MathWorks Math Modeling Challenge 2022

Pine View School

Team #15344, Osprey, Florida

Coach: Mark Mattia

Students: Uday Goyat, John Halcomb, Max Rudin, Nolan Boucher, Lisa Zhang

M3 Challenge THIRD PLACE—$10,000 Team Prize
M3 Challenge Technical Computing Award WINNER—$3,000 Team Prize

JUDGE COMMENTS

Specifically for Team # 15344 —Submitted at the Close of Triage Judging:

COMMENT 1: Nice use of statistics and ML techniques. Good statement of assumptions and their justifications. Good discussion on strengths vs weaknesses. Nice use of python for programming.

COMMENT 2: Good executive summary. I would have liked to see more direct results listed in the executive summary. I like the justification in part 1 about the ARIMA model and how polynomial curve fitting gets away very quickly. Did you consider the positives and negatives of ignoring COVID data years? That data sparked the revolution of remote work, so ignoring those would that overly simplify the model? How sensitive is the model? Do large input changes cause large output changes or is the model robust? For question 3, more development is needed of what constitutes a score of "3" vs a score of "1" There are 3 cities listed as 3rd for people moved in 2024 but the 3 cities all had different amounts.

COMMENT 3: Great at presenting your assumptions and justifications. The report would be even better if you included some analysis or your understanding to the tool ARIMA used. In addition, the figures are nice, if it included some explanation in terms of blue vs orange color in the figure, or analysis of your results/figures.

Specifically for Team #15344 —Submitted at the close of Technical Computing Contention Judging:

COMMENT 1: This paper effectively and expertly used technical computing in answering all three sub-questions of the challenge problem. For Q1, an advanced time series modeling technique (ARIMA) was implemented in Python, and coding was used to create plots that beautifully illustrated the model results. For Q2, the team managed to find and process *real survey data* on remote work choices, which allowed them to train a machine learning model (a Support Vector Machine) that could predict likelihood to work from home based on an individual's demographics. This was an approach taken by many top scoring teams this year: this paper's use of machine learning stood out thanks to a clear justification for the model chosen, a high quality training data set, and discussion of potential weaknesses. Finally, the team rounded out Q3 with a well-implemented Monte Carlo model. We were impressed with the paper's code appendix, which was well-commented. The team performed almost all data loading and pre-processing directly in the code, which would allow for the model to be easily adjusted or modified in the future. One weakness of the paper was a lack of discussion to explain parts of the Q3 model, and limited analysis of the machine learning method (e.g. to understand important features). Overall, this paper was a pleasure to read, and it stood out from the rest as this year's Technical Computing winner!

MathWorks Math Modeling Challenge
A program of siam

***Note: This cover sheet has been added by SIAM to identify the winning team after judging was completed. Any identifying information other than team # on a MathWorks Math Modeling Challenge submission is a rules violation.
***Note: This paper underwent a light edit by SIAM staff prior to posting.

M3 Challenge 2022:

Remote Work: Fad or Future

Team #15344

February 27, 2022

Executive Summary

In light of the COVID-19 pandemic, many companies have temporarily or permanently shifted to remote work. This shift represents the acceleration of a long-standing trend of the option to work from home becoming more acceptable over time. While many of the changes brought about by COVID-19 might reverse when the pandemic is over, the previously established trends in remote work will at least remain, if not increase, in pace after COVID-19. Our team will create a model to determine the percentage of the workforces of five sample cities that will be ready for conversion to remote work by 2024 and 2027, a model to predict how individual employers and employees will react to the option of shifting to remote work, and finally, a model that combines the two aforementioned models to determine the true percentage of the workforces of the sample cities that will have converted to remote work by 2024 and 2027.

Our first model predicted the percentage of the workforces of five sample cities that will be ready for conversion to remote work by 2024 and 2027 by creating an autoregressive integrated moving average (ARIMA) model that predicted how the populations and broad-industry allocations of the workforces of the five cities will change by 2024 and 2027. Then the model applied data about the percentage of each industry that is remote-ready to predict the percentage of each city's total workforce that will be remote-ready in 2024 and 2027. Ultimately, the model predicted that remote-readiness across the five cities will increase from 2022 by an average of 0.06% by 2024 and 0.16% by 2027, which is consistent with increases for previous years.

Our second model used a support vector machine classifier (SVM) to predict whether any individual remote-ready employee will work from home or work in person based on demographic data, work satisfaction data, education-level data, corporate-structure data, etc. We divided our data into two different groups to train and test this SVM, creating and confirming the creation of a fairly accurate model. The test data revealed that the SVM successfully assigned the employees to the at-home or in-person classifications 84.84% of the time.

Our final model used a Monte Carlo simulation to determine whether an individual employee will be classified as remote-ready or not based on the specific industry they work in and how likely they are to work in a given specific industry based on which city they live in. Then our model automatically labels employees who are not remote-ready as not working remotely and uses the second model to determine if individual remote-ready employees will or won't work remotely based on the variables used for the second model. The combination of these models yielded percentage values for remote workers in each city. Based on those percentages, our model determined the impact that the changes in remote work would have on the United States and United Kingdom.

Based on our models, our team concluded that the percentage of remote work in the workforce will likely increase in cities across the United States and United Kingdom. Further, this increase will likely impact our societies by decreasing CO_2 emissions in cities with remote workers, decreasing money spent by businesses on office space and utilities, and increasing the rate of migration out of cities of these two countries.

Contents

1 Ready or Not

1.1 Defining the Problem

This question asks us to create a model that determines what percentage of jobs in five given cities will be capable of switching to remote work by 2024 and 2027. Our model will accomplish this by considering the capabilities of various broad industries to switch to remote work, and by considering how large each broad industry is in each of the five cities. Our model will consider how the remote-work capabilities of each broad industry and how the size of each broad industry in each city will change over time to accomplish predictions for 2024 and 2027.

1.2 Assumptions

1. **Workers don't move between these cities and don't change career industries.**

   Justification: Workers moving out of "the city" in general will be considered in later parts of this problem. However, workers moving from one urban area to another will not be considered as an impact on this model. Further, we will assume that people won't change industries based on whether they will be allowed to work from home. Some people tend to prefer work from home and others prefer going into the office. We will assume that people choose their given fields based on factors other than their work-from-home options.[1]

2. **"Workers" includes people who live outside of a city but work inside of a city.**

   Justification: Workers that live outside of a city but commute into it or work remotely in it will be included as workers in this part of the model. Since this question only asks about remote-ready jobs, not jobs that *actually* go remote, we cannot consider the impact that remote jobs will have on people leaving the city. Thus, including such workers in this part of the model will allow us to ignore the effects of people moving out, because they will still be considered "workers."

3. **2020 and 2021 are significant outliers in the data and will not be considered in predicting long-term trends.**

   Justification: The COVID-19 pandemic resulted in drastic changes to employment and remote employment. However, these changes should not be considered permanent, because they were the result of emergency precautions taken to prevent the spread of illness. For long-term trends, it should be assumed that the impact of COVID-19 will wane in the coming years and the trends in employment and remote employment will mostly follow predictions using data from 2019 and earlier.[2]

4. **The effects of the COVID-19 pandemic will be negligible by 2024.**

   Justification: This is a *ceteris paribus* assumption. In other words, COVID-19 will have a drastic and unpredictable effect on the economy and workforce. Therefore, our model must ignore the impacts of COVID-19 to make a reasonable prediction that isn't based on conjecture about the as-yet-unknown long-term impacts of the COVID-19 pandemic. Further, while it is nearly impossible to predict when the COVID-19 pandemic will end, assuming that it ends in the next two years is a feasible and reasonable assumption.

