Final Project Report
Chris Coble
7/8/2024

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

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 is a classification one, where the features 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.

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:

	Random_Forest	K_Nearest_Standard_Scaled	K_Nearest_MaxMin_Scaled	Gradient_Booster	Deep Neural Network
accuracy	0.9420289855072460	0.8260869565217390	0.7681159420289860	0.9710144927536230	0.7681159420289860
precision	0.9762845849802370	0.9129049904097160	0.8644157774592560	0.9723320158102770	0.8195131192573650
recall	0.9420289855072460	0.8260869565217390	0.7681159420289860	0.9710144927536230	0.7681159420289860
f1	0.9562586590700350	0.8606429934572720	0.805466998366736	0.9712174724962450	0.7754701964439200

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 (how many true positives to the sum of true positives and false negatives) and recall (how many true positives 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 importance in learning, the MaxMin scaled K Nearest Neighbor, a preprocessing technique notable for messing up data with outliers in it, performed worse in this case; this arouses suspicion that there is a presence of outliers within the individual groupings of economic regions. 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 worst performing model should be underfit to the data and might be useful to compare in the final modeling. In the final modeling step, the random forest classifier, gradient boosting classifier, and deep neural network were all selected to attempt answering the question: where should I apply to work given my desired salary range and how I perceive my competitiveness in the job market?

Before getting 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 was not exclusively data analysis, 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.

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. This prediction came at a 70% probability. This prediction probability was higher than the probability for the other classifications, making it the most confident model.

On the other hand, the results with the random forest classifier and the deep neural network were different, even though the random forest was the second best performer and the deep neural net was the worst performing. Both had suggested that I work remotely, which was one of the labels. The query came up with a different set of suggested companies and titles as well as the model promised. However, both of these models had probabilities that were less convincing than the gradient boosted classifier. The random forest classifier was 54% sure of its answer, with the next highest confidence being 18% for another classification within the same prediction. The neural network, with a classification system based on the argument max function, produced its highest node at 1664, with the next highest node at 67% of its value. These results, although different from the most confident model, were at least consistent with one another. Perhaps their ability to come to the same separate, less confident conclusion reflects a pattern in the data that the gradient boosted classifier has overlooked and overfitted around, while the neural network and random forest cannot overlook.

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.

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.