Epidemiology Modeling: Predicting COVID-19 Subvariant Spikes

By: Neha Ravi, Nghi Van Pan (Tyler), and Rachel Azarian

Introduction (Nghi Van Pan (Tyler))

COVID-19 was declared a global pandemic on March 10th, 2020. The US was greatly affected by the pandemic and is still experiencing new cases. In total, the US has had over 96 million cases and over one million deaths. Even after vaccines were introduced, the US still experienced a record number of covid cases. The spike in the number of positive cases is a major concern, because communities often lack the resources to adequately test, quarantine, and treat their affected residents.

The Omicron variant was the latest and most transmissible subvariant to cause a spike in cases, which happened throughout December 2021 to February 2022. This occurred after vaccines were available for almost a year. Since the virus mutates into different variants and subvariants, it is a challenge to predict whether the spike in cases will happen and at what rate the spike will occur. When a new variant emerges there are unknowns regarding its transmissibility, vaccine efficacy, and lethality. Variant spikes not only occur due to the genetics of the variant, but also due to other factors like tourism, mask mandates, and potentially demographic features of a region.

In order to efficiently allocate resources to prepare for subvariant spikes and mitigate cases, there is an interest in the specific features that impact the speed of a subvariant's transmissibility. For our project, we are interested in predicting the number of weeks it will take for a covid subvariant to reach its peak spike from the week that it first arrived in a region. We plan to model this using two data sources, state-level and county-level data, in order to see if COVID-19 spikes are better categorized at a more local or broad level. The state data will model COVID-19 subvariant spikes across all 50 states plus Washington DC to help determine if certain demographic and population features are significant for understanding the spread of COVID-19. The county-level data will focus on the counties of the top 10 highest agricultural producing states. Also, we will determine if certain features pertaining to these counties are significant for understanding the spread of COVID-19. Agricultural counties were selected since future work on our project involves incorporating human trafficking with COVID-19 data, and there is a relationship between rural regions and human trafficking.

Background (Nghi Van Pan (Tyler))

The project was conducted in collaboration with Cameron Feathers, our mentor who is in the field of microbiology, virology, and immunology. She is interested in modeling focused on epidemiology, especially COVID-19 spread in relation to human trafficking. This project just focused on COVID-19 spread, and the results will be used for future research regarding human trafficking. Along with our mentor, there are two advisors, Ryan Maiorano, a Pharmaceutical Data Scientist and Ivan Mera, a Data Optimization Analyst in the technology field.They provided support on the technical side of the project, including model analysis and interpretation. As recommended by the mentor, the primary data for prediction will be at the state level, and then

the models can be further applied to county level data. The counties selected for modeling are counties of the top 10 states with the highest agriculture production. This includes Wisconsin, Minnesota, Nebraska, Indiana, North Carolina, Kansas, Illinois, Texas, California, and Iowa. The data features will be collected based on the literature review and our mentor's suggestions. The features will be evaluated based on their relevance to the target value to determine their impact on COVID-19 subvariant spikes. Some features we plan to explore include political party, population density, number of airports, agriculture production, and population health data.

In order to determine the models that are suitable for the data, exploratory analysis needs to be conducted to identify patterns in the data. Many of the data features do not have a linear correlation to the target value. This provides the fundamental understanding that the best models to apply are supervised learning models including tree-based regression models and Support Vector Regression to predict the discrete target value. For tree-based models, we will utilize Decision Tree Regressor and ensemble learning methods for better prediction like Extra Tree Regressor, Random Forest, and XGBoost. Tree-based regression models work by using binary recursive partitioning to split the data into branches. The partition is determined by minimizing the sum of squared deviations from the mean. This is repeated until terminal nodes are achieved.

Both Extra Trees and Random Forest ensemble methods consist of many decision trees where the prediction is based on the prediction of multiple trees by arithmetic mean. So, they are more robust than Decision Trees. When selecting the points to split the nodes, Extra Trees chooses this randomly while Random Forest uses the optimum split. XGBoost is an ensemble method that utilizes gradient boosting. Trees are added to the ensemble in order to correct the errors of the previous model, known as boosting. As the model is fit, the loss gradient is minimized. We are interested in comparing the results of the different ensemble trees.The final model, Support Vector Regression, makes predictions by creating the best fit line using a hyperplane that maximizes the number of data points within the decision boundary. The kernel of the algorithm can be tuned for non-linear relationships. Using the models, the hyperparameters will be tuned using GridSearchCV and final model performance will be evaluated on the test data. Cross Validation will also be used to validate the models and confirm that they can be applied towards new data. These results will provide insights into determining the best performing model.

Literature Review (Nghi Van Pan (Tyler), Neha Ravi, Rachel Azarian)

We evaluated the following six articles related to COVID-19 prediction and time series anomaly detection: *Is COVID-19 seasonal? A time series modeling approach, Detecting COVID-19 Outbeak with Anomalous Term Frequency, Anomaly detection in time series, Early Stage Machine Learning-Based Prediction of US County Vulnerability to the COVID-19 Pandemic: Machine Learning Approach, Covid-19 classification with deep neural network and belief functions,* and *Time series analysis and predicting COVID-19 affected patients by ARIMA model using machine learning*. The articles provided valuable insights into epidemiological spread and model development. Since COVID-19 is still an evolving issue, the papers were written throughout various points in the pandemic. The earlier articles focused on predicting

COVID-19 spread since little was known about the behavior of the new virus. The newer articles focused on understanding the pattern of COVID-19 spikes throughout the world. The literature review helped guide our model development and evaluation.

The first article, *Is COVID-19 seasonal? A time series modeling approach*, aims to uncover whether “COVID-19 case rates in the United States and Europe followed a seasonal pattern using time series models”(Wiemken 1). They analyzed COVID-19 case data for five European countries and four regions of the United States: North, Midwest, West, and South. In order to measure COVID-19 spikes, they used Twitter’s decomposition method and generalized extreme studentized deviate (GESD) to detect anomalies. They chose this method, because in the past it has reliably identified anomalies in infectious diseases that are transmissible similar to COVID-19. Twitter’s decomposition method “decomposes the time series data into trend, seasonal, and remainder components” (Wiemken 3). Then, GESD was utilized to detect anomalies in the data. To perform a sensitivity analysis, they also used Meta’s Prophet method to decompose the US covid data and included additional features like holidays and variant frequencies. As an additional sensitivity analysis, they applied the first approach and the Prophet model to US influenza data to test if both models could predict the seasonal patterns of influenza. The models did accurately predict the influenza seasonal spikes over six seasons, which matches current knowledge about the patterns of the spikes. For COVID-19, the anomaly detection models proved that the virus spikes between November and March for Europe and the United States. For the US regions, North, Midwest, and West seasonal spikes followed the pattern of the whole country, but the South saw an additional spike in late summer. The method of using Twitter’s decomposition method and generalized extreme studentized deviate (GESD) was able to detect seasonal anomalies in COVID-19 case rates.

The second article, *Detecting COVID-19 Outbeak with Anomalous Term Frequency,* focuses on outbreak detection by predicting “whether a future point is or is not a sign of a large-scale COVID-19 outbreak”(Chen 1). There were three main objectives of the study. First, design different anomaly detection pipelines that can be applied to multivariate time series data. Then, test all pipeline configurations using an automated searcher to find the most effective pipeline for outbreak detection. Finally, prove that real world problems can be solved using the TODS anomaly detection package for building anomaly detection pipelines. To prove this, they used four non-anomaly detection methods to create baseline results and compared the results to 12 different anomaly detection methods to determine if they are a viable solution to detect COVID-19 outbreaks.

The four non-anomaly detection methods were both supervised and unsupervised and include Support Vector Classifiers, Logistic Regression, K-Means Clustering, and Spectral Clustering. The algorithms would be evaluated using recall, precision and F1 scores to determine the anomaly class. None of the supervised models classified any anomaly points correctly, and the unsupervised learning algorithms only predicted one anomaly point correctly. For the anomaly detection algorithms, Local Outlier Factor (LOF), K-Nearest Neighbors (KNN), One Class Support Vector Machine (OCSVM) the DeepLog algorithms also did not predict any anomaly points correctly. Histogram Based Outlier Score (HBOS) and Lightweight On-line

Detector of Anomalies (LODA) had the highest Precision and F1-scores. Since the results were not optimal, smoothing was tested by applying a seven-day moving average in an attempt to improve model performance by removing noise. Some models did worse, while some did better, but there was not a large difference in performance using the smoothed data. This is because smoothing made some of the anomaly points less severe. Overall, when comparing the baseline results to the anomaly detection algorithms, the best performing algorithms were Subspace Outlier Detection (SOD), LODA, HBOS, and Cluster-based Local Outlier Factor (CBLOF). All of the algorithms outperformed the baselines with equal or better precision and much higher recall and F1 scores, so they were better equipped with predicting anomaly points. SOD and LODA performed the best overall when all three metrics were taken into account. Overall, the anomaly detection models were better at classifying outbreaks than traditional machine learning algorithms.

The next article, *Early Stage Machine Learning-Based Prediction of US County Vulnerability to the COVID-19 Pandemic: Machine Learning Approach,* aimed to predict COVID-19 disease movement at a county level. This study focused on the early stages of the pandemic, March 14th to March 30th 2020. Although the time frame of this study is limited, it provided us with modeling ideas for our project. In this study, they developed a three stage model with XGBoost “to quantify the probability of COVID-19 occurrence and estimate the number of potential occurrences for unaffected counties” (Mehta 1). By combining these results, they were able to predict county-level risk, which was used to the vulnerability of a particular county. The data used includes US Census data on age, gender, and population density, CDC and the Global Health Data Exchange for data on various health statistics, the county centroids from ArcGIS, and COVID-19 daily causes from the NYTimes Github. The data used for the model did have correlation between variables like cancer rate and older population percentage. The first model was an XGBoost classifier to predict if a county has a negative or positive instance of COVID-19. Next, an XGBoost regression model would predict the number of cases per county. Then, this was combined to calculate the vulnerability by multiplying the two results together. Total population was by far the most important variable, followed by population density, longitude, hypertension prevalence and then other health risk factors. The accuracy of the first stage, the classification model, was 83%. The overall three-stage model had a sensitivity of 71.5% and a specificity of 94%. It was determined that urban counties with higher overall populations and population densities were more vulnerable and rural counties were least vulnerable. These results can help manage response teams, COVID-19 test distribution, and the allocation of resources to counties that are the most vulnerable for the spread of the virus. This article provided insight into useful county-level features and modeling techniques.

Similar to our research, the next article, *Predicting COVID-19 disease severity from SARS-CoV-2 spike protein sequence by mixed effects machine learning,* focused on the different variants of COVID-19. This study aimed to predict the impact of variant spike protein mutations on severity of the disease. They utilized GISAID data that listed whether a sequence record is associated with a severe or mild case of COVID to model disease severity using patient age and gender data and variant viral genotype. Five models were used including logistic regression with elastic net regularization, random forests, XGBoost, Light GBM and

GPBoost. The variant genetic sequences were tokenized, creating 1,273 features, and the patient features include age and gender. Hyperparameter tuning was performed to maximize accuracy using five-fold cross-validation. The models were trained on samples collected from July 17, 2021 - December 25, 2021 and were tested on samples collected from December 26, 2021 - April 10 2022. The five models were evaluated using Accuracy, Precision, and Recall. GPBoost performed the best, followed by LightGBM and XGBoost. The three models had an accuracy above 75% and a relatively high precision and recall score. This study is very interesting because they identified key mutations for recognizing early sights for risks of viral variants. This article provided insight into useful variant features and modeling techniques.

