

ABSTRACT

RO, YOONCHUL. Multi-State and Individual State Time Series Model Comparison. (Under the direction of Dr. Christopher Healey and Dr. Susan Simmons.)

On December 31, 2019, the Wuhan Municipal Health Commission reported a cluster of pneumonia cases, resulting in the first reported finding of the SARS-CoV-2 novel coronavirus, later renamed COVID-19 by the World Health Organization (WHO) [16]. By late January, South Korea, Hong Kong, and Japan reported their first deaths caused by COVID-19. On March 11, 2020, the WHO labeled COVID-19 a global pandemic. Numerous studies have been conducted to forecast deaths caused by COVID-19. In this thesis, I will also forecast deaths from a statistical data analytics perspective using time series approaches. I focus my research scope on five selected states that exhibit different COVID-19 fatality patterns to investigate how a time series model optimized for an individual state performs compared to a model optimized for multiple states as a whole.

With the advent of the big data era, time series has become a crucial area in data forecasting. However, creating, training, and validating individual models for each dataset requires a significant time investment. In situations like this, it may be convenient to have a single “general” model that provides accurate forecasts for multiple datasets.

My research is focused on: (1) creating predictive COVID-19 fatality time series models for both individual and multiple states, and (2) comparing the performance of the individual and multi-state models based on predictive accuracy. Analysis of variance (ANOVA) identified significant differences in mean absolute error (MAE) for the general model across states, suggesting even a single model can perform statistically differently across states with different COVID-19 fatality patterns. Individual models optimized for specific states outperformed the general model in three cases, and performed equivalently in two cases. This indicates that, in some instances, a general model can rival the performance of a model specialized to a specific domain.

In the future, more in-depth analysis on what determines when an individual model will outperform the general model is needed. This may give insights on what factors affect the spread of a pandemic and what measures may be necessary. Increasing the number of states included in the general model will allow further investigation of its capabilities. Finally, experimenting with lag variables may produce different results. For example, it may be that last week's positivity rate, instead of today's, is a better predictor of today's fatalities.

© Copyright 2021 by Yoonchul Ro

All Rights Reserved

Multi-State and Individual State Time Series Model Comparison

by
Yoonchul Ro

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Computer Science

Raleigh, North Carolina

2021

APPROVED BY:

Dr. Christopher Healey
Co-chair of Advisory Committee

Dr. Susan Simmons
Co-chair of Advisory Committee

Dr. William Enck

DEDICATION

To my family.

BIOGRAPHY

Yoonchul Ro is South Korean born in Vienna, Austria on January 1, 1993. He received his undergraduate degree in Computer Science from Yonsei University (Seoul, South Korea) in Spring 2019. He joined North Carolina State University in Fall 2019 to pursue his Master's degree.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Dr. Christopher Healey and Dr. Susan Simmons for their great support and guidance in the midst of COVID-19 pandemic.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
Chapter 1 INTRODUCTION	1
1.1 Thesis Problem	3
1.2 Proposed Approach	4
1.3 Results	5
1.4 Thesis Organization	5
Chapter 2 BACKGROUND	6
2.1 Related Work	7
2.2 ARIMA Model	7
2.2.1 Autoregressive Models	8
2.2.2 Moving Average Models	8
2.2.3 ARIMA Models	9
2.3 Seasonal ARIMA Model	10
2.4 Exponential Smoothing Model	10
Chapter 3 APPROACH	12
3.1 Sliding Window	13
3.2 Selection of States	14
3.3 Data Preprocessing	15
3.3.1 Model Training	20
Chapter 4 RESULTS	24
4.1 Analysis of Variance	26
4.2 Model Parameters	30
4.3 Prediction Comparisons	30
Chapter 5 CONCLUSION	35
5.1 Future Work	36
BIBLIOGRAPHY	38

LIST OF TABLES

Table 3.1	Dataset columns and descriptions	15
Table 3.1	Continued	16
Table 3.1	Continued	17
Table 3.1	Continued	18
Table 4.1	Average MAEs for forecasted days, l for individual model results, G for generalized model results	25
Table 4.2	Individual SARIMAX model parameters $(p, q, d) \times (P, Q, D)$	28
Table 4.3	Generalized SARIMAX model parameters $(p, q, d) \times (P, Q, D)$	29
Table 4.4	Average fatalities over the 7-day time window: actual fatalities (A), individual predictions (l), and generalized predictions (G)	32

LIST OF FIGURES

Figure 1.1	South Korea annual average temperature	2
Figure 3.1	First Segment	13
Figure 3.2	Second Segment	13
Figure 3.3	Third Segment	13
Figure 3.4	Final Segment	13
Figure 3.5	Map of United States population by state	15
Figure 3.6	Generalized Model Process	23
Figure 4.1	Individual graphs showing actual fatalities (per million) with standard error for each time window forecast	33
Figure 4.2	Individual graphs showing actual fatalities (red), and generalized (orange) and individual (blue) forecasted fatalities per million for each time window forecast	34

CHAPTER

1

INTRODUCTION

This thesis studies the problem of forecasting fatalities caused by COVID-19 from a statistical data analytics perspective using time series approaches for five selected US states. *The goal of this research is to compare the predictive accuracy of a general, multi-state time series model to individual models for each state.*

A time series is a collection of equally spaced data points ordered over time. A classical example of a time series related task is the forecast of future closing stock prices using past daily closing stock prices. Performing a time series analysis consists of model creation and forecasting. The former focuses primarily on analyzing given time series data, and discovering patterns in the data, such as seasonality. For example, the average temperature of South Korea is highest during August as depicted in Figure 1.1. The latter uses information from the past to make a prediction of future activity. Referring back to the stock market

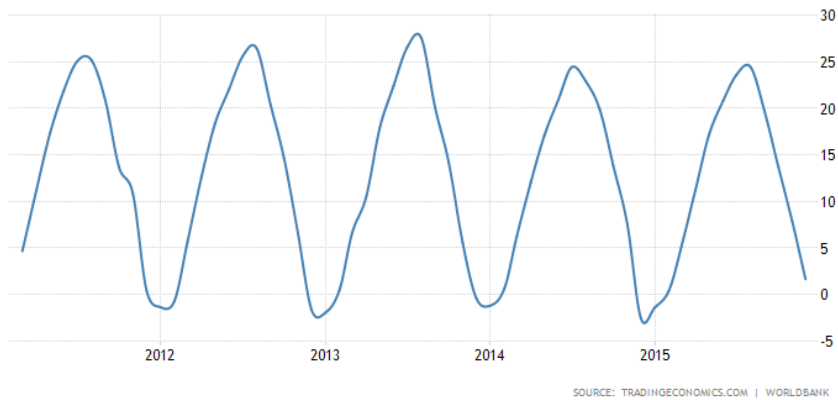


Figure 1.1 South Korea annual average temperature

example, predicting tomorrow’s stock price using six months of past closing stock prices would be a time series forecast.

Time series analysis is important as it allows us to not only detect trends, periodicity, seasonality, cycles, and irregularity, but also make predictions on the future behavior. Time series is widely used in various applications such as economic market analysis and forecasting, earthquake prediction, and more. One caveat in time series is that the data must be dependent on time (for example, daily, weekly, or monthly) and this time interval must be consistent throughout the entire dataset.

In this thesis, I perform time series analysis on COVID-19 datasets and forecast fatalities for five US states with different fatality patterns and volumes. The questions I investigate are the following:

1. How well does a time series model forecast fatalities for the five individual states?
2. How well does a generalized time series model built on the combined data for the five states forecast fatalities for each individual state?
3. How well does a generalized model for an individual state perform compared to a model optimized for the given state?
4. Can we explain the performance of the generalized time series model?

My hypothesis is that individual models will perform at least as well as, or better than, the generalized model. It seems intuitive that using only data from an individual state will produce a more accurate prediction model for that specific state, as opposed to combining data from other states. To give an example, it is in agreement to our intuition to use only the data for California when forecasting its fatality rate, rather than combining the data for California and South Dakota.

