COVID-19 Forecasting: A Data-Driven Approach to Transmission Modeling and Resource Allocation

Team Members:

- Sowpati Raj Kamal
  23110319
- Vadithe Venkat Akhilesh Naik
  23110348
- Pulakurthi Manohar
  23110259
- Praveen Kumar
  23110257
- Yalla Sai Teja
  23110366

Abstract

The COVID-19 outbreak has affected the world in ways never seen before. To understand and lessen its impact, data analysts globally have been working extensively with pandemic-related data. This health crisis has emphasized the value of data analysis and prediction in spotting potential hotspots and planning resource distribution.

Although vast amounts of COVID-19 data have been collected from different platforms, many key questions about how the virus spreads, its high-risk areas, and its overall impact still need answers. This report focuses on examining various data sources such as Kaggle, COVID-19-India, WHO, and Johns Hopkins University. We explore questions like: Are there anomalies in the data? How does the growth of cases relate to other health indicators at the local level? Can we predict how the virus will spread? Which areas should be prepared by increasing hospital capacity?

In this project, our goal is to use publicly available COVID-19 datasets and apply analytical methods to uncover trends, relationships, and potential drivers behind the disease's spread and the corresponding need for healthcare resources.

Keywords: COVID-19, transmission patterns, health data analysis

Acknowledgements: We would like to thank Prof. Anirban Dasgupta and his team from the Computer Science and Engineering department at IIT Gandhinagar for his constant guidance and encouragement throughout the project.

1. INTRODUCTION:

The COVID-19 pandemic has drastically affected global health systems, resulting in over 150 million confirmed cases and more than 3 million deaths by April 2023. A wealth of COVID-19 data is publicly available from multiple reputable sources such as Kaggle, COVID19-India, the World Health Organization (WHO), Analytics India Magazine, the National Institutes of Health (NIH), and Johns Hopkins University.

This extensive data availability enables analysts and researchers to explore critical questions about the pandemic. For instance, analysts can detect inconsistencies in reported figures across sources, study underreporting trends, and explore the relationship between case growth and key healthcare indicators, such as hospital bed availability, the prevalence of pre-existing conditions, and access to medical infrastructure at regional levels.

The pandemic also presents a valuable opportunity to apply data-driven techniques to address complex public health challenges. By examining historical data and demographic variables, we can identify factors contributing to virus spread in specific geographic areas and evaluate regional vulnerabilities.

Given the likelihood of future outbreaks, this analysis serves not only to understand COVID-19 but also to strengthen preparedness for similar public health crises. This project aims to uncover actionable insights that can support decision-makers in developing effective policies, resource allocation strategies, and intervention plans to mitigate the impact of future pandemics.

2. PREDICTION OF COVID-19 CASES USING AUTOREGRESSION MODELS AND FINDING OUTLIERS

Description: In this project, we employed autoregression models to analyze confirmed case trends and predict future patterns. These models enabled us to identify underlying behaviors in the data and make informed forecasts regarding the spread of the virus.

In this project, we applied the Z-score method to detect potential anomalies in the reported data. This analysis enabled us to identify outliers and irregularities that could signal significant changes in the underlying data patterns.

Assumptions Made

1. **Stationarity**:
   We assume that the COVID-19 time series data can be transformed into a stationary series using differencing, enabling ARIMA and SARIMAX modeling.

2. **Predictive Continuity**:
   The models assume that patterns and trends observed in the past data will continue similarly into the near future.

3. **Stable External Environment**:
   We assume there are no sudden, drastic events (like new variants, major policy shifts) that could drastically change the infection dynamics during the prediction period.

4. **Data Accuracy and Completeness**:
   We assume the Indian COVID-19 datasets used (cases, vaccination rates, testing rates) are accurate, consistent, and reliable.

5. **Impact of Exogenous Variables**:
   In SARIMAX, we assume that the external factors (e.g., vaccination data) have a measurable and linear impact on the spread of COVID-19.

6. **Effectiveness of Z-Score Normalization**:
   We assume that applying z-score normalization retains the critical

information in the data without distorting trends needed for forecasting.

2.1 DATASET ANALYSIS

We used three main datasets for COVID-19 data analysis and forecasting in India:

1. **covid_19_india.csv**
   - Daily case counts for confirmed, recovered, and deaths across Indian states.
   - Key columns: `Date`, `State/UT`, `Confirmed`, `Recovered`, `Deaths`.
   - Challenges: Missing entries for some states on certain dates, sudden jumps due to data corrections.
2. **covid_vaccine_statewise.csv**
   - Vaccination statistics (first dose, second dose) for each state over time.
   - Key columns: `Date`, `State`, `Total Vaccinated`.
   - Used as an exogenous variable in SARIMAX to improve prediction accuracy.
3. **StatewiseTestingDetails.csv**
   - Daily testing numbers per state.
   - Key columns: `Date`, `State`, `TotalSamples`, `Positive`.
   - Helped to understand the relationship between testing volume and case detection.