The following article, *Convolution Neural Networks Based on Sequential Spike Predict High Human Adaption of SARS-CoV-2 Omicron Variants*, also attempted to identify high-risk Omicron sublineages from their less harmful counterparts. Because COVID-19 transmissibility is tied to mutations in the virus, they used over nine million genome sequences of the COVID-19 virus from the GISAID database. They focused on Spike sequences and translated them into protein sequences composed of amino acids. After their transformation, Spike protein sequences were represented by 20 types of amino acids. Principal Component Analysis and hierarchical clustering were used on the Spike sequences to reduce the feature matrix of the data. The training data was composed of Gamma, Alpha, Iota, and Epsilon variants and were labeled with a 0, 1, or 2 based on their transmissibility. They ran a 2D-CNN model on the data in an attempt to predict the transmissibility of Omicron sub lineages based on their Spike protein sequences. The model predicted 70% of Omicron's sub lineages were level 1 and 30% were level 2. The predicted transmissibilities were highly consistent with the spread of Omicron subvariant BA.1 and BA.2. The model was able to provide a real-time solution to assess transmissibility of emerging variants based on their genetic sequence. This is helpful because it can help mitigate risks and control emerging variants. This article provided insight into how complex variant related features are, but showed us the importance of including them in our feature set.

The final article, *Time series analysis and predicting COVID-19 affected patients by ARIMA model using machine learning*, took a different approach and attempted to forecast the confirmed number of positive cases using the ARIMA time series model. ARIMA was selected due to its high accuracy for short time series data. The COVID-19 case data was obtained from the World Health Organization. Data was collected for over 198 counties from January 22, 2020 - July 7, 2020. January through April 6th was used for training and the rest for testing. First the rolling statistics were calculated on the dataset. Next, the Dickey-Fuller test was utilized to estimate trends in the data. The values of P & Q were determined by performing the Partial Autocorrelation graph and Autocorrelation graph. After, the AR and MA model were run to predict future case levels. The Residual Sum of Squares of the AR and MA model were both 0.42, which were satisfactory results. The RSS of the ARIMA model was 0.41, which the authors deemed a good result. The confidence level of the predictions was around 95%, which is satisfactory for the task. When training time was increased, the model results improved. For future work, other prediction models should be explored like artificial neural networks, Support Vector Machines, and Baysian networks.

Overall, the literature reviews provided valuable insights into modeling COVID-19 spikes, cases, and variant transmissibility throughout the pandemic. They each studied different approaches for modeling these topics and provided various avenues for future work. Additionally, the articles provided valuable insights into the useful features needed for accurate COVID-19 related predictions. Some methods focused on demographic and health data of a location, while others utilized the genetic make-up of different COVID-19 subvariants. It is also interesting to see how modeling evolved along with the pandemic. Older studies focused on predicting number of cases, spread, and spike detection. Newer studies looked more closely at the differences between variants and their transmissibility. We combined the insights from these articles to develop a new approach to COVID-19 subvariant modeling.

Data Overview (Nghi Van Pan (Tyler), Neha Ravi, Rachel Azarian)

Data was collected from many different sources to create the final datasets that were used to investigate the problem statement. In order to predict the number of weeks for a subvariant to spike in a state or county, we collected data on the daily number of COVID-19 cases by county. The daily cumulative number of cases by county from 2020 to 2022 was scraped using Python from Johns Hopkins COVID-19 Data Repository. After the data was scraped, the daily new number of COVID-19 cases were calculated by taking the difference in cumulative cases per day by county using Python.

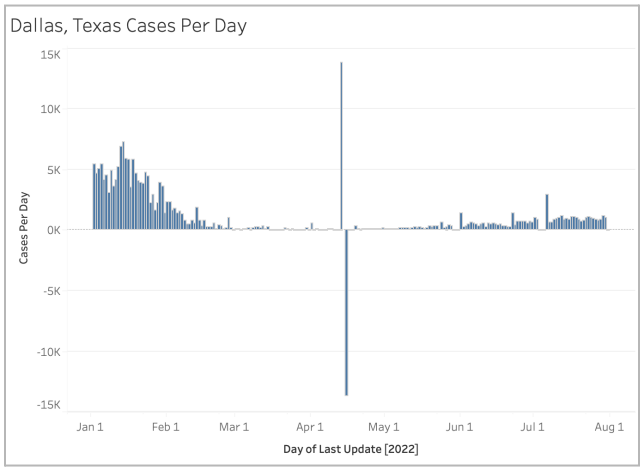


Figure 1. Dallas, Texas daily COVID-19 cases from Johns Hopkins COVID-19 Data Repository

The figure above shows the daily number of cases for Dallas, Texas. There were sharp increases and then decreases in the number of cases reported by many counties. From the data, it appears that a county would report a large number of cases on one day and then a large number of negative cases the next day. This was smoothed to avoid these artificial spikes in cases. To create the state-level daily number of new cases, the counties were summed together by state. Below is a graph of the new daily COVID-19 cases for the entire US once the spikes were smoothed.

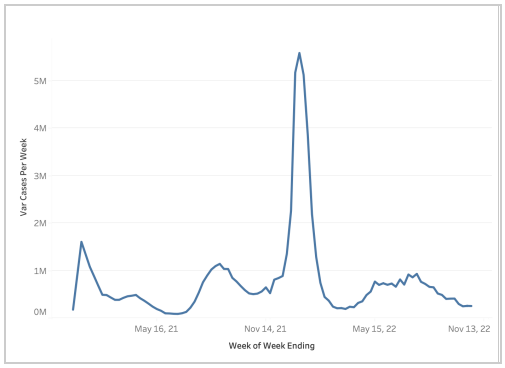


Figure 2. US new daily COVID-19 cases

To investigate the problem statement, data on new cases by subvariant was needed, which is currently not available. However, the CDC provides national and regional sub-variant proportions on a weekly basis starting 1/2/21 through 10/29/22. Although the data is not available by state or county, the CDC breaks the US into 10 regions made up of three or more states, and they report the proportions of subvariants by region. This data listed the proportion of each subvariant found in a region by week, but there were many duplicates of subvariant proportions for each week due to various published dates. To solve this issue, only the data for the most recent published date was kept for each week. Below is our finalized data on sub-variant proportions for all regions.

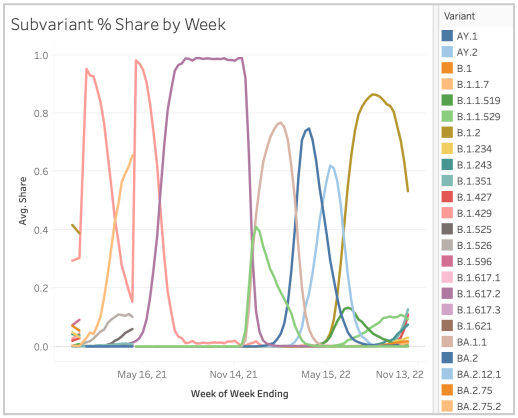


Figure 3. CDC Subariant Proportion data for all regions

In order to combine the subvariant proportions and covid case data, each state and county were assigned a region based on the CDC region assignments. Since the subvariant proportions were only available on a weekly basis, the daily new covid cases by state and county had to be summed by week starting 1/2/21 through 10/29/22 to match the CDC data. Next, the subvariant region proportions were multiplied by the number of new weekly cases to provide the number of new weekly cases by subvariant for each state and county.

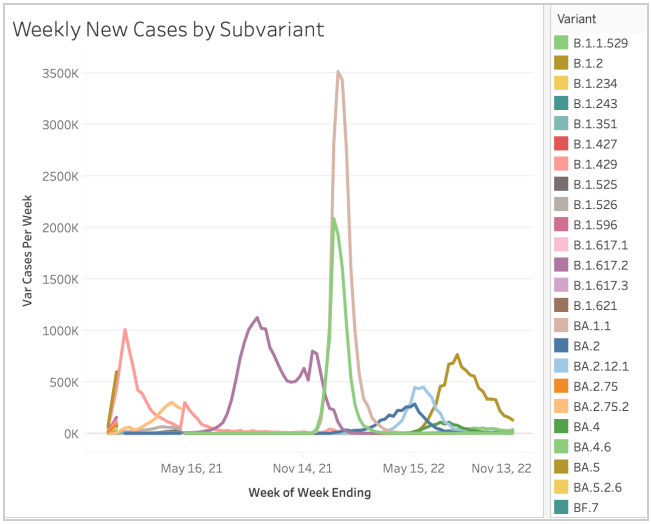
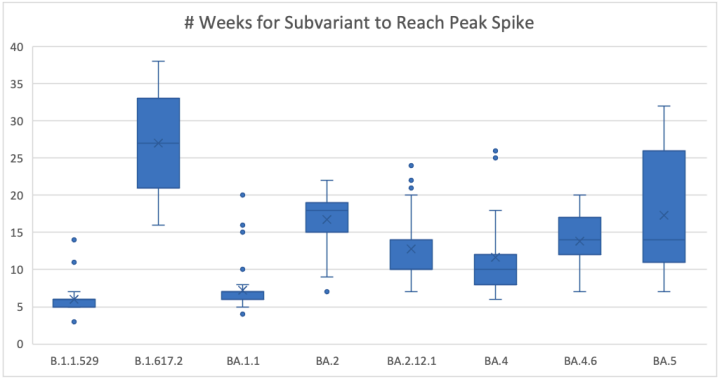


Figure 4. US new weekly cases by subvariant

Here is the number of new weekly cases by subvariant for all counties and states. The CDC data included 35 different sub-variants, but this graph showed which subvariants would be useful for our prediction model. We could only select subvariants that emerged after 1/1/21, so that we could get the date that the subvariant first arrived in a state or county. The final subvariants selected were B.1.1.529, B.1.617.2, BA.1.1, BA.2, BA.2.12.1, BA.4, BA.4.6, and BA.5. While subvariants B.1.1.7 and B.1.526 both produce a material number of new weekly cases, they both began in 2020, which the CDC does not provide subvariant proportion data for.

The goal is to predict the number of weeks it takes a subvariant to spike in a county or state, so we calculated the number of weeks between the date with the maximum number of cases for each subvariant, and the date that the subvariant first arrived, all grouped on a county and state level. We counted the first week of a subvariant's arrival if the subvariant was responsible for at least 10 cases at the state level and at least two cases at the county level. Below is a box and whisker plot showing the means and standard deviations of the number of weeks for the subvariants to reach their peak spike.



# Weeks for Subvariant to Reach Peak Spike								
	B.1.1.529	**B.1.617.2**	**BA.1.1**	**BA.2**	**BA.2.12.1**	**BA.4**	**BA.4.6**	**BA.5**
Mean	6.00	27.04	7.22	16.71	12.75	11.61	13.84	17.25
Std Dev	1.59	6.86	3.00	3.43	5.16	5.76	3.43	7.33

Figure 5. Mean and standard deviations of subvariant spikes

This analysis was helpful for understanding the behavior of the different subvariants and interpreting the results of our models. B.1.1.529 and BA.1.1 were the quickest subvariants to spike, while B.1.617.2 and BA.5 took the longest to reach peak spike and had the largest variation in number of weeks to spike. While this plot does show the outliers in the data, we decided to keep the outliers due to the nature of the problem statement. Outliers can be important for understanding the features that can impact spikes.