1.1 Thesis Problem

Predicting fatalities in a global pandemic crisis is useful in many ways. Such forecasts provide insights on best and worst outcomes in the near future. Additionally, they help policymakers make wise decisions in controlling the pandemic. They provide individuals with situational awareness, giving them time to prepare for the situation. A large body of research has been conducted to attempt to predict upcoming fatalities caused by COVID-19 as accurately as possible. Details are included in the Background section of this thesis. Given these existing models, one may question the purpose and usability of creating an additional approach. One advantage is that access to multiple models can offer advantageous options over a single model, since multiple models may cover more possible future situations. It is extremely difficult to predict the future when a wide range of factors affect it. Moreover, COVID-19 is an unprecedented crisis. When different models agree in their forecasts, it may indicate a measure of reliability. Alternatively, when different models show different predictions, it does not necessarily mean that they are not useful, but rather provides opportunities for further investigation to try to understand why the forecast measurements differ. We can gain insights from similarities and dissimilarities among the different models, such as which factors may or may not affect the spread of the disease.

1.2 Proposed Approach

My data is collected from a variety of sources, including John Hopkins University Coronavirus Research Center, the United Nations Office for the Coordination of Humanitarian Affairs, Our World In Data, and the COVID Tracking Project. This data is aggregated into a single dataset used throughout the thesis. This dataset contains 10,290 rows and 53 data columns including data pertaining to properties like positive and negative COVID tests (*positive* and *negative*), patients hospitalized (*hospitalizedCurrently*), the quality of the data provided (*dataQualityGrade*), and total tests performed (*totalTestsViral*), among others, from January 22, 2020 to September 3, 2020 for all US states.

Using this data, I selected five US states that are different from one another in their COVID-19 fatality volumes and patterns. The individual time series models created for each of these five states will serve as a baseline for comparison with a general time series model that combines data for all five states. **Seasonal Auto-Regressive Integrated Moving Average** with **eXogenous** regressors (SARIMAX) is used for both sets of models. This is considered the most applicable time series model for this dataset for following reasons:

- Ability to forecast fatalities using previous values and errors,
- Ability to detect seasonality, and
- Ability to support multiple independent variables.

Once the models are trained and created, forecasts are made. Accuracy is calculated versus known daily fatalities for each prediction to measure model performance. I compare the general model with known values, and with the individual models to study whether the general model alone is sufficient to substitute for multiple individual models.

1.3 Results

One may ask the purpose of the generalized time series model as it is intuitive that time series models for individual states should indeed perform better than the generalized time series model. However, this was not necessarily true for the five states we examined. Results show that individual models for certain states outperformed the generalized model in terms of mean absolute error (MAE), as well as performing statistically equal in two of the states. In addition, the generalized model itself was statistically different in performance across the five states. In the future, we want to investigate why only certain individual states are outperforming the generalized model (not all the states, as opposed to our intuition), what input is required, and how well each input acts as a predictive characteristic of pandemic fatalities using a generalized model.

1.4 Thesis Organization

The outline for the remainder of the thesis is as follows. Chapter 2 covers background information and related research papers. Chapter 3 provides a more in-depth explanation of my proposed approach. Chapter 4 discusses results and their analysis. Chapter 5 presents conclusions and additional ideas for future work.

CHAPTER

2

BACKGROUND

When it comes to time series analytics, ARIMA (or its variant) and ETS (Error Trend Seasonality, or exponential smoothing) are the two most commonly used methods. ETS applies a weighted average of past values, assigning the most recent values the most importance [9]. ARIMA has a larger number of requirements, but also provides significant flexibility over ETS. There are other potential options like linear regression. Similar to ETS, ARIMA is generally preferred over linear regression as linear regression only incorporates observed values, while ARIMA can incorporate unobserved values.

2.1 Related Work

Data analytics have been used widely in numerous fields. In the field of epidemiology, John Snow's cholera map of London is the first known case where data analytics was applied to the study of disease [5]. Another prominent example is the Google Flu Trend, where Google used web search data to detect patterns and give forecasts on influenza epidemics [6]. This model proved to be effective, having a mean correlation of 0.90 compared to CDC data in its initial application. However, in subsequent years peak levels were significantly under or overestimated [4].

In relation to the study of COVID-19 using data analytics, [11] takes advantage of 2,512 ensemble forecasts made from April 27 to July 20, having 92–96% of observations falling within the rounded 95% prediction intervals. [17] uses different machine learning techniques including logistic regression, support vector machine, random forest, and extreme gradient boosting (XGBoost) to produce high accuracy with area under the curve or AUC scores of 91%. [3] tackles the task by using long short term memory recurrent neural networks (LSTM RNNs) with an accuracy of 77.9%.

2.2 ARIMA Model

Auto-Regressive Integrated Moving Average (ARIMA) is one of the most widely used approaches when it comes to time series forecasting [9]. Before delving into the ARIMA model, the concept of stationarity and differencing must be covered. A stationary time series is one whose properties do not depend on the time at which the series is observed [9]. For example, a seasonal time series is not stationary as seasonality is related to specific periods of times. Differencing, or calculating the mathematical difference between consecutive data, is a technique to convert non-stationary time series sequences into a stationary ones. This technique is useful as it stabilizes the mean of a time series by removing changes in its

level, and therefore eliminating (or reducing) trend and seasonality [9].

2.2.1 Autoregressive Models

Regression models are a type of statistical modeling technique where a dependent variable is expressed as linear combination of one or more independent variables. In autoregression models, the dependent variable is expressed as a combination of past values of one or more variables. Formally, an autoregressive model with p autoregressive terms ϕ_i can be expressed as

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \epsilon_t \quad (2.1)$$

where c is a constant, $\phi_1 \dots \phi_p$ are parameter values, y_{t-i} is the value of y at time $t - i$, and ϵ_t denotes white noise at time t . Intuitively, an autoregressive model is similar to a regression model except that it uses *lagged* values of y_t as independent variables.

2.2.2 Moving Average Models

Moving average models uses past forecast errors as opposed to past values of the dependent variable (Eq. 2.1) for autoregressive models. Formally, a moving average model with q moving average terms θ_i can be expressed as

$$y_t = \mu + \epsilon_t + \sum_{i=1}^q \theta_i e_{t-i} \quad (2.2)$$

where μ is a expectation of y_t (sometimes zero), $\theta_1 \dots \theta_q$ are the parameters of the model, and e_{t-i} denotes error e at time $t - i$.

2.2.3 ARIMA Models

By combining the two aforementioned models, we obtain the ARIMA model. Formally, the full ARIMA model is as follows.

$$y_t = c + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \theta_j e_{t-j} + \epsilon_t \quad (2.3)$$

where c is a constant, $\sum_{i=1}^p \phi_i y_{t-i}$ is the autoregressive component, $\sum_{j=1}^q \theta_j e_{t-j}$ is the moving average component, and ϵ_t is white noise at time t . If differencing is being applied to produce stationarity, it is assumed the y_{t-i} and y_t represent this result.

Mathematically, Eq. 2.3 is referred to as **ARIMA**(p, d, q), where p is the order of the autoregressive component, q is the order of the moving average component, and d is the degree of differencing. Notice that **ARIMA**($p, 0, 0$) is equivalent to the autoregressive model alone, and **ARIMA**($0, 0, q$) is equivalent to moving average model alone. Also notice that the ARIMA model only contains predictor variables created from the response variable in the form of lags of the response variable and/or past error terms. Since there are multiple explanatory variables in our dataset, it is more applicable to use an **Auto-Regressive Integrated Moving Average with eXogenous variables** (**ARIMAX**) model. Exogenous variables are separate independent variables that are assumed to affect the prediction of y_t . Commonly, the formula for ARIMAX is

$$y_t = c + \beta_t X_t + \sum_{i=1}^p \phi_i y_{t-i} + \sum_{j=1}^q \theta_j e_{t-j} + \epsilon_t \quad (2.4)$$

where X_t is the exogenous variable value at time t and β_t is the coefficient. Calculating optimal values of p , q , and d requires solving a non-linear maximum likelihood estimation. Because of its complexity, most statistical languages offer library functions to automatically estimate ARIMA parameters. For example, `auto.arima` in R or `auto_arima` in Python's `pdarima` library.

2.3 Seasonal ARIMA Model

The ARIMA model described above does not take into account of seasonality. In other words, it is restricted to non-seasonal time series data. Box and Jenkins [13] generalized the ARIMA model by incorporating the concept of seasonality, creating the Seasonal ARIMA (SARIMA) model [1]. SARIMA models have three additional seasonal parameters in addition to the (p, q, d) of the ARIMA model, written as follows:

$$ARIMA(p, d, q) \times (P, D, Q)_m \quad (2.5)$$

Tuple (p, q, d) accounts for the non-seasonal part of the model as discussed above, and $(P, D, Q)_m$ accounts for the seasonal part of the model. The subscript m represents the number of observations per year. For instance, $m = 12$ for monthly data, and $m = 24$ for hourly data. It is a convention to write the seasonal parameters as uppercase letters, and non-seasonal parameters as lowercase letters. The seasonal component of the model is similar to that of the non-seasonal component, but it involves backshifts of the seasonal period [9].

