Final Project Report
Chris Coble
7/24/2024

## Introduction

In this economy of many opportunities, it is difficult to know how you rank in the market and where you should apply for work. Especially as a burgeoning data scientist, I feel the pressure of wanting to know exactly where I should work and what pay I should be getting. Entering this new industry is going to be challenging, but with the help of machine learning, patterns in the economy can be revealed extremely quickly. The aim of this project was to create a model that would predict the best places to look for a data science job based off of my desired salary and how competitive

## Exploratory Data Analysis

The approach to this modeling was to generate a location classification model based off of a kaggle dataset containing data science job posts in 2024 and a Bureau of Labor Statics report for data scientist employment statistics in 2023. The features used to construct the classification model were all numerical. These features included: the upper salary offer of the post, the lower salary, the company rating, the company's state data scientist employment number, the ratio of job posting in state to the amount of preexisting data science roles, the annual mean and median wage of professionals in the company's state, and the ratio of the job posting salary to the annual mean and median wage. The region of posting (west, east, south, midwest, remote) was used as the label for classification. In the final model, the region classification was also linked to the most cities hiring within the region, prevalent companies, and job titles. With this kind of data, and the models that were made, I was able to see where I should apply to work based on my desired salary range and my perceived competitiveness (a number from .5 - 1.5 reflecting years of experience, degree, connections, etc.) in the job market.

The data revealed very little correlation amongst the numerical features described above. This lack of correlation amongst the features was fine because the model was then chosen to become a classifier, where the numeric features of a job post will receive a label. If this was a question of regression, then this data would have been very poor in constructing that kind of model, but with classification the data is appropriate.

## Preprocessing and Model Training

As far as the actual machine learning model, several models were examined: Random Forest classifier (RF), K Nearest Neighbor classifier (KNN), Gradient Boosting classifier (GB), and a Deep Neural Network classifier. Please refer to the summary below on the statistics:

| **Training Model Results** | | | | |
|---|---|---|---|---|
| | Models | | | |
| | Random_Forest | K_Nearest | Gradient_Boosted | Deep Neural Network |
| accuracy | 0.9565217391 | 0.8260869565 | 0.9710144928 | 0.2753623188 |
| precision | 0.9612121212 | 0.8309178744 | 0.9723320158 | 0.2615200336 |
| recall | 0.9565217391 | 0.8260869565 | 0.9710144928 | 0.2028985507 |
| f1 | 0.9572763727 | 0.8202731246 | 0.9712174725 | 0.2341322712 |

From this summary, you can see that the Random Forest Classification and Gradient Boosting were the top performers with the classification in regards to all of the performance metrics: accuracy, precision, recall, and f1 score. This may be due to the style of learning. In random forest classification a set number of decision trees is bagged together and weighted in order to produce one final tree; this diversity of logical decision being averaged together most likely explains why the precision (true positive ratio to the sum of true positives and false negatives) and recall (true positive ratio to the sum of the true positives and false positives) were so high for this algorithm. And the gradient boosted classification may have also performed very well because it was able to forgive outliers in the labeled groups as it trained solely on error. In possible confirmation of this outlier significance in learning, the K Nearest Neighbor, performed worse in this case, which arouses outlier suspicion. Because K Nearest Neighbors works by clustering proximity in high dimensional data spaces, if there are outliers then they will be harder to place near clusters of data and label correctly, thus causing data with many outliers to perform worse typically. As an additional measure to ensure that the algorithms were picking up on actual patterns in data, I used a mislabeled dataset to measure performance in the random forest classifier, and it performed very poorly, ensuring that the labels did capture a pattern in the data. It may be that the random forest classifier and gradient boosted classifier are over fit to the data, but if that is the case, then the performance of the less fit models will be interesting in the examination of unseen data.

Before diving into the results, the intention of this model is to help data scientist professionals seeking employment to get the best sites and titles to look for given their desired salary range for their perceived competitiveness. In my case because I live in San Francisco, and I am aware of the median salary for entry level data scientists here based on a site (levels.fyi), I chose to put 110000-130000 dollars for my desired range. However, because I have no degree in computer science and have 2.5 years of experience in a role that involved data analysis and statistics, I put myself at a competitiveness statistic of .8 (the minimum is .5 and the maximum is 1.5). The salary range tells the model to check for numbers that directly match those numbers. The competitiveness ratio translates into the annual mean and median wages that I am looking at: specifically, I am looking for a region where 120,000 is less than the median and mean wages; additionally, the ratio will go into the calculation of the locations opening up the most data science jobs in comparison to the existing amount of data science jobs (lower numbers are relatively easier entry). With this as the input, the model generates the region best suited to my stats.

## Results

My best classifier: the gradient boosted classifier, instructed me that the midwest was the best region for me to apply to.  It provided me with a list of cities including Chicago, Cincinnati, Columbus, Detroit, and Minneapolis.  Along with that list, it gave the top frequency job title posts in the region; some examples of titles that were suggested and fit my skill set were Data Science Analyst, Data scientist,  Data Analyst, and Data engineers.  However, the titles suggested were not in accordance with the competitiveness ratio, rather they were associated with frequency; so the titles proposed may have been for higher incomes and outside of my skill set.  Also, I got recommendations for companies hiring, which included Discover Financial Services, Home Cher, and The Insurance Center.   Since this model had the best in the test performance, it was most likely the best at capturing the patterns in the data.