2.2 DATA PREPROCESSING STEPS

- Converted `Date` columns to datetime format.
- Aggregated daily totals to obtain national-level time series.
- Handled missing values by forward-filling where appropriate.
- Normalized data (e.g., z-score normalization) for certain models to stabilize variance.

2.3 PROCESS OF ANALYSIS

1. **Data Collection**:
   - Gathered COVID-19 data (cases, deaths) from Indian datasets.
   - Collected vaccination data and testing details to use as external factors.
2. **Data Preprocessing**:
   - Cleaned missing values using forward-fill techniques.
   - Aggregated state-wise data into national-level totals.
   - Normalized data using z-score where necessary.
3. **Exploratory Data Analysis (EDA)**:

  - Plotted case trends to identify peaks, seasonality, and sudden drops.
  - Analyzed the correlation between testing, vaccinations, and confirmed cases.

4. **Feature Engineering**:
   - Created lag features (past 10-15 days' cases) for regression models.
   - Used vaccinations as an exogenous input for SARIMAX.

5. **Model Building**:
   - Applied Manual Linear Regression using lagged values.
   - Built Autoregressive (AR) models.
   - Developed ARIMA models to handle trend and seasonality.
   - Built SARIMAX models by integrating external vaccination data.

6. **Model Evaluation**:
   - Evaluated all models using RMSE and MAE metrics.
   - Compared model performances to select the best forecasting method.

7. **Forecasting**:
   - Predicted future case numbers using the trained models
   - Analyzed forecast trends for decision-making insights.

2.3 MODELLING AND CROSS-VALIDATION

Modelling:

- Built four models:
  - Manual Linear Regression (using lag features)
  - Autoregressive (AR) Model
  - ARIMA Model
  - SARIMAX Model (with vaccinations as external input)
- Each model was trained using the COVID-19 confirmed cases time series.
- SARIMAX incorporated additional real-world factors like vaccinations to improve prediction accuracy.

Cross-Validation:

- Used a Train-Test split approach:
  - Trained models on ~80% of the data.
  - Tested models on the remaining ~20%.
- Evaluated performance using RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error).
- Compared models based on their error values to select the best-performing model.

2.4 PREDICTION OF COVID-19 CASES USING AUTOREGRESSION MODELS

2.4.1 Methodology

We implemented autoregression models with a lag of 7 days to predict the confirmed cases trend over the last 14, 30, and 60 days. To achieve this, we divided the dataset into two parts: a training set and a testing set (corresponding to the most recent days we aimed to predict). We also fine-tuned certain hyperparameters within the models to improve their performance for specific configurations. The models used are as follows:

2.4.2 ARIMA:

ARIMA is a time series forecasting technique that combines three components: Autoregression (AR), Integration (I), and Moving Average (MA). The AR component uses lagged values of the dependent variable to predict future outcomes, while the MA component relies on past forecast errors for prediction. The Integration (I) component helps to make the time series stationary by differencing the data. ARIMA models are capable of handling both stationary and non-stationary time series. They are relatively straightforward to implement, requiring the estimation of only a few parameters. However, ARIMA assumes that the underlying data is linear and stationary, which may not always hold in real-world applications. The model can also be extended to incorporate seasonality and external variables, resulting in more advanced models like SARIMAX and ARIMA. Due to its ability to capture short-term and long-term patterns, ARIMA remains a widely used method in time series forecasting.

2.4.3 SARIMAX:

SARIMAX (Seasonal Autoregressive Integrated Moving Average with eXogenous factors) is an extension of the ARIMA model that incorporates both seasonal components and exogenous variables. It is a widely used time series forecasting technique across industries such as finance, economics, and epidemiology. SARIMAX models are capable of capturing both temporal and seasonal patterns in data, making them effective for predicting future values. The model includes parameters for autoregression, differencing, and moving averages, along with additional parameters for their seasonal counterparts. Furthermore, it allows the integration of exogenous variables, such as economic indicators or weather data, to enhance forecasting accuracy. Due to their flexibility and ability to model complex data patterns, SARIMAX models are extensively applied in real-world forecasting scenarios.

2.4.4 Autoregression:

Autoregression (AR) is a time series modeling technique that predicts future values based on past observations. In an AR model, the value of a variable at time *t* is regressed on its previous values up to time *t-1*. The model’s order (p) specifies how many lagged observations are included. AR models are commonly used in time series analysis and forecasting as they effectively capture the autocorrelation structure within the data. However, they assume that the data is stationary and may perform poorly when trends or seasonality are present. In such cases, more advanced models like ARIMA or SARIMAX are often more suitable.

