Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
CHAPTER ELEVEN

Machine learning modeling techniques and statistical projections to predict the outbreak of COVID-19 with implication to India

W. Regis Anne\textsuperscript{a} and S. Carolin Jeeva\textsuperscript{b}
\textsuperscript{a}PSG College of Technology, Coimbatore, India
\textsuperscript{b}Karunya Institute of Technology and Sciences, Coimbatore, India

1 Introduction

The global pandemic, COVID-19, is a human development crisis and the number of deaths due to this virus is increasing rapidly. According to the World Health Organization (2021), due to the spread of COVID-19, countries around the world have implemented many nonpharmaceutical interventions to control the virus. The study by Human Development Group infers that due to the pandemic, the countries have announced lockdown, school closures, and movement restrictions. The World Health Organization during March 2020 announced COVID-19 as a pandemic and it has affected different parts of the globe. This virus easily affects the persons who have less immunity and other health disorders. The people who are affected with this virus suffer from severe cough, cold, fever, tastelessness, and odorlessness. Although this virus originated in Wuhan, China, it has affected the whole world. According to the survey by Worldometer (2021), the coronavirus cases throughout the globe are found to be 222,839,277 and the death rate about 4,601,398 as on September 2021. Also in India, the cases have been steadily rising since March 8, 2020. India has reached 33,096,718 cases and stands in second place out of 223 countries. In order to diminish the spread, lockdown was announced after March 24, 2020 throughout India. Due to the pandemic, the economic growth of countries has dropped due to loss of jobs and nonfunctioning of many industries.
Regression techniques by data modeling are the techniques to solve prediction problems in machine learning. Regression analysis in supervised learning is used for prediction of time series data. This predictive modeling approach explores the relation among the target and independent variable in a dataset and best fit the input and the target variable. The regression approach determines the predictor strength and forecasts the trend of the time series analysis. In this chapter, machine learning regression models such as Lasso, Linear Regression, Ridge, Elastic-Net, Random Forest Regressor, AdaBoost Regressor, LGBM Regressor, and XGBoost Regressor are considered in this study to predict the exponential increase of the mortality rate, number of confirmed cases, and recovery rate. Also the Facebook Prophet Model was used to predict the outbreak and compared with machine learning regression models. For building the models, the real-time dataset was extracted from Johns Hopkins University which includes the number of confirmed cases, death, and recovered daily cases of COVID-19. The models are trained, tested, and compared for their performances based on the parameters $R^2$-squared value, $R^2$-squared modified score, Mean Squared deviation, and Root Mean Square Error. The results are tabulated to observe the best model for pandemic outbreak prediction. Based on the results of predictions from these models, the concerned officials take necessary steps to control the outbreak of COVID-19 pandemic.

2 Literature survey

Francesca et al. (2021) conducted a study on patients hospitalized due to COVID-19 based on geographical area of origin was presented and the results found that there was no difference in the death rate of COVID-19 patients between Europeans and non-European countries. However, the ICU admission rate of patients was found to be high in non-European countries compared to European countries. Williamson et al. (2020) developed a health analytics platform OpenSAFELY that determines the factors resulting in increased mortality rate due to COVID-19 in England. It was found that the death rate of male is higher than females, increase in age, and patients with diabetes, severe asthma, and other medical problems also contribute to the death rate. The study by Balaji et al. (2020) presents the impact of COVID-19 in India. The features considered are gender, age, travel history, and nationality of patients based on the statistical significance. It was found that the age of the person has an association with COVID-19 in India. Yadav et al. (2020) in their study proposed machine learning regression
algorithms by using the tasks namely prediction of COVID-19 based on a particular region, different types of mitigation, the end stage of pandemic, spread rate of COVID-19, and correlation between the virus and the climatic condition. Dharun et al. (2021) in their work proposed machine learning approach to find the relationship between health care, lockdown policies, and the signs of early containment of COVID-19. Vatsal et al. (2020) proposed Prophet model for decision-making based on time series data. The study was carried out to analyze the cases before and after lockdown. Yuri et al. (2020) in their study present the measures taken by countries to face the pandemic situation. The study reports when implementing voluntary, enforceable steps, the COVID cases are reduced. Shaikh et al. (2021) considered machine learning–based model to determine the optimal regression model considering linear and polynomial regression based on India COVID-19 data set. The time series forecasting is used to forecast the COVID-19 cases that occur in future. Nazrul et al. (2021) in their study present the direct and indirect causes for COVID-19. Mohammad et al. (2021) study reports during the COVID-19 how the lockdown has changed health care. The COVID patients are affected with chronic obstructive pulmonary disease. Punn et al. (2020) in their study used various approach in order to develop a mathematical model by learning the daily analysis of COVID-19 cases and thereby predicting the future. Anne and Jeeva (2020) in their study use ARIMA model and Ramanathan and Dhanwant (2020) presented susceptible Infected Recovered model for time series analysis. Kınacı et al. (2021) use a time series analysis and efficiency measurement approach to analyze the pandemic situation due to COVID-19. Namasudra et al. (2021) presented neural network approach in predicting COVID-19 cases and thereby help the health consultants to carry out proper measures to control the COVID-19 outbreak.

