CSE464: Advanced Database Systems
[Fall 2021]

Project Report
Group No. – 5 (Section 2)

Submitted by:

Student ID	Student Name	Contribution Percentage	Signature
2018-1-60-119	Rafina Afreen	25	Rafina
2018-1-60-244	Maria Mehjabin Shenjuti	25	Shenjuti
2018-1-60-242	Md. Tanvir Hossain Joarddar	30	Tanvir
2018-1-60-105	Zubayar Mahatab Md Sakif	20	Sakif

1. Introduction

The COVID-19 has spread rapidly around the world and the current condition is becoming worse day by day. People are infected by this virus at different times of the year. Since it is a global disease, it affects the death rate strongly. Therefore, it needs to be controlled for not spreading rapidly also it is necessary to keep track of that time when this virus spread more, and the number of patients being affected to reduce the damage of this outbreak. But it is difficult to analyse and predict the growth of this disease at different times of the year because with this dataset our current system provides the computerized data in a collective way. So, we need a ML algorithm to map the disease and its progression to overcome this problem. ML are two categories: one is supervised and unsupervised machine learning. Supervised machine learning includes some models like regression model, classification model, times series forecasting model etc. where unsupervised machine learning includes clustering models. In our project we used linear regression and time series forecasting model where for different time, lab-tests, death case etc. as input we perform regression and classification to analyse data for a particular time and predict the number of confirmed cases in future days from this disease. Using this model, we can get early predictions of the status of coronavirus at different times of the year. And get an estimate of how many people are infected with the virus at that time.

2. Data Preprocessing

Here we have used two types of machine learning algorithms Linear Regression and Time series analysis using SARIMA. For this purpose, we performed some pre-processing. They are:

1. Check the null values and dropped it from the dataset.
2. Drop Day column for the Regression model as the date had no significant to predict the output.
3. For time series model, we cast our date column in datatype and set it as our index.
4. We grouped our data by using Day column to merge the same date info together.
5. Then we resampled our data to weekly format.
6. We also dropped Lab test and Death Case column in our time series model.

3. Dataset Characteristics and Exploratory Data Analysis (EDA)

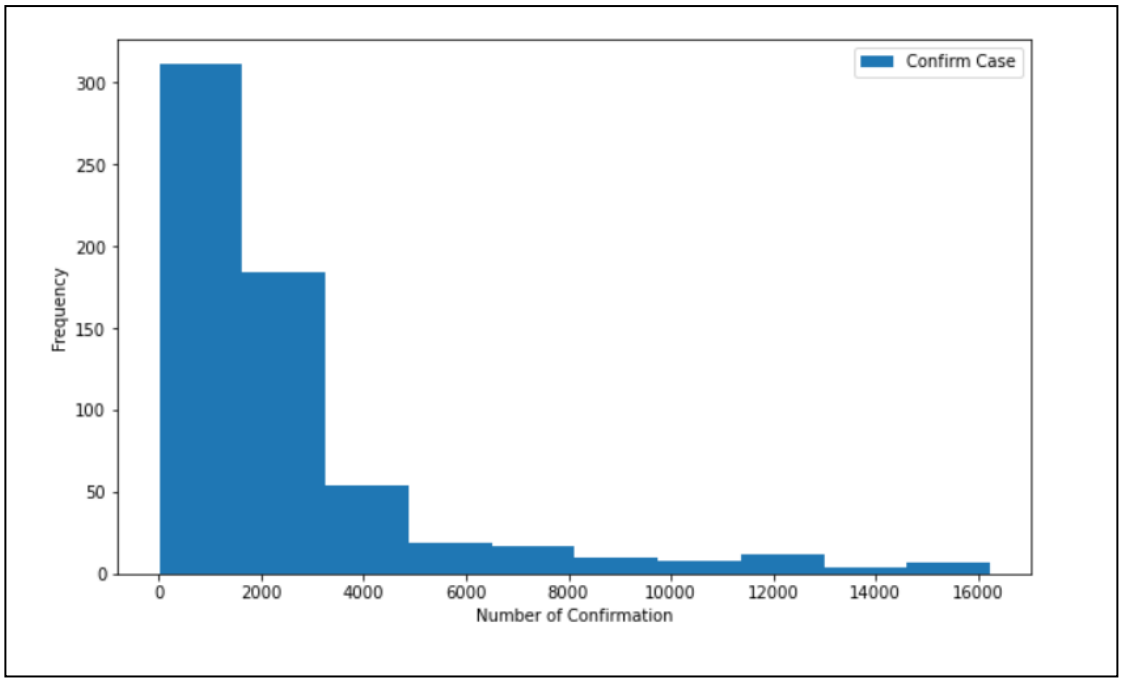
This dataset is taken from the DGHS Bangladesh website.