2.4.5 Manual AutoReg with Linear Regression:

A manual Autoregressive (AR) model using linear regression involves treating the lagged values of the dependent variable as independent predictor variables in a multiple linear regression framework. This method is particularly effective when the AR model order is relatively small. The process includes selecting the order of the AR model, generating new columns for each lagged value up to the specified order, splitting the dataset into training and testing sets, fitting a multiple linear regression model to the training data, making predictions on the test data, and evaluating the model's performance. However, as the order increases, this approach can become impractical due to the growing number of predictors. In such cases, more specialized time series models like ARIMA or SARIMAX offer more efficient alternatives.

2.4.6 Results and Conclusion:

The results for all the autoregressive models applied to the India COVID-19 dataset for confirmed case predictions are summarized in Table 1. Two evaluation metrics were used for comparison: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Based on the results, we can draw the following conclusions:

1. The AutoReg model demonstrated the poorest performance. This is likely because it assumes that the data is stationary (i.e., the mean and variance remain constant) and primarily attempts to project immediate behavior, which may not hold true for the COVID-19 data.

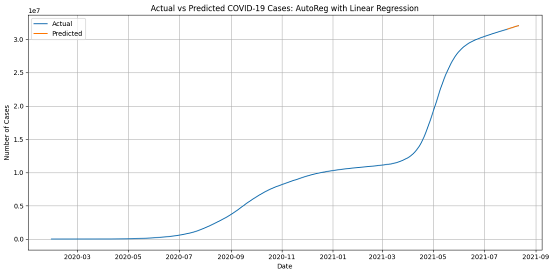
2. The manually implemented AutoRegressive (AR) model using linear regression performed notably well. A possible explanation is that this model is simpler and more direct, yet still effectively captures internal correlations within the dataset. Moreover, it requires less preprocessing and parameter tuning compared to more complex models like SARIMAX. Additionally, in this case, the influence of exogenous variables might not be significant enough to enhance forecasting accuracy, making SARIMAX less suitable.

3. Both the ARIMA and SARIMAX models performed reasonably well over a 30-day prediction horizon but showed a decline in accuracy over longer periods. This degradation could be attributed to suboptimal hyperparameter selection, which limited the models' ability to fully capture the underlying trends in the data. With more careful hyperparameter tuning, these models could potentially achieve better performance.

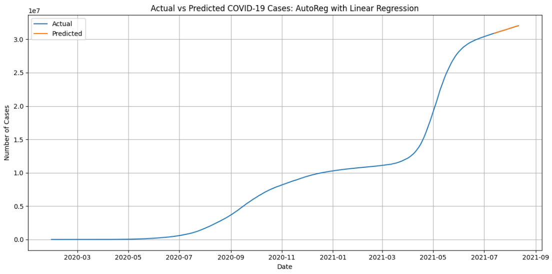
Overall, while SARIMAX and ARIMA models could be improved through further fine-tuning to better capture underlying patterns, the manual AutoReg model with linear regression currently proves to be the most effective for forecasting COVID-19 cases. Its simplicity and ability to capture intricate data structures make it particularly suitable. Furthermore, the presence of significant fluctuations in mean and variance, along with numerous outliers in the real-world data, contributed to the challenges faced by some models, highlighting the unpredictability of the pandemic scenario.

MODEL	NUMBER OF DAYS PREDICTED	RMSE	MAE
Manual AutoReg with Linear Regression	14	6472.987064397512	5002.252077089889
	30	6261.129406271303	4889.239614439259
	60	5139.1214720636535	3899.347637808323
Autoregression	14	38242.5433008433	28300.31845706116
	30	539978.675832496	410254.0821443154
	60	1429269.20690585	950983.7260725963
ARIMA	14	30365.030360179502	23113.359210774037
	30	44762.58581153929	33567.258247119185
	60	567687.3871173041	471350.6003402954
SARIMAX	14	47898.20167850287	36810.347917965744
	30	6170.859421210815	4590.48068803971
	60	943277.2334670689	715452.9929343768

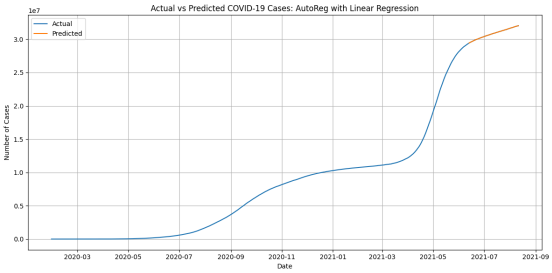
Manual AutoReg with Linear Regression 14 days



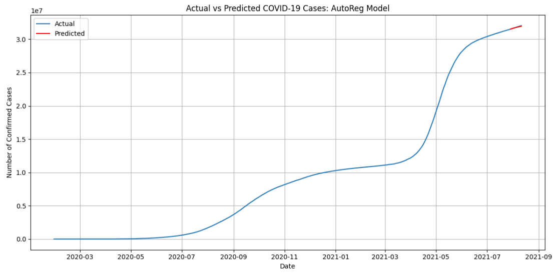
Manual AutoReg with Linear Regression 30 days