### 3 Predicting and analyzing COVID-19 outbreak

The COVID-19 dataset by Philips (2020) has become essential in order to take necessary action to curb the infection. COVID-19 has caused havoc in the lives of people all over the world.

The second wave has caused more damage in certain countries irrespective of the strict social distancing, hand washing, lockdowns, and vaccination. The time frame for this study is from April 2020 to August 2021. The number of reported confirmed cases for few top affected countries is represented in Fig. 1. From Fig. 1 we understand that in the United
States and India, the number of reporting cases is increasing next only to Spain, France, and Italy.

The number of reported, confirmed, recovered, and deaths in India is displayed in Fig. 2. It depicts that in a particular time period, the number of confirmed cases has crossed many lakhs and this calls for the immediate need for analysis and controlling measures in India. The average growth factor, in the case of India, is 1.04252508 that is calculated by the ratio of the sum of growth difference and the length of growth difference. Since the value is above 1, it infers that the new cases are steadily rising. This reflects the collapse of the public health care system in India.

In this chapter, the outbreak is predicted by (1) Machine learning regression models and (2) Facebook Prophet Model. The machine learning regression models were selected for prediction because the models can predict accurately for large dataset and also can learn automatically and predict without any interventions from human side. The COVID-19 dataset generates large amount of time series data on daily basis leading to accurate predictions by these models. Also, the models are able to learn and modify the errors for an optimum cost function. Though the machine learning models predict accurately, the effect of lockdown on the infection rate is explored using Prophet model. The Prophet model works well with daily data and has options for seasonality and lockdown (holidays) that can be explored.

The overall architecture of the model is given in Fig. 3. The COVID-19 dataset was acquired, cleaned, and explored. The dataset was split into the training and the testing dataset. The independent feature \(Y\) is predicted from the dependent features \(x_1, x_2, \ldots, x_n\) using function \(f(x)\). Then the logistic
models are built and trained according to their learning methodology. The logistic models are trained to minimize the bias and variance. Since the bias and variance trade-off is met, high bias and high variance will not affect the performance of the algorithm. After training, the logistic models are evaluated on the test data and various performance metrics are measured. The best logistic models are considered based on the performance metrics to predict the outbreak. Also, the Prophet model was trained with the COVID-19 dataset. The model was trained and cross-validated to tune the hyper parameters. The evaluation metrics are compared with the previously obtained output parameters from the best logistic models. The best optimized model for prediction is selected based on the evaluation metrics.
3.1 Machine learning regression models to predict the outbreak of COVID-19

In this chapter, we are considering the dataset from John Hopkins University which is updated every day at 6 am UTC that contain the cases of COVID-19 (Coronavirus Research Center, 2020). First, decomposition is performed on the dataset to find out the trend and seasonal components in order to predict the time series data with greater accuracy. The time series data show seasonal variation, so additive decomposition is performed which classifies the data of confirmed cases into four components of observed, trend,
seasonal, and residual. The data clearly show a trend in the data with no sea-
sonal components and the residuals show variation with high variability
toward the end.

Machine learning regression models are used to predict the continuous
independent variable based on one or more dependent variables. Various
machine learning regression models such as Lasso, Linear Regression,
Ridge, Elastic-Net, Random Forest, AdaBoost Regressor, LGBM Regres-
sor, and XGBoost Regressor are used to predict the confirmed, death rate,
and recovery rate from the COVID-19 dataset.

3.1.1 Linear regression model
The main advantage of linear regression is to find the linear regression
between the variables. The linear regression is given in Eq. (1).

\[ Y = a + bX \]  

where \( X \) is the independent variable, \( Y \) the dependent variable, \( b \) represents
the slope of the line, and \( a \) is a constant.

3.1.2 Least absolute shrinkage and selection operator model
Lasso is a regression algorithm that is used when two or more parameters are
highly correlated. Regularization is the concept used to avoid overfitting.
The Lasso algorithm uses a shrinkage technique where the coefficients of
determination are shrunk to zero. The main advantages of using Lasso
regression is it reduces overfitting, works well on different data sets, selects
the features, and regularizes the model. When the data set is large with high
correlation, then this algorithm is used. In Lasso, the tuning parameter that
controls the amount of shrinkage is determined by cross-validation. The
cross-validation represents the number of folds considered; in our work,
the cross-validation is set to threefolds to fivefolds. The mathematical form
of Lasso regression is given in Eq. (2).