Next, demographic and subvariant related features were merged onto the state and county subvariant covid case data. Below is a table of our attribute names, data types, descriptive statistics, and data sources for our state and county datasets.

State Data Features

Attribute Name	Data Type	Min	Max	Mean	Std Dev	Source
S1 Mutation	Continuous	9.00	29.13	24.89	6.13	nextrain.org
Mutation Fitness	Continuous	0.87	3.07	2.41	0.67	nextrain.org
22A (Emerging Lineage)	Nominal	0	1	0.38	0.48	nextrain.org
21A (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
21k (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
21L (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
22C (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
Q1_first_case	Ordinal	0	1	0.28	0.45	Calculation w. Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions
Q2_first_case	Ordinal	0	1	0.46	0.50	Calculation using Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions
Q3_first_case	Ordinal	0	1	0.01	0.11	Calculation using Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions
Republican_2020	Nominal	0	1	0.49	0.50	Cook Political
Population Estimate	Discrete	578,803	39,237,836	6,508,559.04	7,333,393.27	Bureau of Transportation
PovertyRate	Continuous	0.07	0.19	0.12	0.03	USDA
Unemployment Rate	Continuous	0.03	0.07	0.05	0.01	USDA
HS Diploma Rate	Continuous	0.84	0.95	0.91	0.02	USDA
Interstate_Pct	Continuous	0.01	0.07	0.03	0.01	Federal Highway Administration
Freeway_Pct	Continuous	0.00	0.03	0.01	0.01	Federal Highway Administration
Arterial_Road_Pct	Continuous	0.07	0.26	0.12	0.04	Federal Highway Administration
Collector_Road_Pct	Continuous	0.09	0.39	0.18	0.05	Federal Highway Administration
Local_Road_Pct	Continuous	0.43	0.77	0.66	0.05	Federal Highway Administration
#_Public_Airports	Discrete	0	12	3.65	2.70	US Airport Codes
Total_#_Airports	Discrete	3	1,494	252.65	223.27	GlobalAir
#_Bridges	Discrete	244	54,131	12,021.88	9,984.42	Bureau of Transportation
#_business_establishments	Discrete	20,967	940,166	153,579.20	172,361.58	Bureau of Transportation
Miles freight railroad	Continuous	4.20	10,856.10	2,834.08	1,979.96	Bureau of Transportation
Miles passenger railroad	Continuous	0.00	2,709.90	536.47	495.55	Bureau of Transportation
1 Region	Ordinal	0	1	0.12	0.32	CDC
2 Region	Ordinal	0	1	0.04	0.19	CDC

3 Region	Ordinal	0	1	0.12	0.32	CDC
4 Region	Ordinal	0	1	0.16	0.36	CDC
5 Region	Ordinal	0	1	0.12	0.32	CDC
6 Region	Ordinal	0	1	0.10	0.30	CDC
7 Region	Ordinal	0	1	0.08	0.27	CDC
8 Region	Ordinal	0	1	0.12	0.32	CDC
9 Region	Ordinal	0	1	0.08	0.27	CDC
Omicron	Nominal	0	1	0.88	0.33	nextrain.org
% Fair or Poor Health	Continuous	12.64	24.34	17.21	2.90	County Health Rankings
% Smokers	Continuous	7.90	26.14	16.76	3.53	County Health Rankings
% Adults with Obesity	Continuous	23.60	41.20	32.06	3.98	County Health Rankings
% Flu Vaccinated	Continuous	37.00	56.00	47.53	4.60	County Health Rankings
% Severe Housing Problems	Continuous	11.13	26.43	15.77	3.39	County Health Rankings
Population_Density	Continuous	1.28	11,675.94	431.43	1,614.75	Census Bureau

County Data Features

Attribute Name	Data Type	Min	Max	Mean	Std Dev	Source
S1 Mutations	Continuous	9.00	29.13	24.85	6.14	nextrain.org
Mutation Fitness	Continuous	0.87	3.07	2.40	0.67	nextrain.org
22A (Emerging Lineage)	Nominal	0	1	0.37	0.48	nextrain.org
21A (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
21k (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
21L (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
22C (Emerging Lineage)	Nominal	0	1	0.13	0.33	nextrain.org
Primary and Commercial Airports	Discrete	0	3	0.15	0.38	Bureau of Transportation
Airports and Seaplane base	Discrete	0	13	1.21	1.33	Bureau of Transportation
Number of Bridges	Discrete	0	3,889	262.87	298.72	Bureau of Transportation
# business establishments	Discrete	103	275,316	3,538.75	13,796.16	Bureau of Transportation
% resident workers commute by transit	Continuous	0.00	0.34	0.01	0.02	Bureau of Transportation
Number of residents	Discrete	10,035	10,057,155	147,731.40	527,134.98	Bureau of Transportation
Miles freight railroad	Continuous	0.00	816.10	57.82	63.05	Bureau of Transportation
Miles passenger railroad	Continuous	0.00	431.50	10.81	29.55	Bureau of Transportation
% HS Diploma	Continuous	0.39	0.92	0.77	0.09	USDA
Total people in poverty (%)	Continuous	0.04	0.30	0.12	0.04	USDA
Poverty under 18 (%)	Continuous	0.03	0.46	0.16	0.07	USDA
Unempl rate	Continuous	0.03	5.60	0.20	0.70	USDA
median Income	Continuous	33,531.	139,462.00	61,550.62	13,301.97	USDA
Region 4	Ordinal	0	1	0.13	0.33	CDC
Region 5	Ordinal	0	1	0.40	0.49	CDC
Region 6	Ordinal	0	1	0.21	0.41	CDC
Region 7	Ordinal	0	1	0.18	0.39	CDC
Q1_first_case	Ordinal	0	1	0.14	0.35	Calculation w. Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions
Q2_first_case	Ordinal	0	1	0.48	0.50	Calculation w. Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions

Q3_first_case	Ordinal	0	1	0.13	0.33	Calculation w. Johns Hopkins Covid cases and CDC data; https://github.com/CSSEGISandData/COVID-19, https://covid.cdc.gov/covid-data-tracker/#variant-proportions
Omicron	Nominal	0	1	0.87	0.33	nextrain.org
% Fair or Poor Health	Continuous	10.50	42.00	19.33	4.76	County Health Rankings
% Smokers	Continuous	8.70	28.70	18.75	3.13	County Health Rankings
% Adults with Obesity	Continuous	16.40	48.10	35.61	3.53	County Health Rankings
% Flu Vaccinated	Continuous	10.00	68.00	46.77	9.04	County Health Rankings
% Severe Housing Problems	Continuous	5.15	32.43	13.47	4.13	County Health Rankings
Population_Density	Continuous	1.80	18,141.24	212.15	792.35	Census Bureau

Once all of the features were merged onto the state and county datasets, the state data ended with 408 instances and 42 features and the county data had 6,080 instances and 34 features. We attempted to collect the same features for both datasets, however some data sources only had data available at the county or state level.

Data was obtained from mostly government and scientific organizations. From nextrain.org, we collected features on the subvariants. This includes the emerging lineage of the subvariant, the number of S1 mutations, and mutational fitness. S1 mutations indicate the mutations of the spike protein that is responsible for transmissibility of the subvariant, and mutational fitness is the subvariant's fitness compared to a reference genotype. Both of these features impact virus transmission, and we expect them to play a significant role in our models' predictions. From the Bureau of Transportation and Federal Highway Administration, state and county features were collected including the number of highways, bridges, business establishments, miles of railroad road and other transportation-related data for 2021. These features were chosen since increased transportation can potentially increase the spread of the virus. The USDA provided general demographic features like poverty rate, high school diploma rate, and income data as of 2020. Although there isn't a clear connection between these features and the target variable, it was tested to see if they play a role in prediction. County Health Rankings provided state and county data on population health like percentage of residents in fair or poor health, percentage of residents who smoke, and percentage of residents vaccinated for the flu as of 2022. Since COVID-19 affects immunocompromised populations differently, we expect these features to be useful for the models. In addition, data was collected on the states' 2020 election results since COVID-19 resulted in a political divide during the pandemic. The Census Bureau provided data on population density as of 2020, to determine if covid spreads faster in more populated states or counties. The reasons listed above indicate the justification for using these sources for modeling.

Project Methods (Rachel Azarian, Neha Ravi)

To predict the number of weeks for a subvariant to spike, the models tested were Decision Tree Regressor, Extra Tree Regressor, Random Forest, XGBoost, and Support Vector Regression. Tree-based models and SVR were chosen because there is not a linear relationship between the features and the target variable. This was observed by evaluating the correlation matrix (Figure 18 in Appendix) and a pairplot of the data. The Decision Tree is the baseline model, because it is relatively simple compared to the other models, and it is easy to

interpret and understand. Also, the state and county data sets contain many features, 42 and 34 respectively. One goal of our modeling is to understand which features are best for predicting the number of weeks for a subvariant to spike, so we wanted to use a model that we could run using all of the features and the model will inherently choose the best performing features based on their relative performance.

Because Decision Trees are prone to overfitting, ensemble methods like Extra Trees, Random Forest, and XGBoost were also selected. Both Extra Trees and Random Forest are composed of many decision trees and the prediction is based on the prediction of multiple trees by arithmetic mean, so they are more robust than decision trees. When selecting the points to split the nodes, Extra Trees chooses this randomly while Random Forest uses the optimum split. Extra Trees are typically faster than Random Forest, but Random Forest tends to have better predictions with noisy features in a high dimensional dataset. We wanted to compare the results of these two similar ensemble methods. XGBoost was chosen because we wanted to test a model that utilizes gradient boosting and learns from its previous predictions. XGBoost can be faster than Random Forest and is also a good predictor when the data set has many features, so we are interested in seeing how XGBoost's results will compare to the other ensemble trees.

The final model is Support Vector Regression to evaluate the results of a non-tree based model. Support Vector Regression performs well when there are many features, and is useful when the solution is not linear due to the ability to tune the kernel parameter. Since there is not a linear relationship between our target variable and the features, Support Vector Regression was selected over linear regression.

For the tree-based models, feature selection is not necessary, however highly correlated features were removed. This is to improve the interpretability of the results. When a feature is utilized in a tree-based model, the importance of the correlated features is reduced. While this makes sense when building the model, it will alter the interpretability of the results. One can conclude that the correlated features are not important for predicting the target variable, however that is incorrect. Since one of the goals of this project is to understand the features that lead to subvariant spikes, correct interpretability of the features is very important. A correlation matrix was created on the attributes to identify features that were strongly correlated to each other for both the state and county data. For features that had a correlation higher than 0.80, the feature that had a lower correlation to the target variable was dropped.

When evaluating the models, the data was first split into testing and training sets, with 20% of the data used for testing and 80% used for training. For both state and county level data, the data was grouped so that the models were trained on 80% of the states/counties and tested on the other 20% of the states/counties to prevent data leakage. This was performed by assigning each state and county a group number and using the sklearn GroupShuffleSplit function to split the data by group.