Covid Dataset				
SN	Column	Number	Data Type	Null Count
1	Day	626	datetime64	0
2	Lab Test	626	int64	0
3	Confirmed case	626	int64	0
4	Death Case	626	int64	0

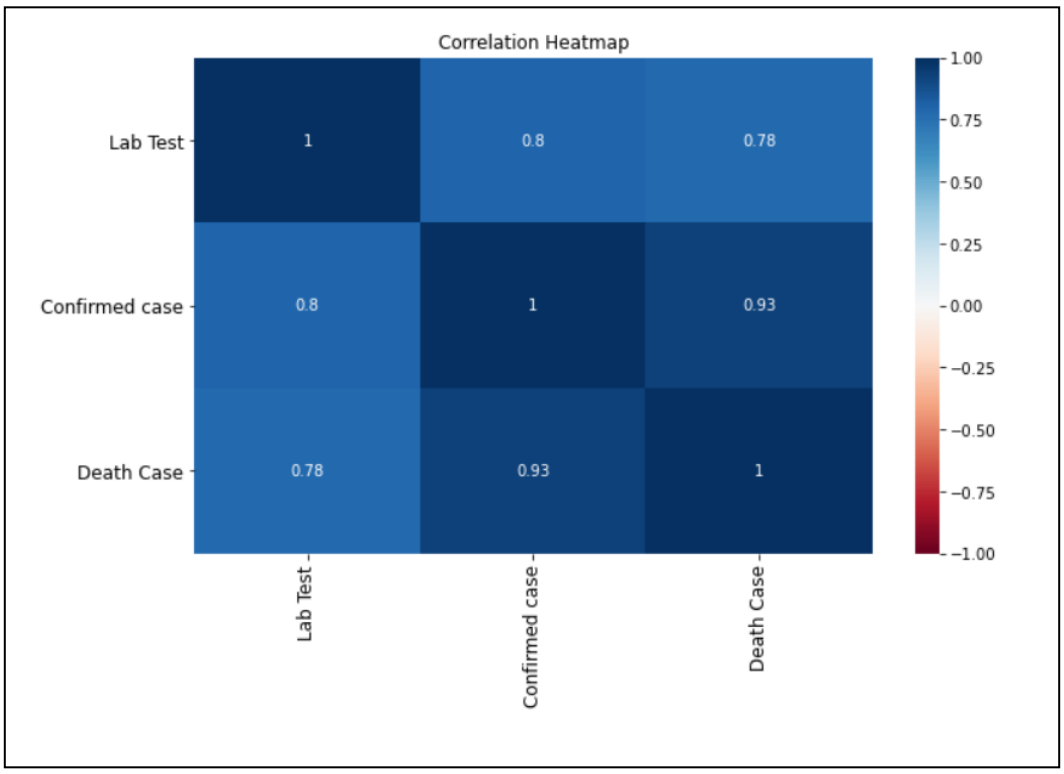
Descriptive statistic of daily cases of covid-19 Dataset.

Dataset	Minimum	Maximum	Sum	Mean	Standard Deviation
Date	04-April-2020	20-December-2021	-	-	-
Lab Test	367	55284	11289181	18033.84	9357.62
Confirmed Case	9	16230	1581282	2526.01	2936.76
Death Case	0	264	28044	44.79	54.53

Histogram of Confirm case in the dataset



Correlation of Covid dataset



4. Machine Learning Models

Linear Regression

Linear regression is a statistical approach and one of the most common approaches to predict research technique. This approach is one of the straightforward machine learning algorithms. Between one or more independent and a dependent variables the linear regression algorithms show the linearly relations. For the linear relation this algorithm finds as a function of the independent values some differences in the dependent variables. There are two types of linear regression a simple linear regression when one independent variable present on the other hand if more than one independent variable present then it called multiple linear regression. In this project we use multiple linear regression model. Using linear predictor function parameters are estimated from the data which unknown model that modelled the relationship between variables. In this model most commonly use the conditional mean of the response that gives independent variables value and less commonly use conditional median or some other methods. Mainly linear regression focusses on conditional probability distributions for the predictors.