\[
\sum_{i=1}^{N} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| 
\]  

where \( \lambda \) represents the quantity of shrinkage. The value \( \lambda = 0 \) considers all
parameters, \( \lambda = \infty \) eliminates the features, bias increases when there is an
increase in \( \lambda \) and variance increases when there is a decrease in \( \lambda \). \( \beta_j \) is the
true parameter that has to be estimated from the sample.
3.1.3 Ridge model
Ridge Regression is a regression model and is used when multicollinearity is found in the dataset. In this regression model, L2 regularization method is used which has the outcome of shrinking the coefficients for the input parameters which does not contribute much in the predicting process. The feature weights are updated as the loss function contains another additional squared term. Ridge Regression avoids overfitting. The mathematical form of Ridge is given in Eq. (3).

\[
\sum_{i=1}^{N} \left( y_i - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} w_j \beta_j^2 \tag{3}
\]

where \( w_j \) is the weights used, and ordinary Least Squares is achieved when lambda is zero.

3.1.4 Elastic-Net model
Elastic-Net overcomes the limitations of both Lasso and Ridge Regression. This algorithm is the mixture of Lasso and Ridge Regression which comprises of L1 and L2 penalties. In this algorithm, variable selection and regularization are done simultaneously and are used when the dimensional data are greater than the samples. The Elastic-Net is given in Eq. (4).

\[
\sum_{i=1}^{n} \left( y_i - x_{i}^T \beta \right)^2 + \lambda \left( 1 - \alpha \right) \frac{1}{2} \sum_{j=1}^{m} \beta_j^2 + \alpha \sum_{j=1}^{m} |\beta_j| \tag{4}
\]

where \( \alpha = 0 \) corresponds to the ridge and \( \alpha = 1 \) to the Lasso.

3.1.5 Random Forest model
Random Forest is a metaestimator that creates multiple decision trees on different samples of the dataset. The Gini index or Entropy is used to branch on a node to create the tree. This is done by calculation of impurity on classification due to randomness of selection of samples. The Gini index is used to conclude how the nodes are added to the branch of the decision tree and is given in Eq. (5). Then finally voting is done to combine all the trees to do final prediction by improving accuracy and to prevent overfitting.

\[
\text{Gini} = 1 - \sum_{i=1}^{C} (P_i)^2 \tag{5}
\]
where $P_i$ denotes the relative frequency of the class and $C$ denotes the total number of classes. The entropy is given in Eq. (6).

$$
\text{Entropy} = \sum_{i=1}^{C} - P_i \cdot \log_2 (p_i)
$$

where $P_i$ denotes the relative frequency of the class and $C$ denotes the total number of classes.

### 3.1.6 Adaptive Boosting model

The AdaBoost Regressor is an adaptive regressor where the data are fit and then the weights are adjusted to minimize the error. Many weak classifiers are generated and are ensembled to form a strong classifier. This model is able to capture the nonlinearity in the real world. First, the data are made to learn with the set weight value and weak classifiers are built. Then the weights are modified to build a strong classifier. The learning rate and the number of weak learners can be controlled to define the model accuracy. The model can be cross-validated to optimize the correctness and accuracy. The growth function $H(x)$ that bounds the generalization error is given in Eq. (7).

$$
H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right)
$$

where $x$ is equal to sign of the weighted sum of the outputs $h_t(x)$ of $T$ weak classifiers with the weights equal to $(\alpha_t)$.

### 3.1.7 Light Gradient Boosted Machine model

The Light Gradient Boosted Machine (LGBM) Regressor selects the features automatically and boosts examples with larger gradients resulting in faster training and accurate predictions. In the architecture of LGBM Regressor, the tree grows leaf wise and chooses the leaf whose delta loss is maximum. The modeling rate controls the growth of selecting the gradients according to the learning graph. The learning rate is 0.15. The accuracy can be augmented by increasing the number of leaves and building deeper trees. The only disadvantage of this method is it leads to overfitting.
3.1.8 Extreme Gradient Boosting model

The algorithm eXtreme Gradient Boosting (XGBoost) is derived from Gradient boosting model. The gradient of the loss function is computed using the Eq. (8).

\[
    r_{im} = -\alpha \left[ \frac{\delta L(y_i, F(x_i))}{\delta F(x_i)} \right]_{F(x) = F_{m-1}(x)}
\]

where \( r_{im} \) is the pseudo residuals, \( L \) is the loss function, \( m \) is the iteration number, \( \alpha \) denotes the learning rate, and \( F(x) \) gives the predictions from the model.

The machine learning algorithms are validated using the COVID-19 dataset and are used for predicting the outbreak. Fig. 4 gives the predicted and observed data for various machine learning algorithms on the COVID-19 dataset. The performance metrics for these algorithms are given in Table 1. The Regression accuracy metrics determine the prediction error rates and the performance of the model.

3.2 Performance metrics

The performance metrics used are Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, Root Mean Squared Log Error, Max Error, and \( R \)-squared to validate the model.