2.4 Exponential Smoothing Model

Unlike ARIMA, exponential smoothing models (ESMs) use exponential smoothing to generate their predictions. Similar to a moving average, exponential smoothing combines values over a time window to produce aggregated estimates at each timestep. Unlike a moving average's equal weights, however, exponential smoothing models use exponentially decreasing weights. Exponential smoothing is relatively simple compared to ARIMA or its variations, as it requires only three inputs:

1. A forecast for the most recent time period.

2. An actual value for the time period.
3. A smoothing constant.

The smoothing constant determines the weight given to the most recent data values. ESMs give more importance to the most recent observations. ESMs have the advantage of not needing to identify a set of “correct” parameters (*e.g.*, the p , d and q used by ARIMA models) and also have fewer constraints, for example, no requirement for the data to be stationary. Despite this, there are also disadvantages to ESMs.

ESMs have an important disadvantage: they are not able to handle exogenous or external attributes that contribute to predicting future values. For example, if *age* were considered important to predicting the likelihood of fatality from COVID-19, an ESM would not be able to include this information, where an ARIMAX model could. Because of this, we chose to use seasonal ARIMAX. This allowed us to incorporate both the seasonal trend and external variables into our model, producing better forecasts. Since we are examining the issue of COVID-19 fatalities using a multivariate times series approach, seasonal ARIMAX is the most appropriate model.

CHAPTER

3

APPROACH

This section describes the detailed approach to achieve the goals described in the Introduction (Chapter 1). To reiterate, we are studying how well a generalized time series model performs in comparison to individual time series models specifically designed for individual states. This comparison is performed by examining mean absolute errors (MAEs) and performing analysis of variance (ANOVA) tests on the MAEs to search for statistically significant differences. ANOVAs are used for two comparisons: one on the MAEs across all states, and another for a state's MAEs from the generalized model versus an individual model¹.

In order to gain meaningful insights from the ANOVA tests, it is necessary to have a sufficient number of MAEs. In other words, multiple forecasts are required. One way to

¹We recognize a t -test could have been used here, but ANOVA with two conditions is equivalent to a t -test, so we use “ANOVA” to simplify terminology usage.

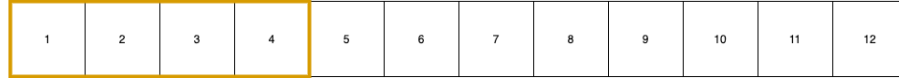


Figure 3.1 First Segment

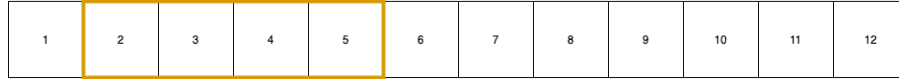


Figure 3.2 Second Segment



Figure 3.3 Third Segment

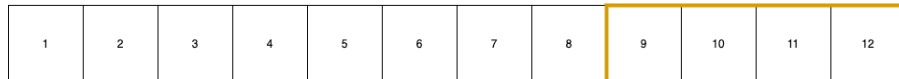


Figure 3.4 Final Segment

achieve this is to train multiple SARIMAX models on different time intervals using the sliding window algorithm.

3.1 Sliding Window

The sliding window algorithm (SWA) is used extensively in time series-related tasks. SWA is a temporal approximation over the values of a segment (or window) of time series values [2]. A single segment S_i is a group of data $S_i = \{d_i, \dots, d_{i+(n-1)}\}$, and the number of data samples n in a segment defines the window size. For each segment, we have a corresponding time series model and its forecasts. Once training the model and generating forecasts is complete, the process is repeated for the next segment S_{i+1} , which starts at d_{i+1} and has length n , $S_{i+1} = \{d_{i+1}, \dots, d_n\}$. This is repeated until the end of the data is reached. This process is visualized in Figures 3.1, 3.2, 3.3 and 3.4 with a window size of $n = 4$. The values d_1, d_2, \dots, d_{12} are the actual time series data (12 days for instance), and the initial segment $S_1 = \{d_1, d_2, d_3, d_4\}$ focuses on the first four observations. A time series model is created based only on these four observations, and is then used to make forecasts. The window

shifts one unit to the right to the second segment $S_2 = \{d_2, d_3, d_4, d_5\}$ portrayed in Figure 3.2. The same process is repeated until the window reaches the final segment as shown in Figure 3.4. The sliding window algorithm is applied in my approach with a window size of 60 days of training data. Using the model created from 60 days, we forecast the next 7 days. The mean absolute error (MAE) is calculated by comparing the forecast values with the actual, known fatality values. MAE is used instead of mean absolute percentage error (MAPE) as this value cannot be calculated due to zero daily deaths for certain days in South Dakota.

3.2 Selection of States

The five states we selected to study are California, Texas, South Dakota, New York, and North Carolina (CA, TX, SD, NY, and NC). These five states were chosen to be “different” from one another, to try to maximize the variability between the generalized and individual time series models. They are spatially distributed throughout the country: California in the west, Texas in the south, South Dakota in the north, New York in the northeast, and North Carolina in the southeast. None of the states share adjacent borders of one another. This is shown in figure 3.5.

The states also have a wide range of populations: 39.5M, 29M, 885K, 19.5M, and 10.5M respectively [14]. California is the most populated state in the US, and South Dakota is one of the least populated [14].

The five states also have different political views. According to US election results for the years 2016 and 2020, South Dakota, Texas, and North Carolina are “red” states as opposed to New York and California, which are “blue” states [7].

There are numerous other differences that can be found across the states. Examples include population density, land size, and percentage of different races that form the states’ populations.

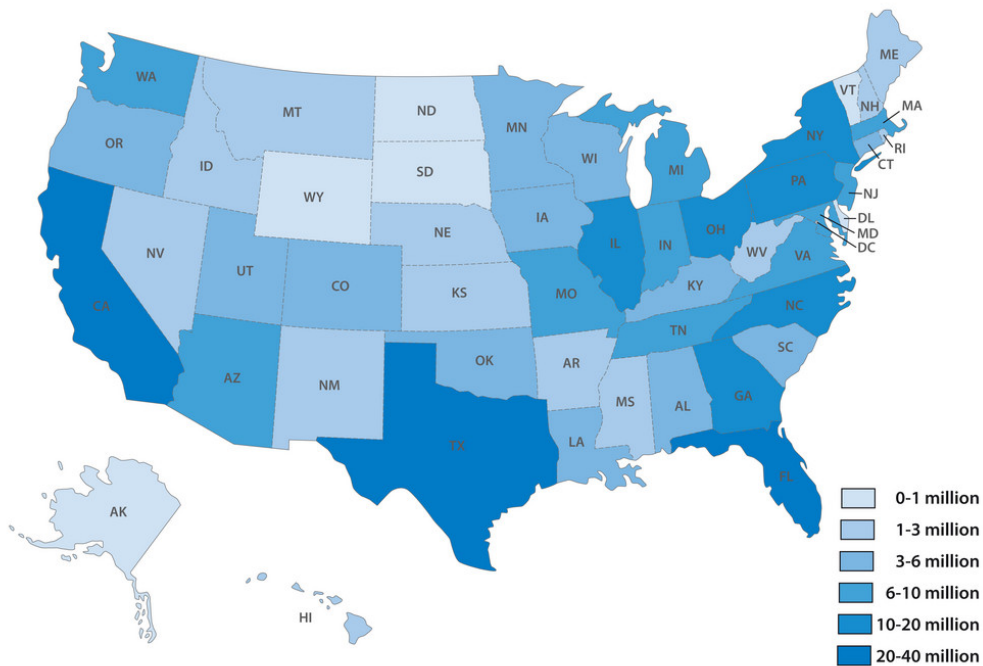


Figure 3.5 Map of United States population by state

3.3 Data Preprocessing

Chapter 1 briefly described the dataset. A full description of the dataset and its columns is shown in Table 3.1.