5. **From city to city, the composition of each broad industry is roughly equal.**

   Justification: Within the given data, the workforce of different cities is divided into broad-industry categories. However, the data on the "estimated percentage of jobs that can be done at home by occupation category" divide the workforce up into specific-industry categories. These specific-industry categories can be grouped within the broad-industry categories, but their weights within the broad categories must be determined. Our model will assume that the distribution of jobs among specific-industry categories within each broad-industry category are equal in each of the five industries.

6. **The composition of each broad industry is roughly equal in the United States and the United Kingdom.**

   Justification: The data used to consider the weights of specific industries within broad industries is from the Bureau of Labor Statistics, which collects data for labor in America. There is no data from the United Kingdom that groups specific and broad industries in the same way that the given data does.[4]

7. **The percentage of each specific-industry category that is remote-ready will not change between 2020 and 2024 or 2027.**

   Justification: This is another *ceteris paribus* assumption. While the development of technology will likely facilitate further remote-readiness across every specific industry, it is impossible to predict the impact of technology that has not yet been invented. Therefore, the given data for specific-industry remote-readiness will be used as a source.[2]

1.3 Variables Used

- Percentage of workforce in each city that works in each broad industry
- Percentage of workforce in each broad industry that is composed of each specific industry
- Percentage of workforce in each specific industry this is remote-ready
- Time (in years)

1.4 Developing the Model

Our team chose to use an autoregressive integrated moving average (ARIMA) model to forecast trends in the numbers of employees that work in each industry in each of the five cities—Seattle, Omaha, Scranton, Liverpool, and Barry—listed. ARIMA models are frequently used to forecast financial markets, such as the stock market, and we noticed that the data for many of the industries as well as the total workforces of each city, roughly follows the business cycle, with fewer employees after a recession and more employees near a peak. For instance, the 12-month moving average for Construction jobs in Seattle reached local minimums in 2004 and 2012,[6] shortly after the financial recessions in 2001 and 2009. Further, we noticed seasonality within the data, as many industries tend to have "peak months" and "trough months," such as Leisure and Hospitality in Omaha, which peaks in June and reaches its minimum in January each year.[7] ARIMA models are used for time series with seasonality for interpolation and prediction. Since our data exhibits some annual seasonality, since it is closely related to a field where ARIMA models are already used, and since it is being used to forecast future trends, an ARIMA model is appropriate for our modeling of Question 1.

Our team considered using one of many regression models; however, those were inferior to the ARIMA model. For instance, a linear regression model could fit the data and use it to predict some future trends, but it would be far too simple, missing much of the detail in the business cycle and the annually recurring changes in various industries. A polynomial regression model improves upon the linear regression model, because it can match the many peaks and troughs in the data. However, polynomial regression models have little ability to forecast, finding their primary use in interpolation. When a polynomial regression model is used to extrapolate data, it becomes highly inaccurate very quickly. Considering the alternatives, an ARIMA model was the clear choice.

1.5 Executing the Model

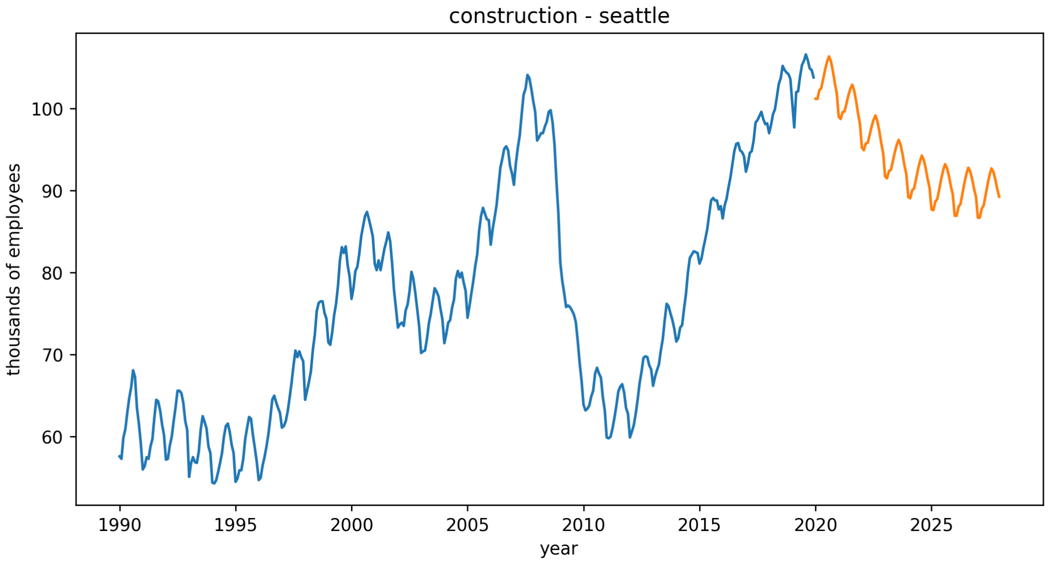
Our team imported the data for each broad industry into Python, then applied the auto-ARIMA function from the library `pmdarima`. The benefit of using auto-ARIMA is that it helps us find the best value for the P, D, and Q constants required by the ARIMA model. After obtaining ARIMA forecasts for each industry in each city up to the end of year 2027, our team summed the data of each industry to create city-wide total predictions. Then we found the mean of the data for January through December 2024 and January through December 2027 for each industry and the city-wide totals to make our predictions of industry and workforce employee counts for both sample years.

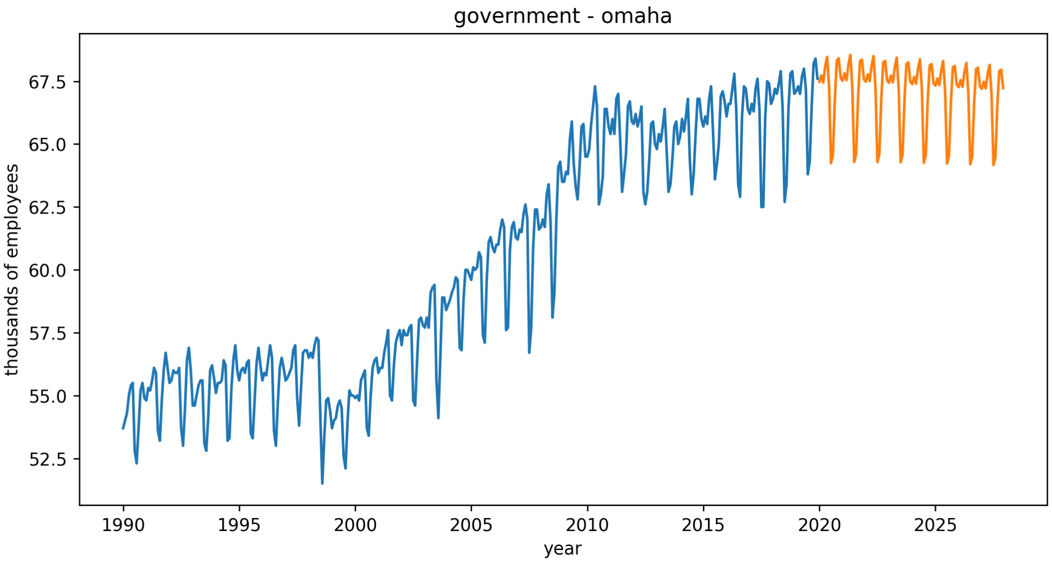
With these predictions, our team multiplied each broad industry by its estimated percentage of jobs that can be done at home, which we found using a weighted average of "estimated percentage of jobs that can be done at home by occupation category" of jobs in each industry;[4] thus, we predicted the total number of remote-ready employees in each industry in each city. Then we summed these industry predictions for each city to find the total remote-ready workforce in each city in the prediction year. Finally, we divided the predicted remote-ready workforce of each city by the total predicted workforce of each city to find the predicted percentage of each city that would be remote-ready in 2024 and 2027. In fact, by averaging each of the twelve months in 2022, 2023, 2025, and 2026, we can create a complete series of annual predictions for the percentage of each city that is remote-ready up to and including 2027.