The Mean Squared Error (MSE) compares the original and the predicted values and obtains the squared variance over the dataset. This metric is given in Eq. (9).

\[
    \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2
\]

where \( y_i \) is the observed value of the \( i \)th attribute, \( \hat{y} \) is the predicted value of the \( i \)th observation, and \( N \) denotes the number of samples in the dataset.

The Mean Absolute Error (MAE) is determined by considering the comparison between the original and the predicted values and is obtained by the averaged absolute difference over the dataset. The metrics is given in Eq. (10).

\[
    \text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}|
\]
Fig. 4 Predicted vs observed Graph.
Table 1 Parameters of the different machine learning regression models.

| Model/metrics           | Score  | MSE    | MAE    | RMSE   | MSLE  | Max error | $R^2$ error |
|-------------------------|--------|--------|--------|--------|-------|-----------|-------------|
| Lasso Model             | 0.9424 | 67,824 | 47,874 | 67,824 | 0.01725| 419,619   | 0.830217    |
| Linear Regression       | 0.9466 | 63,785 | 42,956 | 63,785 | 0.01521| 419,016   | 0.849838    |
| Ridge Model             | 0.9449 | 66,082 | 44,925 | 66,082 | 0.01683| 425,594   | 0.83883     |
| Elastic-Net             | 0.9294 | 81,945 | 61,696 | 81,945 | 0.02391| 419,554   | 0.752162    |
| Random Forest Regressor | 0.9918 | 84,180 | 57,346 | 84,180 | 0.03077| 541,365   | 0.738463    |
| AdaBoost Regressor      | 0.9767 | 718,760,288 | 65,593 | 89,428 | 0.04089| 472,286   | 0.704833    |
| LGBM Regressor          | 0.9785 | 71,363 | 50,532 | 71,363 | 0.02726| 432,701   | 0.812036    |
| XGBoost Regressor       | 0.9969 | 87,367 | 59,759 | 87,367 | 0.01952| 453,285   | 0.718281    |
where $y_i$ is the observed value of the $i$th attribute, $\hat{y}$ is the predicted value of the $i$th observation, and $N$ denotes the quantity of dataset.

The Root Mean Square Error (RMSE) is determined by the square root of MSE. The metrics is given in Eq. (11).

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

where $y_i$ is the actual observation of the $i$th attribute, $\hat{y}$ is the estimated value of the $i$th observation, and $N$ denotes the number of samples in the dataset.

The Mean Square Log Error (MSLE) is proportional measure of true and predicted values in logarithmic scale. The formula is given in Eq. (12).

$$ \text{MSLE} = \frac{1}{N} \sum_{i=1}^{N} \left( \log(y_i + 1) - \log(\hat{y} + 1) \right)^2 $$  

where $N$ denotes the number of observations in the dataset, $y_i$ is the actual observation of the $i$th attribute, $\hat{y}$ is the estimated value, and $\log(x)$ is the natural logarithm of $x$ ($\log_e x$).

The Max Error denotes the maximum residual error that captures the worst case error between the predicted value and the true value.

$R$-Squared Error ($R^2$) denotes the coefficient of values that are fit when compared with the original value. The formula is given in Eq. (13).

$$ R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} $$  

where $y_i$ denotes the experimental value of the dependent variable and $\bar{y}$ denotes the mean value of $y$ and $\hat{y}$ is the estimated time series.

Table 1 specifies the different parameters and it is inferred that the AdaBoost Regressor and LGBM Regressor perform second best to the other algorithms. XGBoost and Random Forest Regressor perform the best and surpass all the algorithms. Though the machine learning regression models provide high accuracy on prediction of high dimensional data, these methods can be compared with Prophet model for better performance.
Facebook developed the Prophet model in the year 2017. The forecasting can be done for the future specified number of days. The Prophet model by Taylor et al. (2018) is constructed on the supervised regression technique given in Eq. (14).

\[ y(t) = g(t) + h(t) + s(t) + e(t) \]  

(14)

where \( g(t) \) denotes the trend function which is used to model the non-periodic differences in time series values that is represented using Fourier series, \( s(t) \) denotes the periodic changes the daily or weekly or monthly or quarterly or yearly COVID confirmed cases, \( h(t) \) denotes the lockdown that affects the pandemic, and \( e(t) \) the error term which is any unconditional changes in a particular situation. The logistic growth model with respect to the time series COVID-19 dataset is given in Eq. (15).

\[ g(t) = \frac{C}{1 + e^{-k(t-m)}} \]  

(15)

where \( C \) is the carrying capacity which represents the maximum number of people affected with the virus at time \( t \), \( k \) is the rate of growth of the virus, and \( m \) is the offset parameter. The linear piece-wise growth model with respect to the time series COVID data is given in Eq. (16).

\[ g(t) = \frac{c(t)}{1 + e^{(-(k+a(t)\delta)(t-(m+a(t)\gamma)))}} \]  