Table 3.1 Dataset columns and descriptions

Column	Description
Date	Date of data collection
State	Two letter state abbreviation
Positive	Total number of confirmed and probable cases reported
Negative	Total number of unique people with negative PCR test result
Pending	Total number of viral tests not yet completed
hospitalizedCurrently	Total number of people who are hospitalized with COVID-19

Table 3.1 Continued

hospitalizedCumulative	Total number of people who have ever been hospitalized with COVID-19
inIcuCurrently	Total number of people who are currently hospitalized in Intensive Care Unit (ICU) with COVID-19
inIcuCumulative	Total number of people who have ever been hospitalized in ICU with COVID-19
onVentilatorCurrently	Total number of people who are currently hospitalized under advanced ventilation with COVID-19
onVentilatorCumulative	Total number of people who have ever been under advanced ventilation with COVID-19
recovered	Total number of people recovered from COVID-19
dataQualityGrade	Estimated quality of data source
lastUpdateEt	Last updated date and time (in Eastern time)
dateModified	Date data was last modified
checkTimeEt	Date data was last checked, in UNIX epoch time format
death	Total number of fatalities with confirmed or probable COVID-19 case diagnosis
dateChecked	Date data integrity was last verified
totalTestsViral	Total number of PCR tests
positiveTestsViral	Total number of PCR tests that returned positive
negativeTestsViral	Total number of PCR tests that returned negative
positiveCasesViral	Total number of unique people with positive PCR or other approved nucleic acid amplification test (NAAT)
deathConfirmed	Total number of fatalities with confirmed COVID-19 diagnosis
deathProbable	Total number of fatalities with probable COVID-19 case diagnosis
totalTestEncountersViral	Total number of unique encounters tested at least once via PCR testing

Table 3.1 Continued

totalTestsPeopleViral	Total number of unique people tested at least once via PCR testing
totalTestsAntibody	Total number of completed antibody tests
positiveTestsAntibody	Total number of antibody tests that returned positive
negativeTestsAntibody	Total number of antibody tests that returned negative
totalTestsPeopleAntibody	Total number of unique people who have been tested at least once via antibody testing
positiveTestsPeopleAntibody	Total number of unique people with completed antibody tests that returned positive
negativeTestsPeopleAntibody	Total number of unique people with completed antibody tests that returned negative
totalTestsPeopleAntigen	Total number of unique people who have been tested at least once via antigen testing
positiveTestsPeopleAntigen	Total number of unique people with antigen test that returned positive
totalTestsAntigen	Total number of completed antigen tests
positiveTestsAntigen	Total number of antigen tests that returned positive
fips	US Census FIPS geolocation code
positiveIncrease	Daily increase in <i>positive</i>
negativeIncrease	Daily increase in <i>negative</i>
total	Total daily increase, positiveIncrease + negativeIncrease
totalTestResultsSource	Total viral tests run
totalTestResults	Addition of <i>positive</i> and <i>negative</i>
totalTestResultsIncrease	Daily increase in <i>totalTestResults</i>
posNeg	Number of presumptive positive results that have yet to be confirmed
deathIncrease	Daily increase in death
hospitalizedIncrease	Daily increase in <i>hospitalizedCumulative</i>

Table 3.1 Continued

hash	Hash to validate data values
commercialScore	Data quality score, recommended to use dataQuality-Grade in place of this field
negativeRegularScore	Data quality score, recommended to use dataQuality-Grade in place of this field
negativeScore	Data quality score, recommended to use dataQuality-Grade in place of this field
positiveScore	Data quality score, recommended to use dataQuality-Grade in place of this field
score	Data quality score, recommended to use dataQuality-Grade in place of this field
grade	Data quality score, recommended to use dataQuality-Grade in place of this field

Each row is uniquely identified by the *(date, state)* columns. In other words, there may be multiple rows that have **2020-06-08** as the date, but there is only one row that contains **2020-06-08** as the date and **TX** as the state.

We use rows from **2020-06-08** to **2020-09-03** for the five aforementioned states. These are the last 86 rows of the dataset. We are using the last 86 rows instead of the first 86 because the first few months of the pandemic had numerous missing values or repeated values from prior days. The data seemed highly inaccurate and we believe that it is not a clear representation of the COVID situation.

After selecting the necessary rows, a similar process is done for the columns. Although the original dataset contains 53 columns, many of them have repeated information or missing values, some for all rows. For example, *total* is simply the summation of *positive* and *negative*, which is equivalent to *totalTestResults*, *posNeg*, and *totalTestsViral*. All the rows contained 0 for *commercialScore*, *negativeRegularScore*, *negativeScore*, *positiveScore*, and *score*, and the entire column of *grade* is left blank.

After compressing the dataset, the result is 86 rows and 8 columns for each state: *date*, *state*, *positive*, *negative*, *hospitalizedCurrently*, *inIcuCurrently*, *recovered*, and *death*. The *date* and *state* columns are used as a key for accessing target rows and are ignored during the time series model creation and training. The data is sorted in chronological order, estimates missing values by either replacing empty values with the column minimum, if present, or 0. The data is then converted to daily data by taking the difference between adjacent values.

For the generalized model, a new dataset is created by summing over all columns for the five states. A new column *positivityRate* is created using *positive* and *negative* based on Equation 3.1. Positivity rate is a preferred metric to the number of positive and negative tests as the former indicates the level of testing relative to the size of the outbreak [10]. The remaining columns are converted to units of *per million* by dividing by the state population for each state and multiplying by one million.

$$positivityRate = \frac{positive}{positive + negative} * 100 \quad (3.1)$$

The last preprocessing step is handling multicollinearity. Multicollinearity occurs when similar information is provided by two or more predictor variables during multiple regression [9]. One way to measure multicollinearity is the variance inflation factor (VIF). As the name suggests, VIF measures how much a given variable's variance is increased in the model. It is calculated using Equation 3.2. R_j^2 is the R^2 value of the regression on the j^{th} independent variable versus other independent variable(s) [9]. In general, a value of greater than 10 indicates problematic multicollinearity [9]. If such a situation arises, the independent variable with the highest value is removed and VIF is recalculated. This process is repeated until all the VIF values are less than or equal to ten. The VIF was used to define the predictor variables in our model.

$$VIF_j = \frac{1}{1 - R_j^2} \quad (3.2)$$

3.3.1 Model Training

We use an SARIMAX time series model as described in the last section of Chapter 2. This is done in Python using the *auto_arima* method from the *pmдарima* [12] Python library. *auto_arima* is implemented so that it behaves similar to R's well-known *auto.arima* [12]. The SARIMAX model can be created in various ways by changing different parameters for *auto_arima*. For instance, the minimum and maximum values for p and q (non-seasonal parameters of SARIMAX) can be set manually. Similar parameter choices can be made for the minimum and maximum values for P and Q (seasonal parameters of SARIMAX) as well. In our models, the maximum values for the non-seasonal SARIMAX parameters are set to 10, and the maximum values for the seasonal SARIMAX parameters are set to the algorithm's default in [12]. The maximum value of $p + q + P + Q$ (*max_order*) is set to 25.

There are two main approaches to finding the optimal parameter values for SARIMAX or any variant of ARIMA. The first is grid search. This method tries every possible combination of (p, d, q) and (P, D, Q) (if seasonality is considered) and returns a single set with the lowest information criterion. The Akaike Information Criterion (AIC) is generally used as the information criterion, but the Bayesian Information Criterion (BIC), Hannan–Quinn information criterion (HQIC), or Out-of-Bag (OOB) are also available. One disadvantage to this method is that it takes a considerably amount of time. The second method is a step-wise algorithm developed by Hyndman and Khandakar [8] to address this disadvantage. This greatly reduces running time to find the optimal parameters for both the seasonal and non-seasonal SARIMAX models.

Unfortunately, there may be cases where the model fails to converge to a set of valid parameters using either of the two aforementioned search methods. This produces model parameters of $(0, 0, 0)$, which is equivalent to white noise. If such a situation arises, there are two potential solutions. The first is to increase the number of iterations, hoping a valid solution can be found. The second is to set the parameter *random* of *auto_arima* to

True. This configures the ARIMA model to randomly search the hyper-parameter space n_fits times, choosing the most optimal parameters found. Using the *random* option and increasing the number of n_fits is the preferred choice to addressing automatic parameter selection results of (0, 0, 0).