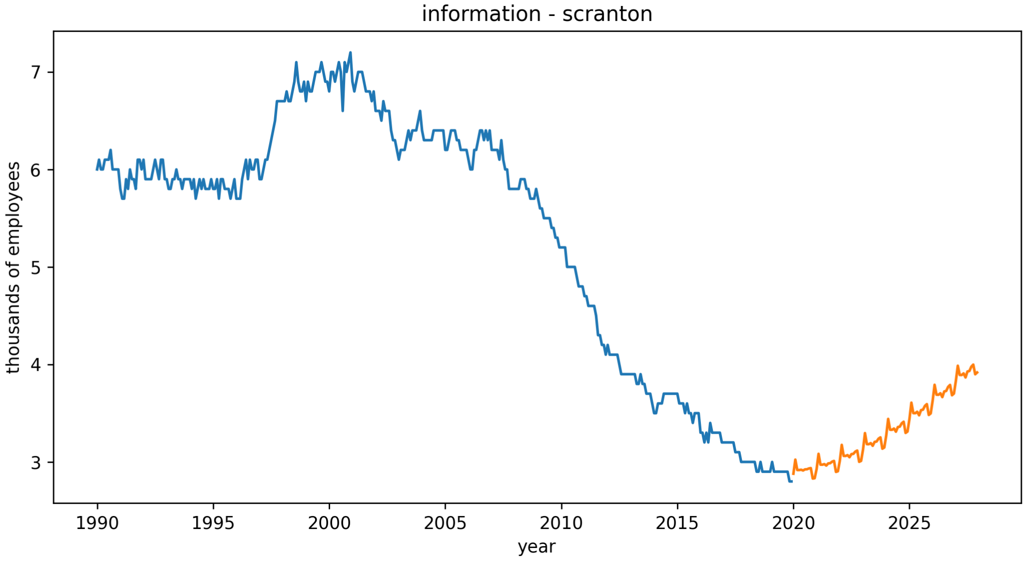
Unfortunately, our data collection was not as successful for Liverpool and Barry as they were for Seattle, Omaha, and Scranton. Therefore, using the given data provided by MathWorks Math Modeling Challenge,[2] we only had four data points per industry per city (because we decided previously to exclude 2020 and 2021 as outliers). For such a limited amount of data, ARIMA could not be used; therefore, we applied linear regressions to each industry in each British city to make our predictions. Then we applied the same process to turn our raw workforce population predictions for British cities into proportions of said cities' workforces that will be remote-ready by year.

1.6 Results and Discussion

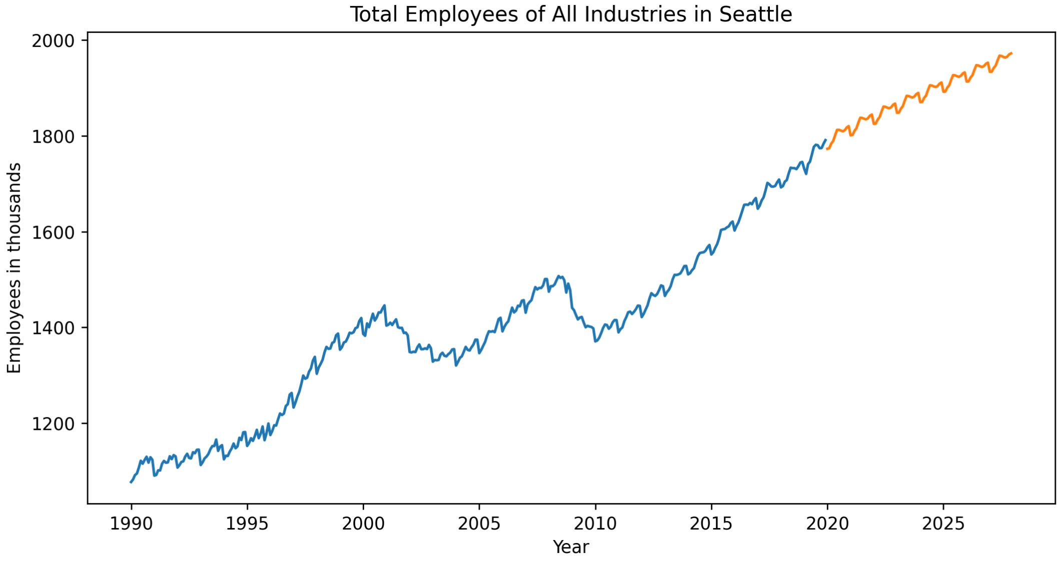
Below are selected graphs for our ARIMA predictions of three different industries in three different American cities.