(16)

where \( \delta, \gamma, a(t) \) are rate adjustments and \( t \) is a series of history points. Internally, the model parameters use Stan (2018) to optimize the model using maximum a priori optimization. The parameters can also be computed using Nonlinear Least Squares Estimation method by using Scipy Curve fit optimization library of python. The COVID-19 dataset is given as input to the Prophet model and the values of the confirmed cases are calculated. The timestamp \( ds \) and the number of confirmed cases \( y \) that is to be predicted is the input data frame to the model. The dataset is used to fit to the model. Later, future data frame, a function of the Prophet model predicts the total confirmed cases for future referenced dates. The parameters \( \beta, \delta, k, m, trend, Y \) and sigma_obs are displayed using the function model.params() from the fbprophet package. Now with these parameters, the forecasting is done. The model parameters are listed below,
Prophet Model Parameters:

{'beta': array([[-0.00021026, -0.00038018, -0.0001145, 0.00028692, 0.00022041, -0.00010876]],
'delta': array([[3.19732253e-07, -2.37462250e-08, 4.51080641e-09, 7.04436707e-08, 6.38785132e-03, 1.29831752e-04, 2.67351316e-02, 1.92588740e-01, 3.81474053e-01, 4.02797917e-01, 1.18428021e-01, 2.71412175e-04, -6.21269821e-09, -5.56374804e-08, -8.55928785e-06, -3.05403869e-01, -3.40221280e-01, -1.72022445e-01, -7779656e-04, -1.40185075e-07, -3.31293438e-09, 2.01034969e-07, 1.07461156e+00, 1.65596694e+00, 2.17609675e-08]],
'k': array([[0.04070934]],
'm': array([[-0.00405455]], 'sigma_obs': array([[0.03471788]],
'trend': array ([-4.05455451e-03, -3.98133627e-03, 3.90811804e03, 1.11838091e+00, 1.12392322e+00])})

The Prophet model shows very good results for linear and nonlinear trends of data. The dataset after the exploratory data analysis is fed to the Prophet model and the configured Prophet object is fitted on to the data. Then the testing data can be given to the configured model to make predictions and it can be visualized and evaluated. The data for prediction are considered for the confirmation cases pertaining to India. The data for India are extracted from the dataset, and the data consist of time series of dates and its number of persons confirmed with COVID-19 for each date. Fig. 5 shows a steep increase in number of positive cases in India.

The Mean Average Percentage Error (MAPE) measures the forecast error by the average number of times the forecast was predicted incorrectly. The Median Average Percentage Error (MDAPE) is the distance between the predicted and actual values. The error diagnostics is presented in Table 2. The minimum error is reported on the sixth day so the future values can be predicted by looking into the past 6 days’ data.

Fig. 6 shows the forecasted values with true cases and the confidence travel. Though the model has predicted accurately, its performance can still be increased by incorporating the other features like lockdown, social distancing, and vaccination.

4.1 Fine tuning lock down in India on Prophet model

The increasing trend of COVID-19 in India and the alarming death rates forced the government to initiate lockdown in various states of India. In this analysis, we have considered a standard lockdown of 21 days and the general
holidays into account. The time period of lockdown considered was from April 1, 2021 to April 21, 2021. The effect of lockdown has certainly reduced the confirmed cases, and the predicted number of cases from the Prophet model accurately captures the lockdown. The number of new cases reached nearly 3 lakhs during the month of May, but the lockdown efforts reduced the cases to 1 lakh from the month of June. This reduction of cases from 3 lakhs to 1 lakh due to lockdown is very well reflected in the model. The Prophet model predicts accurately the decline of positive cases after lockdown as displayed in Fig. 7.

### Table 2 Performance metrics of the Prophet model without lockdown.

| S. no | Horizon (days) | MSE   | RMSE  | MAE   | MAPE  | MDAPE |
|-------|----------------|-------|-------|-------|-------|-------|
| 0     | 3              | 8.72E+12 | 2,953,283 | 1,605,713 | 0.11791 | 0.08859 |
| 1     | 4              | 9.54E+12 | 3,089,469 | 1,683,991 | 0.12526 | 0.09808 |
| 2     | 5              | 1.04E+13 | 3,224,369 | 1,762,106 | 0.13278 | 0.10551 |
| 3     | 6              | 1.13E+13 | 3,359,880 | 1,840,509 | 0.14032 | 0.1175 |
| 4     | 7              | 1.22E+13 | 3,490,687 | 1,917,196 | 0.14814 | 0.12911 |
| 5     | 8              | 1.32E+13 | 3,627,296 | 1,998,473 | 0.15665 | 0.13876 |
| 6     | 9              | 1.41E+13 | 3,753,757 | 2,075,367 | 0.16552 | 0.15286 |
| 7     | 10             | 1.50E+13 | 3,871,026 | 2,147,617 | 0.17382 | 0.1658 |
| 8     | 11             | 1.58E+13 | 3,968,933 | 2,208,384 | 0.18112 | 0.17724 |
| 9     | 12             | 1.65E+13 | 4,064,855 | 2,266,841 | 0.18792 | 0.18793 |
| 10    | 13             | 1.73E+13 | 4,158,632 | 2,323,585 | 0.19396 | 0.19803 |

Fig. 5 Predictions by Prophet model.
The effects of holidays and the different components of weekly, seasonal, and monthly trends are depicted in Fig. 8. The model’s performance can be validated by cross-validation provided by the Prophet model. The cross-validation divides the data into training and testing data and validates the model for the data.