Once the optimal parameters are found and our model has been trained, it makes a forecast of estimated fatalities for the next seven days using the *predict* method. It then calculates the mean average error (MAE) between the forecast and known values. The final step is performing an analysis of variance using the *f_oneway* method to search for statistically significant differences. This is explained in detail in Chapter 4.

Note that this entire model training process represents a single iteration. This is repeated nineteen more times using different data selected with our sliding window approach. Multiple trials are required to produce meaningful ANOVA results. The entire process for the generalized model is visualized in Figure 3.6.

The description given up to this point relate to generalized models, but the algorithm for the individual models is almost identical. The high-level description of the individual model process is given in Algorithm 1.

Algorithm 1 Individual Model

```
1: Preprocess dataset
2: states = ['NC', 'TX', 'CA', 'SD', 'NY']
3: for currentState in states do
4:     Sort the data for currentState in chronological order
5:     Obtain the last 86 rows for currentState
6:     Handle missing values
7:     Create positivityRate column
8:     Convert the units of remaining columns to per millions
9:     Handle multicollinearity
10:    for iteration = 1, 2, ..., 20 do
11:        Obtain data for iteration-th window
12:        Train SARIMAX model
13:        Make 7 day fatality forecasts
14:        Calculate MAE
15:    end for
16: end for
```

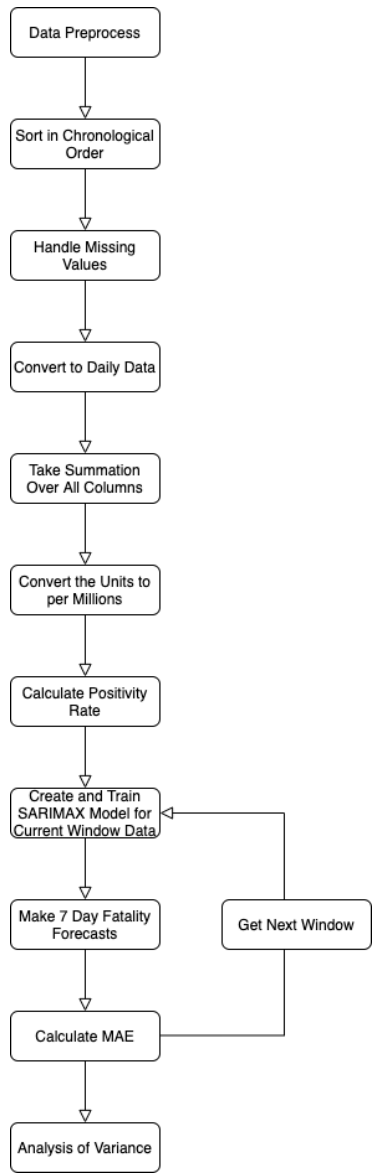


Figure 3.6 Generalized Model Process

CHAPTER

4

RESULTS

As discussed in our Approach Section, we tested both individual models customized for each state, and a single general model built on data aggregated across all five states. Table 4.1 compares the results for individual state models versus the generalized model. Table 4.2 and table 4.3 shows the SARIMAX parameters for all iterations of both the generalized and individual state models. Finally, Table 4.4 compares the average forecasted fatalities for both the individual and generalized models with the actual, known fatalities. Reported accuracies represent the MAEs between the actual and forecasted fatalities.

Table 4.1 Average MAEs for forecasted days, I for individual model results, G for generalized model results

CA		NC		SD		NY		TX	
I	G	I	G	I	G	I	G	I	G
0.66	0.97	1.20	1.59	1.16	1.41	0.14	0.81	4.28	3.29
0.79	0.73	1.07	2.31	0.88	1.56	0.16	0.85	3.74	4.03
0.87	0.90	1.02	2.04	0.82	1.53	0.16	0.93	3.49	3.16
1.07	1.00	0.62	1.68	1.00	1.36	0.12	1.53	4.01	4.39
0.93	0.98	0.84	1.66	0.96	1.29	0.11	1.53	4.39	2.97
1.16	1.21	0.71	1.42	0.99	1.74	0.14	0.79	3.02	2.92
0.88	0.94	0.79	1.33	0.70	1.56	0.14	0.71	3.10	2.46
0.95	0.85	0.80	1.12	0.97	1.39	0.12	0.40	2.30	2.63
0.83	1.09	0.67	1.41	0.93	1.24	0.11	0.53	2.38	2.73
0.69	1.03	0.62	1.24	1.05	1.53	0.10	0.44	2.34	2.91
0.57	0.90	0.71	1.12	1.11	1.30	0.07	0.69	2.75	3.35
0.57	0.87	0.46	1.12	1.10	1.84	0.11	0.78	2.88	3.29
0.52	0.90	0.38	0.75	0.94	1.22	0.11	0.96	2.80	2.27
0.58	0.80	0.43	0.95	1.38	0.97	0.11	1.00	2.58	2.95
0.52	0.63	0.40	0.83	1.37	1.24	0.11	0.59	2.88	3.14
0.57	0.55	0.43	0.83	1.12	1.38	0.10	0.82	3.21	2.66
0.78	0.72	0.41	1.02	1.43	1.25	0.11	0.66	2.55	3.01
0.55	0.49	0.44	1.05	1.26	1.13	0.15	0.78	2.47	2.29
0.63	0.56	0.53	1.15	1.52	1.04	0.13	0.91	2.57	2.11
0.44	0.51	0.49	1.21	1.27	1.01	0.13	0.83	2.52	2.11

4.1 Analysis of Variance

The method `f_oneway` in Python's **scipy** library performs a one-way analysis of variance (ANOVA) [15]. One-way ANOVA tests are a type of hypothesis test designed to determine statistically whether two or more groups have the same mean, the so called *null hypothesis* H_0 . Assuming there are k groups, this is defined as follows.

$$H_0 : \mu_1 = \mu_2 = \dots = \mu_k \tag{4.1}$$

$$H_1 : \mu_i \neq \mu_j, \quad i, j \in [1, k]$$

Equation 4.2 refers to the statistic that is used to test for the null hypothesis. MST is calculated using equation 4.3 and MSE is calculated using equation 4.4.

$$F = \frac{MST}{MSE} \tag{4.2}$$

$$MST = \frac{\sum_{i=1}^k n_i (\bar{x}_i - \bar{x})^2}{k - 1} \tag{4.3}$$

$$MSE = \frac{\sum_{i=1}^k (n_i - 1) s_i^2}{n - k} \tag{4.4}$$

where n is the combined sample size, $n = \sum_1^k n_i$. Applying the `f_oneway` method with the MAEs as parameters returns a p -value used to determine whether to reject the null hypothesis. Typically, if the p -value is below 0.05, we reject the null hypothesis and conclude that at least one group mean is significantly different from the others. Otherwise we conclude that the sample means are not statistically significantly different.

If we compare the MAEs across the five states, $F(4, 95) = 110.207$, $p < 0.001$, indicating a statistically significant difference between the states' results. A Tukey HSD (honestly

significant difference) post-hoc test shows all pairs of states have a significant difference in their generalized model MAEs, $p < 0.05$, except for (CA, NY) and (NC, SD), with $p = 0.9$ for both pairs.

Next, we need to determine whether the individual model for a particular state outperforms the general model. To do this, we can conduct five separate ANOVAs for each state. Since there are only two conditions being tested: individual versus general; this is equivalent to a t -test.

Results for these tests were $F(1, 38) = 37.293, p < 0.001$, $F(1, 38) = 12.556, p = 0.001$, $F(1, 38) = 2.727, p = 0.107$, $F(1, 38) = 119.137, p < 0.001$, and $F(1, 38) = 0.169, p = 0.683$ for the states of North Carolina, South Dakota, California, New York, and Texas, respectively. This shows that the individual models and the general model were different for three of the five states. For the states of California and Texas, performance was statistically significantly equivalent. For the states of North Carolina, South Dakota, and New York, the individual model outperformed the general model. This is interesting, since as we discussed previously, intuition might suggest that an individual model customized for a particular state's properties should be better than a generalized model. This was not the case for two of the five states we examined.

Similar analysis can be done with the actual and forecasted values produced from the generalized and the individual model. Conducting five separate ANOVAs on these three results for each state gives us $F(2, 58) = 23.475, p < 0.001$, $F(2, 58) = 62.757, p < 0.001$, $F(2, 58) = 100.799, p < 0.001$, $F(2, 58) = 1.753, p = 0.18$, and $F(2, 58) = 11.020, p < 0.001$ for the states of California, North Carolina, South Dakota, New York, and Texas respectively. This shows that the actual and forecasted values (from both the generalized and the individual models) showed statistically significant differences except for the state of New York. Performing a Tukey HSD post-hoc test on all pairs of results (actual, individual), (actual, forecasted), and (individual, forecasted) identified statistically significant differences.