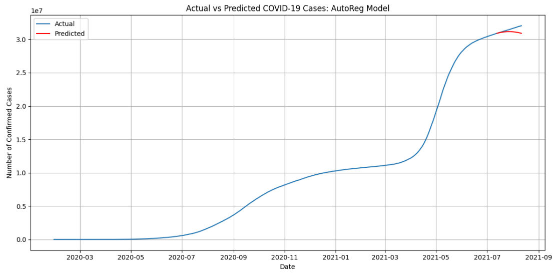
Manual AutoReg with Linear Regression 60 DAYS



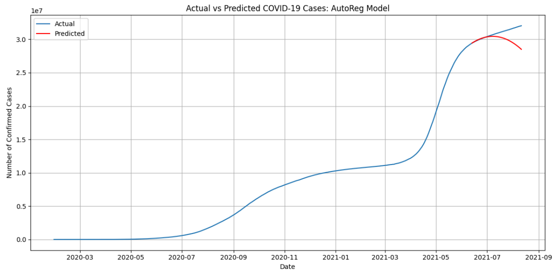
Autoreg 14 days



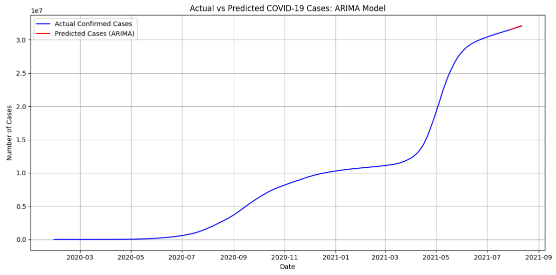
Autoreg 30 days



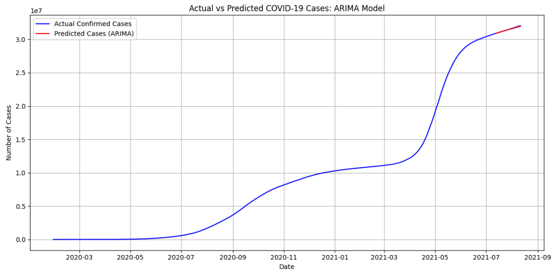
AutoReg 60 days



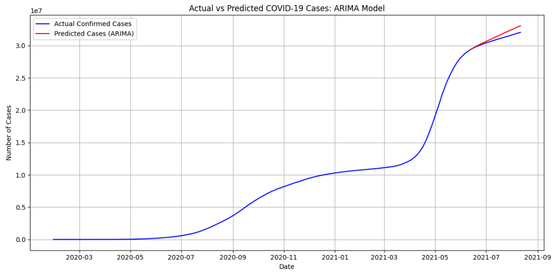
ARIMA 14 DAYS



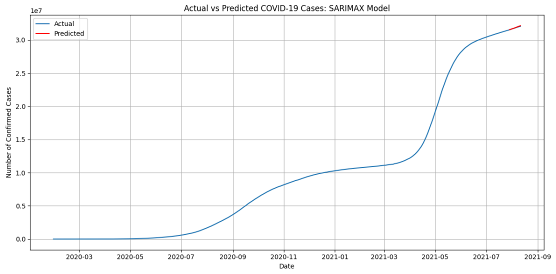
ARIMA 30 DAYS



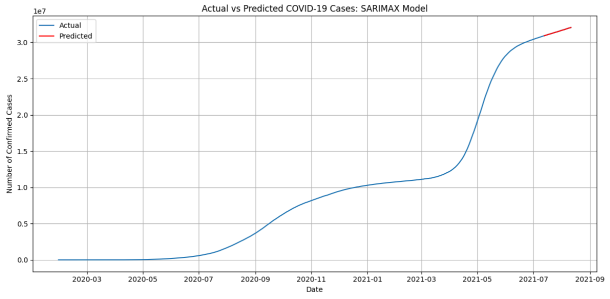
ARIMA 60 DAYS



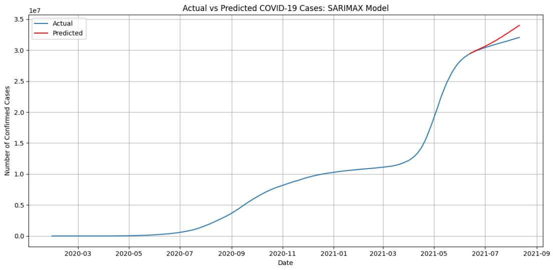
SARIMAX 14 days



SARIMAX 30 DAYS



SARIMAX 60 DAYS



2.6 FINDING OUTLIERS/ANOMALIES IN THE COVID-19 DATASET

2.6.1 Methodology:

In this study, we employed a combination of statistical models and machine learning techniques to analyze trends in COVID-19 confirmed cases and deaths. A key focus of our approach was the careful fine-tuning of model hyperparameters to enhance predictive accuracy and adapt to the unique characteristics of the dataset. By optimizing these parameters, we aimed to improve the reliability of our findings and ensure robust detection of patterns in infection rates and mortality data. Our methodology incorporated multiple analytical techniques, including the Z-score for outlier detection, which helped identify statistically significant deviations that could indicate emerging outbreaks or data anomalies. This systematic approach allowed us to derive meaningful insights while maintaining methodological rigor.