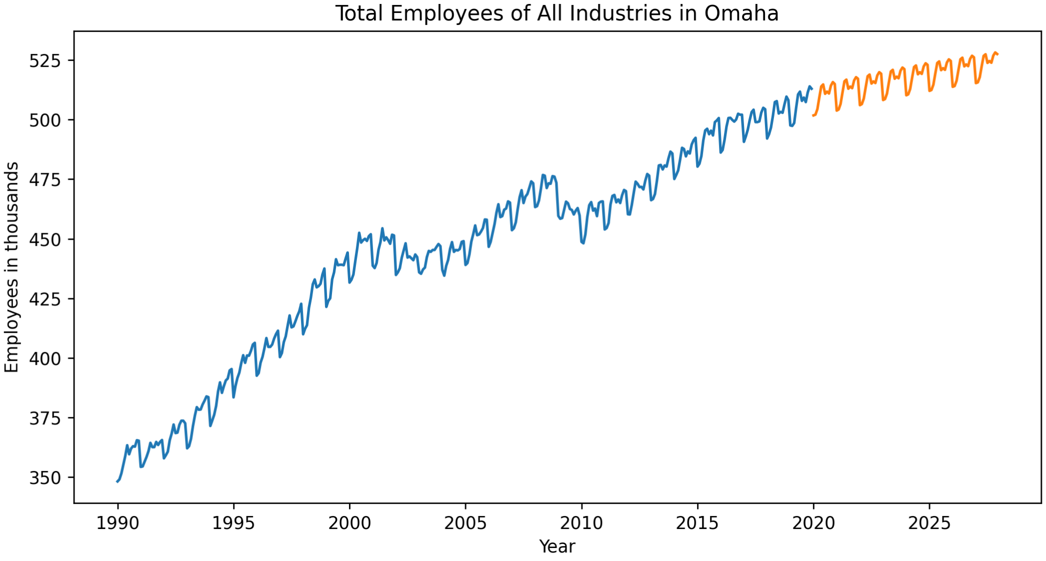


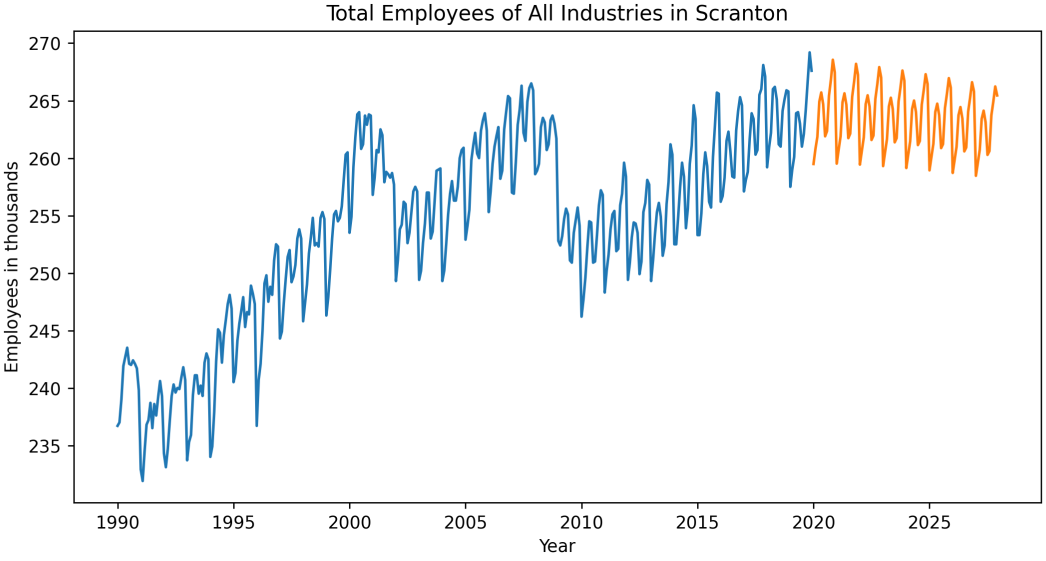




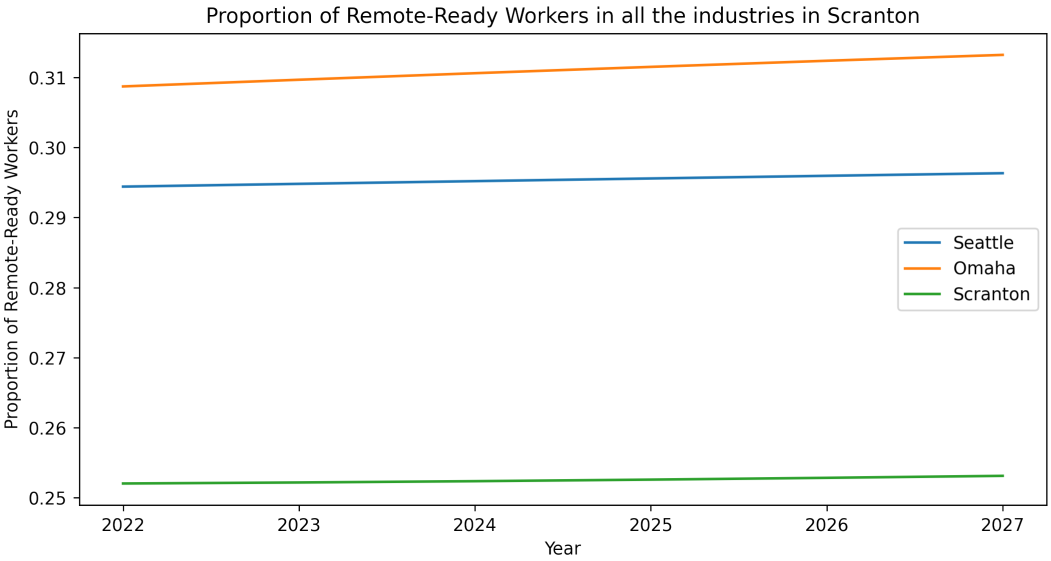
Below are graphs for the total workforce forecasts for Seattle, Omaha, and Scranton.




Total Employees of All Industries in Omaha
Employees in thousands
525
500
475
450
425
400
375
350
1990
1995
2000
2005
2010
2015
2020
2025
Year


Total Employees of All Industries in Scranton
Employees in thousands
270
265
260
255
250
245
240
235
1990
1995
2000
2005
2010
2015
2020
2025
Year

Below is a graph of the proportion of remote-ready workers predicted in the workforces of each American city from 2022 to 2027.



While this graph may appear to show little change over time in each city, there is actually significant change for such a short period, and a table of the data displays this clearly:

Remote-Ready Proportion by City and Year

	Seattle	Omaha	Scranton	Liverpool	Barry
2022	0.29439	0.30868	0.25204	0.20861	0.32509
2023	0.29479	0.30964	0.25218	0.20755	0.32556
2024	0.29517	0.31057	0.25237	0.20651	0.32602
2025	0.29556	0.31148	0.25259	0.20549	0.32647
2026	0.29593	0.31235	0.25285	0.20448	0.32692
2027	0.29631	0.31319	0.25313	0.20349	0.32736

In summary, our model predicts that Seattle will see a 0.26% increase in remote-ready workers from 2022 to 2024, Omaha will see a 0.61% increase over that same period, Scranton will see a 0.13% increase over that period, Liverpool a 1.01% decrease, and Barry a 0.29% increase. Further, our model predicts that Seattle will see a 0.65% increase in remote-ready workers from 2022 to 2027, Omaha will see a 1.46% increase over that same period, Scranton will see a 0.43% increase over that period, Liverpool a 2.45% decrease, and Barry a 0.70% increase.

1.6.1 Strengths

Our team's ARIMA models created many industry graphs that looked like realistic future possibilities. The other, simpler models we considered would all forecast smoother future data that didn't consider the seasonality. ARIMA worked great for our data.

1.6.2 Weaknesses

The ARIMA model requires many data points to make successful predictions. While the given data provided by MathWorks Math Modeling Challenge[2] provides direct links to the data sources for each city on the Bureau of Labor Statistics (BLS) website, it only provides links to the Office for National Statistics' Nomis home page. Therefore, while we were able to find monthly workforce data for each industry in each American city going back to January 1990, we only had the given four data points to work with for each industry in each British city. Therefore, while we were able to create complex ARIMA models that took into account seasonality for each industry in each American city, we could only create simple linear regression models for the British cities.

2 Remote Control

2.1 Defining the Problem

This question asks us to consider how employers and employees in remote-ready jobs will react to this new option. Our model must be able to predict whether individual remote-ready employees will be both willing to work from home and allowed by their employers to work from home.