Linear regression has many real live uses, most commonly use two categories are- Linear regression can be used to fit a predictive model for prediction, forecasting, error reduction by observed dataset values response and independent variables. The fitted model used to make the prediction of the response after acquiring further values of the independent variables collected without accompanying answer values.

In the independent variable there are variation in the response variables that can be attributed to variations. This linear regression can analyse the strength of the relationship between independent variable and dependent variables determined some independent variables may have no liner relationship with answer values or the independent values may consist redundant values in the answer values.

In our project we have use Multiple Linear Regression in this approach using a single feature it can predict a quantitively response. By using this form, we can find out the multiple linear regression-

$$Y = \beta_0 + \beta_1 * Xi_1 + \beta_2 * Xi_2 + ... \beta_n * Xi_n$$

Here, Y is the Output,

Xi, is the feature,
β_0, is the intercept and
β_n is the coefficient. β_0 and β_n are the model coefficient for feature Xi.

Time Series (SARIMA)

A time series approach basic structure to conduct a time series model and use statistic calculation that analyze data in time series structure to predict and mapped at a time point and deciding. Using research on time series forecasting and modeling, previous outcomes the model gives a particular answer according to predictions. One of the most efficient techniques for imaging natural language processing and expression for large dataset is well known as machine learning. A particular sorting algorithm time series research that cannot process image and voice it can process time dependent pattern models. For example, in a food court, when will more visitors come to the food court it will predict based on the number of visitors at a particular time based on recent visitor come to the food court and the specific time. Some of the problems in time series approach are inequalities in temporal scales, personalities, attributes, properties, dimensionality, except though data is obtained from various resources normally not include construed datasets. In machine learning various models are analysis by time series approach. In this project we use time series Seasonal Arima model to our dataset.

Seasonal Arima**,** If S define the number of time periods seasonality in time series define a regular pattern of changes that repeat over S time until the pattern recurs. Further explain in monthly date there is a seasonality in some particular months greater value tend always, on the other specific month the value always tends to low to occur.

Sometimes in seasonality causes the series differences because some particular time the values not same than other times values. But sometimes it always be same higher or lower for example in summer month always higher on sales of cooling fan. So, there are some seasonality and non-seasonality problem to reduce that problem ACF and PACF behavior work in the model.

The seasonal ARIMA model features both non-seasonal and seasonal factors in a time series model. The model denotes is-

$$\text{ARIMA } (p, d, q) * (P, D, Q)\ S$$

Here,

p = non-seasonal AR order,
d = non-seasonal differencing,
q = non-seasonal MA order,

P = seasonal AR order,
D = seasonal differencing,
Q = seasonal MA order, and
S = time span of repeating seasonal pattern.

5. Description of Models and Associated Parameters

Linear Regression model:

Linear Regression model has two coefficients β_0 and β_i.

- β_0 is the intercept.
- β_i is the coefficient for Xi.

Here,

β_0 = - 483.624

for Lab Test, β_1 = 0.0611 and for Death Case, β_2 = 43.3822

Time series model:

SARIMA is the Most common model used for time series forecasting. It has 4 components.

1. Autoregression AR.
2. Moving Average MA
3. Integrated
4. Seasonality

P = order of AR; current value of y is dependent on how many previous lagged values of current Y. (P from PACF).

Q = order of MA; Future values of Y is dependent of previous lagged values of white noise. (Q from ACF).

D = order of Integrated; Integrated means no of times we difference the data then we have to integrated it back to get the original series back.

S = Seasonal period.

Here, value is:

- P = 0
- D = 1
- Q = 5
- S = 24

6. Performance Evaluation

Linear Regression:

Here, the diagram shows the prediction result by the model Linear Regression. For different number of lab test and death cases this model predicts the Confirmed case of Covid Dataset.