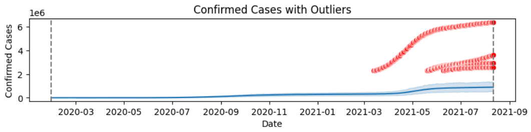
2.6.2 Z-Score Analysis:

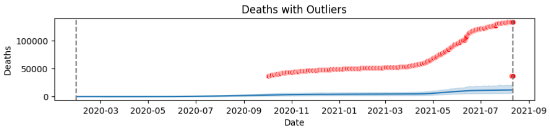
The Z-score is a widely used statistical measure that quantifies how far a given data point deviates from the mean of the dataset, expressed in terms of standard deviations. The formula for calculating the Z-score is as follows:

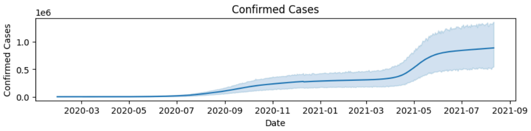
$$Z = (X - \mu)/ \sigma$$

In this equation, X represents the individual data point being evaluated, μ denotes the mean of the dataset, and σ is the standard deviation. A Z-score provides a standardized distance measure from the mean, allowing for consistent comparison across different datasets. Data points with Z-scores exceeding +3 or falling below -3 are generally classified as outliers, as they lie more than three standard deviations away from the mean. However, this threshold can be adjusted based on the analysis's specific requirements or the data's underlying distribution.

In the context of COVID-19 data analysis, the Z-score proved particularly valuable for detecting unusual fluctuations in case counts and death rates. For instance, an abnormally high Z-score in a specific region could signal a potential surge in infections or a reporting anomaly requiring further investigation. While the Z-score is highly effective for normally distributed data, its performance may vary in datasets with significant skewness or multimodal distributions. Despite this limitation, its simplicity and interpretability make it an essential tool for preliminary outlier detection in epidemiological research. Future refinements could involve integrating the Z-score with more advanced anomaly detection methods to improve robustness across diverse datasets.







3. SIR Model:

3.1 Description:

As a foundational step in constructing the model, we first identify the **independent** and **dependent** variables. The **independent variable** in our analysis is time, denoted by *t* and measured in days.

There are two interrelated sets of **dependent variables** in this model:

- The **first set** tracks the number of individuals in each category as a function of time:
    - *S(t)*: number of susceptible individuals,
    - *I(t)*: number of infected individuals,
    - *R(t)*: number of recovered individuals.
- The **second set** represents the **proportional fractions** of the total population *N* in each category. These are defined as:
    - *s(t) = S(t)/N*: the fraction of the population that is susceptible,
    - *i(t) = I(t)/N*: the fraction currently infected,
    - *r(t) = R(t)/N*: the fraction that has recovered.

This dual representation allows us to study the dynamics both in absolute numbers and as a proportion of the total population, which is useful for comparing across different regions or scales.

The differential Equations in the model are:

The SIR model is mainly based on the three differential equations, which tell us about the rate of change of (S, I, R). The three differential equations are:

$dS/dt = -\beta SI$

$dI/dt = \beta SI - \gamma I$

$dR/dt = \gamma I$

where,

S is the number of susceptible individuals, that is, the individuals who are not infected but can get infected.

I is the number of infected individuals

R is the number of recovered individuals, that is, the individuals who are recovered and dead.

β is the rate of transmission (how quickly the disease spreads from infected to susceptible individuals)

γ is the recovery rate (how quickly infected individuals recover and become immune to the disease)

- The first equation outlines how the number of susceptible individuals changes over time. This group decreases as more people contract the infection. The term $-\beta SI$ quantifies the rate at which susceptible individuals become infected through contact with those who are already infected.
- The second equation describes how the infected population evolves. It increases as new infections occur and decreases as infected individuals recover. Here, βSI accounts for the influx of new infections, while γI reflects the rate at which infected individuals recover and exit this category.
- The third equation captures the growth of the recovered population, which increases as people recover from the illness. The term γI represents the recovery rate, indicating how quickly individuals move from the infected to the recovered category, gaining immunity.
- Overall, the SIR model offers a straightforward yet powerful framework for simulating the transmission dynamics of infectious diseases. It has been widely applied to guide public health interventions and policy decisions during epidemic and pandemic outbreaks.