Table 4.2 Individual SARIMAX model parameters $(p, q, d) \times (P, Q, D)$

CA	NC	SD	NY	TX
$(0, 1, 1) \times (2, 0, 0)$	$(0, 0, 2) \times (1, 0, 1)$	$(3, 0, 1) \times (1, 0, 0)$	$(0, 0, 0) \times (2, 0, 1)$	$(2, 0, 1) \times (0, 0, 0)$
$(0, 1, 1) \times (2, 0, 0)$	$(1, 0, 0) \times (0, 0, 2)$	$(0, 0, 3) \times (0, 0, 1)$	$(0, 0, 6) \times (0, 0, 2)$	$(0, 1, 1) \times (0, 0, 1)$
$(0, 1, 1) \times (2, 0, 0)$	$(0, 0, 3) \times (0, 0, 1)$	$(2, 0, 2) \times (0, 0, 0)$	$(2, 0, 6) \times (0, 0, 0)$	$(0, 0, 0) \times (0, 0, 0)$
$(1, 1, 1) \times (1, 0, 2)$	$(2, 0, 0) \times (2, 0, 2)$	$(2, 0, 2) \times (0, 0, 0)$	$(1, 0, 0) \times (0, 0, 2)$	$(1, 0, 1) \times (0, 0, 0)$
$(0, 1, 1) \times (1, 0, 1)$	$(1, 0, 0) \times (1, 0, 1)$	$(2, 0, 1) \times (2, 0, 0)$	$(2, 0, 2) \times (1, 0, 0)$	$(1, 0, 1) \times (0, 0, 0)$
$(1, 1, 1) \times (1, 0, 2)$	$(2, 0, 0) \times (1, 0, 0)$	$(1, 0, 3) \times (0, 0, 0)$	$(0, 0, 6) \times (1, 0, 0)$	$(0, 1, 1) \times (0, 0, 0)$
$(0, 1, 1) \times (1, 0, 1)$	$(2, 0, 0) \times (1, 0, 1)$	$(1, 0, 2) \times (0, 0, 0)$	$(0, 0, 1) \times (2, 0, 1)$	$(0, 1, 1) \times (0, 0, 2)$
$(0, 1, 1) \times (1, 0, 1)$	$(2, 0, 1) \times (1, 0, 1)$	$(3, 0, 1) \times (1, 0, 0)$	$(0, 0, 2) \times (0, 0, 2)$	$(1, 1, 1) \times (0, 0, 1)$
$(0, 1, 1) \times (1, 0, 1)$	$(1, 0, 0) \times (2, 0, 1)$	$(1, 0, 1) \times (0, 0, 0)$	$(0, 0, 0) \times (1, 0, 2)$	$(1, 1, 1) \times (0, 0, 1)$
$(3, 1, 1) \times (1, 0, 1)$	$(2, 0, 0) \times (1, 0, 1)$	$(3, 0, 2) \times (0, 0, 0)$	$(0, 0, 1) \times (1, 0, 0)$	$(1, 0, 0) \times (1, 0, 0)$
$(2, 1, 1) \times (1, 0, 1)$	$(0, 0, 2) \times (2, 0, 0)$	$(3, 0, 1) \times (0, 0, 0)$	$(0, 0, 2) \times (0, 0, 2)$	$(1, 0, 0) \times (0, 0, 1)$
$(2, 1, 1) \times (1, 0, 1)$	$(1, 0, 0) \times (1, 0, 2)$	$(0, 0, 1) \times (0, 0, 0)$	$(0, 0, 1) \times (1, 0, 0)$	$(2, 0, 2) \times (2, 0, 0)$
$(0, 1, 1) \times (1, 0, 1)$	$(3, 0, 0) \times (0, 0, 5)$	$(0, 0, 0) \times (2, 0, 0)$	$(1, 0, 0) \times (2, 0, 0)$	$(0, 0, 0) \times (1, 0, 0)$
$(0, 1, 1) \times (1, 0, 1)$	$(1, 0, 1) \times (2, 0, 5)$	$(0, 0, 1) \times (0, 0, 0)$	$(1, 0, 0) \times (0, 0, 1)$	$(0, 1, 1) \times (0, 0, 2)$
$(0, 1, 1) \times (1, 0, 1)$	$(0, 0, 2) \times (3, 0, 3)$	$(0, 0, 0) \times (0, 0, 0)$	$(0, 0, 6) \times (0, 0, 0)$	$(0, 1, 1) \times (0, 0, 1)$
$(0, 1, 1) \times (1, 0, 1)$	$(1, 0, 0) \times (1, 0, 3)$	$(0, 0, 1) \times (0, 0, 0)$	$(1, 0, 0) \times (2, 0, 0)$	$(0, 1, 1) \times (2, 0, 0)$
$(0, 1, 1) \times (1, 0, 1)$	$(0, 0, 1) \times (2, 0, 1)$	$(3, 0, 3) \times (0, 0, 0)$	$(2, 0, 0) \times (0, 0, 0)$	$(1, 1, 1) \times (1, 0, 1)$
$(0, 1, 1) \times (1, 0, 1)$	$(0, 0, 1) \times (1, 0, 1)$	$(1, 0, 4) \times (1, 0, 1)$	$(1, 0, 1) \times (0, 0, 0)$	$(0, 1, 1) \times (0, 0, 2)$
$(0, 1, 1) \times (1, 0, 1)$	$(0, 0, 2) \times (2, 0, 0)$	$(4, 0, 0) \times (0, 0, 0)$	$(1, 0, 1) \times (0, 0, 0)$	$(1, 0, 2) \times (0, 0, 1)$
$(0, 1, 1) \times (1, 0, 1)$	$(1, 0, 0) \times (2, 0, 0)$	$(0, 0, 6) \times (1, 0, 1)$	$(1, 0, 0) \times (2, 0, 2)$	$(0, 1, 2) \times (0, 0, 1)$

Table 4.3 Generalized SARIMAX model parameters $(p, q, d) \times (P, Q, D)$

n	Generalized
1	$(0, 0, 4) \times (0, 0, 0)$
2	$(0, 0, 1) \times (0, 0, 1)$
3	$(0, 0, 2) \times (0, 0, 0)$
4	$(3, 0, 1) \times (0, 0, 0)$
5	$(0, 0, 0) \times (0, 0, 0)$
6	$(1, 0, 3) \times (0, 0, 2)$
7	$(4, 0, 0) \times (0, 0, 0)$
8	$(4, 0, 1) \times (0, 0, 1)$
9	$(4, 0, 0) \times (0, 0, 1)$
10	$(3, 0, 2) \times (0, 0, 0)$
11	$(4, 0, 0) \times (0, 0, 1)$
12	$(1, 0, 3) \times (1, 0, 0)$
13	$(3, 0, 2) \times (0, 0, 0)$
14	$(0, 0, 2) \times (2, 0, 0)$
15	$(2, 0, 0) \times (1, 0, 0)$
16	$(2, 0, 2) \times (0, 0, 1)$
17	$(1, 0, 1) \times (1, 0, 0)$
18	$(1, 0, 0) \times (1, 0, 1)$
19	$(2, 0, 0) \times (1, 0, 1)$
20	$(2, 0, 2) \times (0, 0, 1)$

1. North Carolina's actual and individual values were statistically equivalent, $p = 0.254$.
2. In South Dakota, all pairs were statistically significantly different.
3. In both California and Texas, actual and generalized values were statistically equivalent, $p = 0.362$ and $p = 0.084$, respectively.

4.2 Model Parameters

Table 4.2 shows the (p, q, d) and (P, Q, D) triples used for each individual SARIMAX model, for each state. Since we ran the models over twenty time windows, there are twenty sets of parameters for each state. Similarly, Table 4.3 shows the twenty $(p, q, d) \times (P, Q, D)$ parameters for the twenty iterations of the generalized model.