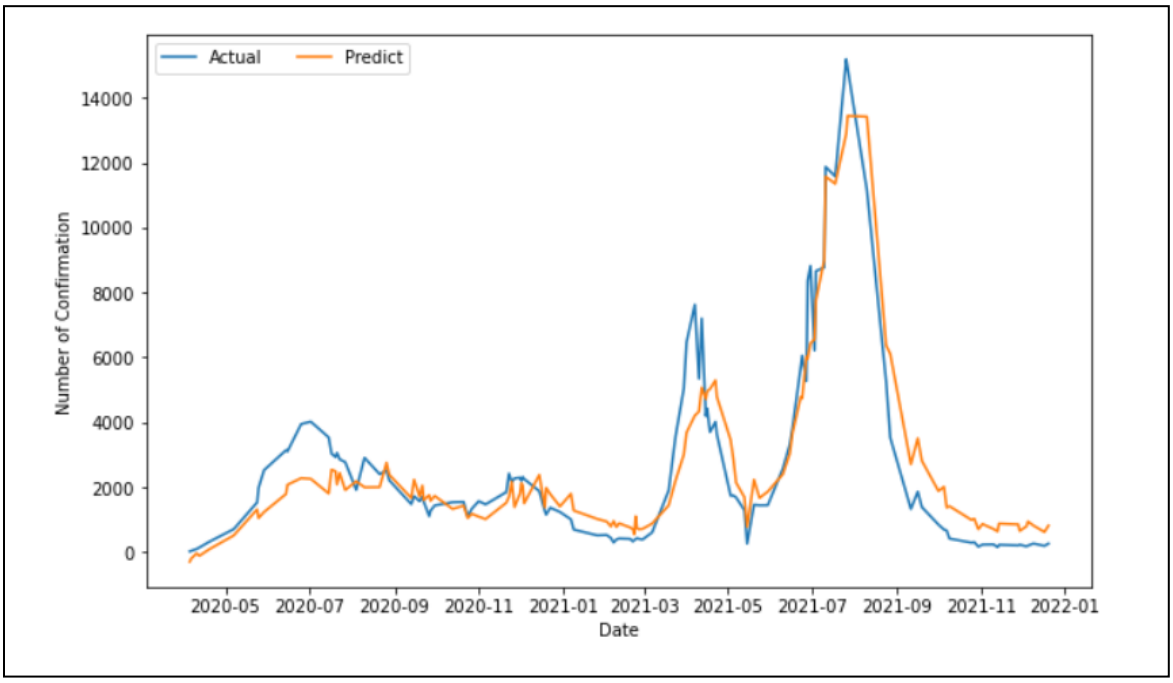
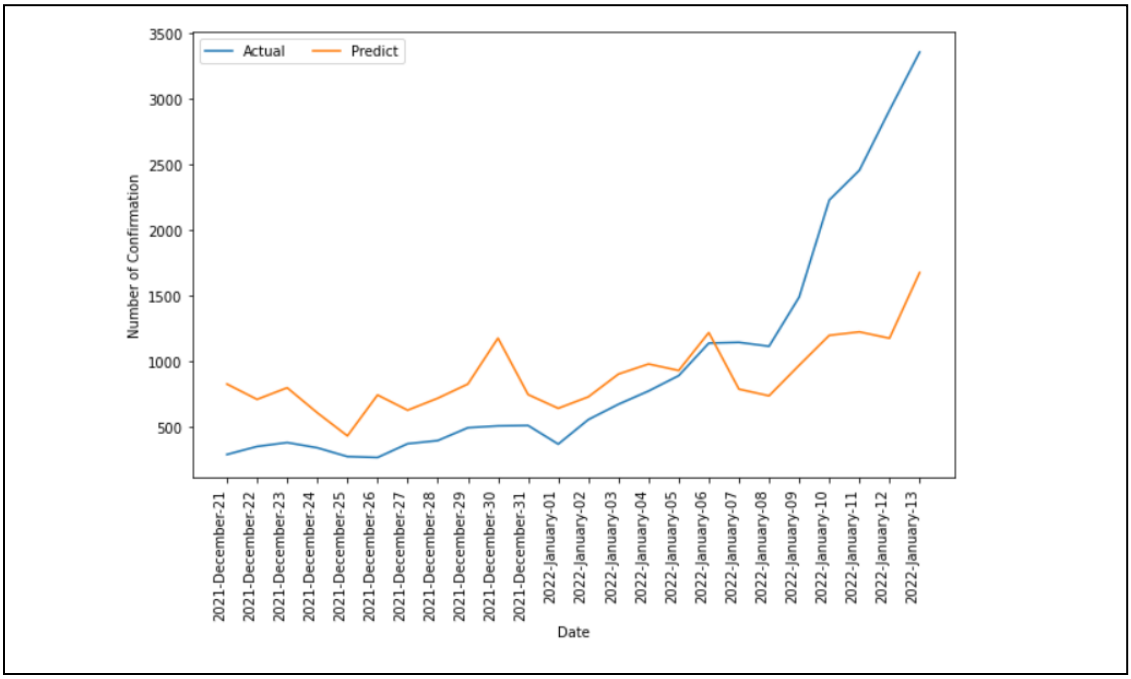


Figure 1: Actual vs Predicted result in LR

And the coefficient of determination of the prediction (r^2 Score) is 0.8823.