The model hyperparameters were tuned in an attempt to improve their performance. Grid Search cross validation on the training data was used to test a variety of hyperparameters for each model. Grid Search cross validation tests the combinations of hyperparameter values provided in the dictionary, and returns the parameters that yield the best result. One drawback is that the grid search is limited to the parameters that are manually inputted. Five-fold Grid Search was selected due to the number of instances for the state-level data. When performing Grid Search cross validation, GroupKFold cross validation was implemented in order to keep the states and counties grouped together and avoid data leakage during cross validation. Grid Search was performed to minimize the root mean squared error. RMSE was chosen as the metric to minimize over Mean Absolute Error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

As shown by the formula above, RMSE squares the errors before they are averaged, which gives a higher weight to large errors. This is helpful for the analysis, because we want to avoid large errors. Large errors in predicting the number of weeks for a subvariant to spike can result in communities lacking resources and being unprepared for a surge in cases.

When performing Grid Search cross validation on the Support Vector Regression model, a pipeline was utilized in order to properly scale the data. SVR requires feature scaling, because it relies on the distance between observations. MinMaxscaler was used, which scales the feature values between 0 and 1. In k-fold cross validation, the data is split into k different groups, and k-1 groups are used for training while the remaining group is used as the validation set. This is repeated until each group is used as a validation set. Since the training data changes at each fold, it is important that the training data is scaled individually to avoid data leakage. This yields results that are more generalizable and will perform similarly to the results of the model when it is applied to the test set.

Next the hyperparameters were tuned. The tree-based hyperparameters were mostly used to pre-prune the model to reduce overfitting. For the Extra Tree Regressor model, the hyperparameters tuned were max leaf nodes, min samples split, and splitter method. Min samples split provides a minimum number of instances needed to split an internal node and max leaf nodes determines the maximum number of nodes, limiting the tree's growth. Updating these parameters can help reduce overfitting. The splitter method hyperparameter can be best or random. This determines which strategy will be utilized to split at each node. For Random Forest, the parameters tuned were max depth, min samples leaf, and n estimators. Max depth limits the depth that each tree can grow to in order to avoid overfitting. N estimators indicates the number of trees utilized in the Random Forest algorithm, which can help reduce the complexity of the model. For XGBoost, the parameters tuned were max depth, min child weight, subsample, colsample bytree, and gamma. Min child weight and max depth are both utilized to help control overfitting and produce a more generalized model. Subsample refers to the fraction

of observations that will be random samples for each tree, also an attempt to prevent overfitting. Subsampling occurs after the trees are constructed, and colsample_bytree determines the ratio of columns when creating each tree. Gamma can range from zero to infinity, and requires the minimum loss reduction needed for a split to occur. For Support Vector Regression, the parameters tuned include the kernel, degree, C, and gamma. The kernel hyperparameter determines which kernel type will be used for the algorithm. The default hyperparameter is radial bias function. When the kernel type is polynomial, the degree hyperparameter determines the degree of the polynomial function. This parameter is ignored when the kernel is not polynomial. Gamma determines the degree of curvature for the line and the C parameter is used to determine regularization.

Once the best hyperparameters were chosen for each model, 10-fold cross validation was performed using the training data to evaluate the generalizability of the tuned models. The average and standard deviation of the mean absolute error, root mean squared error, and mean squared error results were calculated to help determine which model has the best and most generalizable results.

Following cross validation, the tuned models were used to predict the test data and the final model metrics were calculated and compared to also help determine the best model for the data. The model metrics evaluated were mean absolute error, mean squared error, root mean squared error, mean absolute prediction error, and R^2 coefficient, which returns the coefficient of determination. Mean Absolute Error calculates the average of the absolute errors. The Mean Squared Error calculates the average squared difference between the actual and predicted values. There are several benefits of squaring the differences. Squaring the differences ensures that there are no negative values, and increases the impact of large errors. The Root Mean Squared Error takes the square root of the Mean Squared Error.The R-Squared value shows how well the data fits the model. The higher the R-squared value the better the fit. Mean Absolute Prediction Error, MAPE, was also calculated for each model. The lower the MAPE, the better the model is at making predictions.

Results (Nghi Van Pan (Tyler), Neha Ravi, Rachel Azarian)

The models were first evaluated on the county and state level test data before tuning the parameters to understand the baseline model performance. The Random Forest algorithm was the best performing model before tuning the parameters since it had the lowest MAE, MSE, RMSE, MAPE and the highest R^2 coefficient. XGBoost performs similarly, but not as well as Random Forest. Also, the models performed better on the county data, as the county error metrics are better than the state error metrics across all models.

County Test Data Results Before Tuning Parameters					
	Decision Tree	Extra Regressor Tree	Random Forest	XGBoost	SVR
MAE	2.53	2.60	1.96	1.97	2.30
MSE	14.63	15.38	7.61	7.94	10.87
RMSE	3.82	3.92	2.76	2.82	3.30
R^2 Coefficient	0.56	0.57	0.77	0.76	0.67
MAPE	0.22	0.23	0.17	0.17	0.20

State Test Data Results Before Tuning Parameters					
	Decision Tree	Extra Regressor Tree	Random Forest	XGBoost	SVR
MAE	2.85	2.89	2.30	2.60	4.15
MSE	20.76	18.13	11.17	14.94	35.61
RMSE	4.56	4.26	3.34	3.87	5.97
R^2 Coefficient	0.67	0.71	0.76	0.75	0.43
MAPE	0.25	0.25	0.20	0.23	0.36

Figure 6. Tables with model metrics for state and county test data before hyperparameter tuning

Next, the models' hyperparameters were tuned using 5-fold GridSearchCV. The exact hyperparameters tuned can be found in the Approach section.

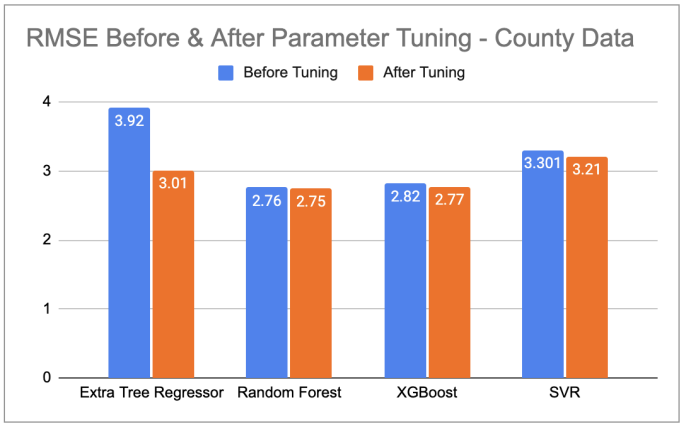


Figure 7. RMSE of models on county test data before and after hyperparameter tuning

The parameters were tuned to minimize the RMSE score. For the county data models, Extra Tree Regressor experienced the best improvement in RMSE with a decrease of 23%. The best hyperparameters for Extra Tree Regression include max leaf nodes of 60, min samples split of 3, and splitter mechanism of best. Random Forest and XGBoost improved the least from hyperparameter tuning, but they are still the best performing models.

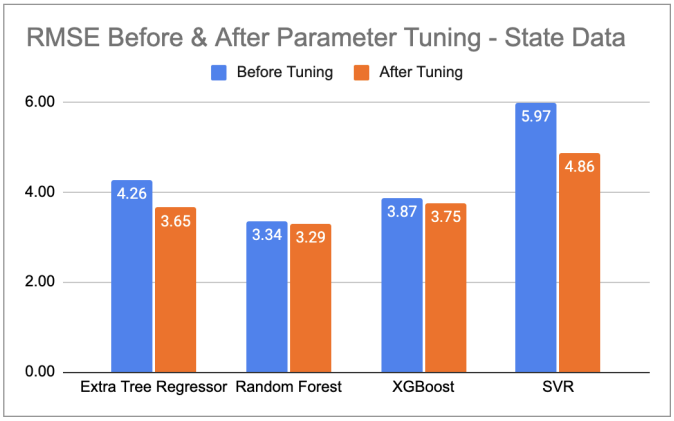
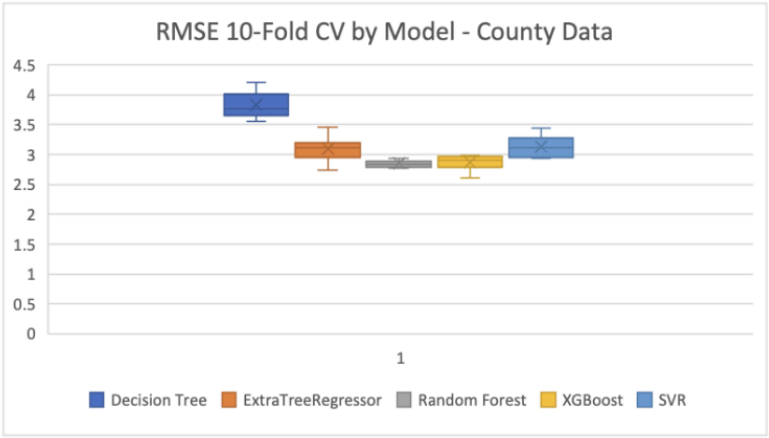


Figure 7. RMSE of models on state test data before and after hyperparameter tuning

Using the state-level data, the same hyperparameters were tuned to minimize the RMSE score. Support Vector Regression had the best improvement in RMSE with a decrease of 19%. The best hyperparameters for the SVR model was a polynomial kernel with a degree of four.

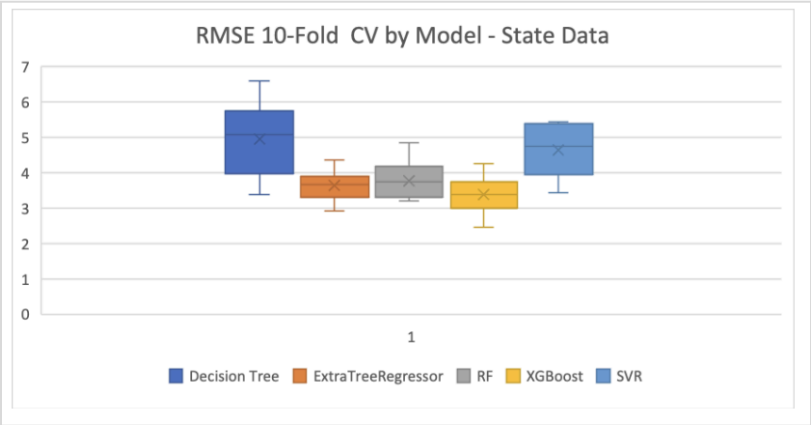
Next, using the tuned models, 10-fold cross validation was performed on the county training data. The chart below displays the mean and standard deviation of RMSE scores by model. The Random Forest model had the least amount of variation in RMSE and lowest average RMSE. The Decision Tree had the most amount of variation and the highest average RMSE.