4.3 Prediction Comparisons

Our final examination looked at the individual and generalized models' forecasted fatalities, versus the actual, known fatalities that occurred over each time window. Table 4.4 shows three columns for each state: actual fatalities (A), the individual models' forecasted fatalities (I), and the general model's forecasted fatalities (G). Each row represents one of the twenty time windows. Figure 4.2 shows the same data as a set of five state graphs with three lines: actual fatalities in red, individual predictions in blue, and general predictions in orange. Since these values are the mean over the 7 day window, it has the effect of smoothing out the irregular bumps that may be present in the original series. Figure 4.1 illustrates the mean absolute error for both the generalized and individual for each states at each time point, where the actual fatalities standard error are shown transparent in the background.

Results for all three models are similar except for Texas, which also exhibited the highest fatality rates. Further study is needed, but we are particularly interested in why results varied in the manner they did for Texas. One possibility is that data reported by Texas

is erroneous or ambiguous in some way. Another possibility is that the models are less accurate for larger fatality rates. This might make sense for the general model, but it seems unlikely for Texas's individual model.

Table 4.4 Average fatalities over the 7-day time window: actual fatalities (A), individual predictions (I), and generalized predictions (G)

CA			NC			SD			NY			TX		
A	I	G	A	I	G	A	I	G	A	I	G	A	I	G
3.56	3.92	3.13	2.44	2.66	1.45	0.97	1.51	-0.44	0.31	0.30	-0.51	7.42	8.53	7.42
3.46	3.94	3.91	2.49	2.50	0.94	0.97	1.74	-0.59	0.36	0.27	-0.49	7.38	11.01	7.40
3.37	3.86	3.39	2.44	2.52	1.18	1.13	1.82	-0.38	0.35	0.37	-0.58	7.51	5.41	6.90
3.19	3.89	3.32	2.40	2.36	1.18	1.13	1.99	0.63	0.38	0.29	-1.15	7.61	4.22	10.02
3.16	3.83	3.33	2.62	2.43	1.35	1.29	2.16	0.74	0.39	0.32	-1.14	7.59	3.85	8.45
3.16	3.88	3.30	2.48	2.33	1.45	1.29	2.29	-0.44	0.38	0.39	-0.41	7.51	5.70	5.95
3.17	3.81	2.93	2.42	2.30	1.71	1.45	2.00	-0.10	0.35	0.36	-0.28	7.41	9.47	5.99
2.98	3.93	3.05	2.47	2.07	1.83	1.44	2.19	0.06	0.34	0.30	0.24	7.13	9.33	5.46
3.04	3.88	2.88	2.42	2.19	1.77	1.30	1.91	0.06	0.28	0.37	0.03	7.03	9.24	5.44
3.29	3.86	3.29	2.51	2.11	1.40	1.29	2.12	-0.24	0.29	0.35	-0.01	6.83	8.94	5.04
3.29	3.67	3.11	2.55	1.87	1.75	1.29	2.22	-0.01	0.28	0.31	0.19	6.71	6.52	5.32
3.31	3.70	2.93	2.37	2.44	1.28	1.13	2.19	-0.41	0.24	0.33	-0.53	6.53	9.13	5.30
3.20	3.35	2.87	2.38	2.00	2.09	1.13	1.83	0.86	0.26	0.33	0.48	6.14	8.94	5.25
3.12	3.33	3.60	2.25	2.44	2.01	0.81	2.19	0.70	0.25	0.32	0.95	6.29	8.82	4.44
3.14	3.24	3.34	2.15	2.11	1.78	0.97	1.82	0.34	0.25	0.29	0.69	5.99	8.83	4.33
3.06	3.31	3.29	2.21	2.11	1.91	1.13	1.59	0.51	0.27	0.32	0.83	5.69	8.75	4.12
2.79	3.57	3.37	2.19	2.19	1.82	0.97	1.49	0.36	0.29	0.33	0.87	5.62	7.90	4.05
2.82	3.37	3.26	2.27	2.29	1.89	0.98	1.10	0.64	0.24	0.33	0.91	5.62	7.73	3.63
2.75	3.38	3.30	2.33	2.24	1.95	0.98	1.72	0.70	0.25	0.31	1.01	5.44	7.62	3.54
2.73	3.18	3.17	2.36	2.27	1.87	1.13	1.10	0.95	0.23	0.35	0.95	5.24	7.31	3.40

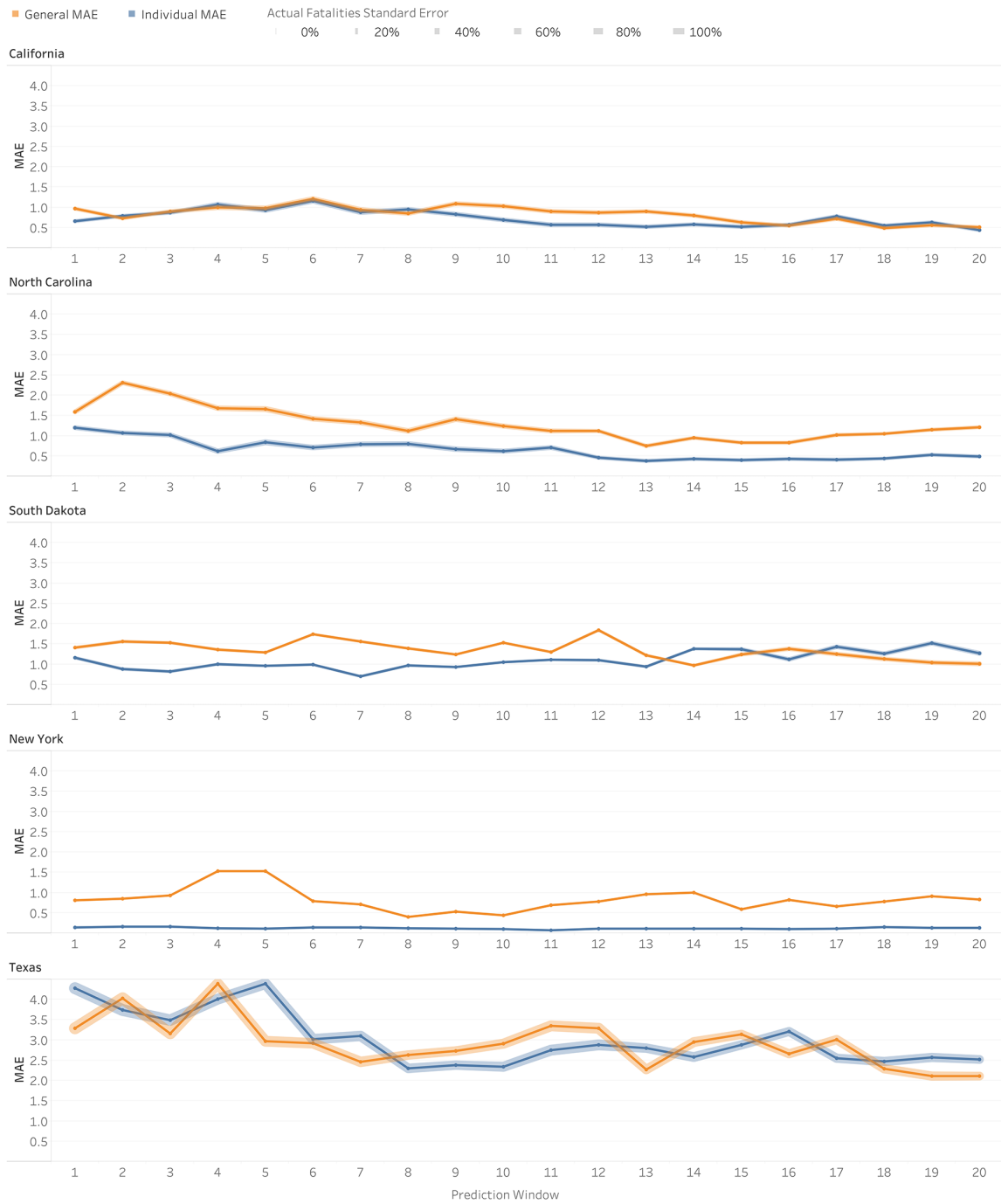


Figure 4.1 Individual graphs showing actual fatalities (per million) with standard error for each time window forecast