2.2 Assumptions

1. **Employees in "Computer and Mathematical" industries are representative of the population of employees in remote-ready jobs.**

   Justification: According to the given data, 100% of "Computer and Mathematical" jobs are already remote-ready, so the preferences of employees in this field already consider the remote work option. In other industries, where remote work is fairly new, employees haven't had time to fully adjust to remote work and decide what their preferences about it are.[2]

2. **The opinions of employees in "Computer and Mathematical" industries have not changed since 2019.**

   Justification: Our data source is a survey of employees in "Computer and Mathematical" industries that was taken in 2019, so our model must assume that this data is up to date. While COVID-19 did push more employees to remote work, jobs in "Computer and Mathematical" industries were already remote-ready before the pandemic, so those employees that preferred to work in the office or work from home held those preferences with the option present already. Therefore, while COVID-19 has forced some of these employees to work from home, their preferences likely haven't changed, as they already had time before the COVID-19 pandemic to think about whether they preferred either option.[3]

3. **If an employee is already working from home, then they both chose to work from home and are allowed to work from home by their employer.**

   Justification: If an employee was not allowed to work from home or didn't want to work from home, then they would not be working from home. Since our data asks employees whether they *already* work from home, rather than whether they would like to work from home, the respondents working from home must have chosen to and must have received permission from their employers.[3]

2.3 Variables Used

- Whether an employee has dependants
- Age of employee
- Gender of employee (including "Man," "Woman," "Non-binary," and "Other"

- Size of the company that an employee works for (number of employees)
- Employee's career satisfaction (from "Very satisfied" to "Very dissatisfied")
- Employee's job satisfaction (from "Very satisfied" to "Very dissatisfied")
- Employee's compensation frequency ("Weekly," "Monthly," or "Yearly")
- Employee's education level (from "No education" to "Doctoral")
- Employee's outlook on the world ("People born today will have a better life than their parents" or "People born today will not have a better life than their parents")
- Employee's perceived competency relative to peers (from "Far below average" to "Far above average")

2.4 Developing the Model

The given data provided by MathWorks Math Modeling Challenge[2] lists that "Computation and Mathematical" industry jobs are 100% remote-ready. Therefore, we used a survey of "Computation and Mathematical" industry employees[3] to develop a support vector machine classifier (SVM) to predict whether an employee is willing and permitted to work from home based on the aforementioned variables. This is a machine learning algorithm, so our team initially split the data into "train" and "test" groups so that we could both train the machine and test its efficacy afterward (as explained further in Results and Discussion, our SVM had an accuracy 84.84%).

Essentially, an SVM takes a distribution of data points in $\mathbb{R}^m$, where m is the number of independent variables, and finds the equation of a hyperplane in $\mathbb{R}^m$ that delimits the observations of two classes. In the case of this model, the classes of interest were "remote workers" and "non-remote workers." The optimal hyperplane determined in the training of the model is then applied to a fresh set of testing data; the higher the percent of correctly classified observations, the better the SVM model.

Our decision to use SVM specifically in the creation of a binary regression model was due to the data available to us: SVM models typically perform well for large sets of data, and are good at finding complex relationships among many input variables. The data set used in the training and testing of the model included 17,000 observations, with 10 independent variables. Additionally, when other models such as random forest and multiple logistic regression were used on the same data set, they resulted in a lower sorting accuracy.

A preprocessing step taken in the development of the model was one hot encoding. Identical to the use of "dummy variables" in typical regression, one hot encoding splits a qualitative independent variable with N categories into binary $N-1$ variables. For instance, the encoding of a "gender" variable would require one binary variable: 0 for males and 1 for females, or vice versa.

2.5 Executing the Model

There were approximately 90,000 entries in the Stack Overflow's survey. After filtering and cleaning the data, approximately 17,000 entries remained. Ninety percent of this data was used to train the model, and the other ten percent to test it.

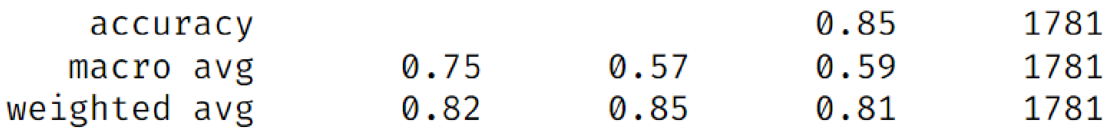
For the preprocessing steps, we first converted the data into a Pandas Dataframe and then scaled the data with Scikit-learn's Standard Scaler for better results.

The Python libraries Scikit-learn, NumPy, and Pandas were used to perform computational transformations on the filtered data. We used Scikit-learn's implementation for the SVM classifer. Additionally, we used all the default values for the fine-tuning parameters for the classifier.

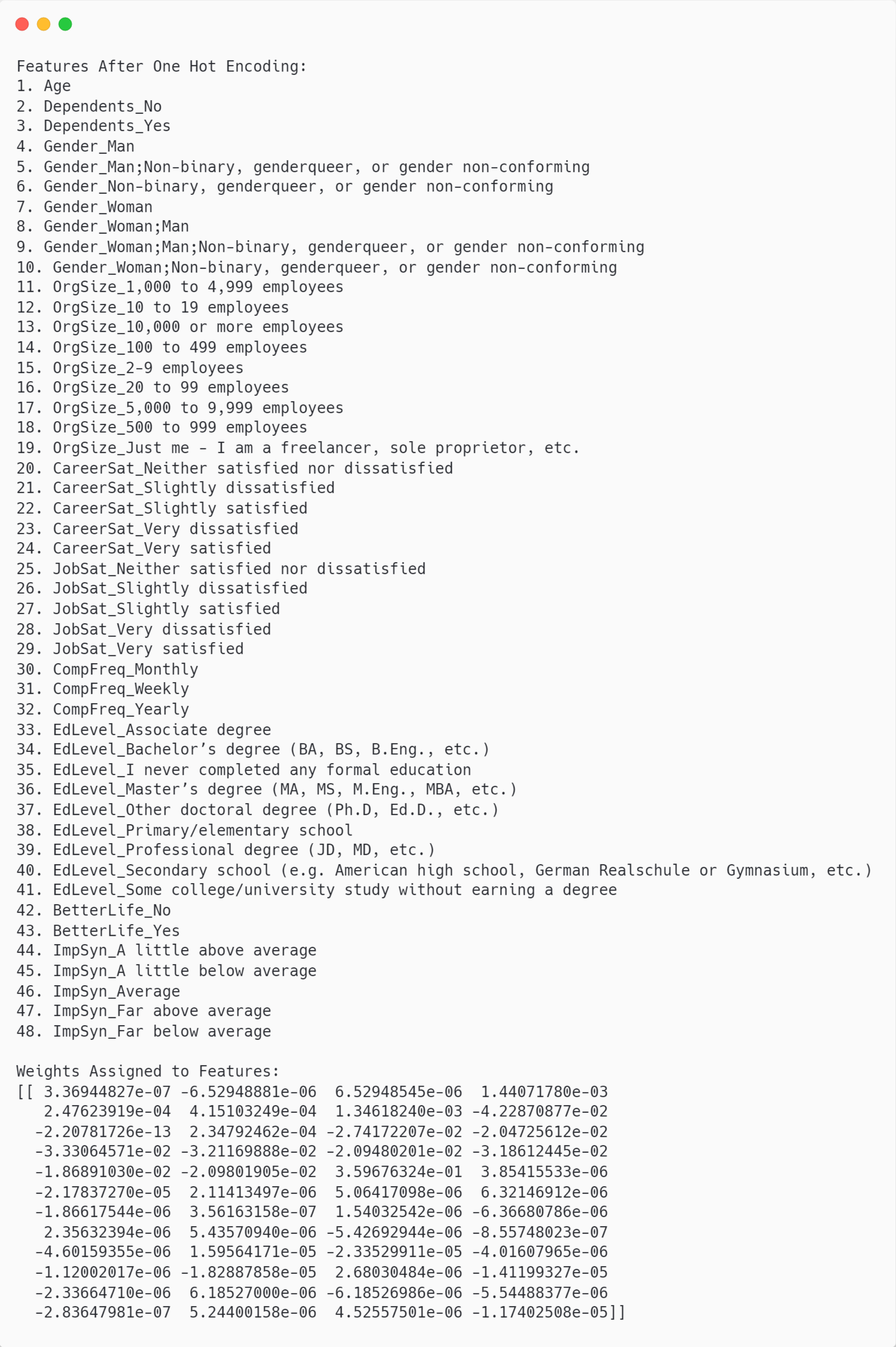
Finally, we applied multiple metrics to measure the performance of the model. The well-performing metrics of the model are provided below.