	Decision Tree	Extra Tree Regressor	Random Forest	XGBoost	SVR
Mean RMSE	3.83	3.09	2.83	2.87	3.13
Std Dev RMSE	0.22	0.20	0.10	0.13	0.17

Figure 8. Mean and standard deviation of 10-fold cross validation RMSE results across all models on county training data

As shown below, 10-fold cross validation was also performed on the state training data to evaluate the RMSE scores by model. The Decision Tree model had the most amount of variation in RMSE results and the highest average RMSE. XGBoost had the lowest average RMSE and Extra Tree Regressor had the smallest standard deviation of results, but XGBoost, Random Forest, and Extra Tree Regressor all had similar results.



	Decision Tree	Extra Tree Regressor	Random Forest	XGBoost	SVR
Mean RMSE	4.94	3.64	3.78	3.39	4.65
Std Dev RMSE	1.07	0.45	0.54	0.54	0.69

Figure 9. Mean and standard deviation of 10-fold cross validation RMSE results across all models on state training data

Once the optimal hyperparameters were identified for each model and model generalizability was tested, the tuned models were evaluated on the test set. Similar to before tuning the parameters, Random Forest performed the best for both the state and county level data sets. Random Forest has the lowest MAE, MSE, RMSE, and MAPE and the highest R^2 coefficient. Also, the results were better across all models on the county data than the state data.

County Test Data Results After Tuning Parameters					
	Decision Tree	**Extra Regressor Tree**	**Random Forest**	**XGBoost**	**SVR**
MAE	2.53	2.04	1.95	2.01	2.22
MSE	14.63	9.03	7.57	7.67	10.32
RMSE	3.82	3.01	2.75	2.77	3.21
R^2 Coefficient	0.56	0.73	0.78	0.77	0.69
MAPE	0.22	0.18	0.17	0.18	0.20

Figure 10. Final model metrics using tuned models for county test data

State Test Data Results After Tuning Parameters					
	Decision Tree	Extra Regressor Tree	Random Forest	XGBoost	SVR
MAE	2.85	2.48	2.31	2.49	3.54
MSE	20.76	13.36	10.83	14.03	23.65
RMSE	4.56	3.65	3.29	3.75	4.86
R^2 Coefficient	0.67	0.79	0.77	0.77	0.62
MAPE	0.25	0.22	0.20	0.22	0.31

Figure 11. Final model metrics using tuned models for state test data

To understand the features that influenced the predictions, feature importance was evaluated for the best performing model.The chart below displays the feature importance for the Random Forest algorithm using county data. S1 mutations are the most important feature, which deals with a spike protein that causes the virus to spread faster, but decreases its lethality. The second most important feature is a dummy variable for subvariants that emerged in Q1, and the third most important feature is the number of residents.

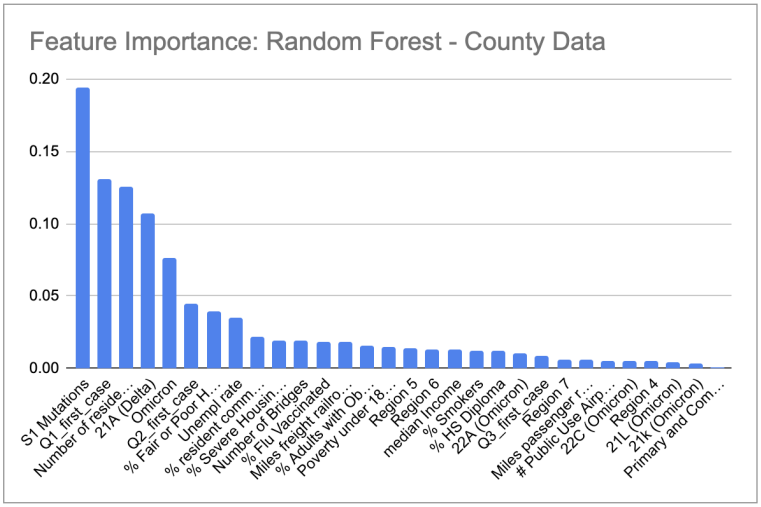


Figure 12. Feature importance of Random Forest tuned model on county data

The chart below displays the feature importance for the XGBoost algorithm using county data. The top three important features are the same as the Random Forest model, however in a different order. The most important feature is the dummy variable for subvariants that emerged in Q1, followed by the number of residents, and then S1 mutations.

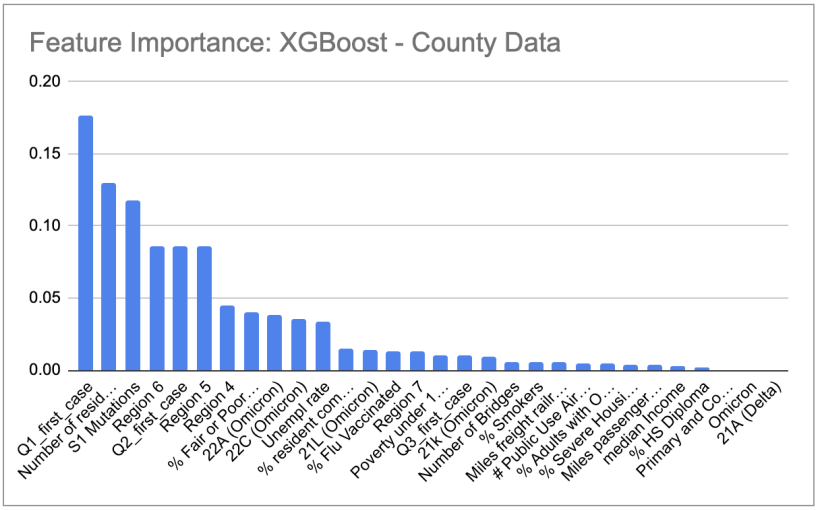


Figure 13. Feature importance XGBoost tuned model on county data

To better understand the usability of the best model in practice, the mean and standard deviation of absolute errors by subvariant was calculated for Random Forest errors on county test data. These metrics were then divided by the average number of weeks the subvariant took to spike to normalize the results. B.1.1.529, BA.1.1, and BA.5 subvariants performed best with the lowest Mean Absolute Error and the smallest variance in absolute error.

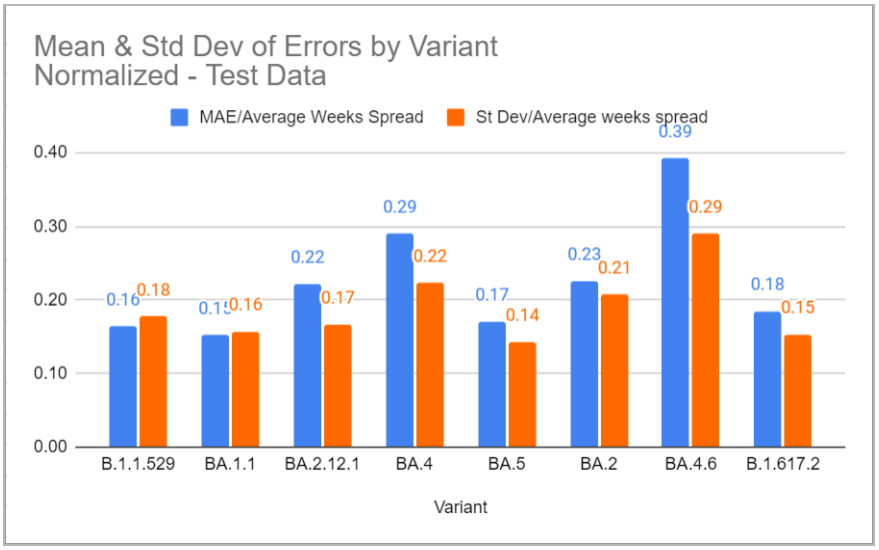


Figure 14. Mean and standard deviation of Random Forest county test data errors by subvariant

The mean and standard deviation of absolute errors by subvariant was calculated on Random Forest using county training data as well. Like the test data results, B.1.1.529, BA.1.1, and BA.5 subvariants had the lowest average errors.

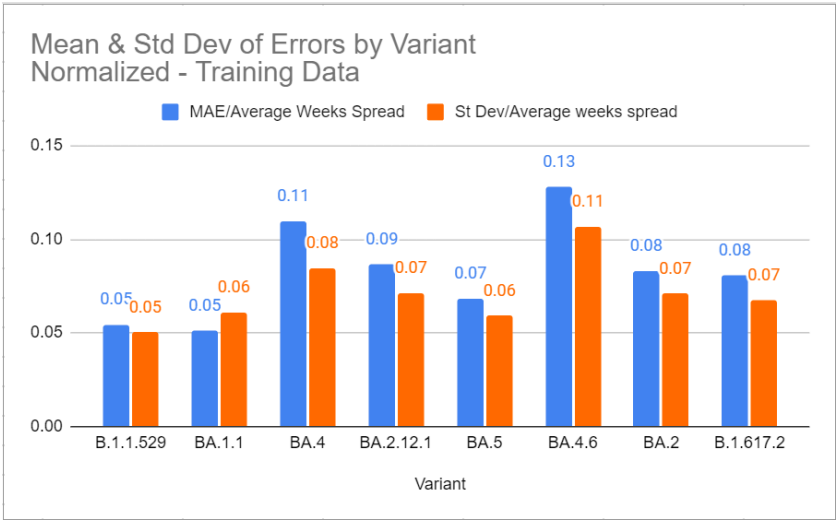


Figure 15. Mean and standard deviation of Random Forest county training errors by subvariant

Additionally, the mean and standard deviation of absolute errors by state was calculated for Random Forest using county test data to understand if the model predicted weeks to spike better in some states than others. This did not have to be normalized by the average number of weeks of subvariant spread by state, because it was similar across all states.

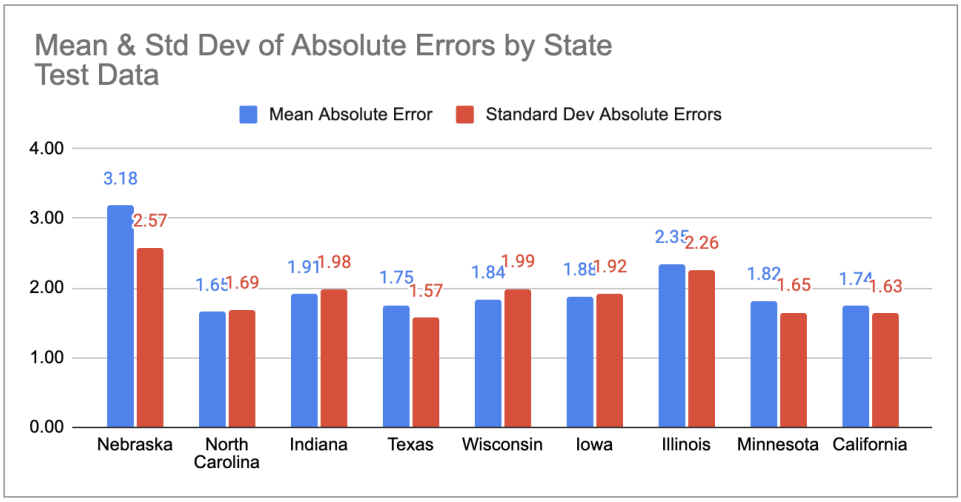


Figure 16. Mean and standard deviation of Random Forest county test errors by state

The mean absolute error and standard deviation of errors was comparable across all states, except Nebraska, which had a noticeably larger mean absolute error and standard deviation of absolute errors.