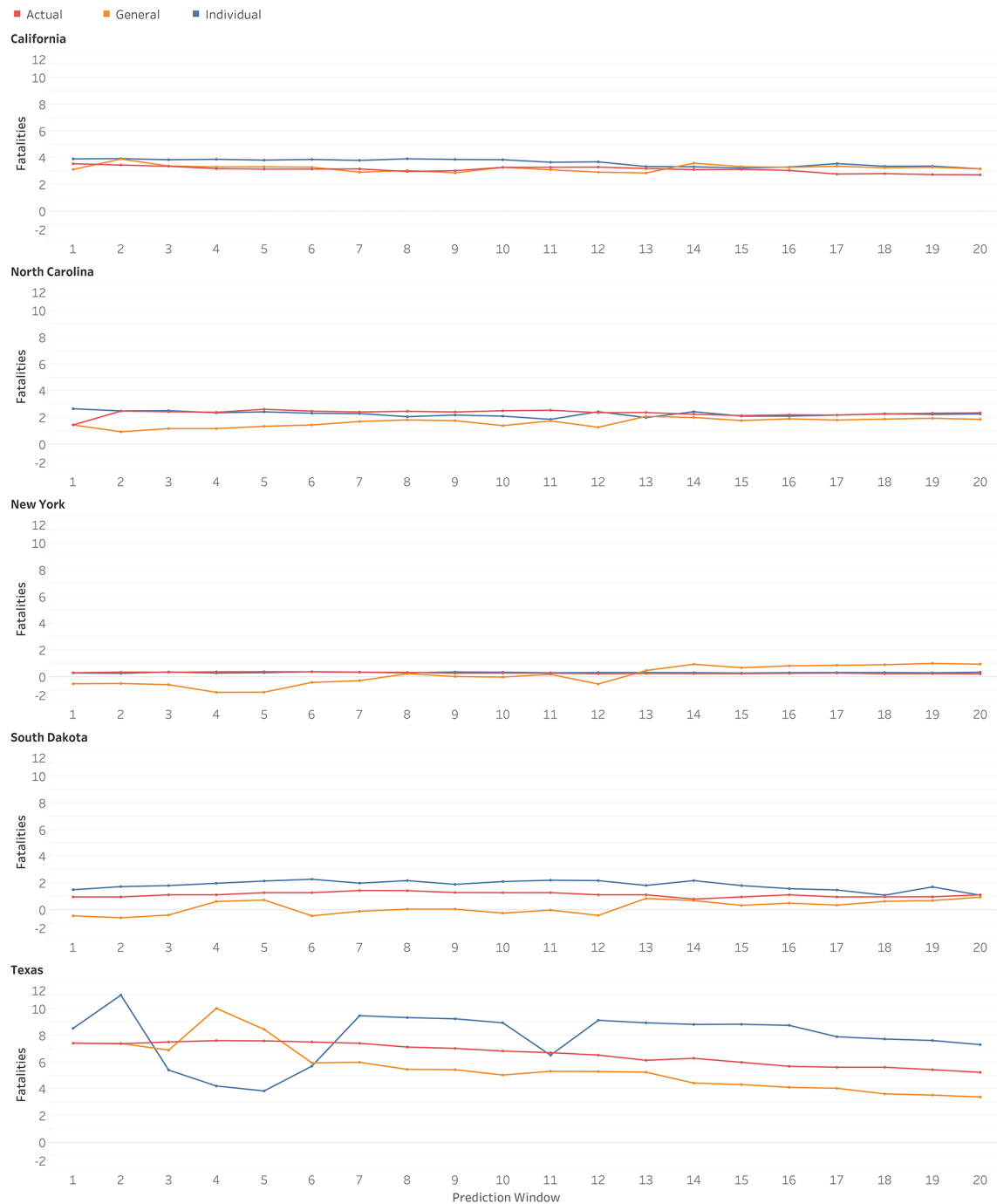


Figure 4.2 Individual graphs showing actual fatalities (red), and generalized (orange) and individual (blue) forecasted fatalities per million for each time window forecast

CHAPTER

5

CONCLUSION

This paper compares the performance of a generalized time series model to individual models for five separate states in an attempt to predict COVID-19 fatalities. The states we selected were included due to their differences in properties with one another: geographic location, population, political party in power, and so on. We first compared mean average errors or MAEs across all five states using the generalized model. This allowed us to assess the performance of a single general model for different states. Analysis of variance results showed the model performed statistically differently across the five states. Only in two cases did the model perform statistically equivalently: California and New York, and North Carolina and South Dakota. All other pairs had significant Tukey HSD values with $p < 0.05$.

We next constructed individual prediction models for each state based only on that state's properties. MAE results from the individual model were compared to the general

model's MAEs, to determine if a model specialized for a specific state outperformed a general model. Although intuitively one might expect a specialized model to outperform a generalized model, results were mixed: for the states of California and Texas, the individual model was statistically equivalent to the generalized model. However, for the states of North Carolina, South Dakota, and New York, the individual model outperformed the general model. This suggests that, in certain situations, a generalized model is as good as a specialized model.

Although there was statistically equivalent performance (based on MAEs) for only two out of the five states we examined, this is a promising sign that a generalized model can be a potential substitute for individual models. One avenue for further investigation is to first cluster states based on the similarity of their fatality curves, then produce a generalized model for each cluster. Given clusters of reasonable size, this would require only a relatively small set of generalized models that have the potential to produce results equivalent to individual, per-state models.

Another promising results was the finding that actual values versus individual models values for the state of North Carolina and actual versus generalized model predictions for the states of California and Texas were statistically equivalent. This shows that accurate predictions are being produced.

5.1 Future Work

In our time series models, we have not experimented with lags of different variables. For example, it may be the case that last week's positivity rate, instead of today's, is a better predictor of today's fatalities. In addition, we have fixed 60 days for training and 7 days for forecasting for both the generalized and the individual models. Experimenting with different combinations of training and forecast sizes may provide more insights into the limits and potential of the models' capabilities.

As noted above, another potential area for future work includes a more in-depth analysis on when an individual model will or will not outperform a generalized model. This may provide a better understand of how each variable acts as a predictor of fatalities and spread during a pandemic.

BIBLIOGRAPHY

- [1] Adhikari, R. & Agrawal, R. “An Introductory Study on Time Series Modeling and Forecasting”. *ArXiv abs/1302.6613* (2013).
- [2] Benyahmed, Y. et al. “Adaptive sliding window algorithm for weather data segmentation”. *Journal of theoretical and applied information technology* **80** (2015), pp. 322–333.
- [3] Bodapati, S. et al. “COVID-19 Time Series Forecasting of Daily Cases, Deaths Caused and Recovered Cases using Long Short Term Memory Networks”. *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. 2020, pp. 525–530.
- [4] Butler, D. “When Google Got Flu Wrong”. *Nature* **494**.7436 (2013), pp. 155+.
- [5] Disease Control, C. for & Prevention. “Section 2: Historical Evolution of Epidemiology” (2012).
- [6] Ginsberg, J. et al. “Detecting influenza epidemics using search engine query data”. *Nature* **457** (2009). doi:10.1038/nature07634, pp. 1012–1014.
- [7] Hall, M. & Gal, S. *How the 2020 election results compare to 2016, in 9 maps and charts*. 2020.
- [8] Hyndman, R. & Khandakar, Y. “Automatic Time Series Forecasting: The forecast Package for R”. *Journal of Statistical Software* **26** (2008).
- [9] Hyndman, R. & Athanasopoulos, G. *Forecasting: Principles and Practice*. English. 2nd. Australia: OTexts, 2018.
- [10] Max Roser Hannah Ritchie, E. O.-O. & Hasell, J. “Coronavirus Pandemic (COVID-19)”. *Our World in Data* (2020).
- [11] Ray, E. L. et al. “Ensemble Forecasts of Coronavirus Disease 2019 (COVID-19) in the U.S.” *medRxiv* (2020). eprint: <https://www.medrxiv.org/content/early/2020/08/22/2020.08.19.20177493.full.pdf>.
- [12] Smith, T. G. *1. About the Project*. 2017-2020.
- [13] Tunnicliffe Wilson, G. “Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1”. *Journal of Time Series Analysis* **37** (2016), n/a–n/a.
- [14] U.S. Census Bureau. *U.S. and World Population Clock*. 2021.

- [15] Virtanen, P. et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. *Nature Methods* **17** (2020), pp. 261–272.
- [16] World Health Organization. “Listings of WHO’s response to COVID-19” (2020).
- [17] Yadaw, A. S. et al. “Clinical features of COVID-19 mortality: development and validation of a clinical prediction model”. *The Lancet Digital Health* **2.10** (2020), e516–e525.