In order to achieve the best possible performance, first a model without optimization is used in order to establish a baseline, followed by Bayesian optimization with f1-score as the optimization metric.
- The validation results for the model without and after optimization are presented below. Our focus is on Class 1 precision, first, and Class 1 recall, second. As the results show, the model after optimization achieves better performance providing Class 1 precision increase of 2%, Class 1 recall increase of 10%, and Overall f1-score increase of 2%

XGBClassifier Model Validation Performance

Confusion Matrix:

17	32
10	82

Classification Report:

	precision	recall	f1-score	support
1	0.63	0.35	0.45	49
2	0.72	0.89	0.80	92
accuracy			0.70	141
macro avg	0.67	0.62	0.62	141
weighted avg	0.69	0.70	0.67	141

Model w/o Optimization

Confusion Matrix:

22	27
12	80

Classification Report:

	precision	recall	f1-score	support
1	0.65	0.45	0.53	49
2	0.75	0.87	0.80	92
accuracy			0.72	141
macro avg	0.70	0.66	0.67	141
weighted avg	0.71	0.72	0.71	141

Model w/ Optimization

Data Modeling: Class 1 Predictions Confidence Interval

- The results presented in the previous slide are not sufficient to establish a confidence interval in predicting the Class 1 precision and recall. The reason for this is twofold:
  - **Optimization metric**: In the optimization process the metric is the overall f1-score which reflects the precision and recall of both classes; it is not possible to assign the score for a particular class as an optimization metric. Therefore, a model with a good optimization score does not imply good Class 1 scores.
  - **Class imbalance**: Here, Class 1 represents only 34% of all data points. In cases of class imbalance, the minority class is susceptible to greater predictions accuracy variations – small absolute variations in the predictions would cause small overall score variations, but could result in significant changes in the performance metrics associated with the minority class.
- That's why we have ran ten times the optimization and tracked the Class 1 precision and recall together with the overall f1-score. Results show:
  - **Convergence (f1-score)**: All optimization runs converge well to provide an overall f1-score close to 0.72. The only exception is optimization run #5 which is likely due to an extremely misfortunate choice of a starting point.
  - **Class 1 confidence interval**: Precision values – 50-75%, 65% average; recall values – 39-47%, 43% average. The average values here set the typical expectations from predictions with unknown data.

Model Performance Metrics Results from Ten Optimization Runs

Optimization #	C1 Precision	C1 Recall	Overall f1-score
1	0.65	0.45	0.72
2	0.75	0.43	0.75
3	0.66	0.47	0.73
4	0.66	0.43	0.72
5	0.50	0.41	0.65
6	0.66	0.39	0.72
7	0.70	0.47	0.74
8	0.66	0.43	0.72
9	0.62	0.43	0.71
10	0.63	0.39	0.71
Average	**0.65**	**0.43**	**0.72**