Interpretation of results (Nghi Van Pan (Tyler), Neha Ravi, Rachel Azarian)

Valuable insights were learned from the results of our modeling efforts to predict the number of weeks for a subvariant to spike in a state or county. One part of the problem statement was to test whether COVID-19 subvariant spikes are better predicted at a more broad or localized level by comparing the modeling results using state and county data. While county-level data had many more instances, the reporting of covid cases by state was more standardized. There was more variability in how counties reported daily cases, especially in more rural areas. Additionally, eight more features were collected for the state data.

However, based on the models' results the county-level data resulted in more accurate predictions. All of the county models resulted in lower error metrics, including Mean Absolute Percentage Error, and a higher R^2 coefficient, which indicates how well the data fits the model. Also, as shown by the 10-fold cross validation results, the mean and standard deviations of RMSE scores of the state data were higher across all models. For example, Random Forest cross validation on the county data had an average RMSE score of 2.83 with a standard deviation of 0.10 and the state model had an average RMSE score of 3.78 with a standard deviation of 0.54. The lower standard deviations of RMSE scores indicates that the county-level models are not only more accurate, but also more generalizable and will perform better on new data. Due to these results, it is apparent that COVID-19 subvariant spikes are more accurately predicted at the county level.

We expected these results, since the county-level data had many more instances than the state data, 6,080 and 408 respectively. This is due to the fact that there are many more counties than states. Generally, the more data the models are trained on, the less overfit they are and the better they will perform. This is indicated by the cross validation results mentioned above. Also, more data will provide a better distribution for the model's learning process. Additionally, applying knowledge based on how COVID-19 has spiked during the past two years of the pandemic, the virus generally creates more localized spikes. In terms of using the model in practice, obtaining subvariant spike information on a county level is more helpful for allocating resources to treat, test, and mitigate cases.

Focusing on the county data model results, Random Forest performed the best across all metrics, and XGBoost was a close second. Decision Trees performed the worst, and this was expected when comparing the results of the Decision Tree to ensemble trees. Decision trees have a tendency to overfit the data, which is shown by the higher standard deviation of RMSE scores during cross validation. Support Vector Regression was the next worst performing model. Since our features did not have a linear relationship to the target variable, we chose Support Vector Regression to tune the kernel parameter, and the best performing kernel was polynomial with a degree of 4. As shown by Figure 5, the data does include outliers, which we kept due to the nature of the problem statement. SVR is sensitive to outliers, and with a polynomial degree of 4, this may have led to overfitting.

The two best performing models were XGBoost and Random Forest, with Random Forest slightly beating XGBoost across all performance metrics. Since Random Forest uses parallel, independent trees and takes the average result, this helps mitigate the effects of

overfitting. We believe this is the reason the model outperformed XGBoost. XGBoost builds trees sequentially, so they learn from the previous trees' mistakes, which can inherently lead to more overfitting. XGBoost did have a slightly higher standard deviation of RMSE scores during 10-fold cross validation than Random Forest, so slight overfitting caused by the sequential trees of XGBoost led to the model's slightly worse results on the county data than Random Forest.

Next, we evaluated Random Forest's county data results by subvariant and by state to gain more insight into the usability of the model. Figure 14 shows how the model performed by subvariant. We evaluated the mean and standard deviations of the absolute errors by subvariant, normalized by the average number weeks it takes for each subvariant to spread. The two BA4 subvariants performed the worst, with BA4.6 being the worst performing subvariant. This is helpful to understand, because future variants that have similar features to BA4.6 or share an emerging lineage may experience errors worse than average. Additionally, Figure 16 shows the model's performance by state. Nebraska stands out as the outlier with the highest error metrics. After evaluating Nebraska's covid case data by county, we noticed that many counties did not report case data in a standardized way. Some counties would report cases every day and then switch to reporting cases once every two or three weeks, which created artificial spikes in cases. The quality of Nebraska's daily covid case data impacted the results. This displays how our models are only as good as the data used to train them.

Interpreting the feature importance of the best performing model, Random Forest on county data, provides insight into the features that are the most useful for predicting subvariant spikes. Figure 12 in the results section provides the importance of all features, and the table below is the top 10 features ranked by their importance.

Feature	Importance
S1 Mutations	0.194
Q1_first_case	0.131
Number of residents	0.126
21A (Delta)	0.107
Omicron	0.076
Q2_first_case	0.044
% Fair or Poor Health	0.040
Unempl rate	0.035
% resident commute by transit	0.021
% Severe Housing Problems	0.019

Of the top five most important features, 80% of them are subvariant-related features. The most important feature is S1 mutations of the subvariant, which indicates the average number of mutations on the S1 spike protein. This impacts the transmissibility of the virus, so it is clear why this is the most important feature. The next important feature is a dummy variable that indicates if the first case of the subvariant in the county occurred in the first quarter of the year. This makes sense since COVID-19 tends to be more transmissible in the winter. The third most important feature is the number of residents, which can impact the transmissibility and introduction of new variants. The next two features, 21A (Delta) and Omicron are dummy variables relating to the subvariant. 21A indicates if a subvariant emerged from that lineage, and Omicron indicates if the subvariant is an Omicron subvariant. Further down on the list are demographic features like the percentage of residents in fair or poor health, unemployment rate, percentage of residents who commute by

transit, and percentage of residents with severe housing problems. Overall, the feature importance results imply that subvariant related features are the most important for prediction, however demographic features do play a role. Our mentor can utilize these features to understand their relationship with human trafficking and the spread of COVID-19 for future projects.

The Random Forest model on county data had a Mean Absolute Error of 1.95 weeks and a Mean Absolute Prediction Error of 17%, meaning that on average, our predictions were 1.95 weeks away from predicting the actual number of weeks to spike, which is about a 17% error. We believe that these are acceptable results for model implementation. However, we only modeled subvariants that did spike. There are many subvariants that do not create a spike in cases. We recommend creating a binary classification model to determine if a subvariant will spike, and if so, using our model to determine how many weeks until peak spike is reached. More detail on this is provided in the Recommendations for Future Work section of the report.

All Additional Modeling and Work Done (Nghi Van Pan (Tyler), Rachel Azarian)

Before deciding on the final problem statement, a lot of background research was required. The topic for this project was very vague, which allowed us to explore topics we were interested in. However, it also required exploration into different problem statements in order to find one that had sufficient data for modeling. One area we explored was modeling COVID-19 spread during the first six months of the pandemic for counties in the ten highest producing agricultural states. This topic was suggested by our mentor since many factors were stable during the first six months of the pandemic that were variable throughout 2021 and 2022. For example, there was a stay at home order, most events were canceled, and many citizens wore masks. With all of these variables constant during the first six months, we may be able to better determine the demographic or population features that contribute to COVID-19 spikes. We collected data on covid cases and demographic features on the counties. After the data was analyzed, it was apparent that the majority of the rural counties in the 10 states had an insignificant number of COVID-19 cases during the first six months of the pandemic. This makes sense as COVID-19 initially spiked in big cities during the early months of the pandemic. Because of this, testing resources were scarce and concentrated in bigger cities, so the case data was likely understated in rural counties. So, there was not sufficient data to model COVID-19 spikes during the first six months of the pandemic.

Additionally, using the same data we collected for our problem statement, we were also able to gather the weekly number of COVID-19 cases per state and county at the peak of the spike. Understanding the weekly number of positive cases and when the peak is expected to happen can help with allocating testing materials, masks, hospital beds, hospital staff, and other resources to help mitigate the effects of the spike. We used the same models, Decision Tree, Extra Decision Tree, Random Forest, XGBoost, and Support Vector Regression, and took a similar project approach. Below are the results of our initial modeling efforts using the county data.

County Test Data Results					
	Decision Tree	Extra Regressor Tree	Random Forest	XGBoost	SVR
MAE	409	463	347	351	745
MSE	21,320,922	27,816,919	21,271,639	20,875,208	41,975,318
RMSE	4,617	5,274	4,612	4,568	6,478
R^2 Coefficient	0.49	0.33	0.49	0.50	0.25

Figure 17. Table with model metrics on county data to predict number of peak weekly cases

We first evaluated the untuned models to get a baseline of scores. To compare the results to our models that predicted the number of weeks for a subvariant to spike, the R^2 score was evaluated. This is the coefficient of determination and indicates how well the model predicts the data. The R^2 coefficients are much lower when trying to predict peak number of weekly cases. This is especially true for Random Forest and XGBoost. When predicting the number of weeks for a subvariant to spike, Random Forest and XGBoost had R^2 coefficients of 0.77 and 0.76 respectively. So, the models performed much better when predicting the number of weeks for a subvariant to spike. We also evaluated the feature importance for the XGBoost model.

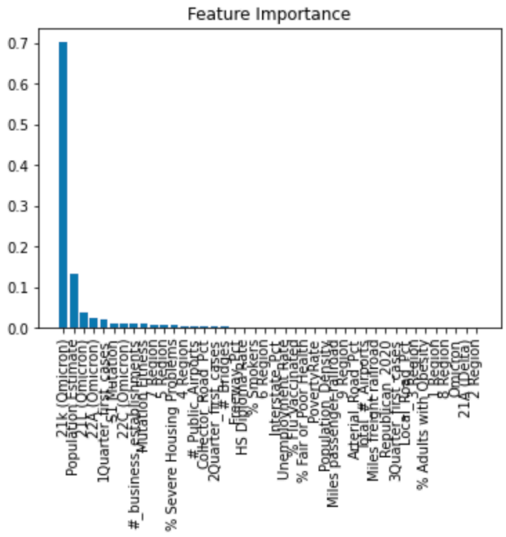


Figure 18. Feature importance of XGBoost model on county data to predict number of peak weekly cases

The most important feature by far was a dummy variable for the emerging lineage of several omicron subvariants. Since the Omicron variant produced the largest number of cases by far, this makes sense. The second most important feature is the population estimate. This is also logical, since the larger the population, the larger the expected spike in cases will be. The remainder of the features did not have any material importance. Due to this, there wasn't much to learn or interpret from this model and it wouldn't provide any useful insights for our mentor. From these results, we decided to focus on predicting the number of weeks for a subvariant to spike.

Additional work was done for the Support Vector Regression model to test the impact of feature selection. Support Vector Regression does perform well in high-dimensional spaces, but reducing noise can improve the results. Random Forest was utilized for feature selection. Below are the Support Vector Results with and without feature selection using the county dataset.

Testing Feature Selection on SVR Model - County Data		
	All Features	**RF Feature Selection**
MAE	2.22	2.75
MSE	10.32	14.51
RMSE	3.21	3.81

Figure 19. SVR county error results before and after performing Random Forest feature selection

The models' parameters were tuned separately to obtain the best hyperparameters. The Random Forest feature selection chose only eight of the 34 features available. As a result, the model had worse performance with reduced features. Due to these results, we decided to use all of the features when evaluating SVR's performance. For future work, testing out different feature selection methods would be beneficial.

In addition to the five models evaluated in the results section, neural networks were also tested on the data. We tested multilayer perceptron to compare to our other model results. Once the parameters were tuned, the R^2 coefficient was only 0.6, which did not perform as well as the other models. We did not have ample time to fully research the utilization of deep learning on the data, so we decided to leave this model out of the results and include it in the suggestions for future work.