On the other hand, the results with the random forest classifier K Neighbors, and Deep Neural Network were all the same: remote work.  It is an interesting result because the Random Forest Classifier and K Neighbors performed much better than the neural network, but landed on the same conclusion.  These results, although different from the most confident model, were at least consistent with one another, but it raised the question: what would the results look like for other inputs?  Which I generated below:

| Trial | Input Competitiveness | Input Lower Salary | Input Upper Salary | RFC Result | KNNC Result | GBC Result | DNN Result |
|---|---|---|---|---|---|---|---|
| 1 | 0.5 | 50000 | 80000 | Remote | South | Remote | Remote |
| 2 | 0.6 | 70000 | 90000 | Remote | Midwest | Remote | Remote |
| 3 (personal) | 0.8 | 110000 | 130000 | Remote | Remote | Midwest | Remote |
| 4 | 1 | 100000 | 110000 | None | East | Remote | Remote |
| 5 | 1.1 | 130000 | 170000 | None | South | West | Remote |
| 6 | 1.3 | 180000 | 200000 | West | South | West | Remote |

Firstly, the DNN is clearly biased towards classifying as remote; so that error is good to know as the DNN is picking up only on the uniqueness of the remote work characteristics.  With DNN disregard, We can see that the GBC and RFC, the best performing models were consistent for three of the six trials: the low salary/competitiveness inputs and the high salary/competitiveness input.  However, the GBC and RFC are somewhat red flag raising in how they are limited on outputs, specifically, the GBC mainly output Remote for lower salaries and West for higher salaries, while RFC does the same plus it produced a none classification for those middle of the road salaries.  The KNNC produces 4 out of the 5 classification based on this strata of input, but may contain a bias towards the south as that is 3 of its answers, and those outputs are for the lowest and highest salaries.  From these results, I believe that the each model, minus the DNN, picks up on salary and translates it to a region, however, the respective models are not consistent with each other on how they view salary.  This may be due to the competitiveness ratio, which I experimented with below, holding the salary ranges consistent:

| Trial | Input Competitiveness | Input Lower Salary | Input Upper Salary | RFC Result | KNNC Result | GBC Result |
|---|---|---|---|---|---|---|
| 1 | 0.7 | 50000 | 70000 | Remote | South | Remote |
| 2 | 1 | 50000 | 70000 | Remote | South | Midwest |
| 3 | 1.4 | 50000 | 70000 | Remote | South | Remote |
| 4 | 0.7 | 100000 | 120000 | Remote | South | Remote |
| 5 | 1 | 100000 | 120000 | None | East | Remote |
| 6 | 1.4 | 100000 | 120000 | West | South | West |
| 7 | 0.7 | 150000 | 170000 | None | East | Remote |
| 8 | 1 | 150000 | 170000 | West | West | West |
| 9 | 1.4 | 150000 | 170000 | West | West | West |

Firstly, the RFC and GBC were the same in identification for 6/9 predictions, and all three were consistent in two predictions.  These two predictions were in the high salary range and high competitiveness range, which indicates that all three models have a distinct characterization of the West's economy.  However, the RFC and GBC seem to be biased on Remote and West classifications.  The general trend is that if the salaries are low, regardless of competitiveness, remote work will be the label; conversely, if the salary is high and the competitiveness is high, then the west is the label.  The KNNC again is able to classify the south and the east, which the other models have not yet done, but it understands the south as low salary job posts, rather than the other model's understanding that remote is the low salary income.  These two examination of inputs to outputs reflects the current iteration of the models' abilities to predict, which will be subject to change based on updates in the future.

In the future, I hope to update the models with new training data.  This new data would consist of the future job posts in 2024 and the new BLS report on data scientist professionals in the U.S.   With this, new statistics could be added, such as, yearly growth in total jobs per state or yearly growth in mean/median wages per state.  Inclusion of these metrics would capture more qualities about that state of the data science economy in the listed regions.  Additionally, I would like to make a version of the model, in which there are no remote work jobs, just to see how the model deals with this lack of dimension; the hope is that the model's will be able to identify jobs in the south, east, and midwest more appropriately without misunderstandings caused by the remote work.  With new data and a new classification schema, the model will improve in its applicability.

Currently, the ways that clients could use this model are: find the best cities to work at, best companies to apply for, and inversely what salaries to expect based on location and competitiveness in the job market. The model is naturally designed to take a desired salary range as an input along with the self reflection of your competitiveness in the job market and output cities and companies that are posting jobs in their respective region of the US. However, clients could reverse the function and try a variety of salary expectations, keeping their confidence constant, and see what regions fit with which salary ranges. Maybe if someone felt that they were very confident in their ability to score a senior data scientist role in their local city with 5+ years of experience, they could put their confidence at 1.3, and keep moving around the salary range until their city shows up in the output. This could be useful for those who want to see the average amount of compensation for people of their standings in their economic region. With this machine learning model, data scientists can be more confident in their financial expectations and living situations.

## Conclusion

It is paramount to have a thorough understanding of the economy for those looking to find work. By providing information on job posts, incomes, company ratings, supply and demand of professionals, one is able to incorporate this into predictive machine learning models, from which the machine will learn and guide you based on its data-driven predictions. In this specific instance, I incorporated the upper and lower salaries I am looking for as a burgeoning data scientist, along with a self assigned competitiveness factor; with this information I asked three classifiers where the best place would be to look for work. And from the machine learning models, I found that remote work most consistently described the kind of input I fed the model. However, because I do not want to work remote, I am able to take the midwest prediction over the others. I feel confident going into the job market knowing that I am able to harness the power of data to understand the world and its patterns.