Fig. 6 Trend and weekly components of Prophet model.

Fig. 7 Predictions after lockdown for India on the Prophet model.

The effects of holidays and the different components of weekly, seasonal, and monthly trends are depicted in Fig. 8. The model’s performance can be validated by cross-validation provided by the Prophet model. The cross-validation divides the data into training and testing data and validates the model for the data.
The error diagnostics with induced lockdown is presented in Table 3. The average of MSE of the model is $1.63584 \times 10^{13}$, RMSE is $3.54115 \times 10^6$, MAE is $2.05495 \times 10^6$, MAPE is 0.194318, and MDAPE is 0.175774. Though the model metrics are higher at the beginning, after the training of the model, the metrics shows considerable performance after tuning.

4.2 Training hyper parameters in the Prophet model

The hyper parameters are adjustable parameters in the fbprophet package that can be tuned for optimized models and thereby make accurate predictions. The tuneable parameters in this package are,
changepoint_prior_scale: This effective parameter establishes the trend at various trend change points. The prior scale value is normalized so the most probable value is 0.1. The default of 0.05 works for many time series, but this could be tuned to the range of [0.001, 0.5].

seasonality_prior_scale: This parameter regulates the seasonal pattern of the rise and fall of confirmed cases. The possible values for tuning would probably be [0.01, 10].

holidays_prior_scale: This controls the flexibility to include lockdown effects. This default value can be set to 10. The probable values are tuned on a range of [0.01, 10].

seasonality_mode: Options are ['additive', 'multiplicative']. Default is “additive,” but many business time series will have multiplicative seasonality.

These parameters are tuned using cross-validation to get the best parameter values. The Prophet model has a function cross_validation() to find out the best hyper parameter values for the tunable parameters. For this, a parameter grid of values is formed with the possible values. Then a new model is trained for all the possible combination of values and the best hyper parameter values are selected. These best values can be used to train the model for predictions. Table 4 gives the possible hyper parameter values.

The best fit values for the hyper parameters by cross-validation are: changepoint_prior_scale is 0.5, seasonality_prior_scale is 0.01, and holidays_prior_scale is 0.01. The RMSE error that resulted after the model was trained with the best fit parameters is 1.3186e+13. This proves that

| S. no | Horizon (days) | MSE    | RMSE   | MAE    | MAPE   | MDAPE  |
|-------|----------------|--------|--------|--------|--------|--------|
| 0     | 3              | 6.27E+11 | 791,735 | 523,706 | 0.06732 | 0.063846 |
| 1     | 4              | 8.67E+11 | 931,006 | 606,786 | 0.07712 | 0.068728 |
| 2     | 5              | 1.14E+12 | 1,068,044 | 687,181 | 0.08583 | 0.075562 |
| 3     | 6              | 1.49E+12 | 1,220,729 | 773,787 | 0.094408 | 0.082972 |
| 4     | 7              | 1.89E+12 | 1,375,043 | 862,597 | 0.103471 | 0.090085 |
| 5     | 8              | 2.3E+12  | 1,517,478 | 948,903 | 0.113134 | 0.099895 |
| 6     | 9              | 2.67E+12 | 1,632,800 | 1,020,493 | 0.122923 | 0.106873 |
| 7     | 10             | 3.01E+12 | 1,736,261 | 1,085,523 | 0.13209 | 0.11212 |
| 8     | 11             | 3.37E+12 | 1,835,882 | 1,146,395 | 0.140436 | 0.115872 |
| 9     | 12             | 3.75E+12 | 1,937,323 | 1,206,764 | 0.148252 | 0.119255 |

Table 3 Performance metrics of the Prophet model with induced lockdown.
cross-validation of best fit parameters reduces the MSE by 18%. The code and dataset for this chapter is provided in Anne and Jeeva (2021a,b).

4.3 Execution time complexity of the machine learning models and the Prophet model

The model complexity of machine learning algorithms for prediction can depend on many criteria like the number of features considered, size of the training and testing data, parameters used for optimization, the depth the model achieves, the execution time of the algorithm based on the processing capabilities, the language used to implement the algorithm, and the operating software. The execution time complexities of the models used for prediction are given in Table 5. It is observed that Elastic-Net, LGBM Regressor, and Prophet Model need more execution time than other regressors because of the high prediction accuracy. Techniques like Dimensionality reduction, Principal Component Analysis, and Multiprocessing can be used to reduce the execution time.