Recommendations for Future Work (Nghi Van Pan (Tyler))

As shown from the results of this project, there are areas for improvement. Since we only looked at subvariants that did spike, it would be helpful to look into creating a binary classification model based on subvariant features that will determine if a subvariant will create a spike in cases. There are many subvariants that do not produce a material number of cases, so this will be an important first step to understanding the potential impact of a subvariant. If the binary classification model indicates that the subvariant will produce a spike in cases, our model can be applied to different counties to determine the number of weeks until the subvariant reaches its peak spike.

Additionally, the county data collected is limited to the counties of the top 10 agricultural states, which includes about 6,000 counties. Adding additional data would improve prediction and make the model more generalizable to predict spikes for counties in all states. Also, additional demographic and subvariant-related features would improve prediction. S1 mutations of the subvariant was the most important feature, so it would be interesting to see how

additional subvariant features can improve the model. For an improved learning process, more data and features should be included.

For modeling, the project focused only on supervised learning and mainly on Regression Tree based methods. As the variance between data is significant enough for unsupervised learning methods, such as Principal Component can be utilized to create a low dimensional transformation on the data. Then, this can be applied to support better performance of the supervised learning models. Clustering should also be considered by using Euclidean Distance and group data into k number of group assignments. Then supervised learning can be applied to different groups with similarities. Finally, prediction of a target using time based data with demographic features has challenges with scikit-learn. The process can be further improved with Deep Learning to consider all parameters for a further improved prediction.

Conclusion (Nghi Van Pan (Tyler))

In conclusion, the results of the modeling efforts indicate that the county-level data tends to perform better than state-level data. This is due to the fact that there are many more instances in the county data, so there is more data for the models to learn from. Also, when thinking about how COVID-19 has spread throughout the US over time, it usually spreads and spikes at a more localized level. In practice, we recommend modeling subvariant spikes at a county-level whenever applicable.

As for modeling, Random Forest Regression is the best performing model. This is due to the fact that Random Forest combines all of the predictions from many trees in order to yield the best result, which mitigated overfitting. This model produced the lowest error and highest tree accuracy score before and after tuning the model. When performing 10-fold Cross Validation, Random Forest proved to be the most generalizable model. To gain deeper insights into model performance, we evaluated the model errors by state and by subvariant. It was determined that the model performed the worst on the BA.4 group of subvariants, which can guide future analysis and interpretation of model results. Additionally, the counties of Nebraska yielded the highest errors, due to their inconsistent methods for reporting cases.

Understanding the feature importance of our models provides insights into the demographic features that impact COVID-19 spikes. These features can be evaluated based on their relationship with human trafficking for future work. While subvariant related features proved to be the most important, demographic and population features still did provide valuable insights for the models. The most important demographic features include the percentage of the population in fair or poor health, unemployment rate, percentage of residents who commute by transit, and percent of residents with severe housing problems.

Some future recommendations include the re-curation of data to achieve a larger and more generalized dataset that can be further evaluated with the models. Unsupervised learning should also be considered to support the performance of the models. Deep learning must also be considered for the spatial epidemiology analysis. While our best performing model did provide acceptable error metrics, we recommend combining it with a binary classification model

for predicting if a subvariant will spike or not and if so, then using our model to predict the weeks until peak spike.

Appendix

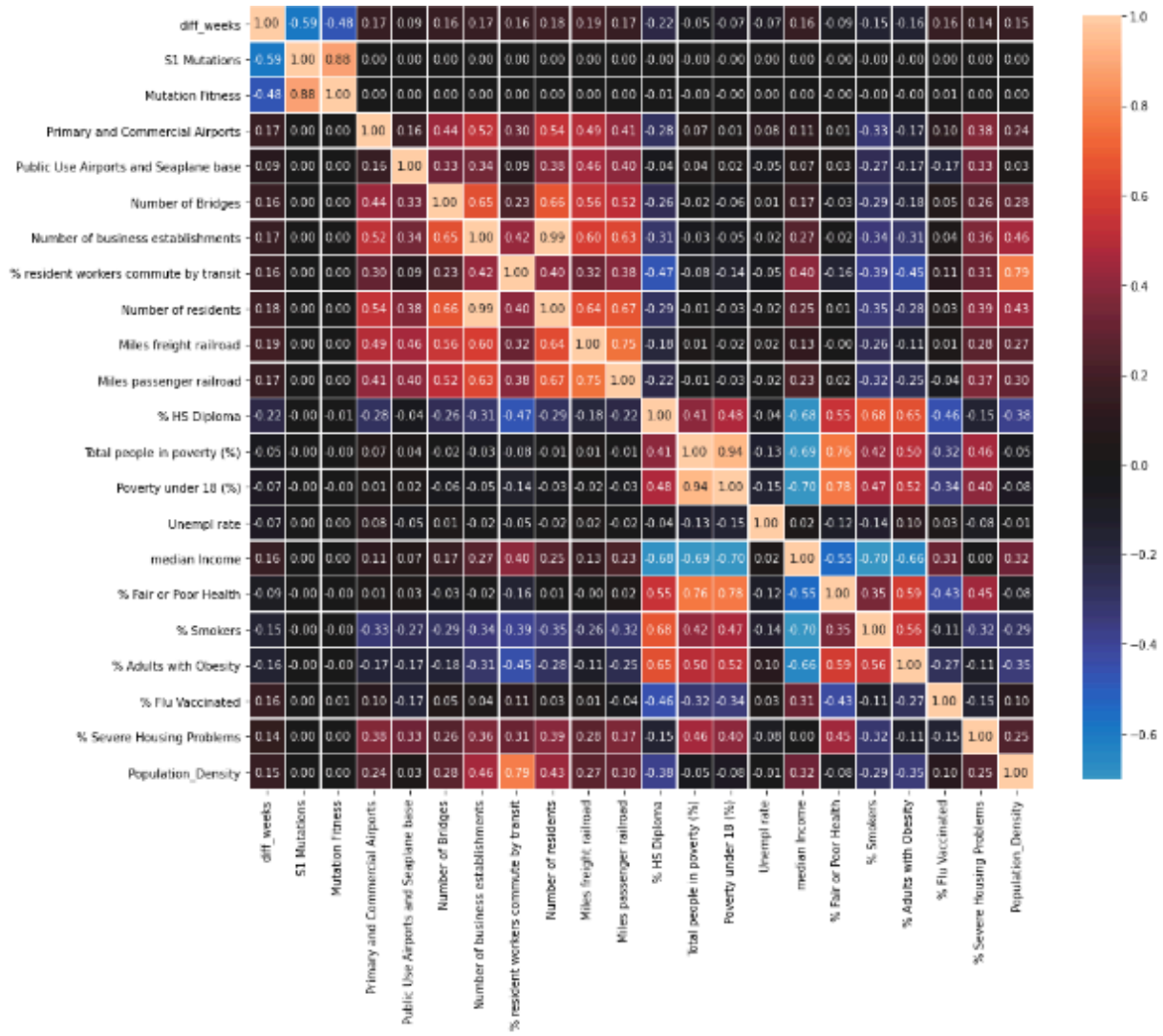


Figure 18. Correlation Matrix with county data features

References

Chen, Y. (2022). Detecting COVID-19 Outbeak with Anomalous Term Frequency. *OAKtrust Texas A&M University Libraries*.

Chyon FA, Suman MNH, Fahim MRI, Ahmmed MS. Time series analysis and predicting COVID-19 affected patients by ARIMA model using machine learning. J Virol Methods. 2022 Mar;301:114433. doi: 10.1016/j.jviromet.2021.114433. Epub 2021 Dec 14. PMID: 34919977; PMCID: PMC8669956.

Mehta M, Julaiti J, Griffin P, Kumara S. Early Stage Machine Learning-Based Prediction of US County Vulnerability to the COVID-19 Pandemic: Machine Learning Approach. JMIR Public Health Surveill. 2020 Sep 11;6(3):e19446. doi: 10.2196/19446. PMID: 32784193; PMCID: PMC7490002.

Nan, B.-G., Zhang, S., Li, Y.-C., Kang, X.-P., Chen, Y.-H., Li, L., Jiang, T., & Li, J. (2022). Convolutional neural networks based on sequential spike predict the high human adaptation of SARS-COV-2 omicron variants. *Viruses*, *14*(5), 1072. https://doi.org/10.3390/v14051072

Sokhansanj, B. A., & Rosen, G. L. (2022). Predicting covid-19 disease severity from SARS-COV-2 spike protein sequence by mixed effects machine learning. *Computers in Biology and Medicine*, *149*, 105969. https://doi.org/10.1016/j.compbiomed.2022.105969

Wiemken, T. L., Khan, F., Nguyen, J. L., Jodar, L., & McLaughlin, J. M. (2022). Is covid-19sseasonal? A time series modeling approach. *MedRxiv*. https://doi.org/10.1101/2022.06.17.22276570

Time Log:

Rachel Azarian:
Meeting with group, mentor, and advisor 9/28 7-8:30 1.5hours
Meeting with team 9/30 6-7 1 hour
Literature Review 9/30 10-12 2 hours
Writing Proposal 10/1 11-1:30 1.5 hours
Meeting with group, mentor, and advisor 10/5 7-8 1 hours
Meeting with advisor for advice and internal discussion, 10/7, 16:30-17:30, 1 hour
Updating proposal 10/9 10:30-11 0.5 hours
Work on proposal slides, 10/10, 4:30 - 5:00, 30 minutes
Met with group to rehearse, 10/10, 5:00 - 6:30, 1.5 hours
Research demographic data 10/11 7:00-8:00, 1 hour
Meet with team before mentor meeting 10/12 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/12 7-8, 1 hour
Meeting with both subteams 10/14 6:00 - 7:00, 1 hour
Research datasets 10/16 11:00-1:00, 2 hours
Research datasets 10/18 3-5:30, 2.5 hours
Tableau visualizations 10/18 10:00 - 11:00, 1 hours
Meet with team before mentor meeting 10/19 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/19 7-8:30, 1.5 hours
Meeting with subteam 10/19 8:30-9:00, 0.5 hours
Research datasets 10/24 12:00-1:00, 1 hour
Meet with subteams and Dr. Khan 10/24 5:30 - 7:30, 2 hours
Create updated proposal presentation 10/24 10:00 - 11:30, 1.5 hours
Practice updated proposal presentation 10/25 4:30 - 5:00, 0.5 hours
Meet with team before mentor meeting 10/26 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/26 7-8, 1 hours
Research California Covid/Variant data 10/28 6-8, 2 hours
Research Iowa Covid/Variant data 10/30 11-12:30, 1.5 hours
Research Illinois and Texas Covid/Variant data 10/30 1-4, 3 hour
Clean and check quality of New York Times Covid Data 11/2 2-5, 3 hours
Meet with team before mentor meeting 11/2 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 11/2 7-8, 1 hours
Clean and check quality of USA Facts Covid Data 11/3 8-9:30, 1.5 hours
Clean and check quality of Covid Act Now Data 11/3 9:30-11, 1.5 hours
Merge and create tableau visualizations of Covid case and variant data 11/7, 6-7, 1 hours
Meet with team to determine how to set up data for modeling 11/7 4-6, 2 hours
Meet with team to create and practice update presentation 11/7 10-12, 2 hour