3.2 Assumptions:

1. Constant Population Size: The total population remains unchanged throughout the epidemic, with no births, deaths, or migration affecting the system.
2. Homogeneous Mixing: It is assumed that all individuals mix uniformly, meaning everyone has an equal probability of coming into contact with anyone else in the population, regardless of geographic or social factors.
3. Uniform Infectivity: Each infected individual is equally infectious for the entire duration of their illness—there is no variation in how the disease spreads from one person to another.
4. Permanent Immunity After Recovery: Once individuals recover, they acquire lifelong immunity and cannot contract the disease again.
5. Immediate Infectiousness: There is no delay between infection and the ability to transmit the disease; individuals become infectious instantly upon being infected.
6. No External Interventions: The model assumes that there are no interventions such as vaccinations, quarantines, or treatments that alter the natural course of the disease spread.

3.3 DataSet Analysis:

The dataset is taken from Kaggle and the WHO website. The dataset for different

Countries consist of characteristics like:

- Observation Date
- confirmed cases
- deaths
- recovered people

We took datasets of nearly 100 countries and nearly all of their states/provinces.

3.4 Process of Analysis:

3.4.1. Importing Libraries

Begin by importing all necessary Python libraries required for data manipulation, modeling, and visualization.

3.4.2. Data Preprocessing:

- **Cleaning:** Remove unnecessary columns such as Province/State, Country/Region, Last Update, etc., to focus on relevant fields.
- **Aggregation:** Group the dataset by date and compute the total number of confirmed cases, recoveries, and deaths by summing values across all regions.
- **Cumulative Counts:** Compute the cumulative totals for confirmed cases, deaths, and recoveries over time to track the progression of the outbreak.

3.4.3. Modeling and Forecasting with the SIR Framework

The **SIR model** classifies the population into three compartments: *Susceptible (S)*, *Infected (I)*, and *Recovered (R)*. It simulates how individuals transition between these groups based on specific rates.

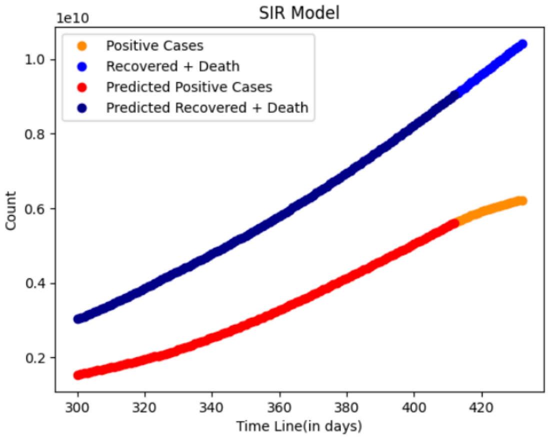
Steps:

- Extract the number of **confirmed**, **recovered**, and **deceased** cases from the processed dataset.
- Calculate the number of **susceptible individuals** as:
  $S(t)=N-I(t)-R(t)$
  where N is the total population.
- Estimate the **transmission rate (β)** and **recovery rate (γ)** from the historical data.
- Compute the **basic reproduction number (R_0)** as:
  $R_0=\beta/\gamma$

3.4.4. Predictive Modeling with Ridge Regression

To forecast how the disease will spread in the coming days:

- Use **Ridge Regression** to model the **β (transmission rate)** and **γ (recovery rate)**.
- Train two separate Ridge models on the **past 15 days** of β and γ data.
- Predict the next **20 days** of transmission and recovery rates.
- Plug the predicted rates into the **SIR equations** to simulate future values of S(t), I(t), and R(t).



Plot: the actual and predicted number of infected and recovered cases over time

The plot displays both the actual data up to the current date and the projected figures for the next 20 days. These predictions are based on estimated transmission and recovery rates. The graph also highlights the expected confirmed cases, which are calculated by summing the predicted infected and recovered individuals.

It appears that the number of deaths is increasing in a roughly linear fashion, while the confirmed cases are beginning to stabilize. This trend aligns with the assumptions of the SIR model, which considers a constant and uniformly mixed population. As more individuals become infected, the number of susceptible people decreases, reducing the likelihood of new infections. Meanwhile, those who were infected earlier either recover or pass away, resulting in the observed patterns.

4. Ridge Regression Model

4.1 Description:

Ridge Regression is a linear regression technique enhanced with L2 regularization to prevent overfitting by penalizing large coefficients. In our implementation, we use polynomial features to allow the model to capture non-linear relationships between inputs and the target variable.

We started with an XGBoost regression model, which predicted ICU demand very accurately. However, due to its complexity, there was a chance it might overfit the data. To check and balance this, we also used Ridge Regression with polynomial features. Ridge helps prevent overfitting by limiting extreme coefficients. It works well with related features and provides more stable, interpretable predictions . Using both models ensured our results were accurate, reliable, and not just due to model complexity.

In this analysis, we treat regional-level healthcare indicators (like population and available hospital beds) and projected infection statistics as features to predict ICU bed requirements.

Modeling Objective: Predict the number of ICU Beds Needed after 12 months, given early indicators and infection rate scenarios.