Table 4 Hyper parameters for the Prophet model.

| S. no | Hyper parameters       | Values                  |
|-------|------------------------|-------------------------|
| 1.    | changepoint_prior_scale| 0.001, 0.01, 0.1, 0.5   |
| 2.    | seasonality_prior_scale| 0.01, 0.1, 1.0, 5.0, 10.0 |
| 3.    | holidays_prior_scale   | 0.01, 0.1, 1.0, 5.0, 10.0 |

Table 5 Execution time complexity of the prediction models.

| S. no | Models                      | Execution time complexity (s) |
|-------|-----------------------------|-------------------------------|
| 1.    | Lasso Regressor             | 0.060781                      |
| 2.    | Linear Regression           | 0.001045                      |
| 3.    | Ridge Regressor             | 0.00145                       |
| 4.    | Elastic-Net Regressor       | 0.163782                      |
| 5.    | Random Forest Regressor     | 0.073406                      |
| 6.    | AdaBoost Regressor          | 0.033182                      |
| 7.    | LGBM Regressor              | 0.183558                      |
| 8.    | XGBoost Regressor           | 0.011524                      |
| 9.    | Prophet Model               | 0.10627                       |
5 Discussion and implications

COVID-19 is an evolving contagious infection that causes a significant mortality rates in many parts of the world. India stands in second position next to the United States. Initially in the absence of vaccines, public adherence to the control measures failed poorly in India which was a result of the attitude and practices of the people. The number of positive cases numbered to 33,096,718 and the number of deaths to 441,443. This exponential growth due to lack of control measures are predicted accurately by the machine learning regression models and Facebook Prophet Model. The Prophet model has given the predictions on weekly and monthly basis with high consistency. This alarming infection rate and mortality rate resulted in Government of India to intervene in imposing lockdown to control the spread. Other efforts include suspension of domestic and international flights, awareness campaigns, guidelines, and national curfew on the citizens. This impact caused the reduction of infection rate and mortality rate and is well captured by the Facebook Prophet Model. In India, as on September 8, 2021, 707,543,018 have been vaccinated and the number of new cases has reduced drastically. The effect of vaccination can also be included as input parameter and prediction model can be explored. These study findings can be used by policy makers and health care industry to predict in advance the infection ratio and to further take appropriate action for hospital preparedness and relief health care facilities. Tuning the hyper parameter for accurate prediction requires prior knowledge and skills but using Facebook provides easy to use tuneable parameters. In this chapter, the tuned parameters using cross-validation predict the infection rate and mortality rate more accurately and can be extended for other business scenarios. This pandemic has caused financial crisis, work loss, and psychological effects. By the predictions made, people can intuitively expect the nature and length of lockdowns and appropriately plan their work and activities. Implications of lessons learnt from the first wave and second wave can be applied for any other infectious diseases.

6 Conclusion

In this chapter, first, we have analyzed the COVID-19 dataset from the John Hopkins University and have analyzed the top countries affected by COVID-19 during the first and second waves. Second, we have used
machine learning regression models to predict the outbreak. Out of all the models, Random Forest Regressor, AdaBoost Regressor, and LGBM perform better than the other regression algorithms. Third, the Facebook Prophet Model is used to predict the outbreak, and the performance is better than the regression models. Also, the effect of lockdown is simulated in the Prophet model and the model has correctly predicted the current data with minimum error. The growth of confirmed cases has been predicted as stable after lockdown as predicted accurately by the Prophet model.