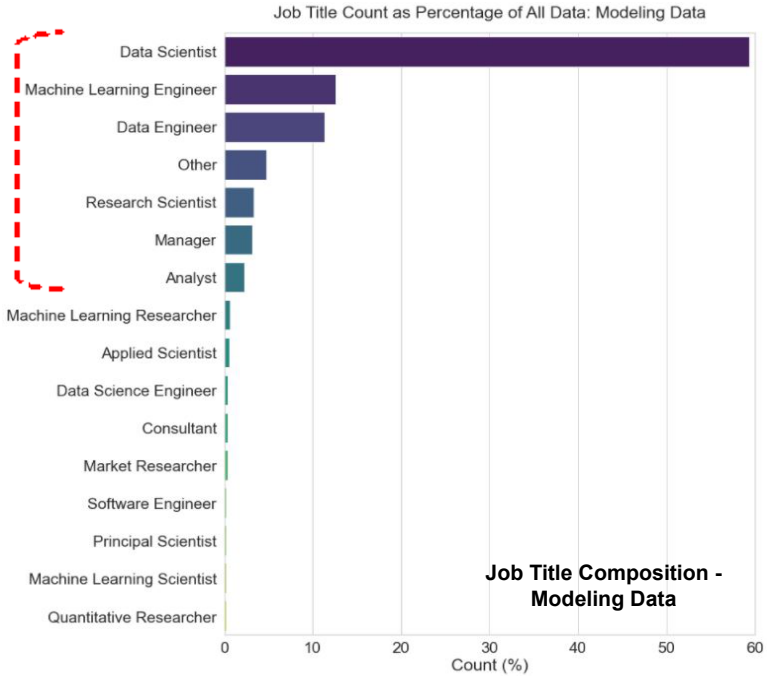
Data Modeling: The AI Advantage

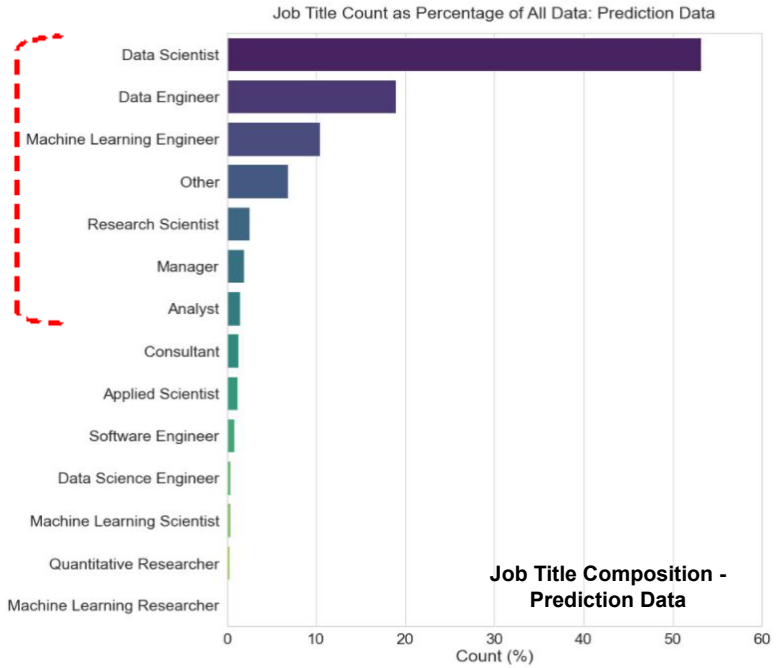
- The results from the model validation presented here point to twofold advantage in using the model predictions versus simply using the data as is:
  - **Advantage 1 – Increase in Search Efficiency**: If we use the data as is, since Class 1 represents 34% of the data, under the assumption that the class ratio does not change over time, browsing through 100 daily postings would yield 34 records belonging to the category we are interested in. When using the model's predictions, 100 postings will have 65 records in the desired class (for the example here the average precision value is assumed). ***Thus, an increase of search efficiency by a factor of 1.9 is achieved.***
  - **Advantage 2 – Speed Factor**: Please, note that advantage 1 holds in the case of an infinite number of postings. If the number of postings is finite, the low recall penalizes the AI performance by discarding 57% of the desired postings during classification. Therefore, after a significant (infinite) amount of time spent, a person will eventually come up with a larger number of 'good' postings. However, the possibility of a person to get to that point is nullified because by the time one could possibly scan through the amount of postings available a large portion of these postings will be outdated. ***Thus, although sometimes overlooked at the expense of greater accuracy/precision, one should not underestimate the extreme value of time saving the AI provides.***

VI. Predictions and Analysis of Prediction Results

Predictions: Model and Data Used

- The model used for predictions is the optimized model with performance closest to the average values (run # 8).
- Regarding predictions, the ultimate goal for us is to be collecting daily postings and to use the model to predict which of these postings will have the smallest number of applicants (Class 1) five to seven days from the date of posting (40 to 60 total number of applicants). Therefore, for predictions we use the subset from Data B (unknown number of applicants) with a 'Time Posted' value of one day.
- The feature with most similar composition in both the Modeling and Prediction data, 'Job Title', will be used to examine whether the predicted Class 1 records are consistent with the expectations set by the Modeling data.