4.2 Assumptions

- **Independent Regions:** Each hospital referral region (HRR) is considered an independent data point.
- **Well-Represented Infection Scenarios:** The 20%, 40%, and 60% infection scenarios are assumed to reflect realistic severity levels.
- **Fixed Features:** Data points such as population and hospital beds are assumed to remain constant over the modeled period.
- **Accurate Infection Projections:** The projected infection-related values used for training are assumed to be derived from valid epidemic models or historical analysis.
- **Numeric Inputs:** All features are numerical and preprocessed to be model-ready.

4.3 Dataset Analysis

The dataset includes data across 300+ hospital referral regions (HRRs) and covers three modeled scenarios of pandemic infection levels (20%, 40%, 60%). It includes:

- Adult and senior population counts
- Available and total hospital/ICU beds

- Projected infected and hospitalized individuals
- ICU beds needed across time horizons

Source: Data extracted from HRR-level healthcare scorecards and simulated infection scenarios.

4.4 Process of Analysis

4.4.1 Importing Libraries

Used Python packages: `pandas`, `numpy`, `matplotlib`, `sklearn`

4.4.2 Data Preprocessing

- Removed invalid rows (e.g., notes in data like "*Based on a 50% reduction in occupancy").
- Converted numeric fields stored as strings (with commas) into float.
- Removed missing data and clipped outliers in the target using IQR-based filtering.

4.4.3 Feature and Target Selection

Features Used:

- Adult Population
- Population 65+
- Available Hospital Beds
- Available ICU Beds
- Total ICU Beds
- Projected Infected Individuals
- Projected Hospitalized Individuals
- Infection Rate

Target Variable:

- ICU Beds Needed (Twelve Months)

4.4.4 Train-Test Splitting

The dataset was split into 80% training and 20% test sets. Stratification was done based on the Infection Rate to ensure each severity level is represented in both sets.

4.4.5 Modeling and Cross-Validation

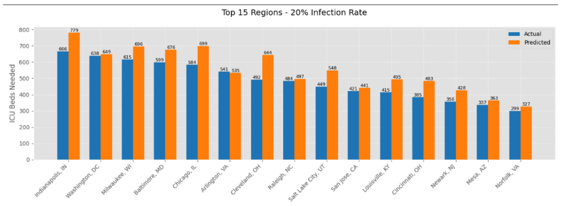
A machine learning pipeline was created using:

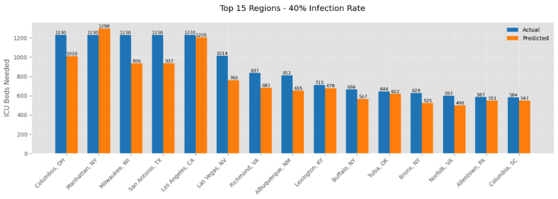
- `StandardScaler()` – to normalize features
- `PolynomialFeatures(degree=2)` – to allow the model to capture interaction and non-linear effects
- `RidgeCV()` – Ridge regression with automatic alpha tuning via 5-fold cross-validation
- Test metrics (RMSE, MAE, MAPE) were used for evaluation.

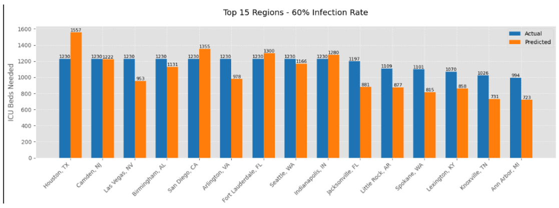
4.5 Results

- The model demonstrated consistent performance across all infection rate categories.
- Cross-validation and test set metrics indicated strong generalization.
- Key influential features included: Projected Infected Individuals, Adult Population, and Available ICU Beds.

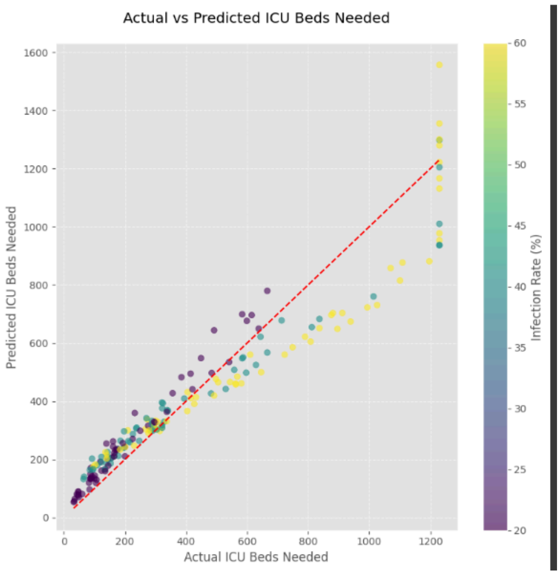
The grouped bar charts for the 20%, 40%, and 60% infection scenarios show comparisons between actual and predicted ICU bed needs across the top 15 most impacted regions.