Predicting the Confirmed case on new data from 21.12.21 to 13.01.22 with Linear Regression.



Time series (SARIMA):

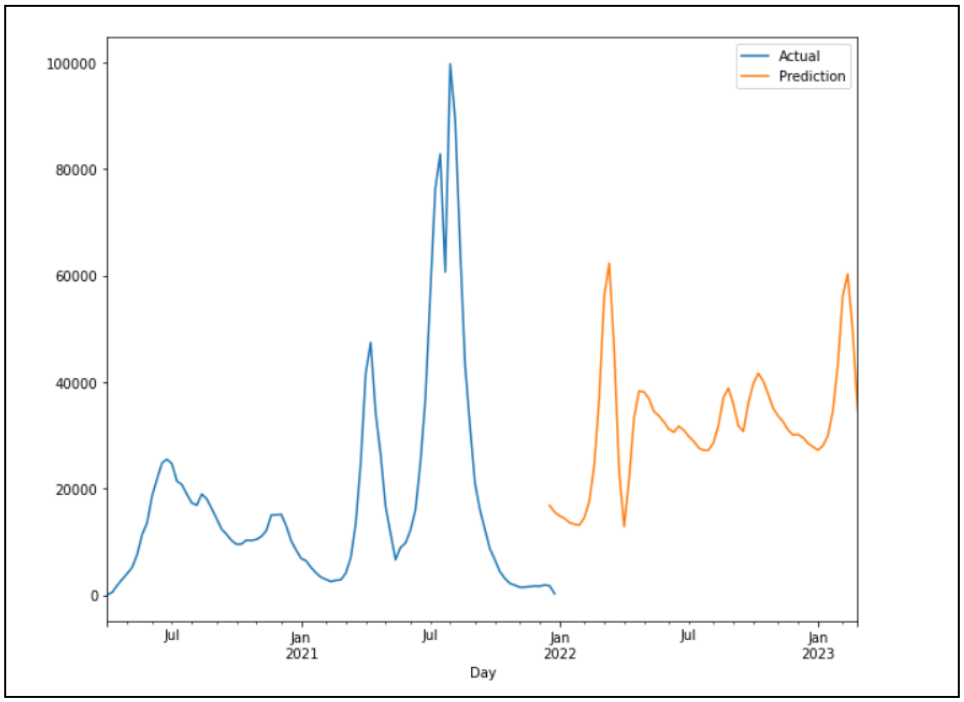


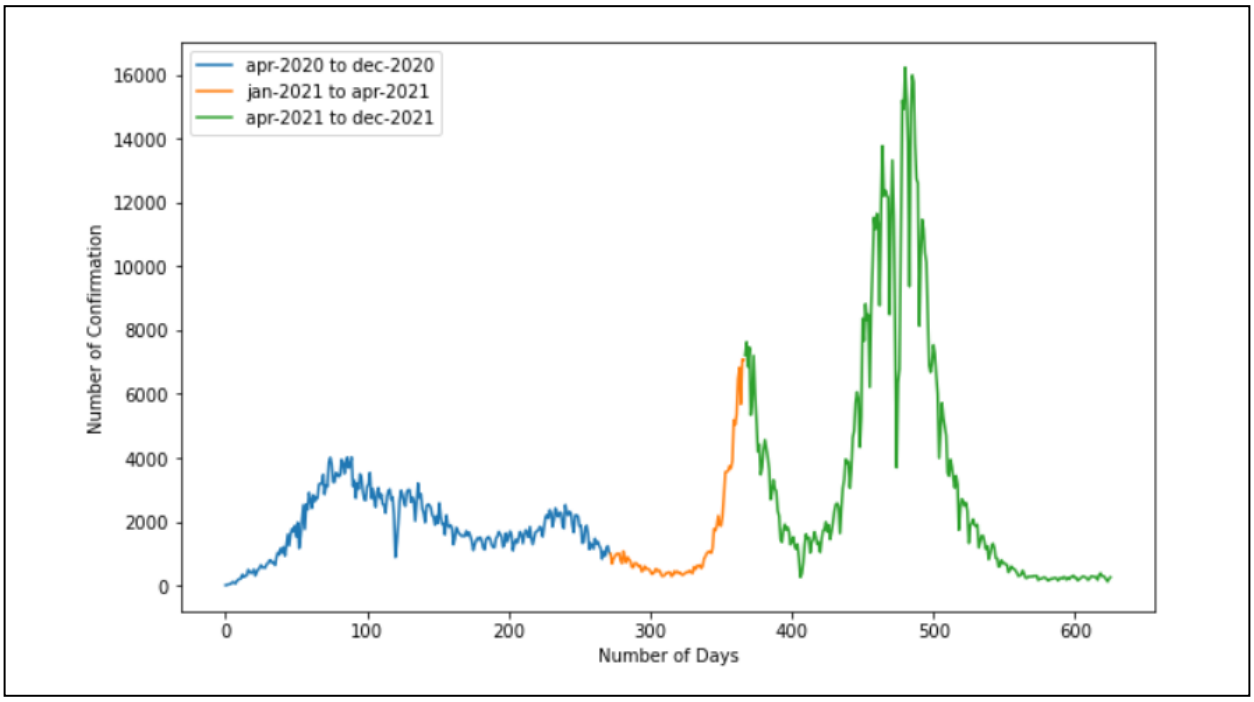
Figure 2: Predicting some future value from the past value covid dataset using SARIMA

7. Discussion

Here we have used two types of models, one is Linear Regression and the other Time Series (SARIMA) model.

In Linear Regression model we predicted the total number of Confirmed case for different Lab test and Death case.

in our dataset we see that there is a huge difference among the confirmed case data between 2020



and 2021. the data in the blue area (from apr-2020 to dec-2020) have lower curve then the data in green area (from apr-2021 to dec-2021). So, it is very difficult for predicting the accurate confirmed case value with the model.

In Linear Regression actual and prediction result graphs (Figure: 1) show much similarity. but at some point, it gives a dissimilar result which is negligible. The Regression model we get the Model Score = 0.8823 means 88% (Figure: 1). So, it can be said that the model quite good for this dataset.

Moreover, for the new real data from 21.12.21 to 13.01.22 the model predicted almost a good result (Figure: 2).

Again, in Time series model where we predict the future data for Confirmed Case. Here also our dataset is not quite good for time series model. because Time series is a special type of model in which one or more variables are measured over time. Each data point in a data set corresponds to a point in time means there need a relationship between different data point in a dataset.