2.6 Results and Discussion

The SVM successfully predicts whether a remote-ready employee will actually be remote based on the aforementioned data with an accuracy of 84.84%. Additional summary statistics for the final model are included below:



```
    accuracy                           0.85      1781
   macro avg       0.75      0.57      0.59      1781
weighted avg       0.82      0.85      0.81      1781
```



```
Features After One Hot Encoding:
1. Age
2. Dependents_No
3. Dependents_Yes
4. Gender_Man
5. Gender_Man;Non-binary, genderqueer, or gender non-conforming
6. Gender_Non-binary, genderqueer, or gender non-conforming
7. Gender_Woman
8. Gender_Woman;Man
9. Gender_Woman;Man;Non-binary, genderqueer, or gender non-conforming
10. Gender_Woman;Non-binary, genderqueer, or gender non-conforming
11. OrgSize_1,000 to 4,999 employees
12. OrgSize_10 to 19 employees
13. OrgSize_10,000 or more employees
14. OrgSize_100 to 499 employees
15. OrgSize_2-9 employees
16. OrgSize_20 to 99 employees
17. OrgSize_5,000 to 9,999 employees
18. OrgSize_500 to 999 employees
19. OrgSize_Just me - I am a freelancer, sole proprietor, etc.
20. CareerSat_Neither satisfied nor dissatisfied
21. CareerSat_Slightly dissatisfied
22. CareerSat_Slightly satisfied
23. CareerSat_Very dissatisfied
24. CareerSat_Very satisfied
25. JobSat_Neither satisfied nor dissatisfied
26. JobSat_Slightly dissatisfied
27. JobSat_Slightly satisfied
28. JobSat_Very dissatisfied
29. JobSat_Very satisfied
30. CompFreq_Monthly
31. CompFreq_Weekly
32. CompFreq_Yearly
33. EdLevel_Associate degree
34. EdLevel_Bachelor’s degree (BA, BS, B.Eng., etc.)
35. EdLevel_I never completed any formal education
36. EdLevel_Master’s degree (MA, MS, M.Eng., MBA, etc.)
37. EdLevel_Other doctoral degree (Ph.D, Ed.D., etc.)
38. EdLevel_Primary/elementary school
39. EdLevel_Professional degree (JD, MD, etc.)
40. EdLevel_Secondary school (e.g. American high school, German Realschule or Gymnasium, etc.)
41. EdLevel_Some college/university study without earning a degree
42. BetterLife_No
43. BetterLife_Yes
44. ImpSyn_A little above average
45. ImpSyn_A little below average
46. ImpSyn_Average
47. ImpSyn_Far above average
48. ImpSyn_Far below average

Weights Assigned to Features:
[[ 3.36944827e-07 -6.52948881e-06  6.52948545e-06  1.44071780e-03
   2.47623919e-04  4.15103249e-04  1.34618240e-03 -4.22870877e-02
  -2.20781726e-13  2.34792462e-04 -2.74172207e-02 -2.04725612e-02
  -3.33064571e-02 -3.21169888e-02 -2.09480201e-02 -3.18612445e-02
  -1.86891030e-02 -2.09801905e-02  3.59676324e-01  3.85415533e-06
  -2.17837270e-05  2.11413497e-06  5.06417098e-06  6.32146912e-06
  -1.86617544e-06  3.56163158e-07  1.54032542e-06 -6.36680786e-06
   2.35632394e-06  5.43570940e-06 -5.42692944e-06 -8.55748023e-07
  -4.60159355e-06  1.59564171e-05 -2.33529911e-05 -4.01607965e-06
  -1.12002017e-06 -1.82887858e-05  2.68030484e-06 -1.41199327e-05
  -2.33664710e-06  6.18527000e-06 -6.18526986e-06 -5.54488377e-06
  -2.83647981e-07  5.24400158e-06  4.52557501e-06 -1.17402508e-05]]
```

2.6.1 Strengths

Our SVM had real data to train with *and* real data to test its validity Additionally, the code is highly modular. The survey also contained many other questions which reveal more traits about the working environment of the individual. More insightful data about the office environment could offer more impactful data to predict whether or not a worker would choose to be online.

2.6.2 Weaknesses

Our model only used employees in "Computer and Mathematical" industries as a basis for how all remote-ready employees feel in remote work. With an increasing number of jobs becoming remote-ready, the employees in remote-ready jobs will become more diverse, so their opinions about remote-work will become more diverse. Considering this newfound diversity of opinions in our model would have strengthened it, especially considering that many of the newly remote-ready jobs are highly community-oriented and public-facing, while "Computer and Mathematical" jobs are less so.

In order for our model to work, we must have all the input data on an employee to put into the SVM, meaning that employees need to answer all of the questions in the specific survey we used for our model to figure out whether they will be willing and permitted to work from home or not. For some of the survey questions, that is not a great burden, as age and gender data can be found for many cities outside of this specific survey. However, gathering certain data, e.g., an employee's outlook on the world, as determined by the specific phrasing of the survey question, would be difficult for general populations.

3 Just a Little Home-work

3.1 Defining the Problem

This question asks us to combine our two previous models to create a final model that predicts the true percentage of employees that will be working from home in the five given cities in 2024 and 2027. This model will be an augmented version of the first model, which not only predicts the percentage of jobs in a city that are remote-ready but also predicts the percentage of jobs in a city that are *actually* remote. This augmentation will be accomplished using the second model. The question also asks us to evaluate and rank the impact of remote working on the cities.

3.2 Assumptions

1. **Remote working will not impact the rate at which people move *into* "the city."**

   Justification: Remote working allows people to live in a different place from where they work without commuting. While most jobs are in cities, the cost of living in a city is much higher than in a suburban or rural area. Therefore, people who move into a city are likely doing so for reasons other than the newfound freedom that remote working provides them. While remote work will enable a very small number of people to move into cities than would have otherwise, we will consider that impact negligible as compared to the relatively larger impact that remote working has on the rate at which people move *out* of cities.[10]

2. **Where people prefer to live is not affected by where they currently live.**

   Justification: When choosing a place to live, people want to move to the singular best place possible according to their own goals or factors. These goals or factors, such as number of rooms or proximity to public services, are non-comparative. Therefore, a person's current living situation is not factored into their preference of best place to live.

3. **Other than employee age distribution and frequency of dependent household members among a city's workforce, additional factors used in the development of the SVM model do not vary significantly between cities.**

   Justification: Data on statistics such as career satisfaction and perceived relative competency are not readily available for each industry in each city of interest.

3.3 Variables Used

- Additional considered migration out of city caused by remote working
- Additional migration out of city caused by remote working
- Cost of office maintenance for remote and non-remote employees
- CO_2 emissions saved by going remote

3.4 Developing the Model

Our team decided to consider the effect of remote work on the cities through CO_2 emissions, cost of office maintenance, and migration out of cities. CO_2 emissions are an important factor in determining the environmental quality of the city. We considered the cost of office maintenance because switching to remote work would cut a significant amount of cost for businesses, which is an important factor for employers when deciding whether to allow remote working. Remote working has a significant effect on migration out of cities because it encourages more people to move to where they prefer to live, instead of forcing them to live near their workplace.