Meet with team before mentor meeting 11/9 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 11/9 7-8, 1 hours
Meet with team to discuss model/data next steps 11/11 5-6, 1 hours
Clean check quality, create visualizations of Johns Hopkins Covid Data 11/13 8-10, 2 hours
Looking into duplicate values in CDC region data 11/14 3-5, 2 hours
Merging cleaned CDC and JH data and create tableau visualizations 11/15 6-9, 3 hours
Meet with team before mentor meeting 11/16 5:30 - 7:00, 1.5 hours
Meeting with group, mentor, and advisor 11/16 7-8, 1 hours
Meet with team to discuss updated modeling plan based on mentor feedback 11/18 5-6:30, 1.5 hours
Cleaning negative covid case values in state data 11/19 2-5, 3 hours
Meet with group to discuss daily covid case data issues 11/20 5-6 1 hour
Cleaning negative covid case values in state & county data 11/20 6-11:30, 5.5 hours
Merging first and max variant spike dates and preparing data for modeling 11/21 4-8, 4 hours
Meet with group to discuss data and preparing for modeling 11/21 9-10:30, 1.5 hours
Adding some features to state data and one hot encoding 11/22 10-1, 3 hours
Meet with group to discuss updated modeling plan 11/22 5-6, 1 hours
Ran initial random forest model on state data and evaluate performance 11/22 7-9, 2 hours hours
Adding some features to county data and one hot encoding 11/23 9-11, 2 hours
Merging, cleaning, and checking county data 11/24, 10-12, 2 hours
Ran initial random forest model on county data and evaluate performance/errors 11/25 1-3, 2 hours
Met with group to discuss model performance and prepare update presentation 11/28, 10-12, 2 hours
Meeting with group, mentor and advisor, 11/30, 7-8, 1 hours
Add additional features and research groupkfold 12/1 4-6 2 hours
Implement groupkfold for test/train and grid search 12/2 4-5 1 hour
Work on model parameter tuning 12/4 8-9:30 1.5 hours
Work on data section of paper 12/5 1-3:30 2.5 hours
Final model tuning and performance 12/6 2-4 2 hours
Meet with subteam to discuss findings, 12/05, 17-19, 2 hour
Meeting with group, mentor and advisor, 12/7, 7-8, 1 hours
Work on approach section of the paper 12/9 1-3, 2 hours
Meet with team to go over presentation and report, 12/12, 5-7, 2 hours
Presentation Rehearsal , 12/12, 9-11, 2 hours

Work on paper 12/13 9-12, 3 hours
Work on paper 12/15 11-1, 3 hours
Work on paper and final submission 12/16 8-11, 3 hours

Total 120.5 hours

Neha Ravi:
Met with group, mentor, and advisors 9/28 7:00 - 8:30 1.5 hours
Met with both sub teams 9/30 6:00-7:00 1 hour
Feature Selection/Analysis 10/1 9:00 - 10:00 1 hour
Exploratory Data Analysis 10/1 10:00 - 10:30 30 minutes
Meeting with group, mentor, and advisor 10/5 7-8 1 hour
Meeting with advisor for advice and internal discussion, 10/7, 16:30-17:30, 1 hour
Updated proposal, 10/7, 18:00 - 18:30, 30 minutes
Work on proposal slides, 10/10, 4:40 - 5:00, 30 minutes
Met with group to rehearse, 10/10, 5:00 - 6:30, 1.5 hours
Mad Tableau visualizations, 10/12. 4:00 - 5:00, 1 hour
Meet with team before mentor meeting 10/12 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/12 7-8, 1 hour
Meeting with both subteams 10/14 6:00 - 7:00, 1 hour
Research datasets 10/16 10:00-11:00, 1 hours
Meet with team before mentor meeting 10/19 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/19 7-8:30, 1.5 hours
Meeting with subteam 10/19 8:30-9:00, 0.5 hours
Research datasets 10/24 12:00-1:00, 1 hour
Meet with subteams and Dr. Khan 10/24 5:30 - 7:30, 2 hours
Create updated proposal presentation 10/24 10:00 - 11:30, 1.5 hours
Practice updated proposal presentation 10/25 4:30 - 5:00, 0.5 hours
Meet with team before mentor meeting 10/26 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 10/26 7-8, 1 hours
Meet with team before mentor meeting 11/2 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor 11/2 7-8, 1 hours
Extracted John Hopkins data 11/4 7 PM - 1 AM, 6 hours
Meet with team to determine how to set up data for modeling 11/7 4-6, 2 hours
Created Exploratory Data Visualizations 11/7 8-9:30, 1.5 hours
Meet with team to create and practice update presentation 11/7 10-12, 2 hour
Meet with team before mentor meeting 11/9 6:00 - 7:00, 1 hours
Meeting with group, mentor, and advisor 11/9 7-8, 1 hours

Meet with team to discuss model/data next steps 11/11 5-6, 1 hours
Worked on Lightning Slides 11/14 5-6, 1 hours
Meet with team before mentor meeting 11/16 5:30 - 7:00, 1.5 hours
Meeting with group, mentor, and advisor 11/16 7-8, 1 hours
Meet with team to discuss updated modeling plan based on mentor feedback 11/18 5-6:30, 1.5 hours
Researched and Extracted Demographic Data 11/19 9-3, 6 hours
Cleaned Demographic Data and joined it to the reset of the data 11/20 10-12, 2 hours
Meet with group to discuss daily covid case data issues 11/20 5-6 1 hour
Meet with group to discuss data and preparing for modeling 11/21 9-10:30, 1.5 hours
Ran initial XGboost model on State Data, 11/22 3-5, 2 hours
Evaluated Metrics, 11/22 7-8, 2 hours
Meet with group to discuss updated modeling plan 11/22 5-6, 1 hours
Re-ran XGboost model on State Data and tune parameters 11/23 10-11, 1 hours
Ran XGboost model on County Data 11/26 9-11, 2 hours
Evaluated Metrics 11/27, 10-12, 2 hours
Created Exploratory Data Analysis charts 11/27, 4-6, 2 hours
Met with group to discuss model performance and prepare update presentation 11/28, 10-12, 2 hours
Meeting with group, mentor and advisor, 11/30, 7-8, 1 hours
Tested SVR model to compare with Tyler, 12/3, 4-6, 2 hours
Build GridSearch for XGBoost, 12/4, 12/4, 7-10, 3 hours
Meet with subteam to discuss findings, 12/05, 17-19, 2 hour
Meeting with group, mentor and advisor, 12/7, 7-8, 1 hours
Final Report, 12/9, 6-12, 6 hours
Final Report, 12/10, 8-11, 3 hours
Meet with team to go over presentation and report, 12/12, 5-7, 2 hours
Presentation Rehearsal , 12/12, 9-11, 2 hours

Total 92 hours

Tyler Pan
Kickoff meeting, 9/21, 19:00-20:00, 1 hour
Official meeting, 9/28, 19:00 - 20:30, 1.5 hours
Sub Team meeting, 9/30, 18:00- 19:30, 1.5hours
Proposal individual work (part 7-10), 9/30, 23:00-00:00, 1 hour
Data creation, 10/1, 22:00- 23:00, 1 hour
Proposal checking and Finalize, 10/2, 20:30-21:00, 0.5 hour
Meeting with group, mentor, and advisor 10/5 7-8 1 hours

Meeting with advisor for advice and internal discussion, 10/7, 16:30-17:30, 1 hour
Proposal revised and updated, 10/08, 15:00-16:00, 1 hour
Met with group to rehearse, 10/10, 5:00 - 6:30, 1.5 hours
Collect USDA data, 10/11 7:00-8:00, 1 hour
Meet with subteam, 10/12, 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor, 10/12 7-8, 1 hour
Meeting with both subteams, 10/14, 6:00 - 7:00, 1 hour
Research on the topic and connect dataset, 10,15, 3:30-6:00, 2.5 hour
Research datasets, 10/16, 11:00-1:00, 2 hours
Research datasets, 10/18, 3-5:30, 2.5 hours
Tableau visualizations, 10/18, 10:00 - 11:00, 1 hours
Meet with sub team, 10/19, 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor, 10/19, 7-8:30, 1.5 hours
Meeting with subteam, 10/19, 8:30-9:00, 0.5 hours
Find new dataset and merge to match due to needs change, 10/21, 5:00-7:00, 2 hours
Research datasets, 10/24, 12:00-1:00, 1 hour
Meet with subteams and Dr. Khan, 10/24, 5:30 - 7:30, 2 hours
Create updated proposal presentation, 10/24, 10:00 - 11:30, 1.5 hours
Practice updated proposal presentation, 10/25, 4:30 - 5:00, 0.5 hours
Demographic Data search, 10/30, 10-12:30, 2.5 hour
State Data collection, 10/30, 13:00-15:00, 2 hour
Meet with team before mentor meeting, 11/2 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor, 11/2 7-8, 1 hours
Meet with team to determine how to set up data for modeling 11/7 4-6, 2 hours
Meet with team to create and practice update presentation 11/7 10-12, 2 hour
Meet with team before mentor meeting, 11/9 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor, 11/9 7-8, 1 hours
Clean and check data for baseline modeling, 11/11, 2 hours
Encoding columns and feature engineering, 11/12, 10:00-13:00, 3 hours
Summarize finding for updated, 11/12, 21:30-23:00, 1.5 hour
Meet with team before mentor meeting, 11/16 6:00 - 7:00, 1 hour
Meeting with group, mentor, and advisor, 11/16 7-8, 1 hours
Finalize Dataset for Spike detection, 11/18, 9-10:30, 1.5 hour
Meet with team to discuss updated modeling plan based on mentor feedback 11/18 5-6:30, 1.5 hours
Merging Spike Detection with State Data, 11/20, 10:00-12:30, 2.5 hour
Build Baseline Model, Decision Tree, 11/20, 11:30-13:30, 2 hour
Meet with group to discuss daily covid case data issues 11/20 5-6 1 hour
Detect Data pattern and applied other ML algorithm, 11/21, 13:00-15:00, 2 hour
Meet with group to discuss data and preparing for modeling 11/21 9-10:30, 1.5 hours

Meet with group to discuss updated modeling plan 11/22 5-6, 1 hours
Measure and model valuation, 11/23, 14:00-15:30, 1.5 hour
Applied Ensemble Learning and evaluate, 11/25, 20:00-23:00, 3 hours
Testing for Scaling and transformation, 11/27, 10:00-12:00, 2 hours
Meeting with team to summarize findings, 11/28, 22:00-00:00, 2 hours
Meeting with group, mentor and advisor, 11/30, 7-8, 1 hours
Build a SVR with and without scaling data, 12/2, 14-16, 2 hours
Build a GridSearch with parameters for models, 12/3, 15-17, 2 hours
Meet with subteam to discuss findings, 12/05, 17-19, 1 hour
Meeting with group, mentor and advisor, 12/7, 7-8, 1 hours
Working on the Final Report, 12/9, 10-13, 3 hours
Working on the Final Report, 12/11, 9-11:30, 2.5 hours
Meet with team to go over presentation and report, 12/12, 5-7, 2 hours
Presentation Rehearsal , 12/12, 9-11, 2 hours

Total 93 hours