For the 20% infection rate scenario, predictions generally align well with actual ICU bed needs, with only minor overestimations in a few regions. As the infection rate increases to 40%, the model maintains good performance, though a few instances of underprediction emerge, especially in regions with moderate actual demand. At the 60% infection rate, the model continues to follow the actual trend but tends to be more conservative, slightly overpredicting in several high-demand areas. Overall, the model exhibits strong consistency across all infection rate scenarios, effectively capturing both linear and nonlinear patterns in ICU demand, though extreme values occasionally show a slight prediction bias.



The scatter plot comparing predicted vs actual ICU beds further validates model accuracy. The majority of points align along the red dashed line (representing perfect prediction), indicating a strong correlation. Points are colored by infection rate, showing consistent prediction quality across infection scenarios. Slight deviations above or below the line suggest modest over- or under-estimation in a few regions, but overall, the distribution confirms that the model is effective at capturing the true demand for ICU beds.

In cases where larger deviations occur, such discrepancies may be attributed to real-world variables not present in the dataset. For example, successful early lockdowns, efficient public health measures, variations in hospital capacity mobilization, or regional reporting inconsistencies can significantly affect the actual ICU bed utilization, making it diverge from the predicted value.

5. XGBoost Regression:

5.1 Description:

XGBoost uses decision trees as weak learners, training a series of trees where each one attempts to correct the errors made by the previous one. The method minimizes a loss function that measures the discrepancy between predicted and actual values, employing gradient descent for optimization. This approach is versatile, as it can work with any differentiable loss function and leverages gradient descent to fit the model. The term "gradient boosting" comes from this process, where the model iteratively minimizes the gradient of the loss, similar to how a neural network operates. XGBoost enhances traditional gradient boosting in several key areas.

5.2 Assumptions:

- Since there is no direct relationship between health indicators such as cancer, respiratory diseases, and other similar factors, we expect these features to have lower importance when training the model using Extreme Gradient Boosting.
- Features like the number of confirmed cases in the first 10 days or first 50 days should be more influential due to their direct relevance.

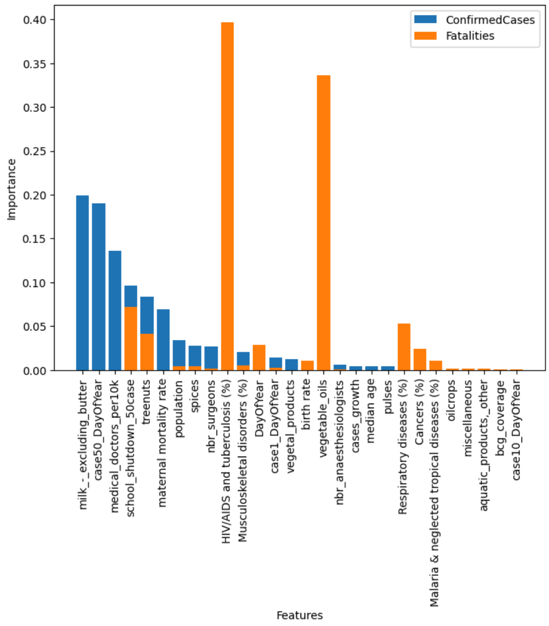
5.3 Dataset Analysis:

5.3.1. Preprocessing:

- Remove the "Province_State" column from the DataFrame.
- If the DataFrame contains a "ConfirmedCases" column, compute the cumulative maximum of confirmed cases for each location. This ensures that the "ConfirmedCases" column reflects only the highest number of confirmed cases for each location up to the current date. Similarly, if a "Fatalities" column exists, calculate the cumulative maximum of fatalities for each location to ensure that the "Fatalities" column shows the maximum number of fatalities up to the current date.
- If the DataFrame lacks a "DayOfYear" column, create one by extracting the day of the year from the "Date" column using the dayofyear attribute of the Pandas datetime object.

5.3.2 Fit an XGBoost model to the training data and return the root mean squared logarithmic error (RMSLE) along with the trained model object.

5.3.3 Plot the top 20 most important features for predicting both "ConfirmedCases" and "Fatalities".



Plot: Top 20 important features identified by XGBoost for predicting COVID-19 *Confirmed Cases* and *Fatalities*.

5.4 Observations:

The bar plot highlights the top 20 features contributing to the XGBoost regression model for predicting both *ConfirmedCases* (blue) and *Fatalities* (orange). It shows that variables like **"milk_excluding_butter"**, **"cases10_Day0Total"**, and **"medical_doctors_per10k"** are influential in predicting confirmed cases, while **"HIV/AIDS and tuberculosis (%)"** and **"Musculoskeletal disorders (%)"** have a significant impact on predicting fatalities. Health indicators related to specific diseases (e.g., cancer, respiratory illnesses) show relatively lower importance, supporting our initial assumption about their weaker direct correlation with case/fatality numbers.

References:

- https://github.com/CityOfLosAngeles/covid19-indicators
- https://www.kaggle.com/datasets/allen-institute-for-ai/CORD-19-research-challenge
- https://www.kaggle.com/code/nxpnsv/logistic-xgb-hybrid#Prepare-submission