Our team decided to use a Monte Carlo simulation incorporating the first and second models to predict the percentage of cities' workforces that will actually be remote by 2024 and 2027. In this simulation, a sample of random individuals are created and moved through a decision tree to determine which industry they work in and what kind of person they are according to the variables described in Question 2. Based on these assignments, the second model determines whether the person is likely to work from home or not.

3.5 Executing the Model

Our team used a data set of time spent commuting in each city to calculate the CO_2 emissions saved. We multiplied the time spent commuting for work in each city by the amount of CO_2 emitted in tons per minute.

For our Monte Carlo simulation, we generated a sample of 1000 sample individuals and pushed them through the two decision trees as previously described to determine the total remote workers for each city out of the sample of 1000. Then, from this sample, we determined how many people considered moving, how many actually moved, how much money was saved by not renting office space, and how much CO_2 wasn't emitted.

3.6 Results and Discussion

Below are the results of our Monte Carlo simulations for each city. Note that the sample size of each simulation is 1000 people. Therefore, the categories "People considering moving" and "People who moved" are measured in people per thousand for each city.

Seattle 2024
People considering moving: 46.17
People who moved: 12
Money saved through space: $71250
CO_2 emissions avoided: 11.74 tons
Seattle 2027
People considering moving: 34.02
People who moved: 9
Money saved through space: $52500
CO_2 emissions avoided: 8.65 tons

Scranton 2024
People considering moving: 38.88
People who moved: 10
Money saved through space: $60000
CO_2 emissions avoided: 6.92 tons

Scranton 2027
People considering moving: 70.47
People who moved: 18
Money saved through space: $108750
CO_2 emissions avoided: 12.55 tons

Omaha 2024
People considering moving: 40.5
People who moved: 10
Money saved through space: $62500
CO_2 emissions avoided: 6.98 tons

Omaha 2027
People considering moving: 45.36
People who moved: 12
Money saved through space: $70000
CO_2 emissions avoided: 7.82 tons

Liverpool 2024
People considering moving: 37.26
People who moved: 10
Money saved through space: $57500
CO_2 emissions avoided: 0.13 tons

Liverpool 2027
People considering moving: 34.83
People who moved: 9
Money saved through space: $53750
CO_2 emissions avoided: 0.12 tons

Barry 2024
People considering moving: 51.84
People who moved: 13
Money saved through space: $80000
CO_2 emissions avoided: 0.37 tons

Barry 2027
People considering moving: 55.89
People who moved: 14
Money saved through space: $86250
CO_2 emissions avoided: 0.39 tons

Below is a data table showing the ranking of each city in terms of which is the most and which is the least impacted (most being a 1 and least being a 5) by each metric in 2024.

Impacts Ranked by City in 2024

Category	Seattle	Omaha	Scranton	Liverpool	Barry
People Moved	2	3	3	3	1
Cost of Maintenance	2	3	4	5	1
CO_2 avoided	1	2	3	5	4

Below is a data table showing the ranking of each city in terms of which is the most and which is the least impacted (most being a 1 and least being a 5) by each metric in 2027.

Impacts Ranked by City in 2027

Category	Seattle	Omaha	Scranton	Liverpool	Barry
People Moved	4	3	1	4	2
Cost of Maintenance	5	3	1	4	2
CO_2 avoided	2	3	1	4	5

One thing to note is that the metrics for the people who left the city is per 1000 people. However, the absolute value isn't as important as the relative value. All of these numbers aren't representative of the city's net impact but rather of how they rank with each other.

3.6.1 Strengths

The use of a Monte Carlo simulation for this model makes it a strong predictor as Monte Carlo simulations account for many possibilities and help reduce uncertainty. The code is highly modular because we utilize the output from Questions 1 and 2. Additionally, we utilized probability distribution for the different traits used as the input for the SVM model while sampling for the Monte Carlo Simulation.

3.6.2 Weaknesses

The third assumption made in this model exists primarily due to a lack of data. Individual UK cities do not have or publish data such as city-wide career satisfaction, which hurts our model as such data would make it more accurate. Instead, we used the same patterns of workforce growth for each city, which limits the variability of results per city.

4 References

1. https://www.nytimes.com/2020/05/05/business/pandemic-work-from-home-coronavirus.html

2. Remote Work: Fad or Future, MathWorks Math Modeling Challenge 2022, https://m3challenge.siam.org/node/559

3. https://insights.stackoverflow.com/survey/2019

4. https://www.bls.gov/ooh/about/data-for-occupations-not-covered-in-detail.htm#

5. https://conference.iza.org/conference_files/lmi2005/ngai_r2115.pdf

6. https://www.bls.gov/regions/west/wa_seattle_md.htm

7. https://www.bls.gov/regions/midwest/ne_omaha_msa.htm

8. https://www.constellation.com/solutions/for-your-small-business/small-business-resources/commercial-real-estate.html#:~:text=The%20overall%20operating%20costs%20for,about%20%2417.68%20per%20square%20foot

9. https://poc-system.com/article/seeing-the-big-picture-the-true-cost-of-working-space/

5 Appendix

5.1 Code Used

5.1.1 Question 1

```
    # -*- coding: utf-8 -*-
"""Q1

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1qxk2XRryMF5YxxyyGGTNExau8g6qfU2j
"""

!pip3 install pmdarima statsmodels

# import libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os
from statsmodels.tsa.arima.model import ARIMA
from pmdarima.arima import auto_arima
import pickle

# change directory to where the data is situated
data_dir = '/content'

# data for the previous observations
data = {
    "seattle": {},
    "scranton": {},
    "omaha": {}
}

# in order to store the extrapolation results
data2 = {
    "seattle": {},
    "scranton": {},
    "omaha": {}
}

# read and parse the data
for industry_city in os.listdir(data_dir):
    if industry_city[0] == '.' or os.path.isdir(os.path.join(data_dir,
    industry_city)) or not industry_city.endswith('.txt'):
        continue

    industry = industry_city.split('_')[0]
    city = industry_city.split('_')[1]
    city = city[:city.index('.')]
    cdf = pd.read_csv(os.path.join(data_dir, industry_city))

    data[city][industry] = cdf

# the auto AMIRA code to forecast the results
def forecast_pattern(cdf, city, industry):
    x = []
    y = cdf['Value'].to_numpy()

    for t1 in range(1990, 2020):
        for t2 in range(12):
            x.append(t1 + t2/12)

    x = np.array(x)

    plt.figure(figsize=(10, 5))
    plt.plot(x, y)
    plt.title(f"{industry} - {city}")
    plt.ylabel("thousands of employees")
    plt.xlabel("year")

    model = auto_arima(y,
                       m=12,
                       seasonal=True,
                       stationary=True,
                       trace=True, error_action='ignore', suppress_warnings=True)
    model.fit(y)

    x2 = []

    for t1 in range(2020, 2028):
        for t2 in range(12):
            x2.append(t1 + t2/12)

    x2 = np.array(x2)

    forecast = model.predict(n_periods=len(x2))
    plt.plot(x2, forecast, label='Prediction')
    data2[city][industry] = (np.concatenate((x, x2)), np.concatenate((y, forecast)))
    plt.savefig(f"{industry}_{city}.png", dpi=250)

    plt.show()

for industry, cdf in data["seattle"].items():
  forecast_pattern(cdf, "seattle", industry)

with open('seattle_extrapolation.pickle', 'wb') as handle:
    pickle.dump(data2, handle)

total_y = []

time_size = len(data2['seattle']['mining'][0])
for i in range(time_size):
  ctotal = 0
  for industry, cdata in data2['seattle'].items():
    ctotal += cdata[1][i]
  total_y.append(ctotal)

# the code for combining all industries
cx = np.array(data2['seattle']['mining'][0])
cy = np.array(total_y)

plt.figure(figsize=(10, 5))
plt.plot(cx[cx < 2020], cy[cx < 2020])
plt.plot(cx[cx >= 2020], cy[cx >= 2020])
plt.ylabel("Employees in thousands")
plt.xlabel("Year")
plt.title("Total Employees of All Industries in Seattle")
plt.savefig("total_seattle.png", dpi=250)
```