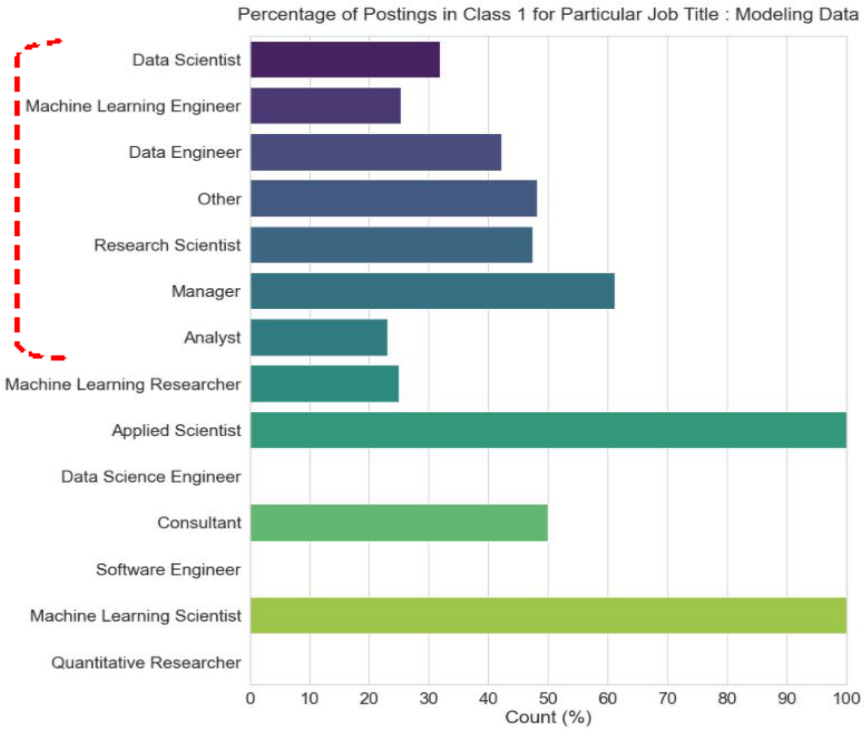




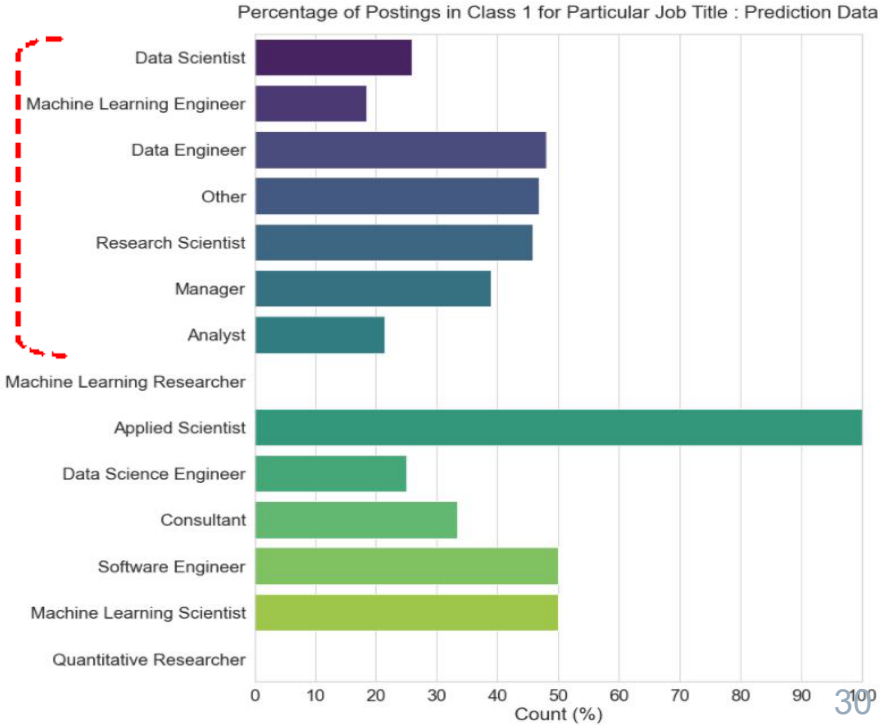
Class 1 Predictions vs. Expectations

- Since there is no way of knowing the possible classes for the prediction data, the best we can do is to examine whether the Class 1 count from the predictions is consistent with that in the modeling data. Significant discrepancies will raise a red flag about our modeling and predictions. Similar behavior will give us a certain degree of confidence in the prediction results. The prediction results show similar behavior for the top titles, in particular, thus increasing our confidence in the model and its predictions.

Job Title Class 1 Percentage - Modeling Data



Job Title Class 1 Percentage - Prediction Data



VII. Project Summary

Project Summary

Summary of Performed Data Collection, Processing and Modeling

- We have built an in-house custom internet scraping tool which has been used to collect LinkedIn weekly job search results for Data Scientist over a period of five weeks.
- The collected data has been processed and analyzed to filter out irrelevant results which has resulted in search efficiency improvement by a factor of two.
- A model has been created and optimized using a portion of the data to predict which postings of interests will have the least number of applicants. The model predictions improve the selection rate of desired postings by a factor of two in comparison with a random choice.

Potential for Improvement and Next Steps

- Collect more data to examine for possible changes in search results over time and update the model.
- Collect limited data from 'internal' (logged-in) LinkedIn job search and use it to calibrate our filtering and modeling procedures.
- Investigate different job search options - Glassdoor, Indeed, etc.