References
Anne, W.R., Jeeva, S.C., 2020. ARIMA modelling of predicting COVID-19 infections. medRxiv, 1–4. https://doi.org/10.1101/2020.04.18.20070631. Accessed 4 January 2021.
Anne, W.R., Jeeva, S.C., 2021a. Statistical Analysis of COVID-19. Google Colab. 22 April. Available from: https://colab.research.google.com/drive/1dEzTxFmjTdhiSnxxuFCx2u-Z1PqxBt2. (Accessed 10 August 2021).
Anne, W.R., Jeeva, S.C., 2021b. Machine Learning and Prophet Model for COVID-19 Prediction. Google Colab. 22 April. Available from: https://colab.research.google.com/drive/1uwXHPuIenfYlj77xPQRT4Q0v9o9o7t. (Accessed 10 August 2021).
Balaji, M.K., Sankararaman, G., Suresh, S., 2020. A study on impact of Covid-19 in India. Test Eng. Manage. 83, 16056–16062. Available from: https://www.academia.edu/43352226/A_Study_on_Impact_of_Covid_19_in_India. (Accessed 5 February 2021).
Coronavirus Research Center, 2020. COVID-19 Data in Motion. Coronavirus Research Center, John Hopkins University, Maryland. Available from: https://covidinfo.jhu.edu/diagnostic-testing/testing-dashboard/. (Accessed 4 June 2021).
Dharun, K., et al., 2021. Exploring the growth of Covid-19 cases using exponential modelling across 42 countries and predicting signs of early containment using machine learning. Transbound. Emerg. Dis. 68, 1001–1018. https://doi.org/10.1111/tbed.13764 (Accessed 4 March 2021).
Francesca, F., et al., 2021. COVID-19 and geographical area of origin. Clin. Microbiol. Infect. 27, 632.e1–632.e5. Available from: https://www.clinicalmicrobiologyandinfection.com/article/S1198-743X(20)30707-2/fulltext. (Accessed 4 March 2021).
Kmiet, H., Unsal, M.G., Kaap, R., 2021. A close look at 2019 novel coronavirus (COVID-19) infections in Turkey using time series analysis & efficiency analysis. Chaos Solitons Fractals 143. https://doi.org/10.1016/j.chaos.2020.110583 (Accessed 23 April 2021).
Mohammad, A., Shanya, S., Kennedy, S., Davies, G.A, 2021. Impact of COVID-19 lockdown on the incidence and mortality of acute exacerbations of chronic obstructive pulmonary disease: national interrupted time series analyses for Scotland and Wales. BMC Med. 19 (1). https://doi.org/10.1186/s12916-021-02000-w.
Namasudra, S., Dhamodharavadhani, S., Rathipriya, R., 2021. Nonlinear neural network based forecasting model for predicting COVID-19 cases. Neural. Process. Lett., 1–21. Available from: https://link.springer.com/article/10.1007%2Fs11063-021-10495-w. (Accessed 5 June 2021).
Nazrul, et al., 2021. Excess Deaths Associated With Covid-19 Pandemic in 2020: Age and Sex Disaggregated Time Series Analysis in 29 High Income Countries. 373. Available from: https://www.bmj.com/content/373/bmj.n1137. (Accessed 24 April 2021).
Philips, T., 2020. Oxford Covid-19 Government Response Tracker. Github. 8 April. Available from: https://github.com/OxCGRT/covid-policy-tracker/blob/master/data/OxCGRT_latest.csv. (Accessed 4 June 2021).
Punn, N., Sonbhadra, S., Agarwal, S.K., S., 2020. COVID-19 epidemic analysis using machine learning and deep learning algorithms. medRxiv, 1–9. https://doi.org/10.1101/2020.04.08.20057679 (Accessed 5 June 2021).

Ramanathan, V., Dhanwant, J., 2020. Forecasting COVID-19 Growth in India Using Susceptible-Infected-Recovered (SIR) Model. 2004.00696. Available from: https://arxiv.org/abs/2004.00696. (Accessed 5 March 2021).

Shaikh, S., et al., 2021. Analysis and prediction of COVID-19 using regression models and time series forecasting. In: Proceedings of the Eleventh International Conference on Cloud Computing, Data Science & Engineering, pp. 989–995, https://doi.org/10.1109/Confluence51648.2021.9377137 (Accessed 5 May 2021).

Stan, 2018. Open-Source Software. 18 September. Available from: https://mc-stan.org/. (Accessed 4 April 2021).

Taylor, S.J., et al., 2018. Forecasting at scale. Am. Stat. 72 (1), 37–45. https://doi.org/10.1080/00031305.2017.1380080 (Accessed 5 May 2021).

Vatsal, T., Dolly, S., Mamta, M., 2020. An eye on the future of COVID-19: prediction of likely positive cases and fatality in India over a 30-day horizon using the prophet model. Disaster Med. Public Health Prep., 1–7. Available from: https://pubmed.ncbi.nlm.nih.gov/33203489/. (Accessed 24 April 2021).

Williamson, E.J., et al., 2020. Factors associated with COVID-19 related death using OpenSAFELY. Nature 584 (7821), 430–436. Available from: https://pubmed.ncbi.nlm.nih.gov/32640463/. (Accessed 5 May 2021).

World Health Organization, 2021. Coronavirus Disease (COVID-19) Pandemic. World Health Organization. 20 March. Available from: https://www.who.int/emergencies/diseases/novel-coronavirus-2019. (Accessed 2 April 2021).

Worldometer, 2021. COVID Live - Coronavirus Statistics. Available from: https://www.worldometers.info/coronavirus. (Accessed 5 May 2021).

Yadav, M., Perumal, M., Srinivas, M., 2020. Analysis on novel coronavirus (COVID-19) using machine learning methods. Chaos Solitons Fractals 139, 110050. https://doi.org/10.1016/j.chaos.2020.110050 (Accessed 6 May 2021).

Yuri, B., Anne-Sophie, L., McCourt, J., 2020. Initial impact of global risk mitigation measures taken during the combating of the COVID-19 pandemic. Saf. Sci. 128. https://doi.org/10.1016/j.ssci.2020.104773 (Accessed 4 May 2021).