5.1.2 Question 2

```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn import svm
from sklearn.metrics import classification_report

# used for creating the one hot encodings for the data
def encode_and_bind(original_dataframe, new_dataframe, feature_to_encode):
    dummies = pd.get_dummies(original_dataframe[[feature_to_encode]])
    df = dummies.add_suffix("_" + feature_to_encode)
    res = pd.concat([new_dataframe, dummies], axis=1)

    return(res)

# read the dataset
df = pd.read_csv("data/developer-survey/survey_results_public.csv")

print(df.shape)

countries = ["United States", "United Kingdom"]
df = df[df['Country'].isin(countries)]

# filtered input factosr to consider
inputFactors = [
                'Dependents',
                'Age',
                'Gender',
                'OrgSize',
                'CareerSat',
                'JobSat',
                'CompFreq',
                'EdLevel',
                'ImpSyn',
                'BetterLife'
               ]

# try using work loc instead of WorkRemote
df = df[inputFactors + ["WorkRemote"]].dropna()
X = df["Age"]
y = df["WorkRemote"]

# manually add the chosen factors
X = encode_and_bind(df, X, "Dependents")
X = encode_and_bind(df, X, "Gender")
X = encode_and_bind(df, X, "OrgSize")
X = encode_and_bind(df, X, "CareerSat")
X = encode_and_bind(df, X, "JobSat")
X = encode_and_bind(df, X, "CompFreq")
X = encode_and_bind(df, X, "EdLevel")
X = encode_and_bind(df, X, "BetterLife")
X = encode_and_bind(df, X, "ImpSyn")

fulltime_options = [
    "All or almost all the time (I'm full-time remote)",
    "About half the time"
]

selected_rows = y.isin(fulltime_options)
y.loc[:] = 0
y.loc[selected_rows] = 1

y = y.astype('int')

# for scaling the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, stratify=y, test_size=0.10, random_state=42
)

classifier = svm.SVC(kernel='linear') # Linear Kernel
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))

print(classification_report(y_test, y_pred))
```

5.1.3 Question 3

```
    # Seattle

# 1000 generated sample points for monte carlo
N = 1000

# Seattle - 2024
# the data must be manually changed for each city
industries_distribution = {
    "mining": 0.00048615853578168236,
    "financial": 0.046734283213312494,
    "trade": 0.19452191412137715,
    "manufacturing": 0.09298998083316522,
    "professional": 0.1538692099317581,
    "other": 0.03763897953658071,
    "government": 0.11333016408489403,
    "construction": 0.048309441975530096,
    "information": 0.08161487395898791,
    "leisure": 0.09855043577208027,
    "education": 0.1319545580365324
}

# remote-ready distribution
remote_distribution = {
    "mining": 0.1814994,
    "construction": 0.1814994,
    "manufacturing": 0.01,
    "trade": 0.0237826,
    "information": 0.6413288,
    "financial": 0.88,
    "professional": 0.5828773,
    "education": 0.3765199,
    "leisure": 0.2743978,
    "other": 0.0010478,
    "government": 0.1838977
}

# the constant remote array distribution
remote_array_distribution = [
    0.1814994,
    0.1814994,
    0.01,
    0.0237826,
    0.6413288,
    0.88,
    0.5828773,
    0.3765199,
    0.2743978,
    0.0010478,
    0.1838977
]

# get people sampled in different distributions
people = np.random.choice(11, 1000, p=list(industries_distribution.values()))
remote_ready_people = 0
for person in people:
    if random.random() < remote_array_distribution[person]:
        remote_ready_people += 1


# data for different cities for carbon emissions
"""
Seattle: 0.206 ton
Omaha: 0.1396 ton
Scranton: 0.1442 ton
"""

# manually change the functino with different probability densities for different cities
def generate_features_seattle(N):
    # probability distribution
    p_age = [0.09, 0.09, 0.21, 0.20, 0.13, 0.11, 0.09, 0.05, 0.03]
    p_imp = [0.47281204, 0.23373778, 0.22042467, 0.06150994, 0.01151556]
    p_ed_level = [0.575, 0.189, 0.124, 0.038, 0.035, 0.029, 0.004, 0.003, 0.003]
    p_comp = [0.829, 0.105, 0.066]
    p_job = [0.399, 0.328, 0.133, 0.083, 0.057]
    p_career = [0.517, 0.320, 0.079, 0.054, 0.030]
    p_org_size = [0.206, 0.187, 0.183, 0.127, 0.070, 0.070, 0.066, 0.054, 0.037]


    # combine the features
    features_packed = []
    for i in range(N):
        age = float(np.random.choice(len(p_age), 1, p=p_age)[0] * 10 + 5)
        gender = random.random() > 0.5
        dependents = random.random() < 0.17
        better_life = random.random() < 0.60
        impsyn = np.random.choice(len(p_imp), 1, p=p_imp)[0]
        edlevel = np.random.choice(len(p_ed_level), 1, p=p_ed_level)[0]
        comp = np.random.choice(len(p_comp), 1, p=p_comp)[0]
        job = np.random.choice(len(p_job), 1, p=p_job)[0]
        career = np.random.choice(len(p_career), 1, p=p_career)[0]
        org_size = np.random.choice(len(p_org_size), 1, p=p_org_size)[0]

        org_size_final = [0 for i in range(len(p_org_size))]
        org_size_final[org_size] = 1

        career_final = [0 for i in range(len(p_career))]
        career_final[career] = 1

        job_final = [0 for i in range(len(p_job))]
        job_final[job] = 1

        comp_final = [0 for i in range(len(p_comp))]
        comp_final[comp] = 1

        impsyn_final = [0 for i in range(len(p_imp))]
        impsyn_final[impsyn] = 1

        ed_final = [0 for i in range(len(p_ed_level))]
        ed_final[edlevel] = 1

        cfeatures_packed = [age, int(not dependents),
             int(dependents),
             int(gender),
             int(not gender),
             0,
             0,
             0,
             0,
             0] + org_size_final + career_final + job_final + comp_final + ed_final +
             [int(better_life), int(not better_life)] + impsyn_final

        features_packed.append(cfeatures_packed)

    return classifier.predict(features_packed)


    return features_packed

# calculate the impact on society by the change to remote
# change the constants for different cities manually
CO2 = 0
money_saved = 0

full_remote = 0

for person in generate_features_seattle(remote_ready_people):
    if person == 1:
        CO2 += 0.206
        money_saved += 1250
        full_remote += 1
```

```
print("Seattle 2024")
print("\tPepole considering moving:", round(full_remote * 0.81, 2))
print("\tPeople who moved:", round(0.210 * full_remote))
print("\tMoney saved through space: $" + str(money_saved))
print("\tCO2 emissions avoided:", round(CO2, 2), "tons")
```