In our dataset the similarity of different point is very must low. It doesn't have a strong upward or downward curve, and there is a minor seasonality in 5/6 months. The dataset is noisy, and Autocorrelation is not so high.

So, the test accuracy of time series model is not very good in our model. but the advantage of this model is we can predict the future value using the past value and there is no need of new data for prediction. By this model we can see the future pick point for a variable in the dataset.

Compare to these model Regression model predicts much more accurate result then the Time Series model. but the Regression can't measure the seasonality. So, at the current situation of the COVID -19 rising the Time Series model predicts more accurate result then the Regression.

Days	Actual	Regression	SARIMA
2021-12-26	1910.0	3848.730374	15587.335680
2022-01-02	3213.0	5182.743082	14871.413302
2022-01-09	7234.0	6291.040766	14412.089361
2022-01-16	10964.0	5225.043725	13588.472324

Figure 3: Predicting some value using both model

8. Conclusion

In our project we have discussed how we can use machine learning to predict the growth of an epidemic called COVID-19. We have used linear regression model and time-series forecasting of supervised machine learning. Mainly, COVID-19 outbreaks are predicted based on past trends.

Challenges:
To predict the number of confirmed cases and death cases many machine learning methods are used. For the accurate prediction by machine learning methods, we face many challenges. with datasets of poor quality will lead to misleading conclusions while training machine learning algorithms. by using any of the machine learning models it will not give significant results if the dataset is not good enough. In our dataset the present values are less similar to past values. In the time series approach we found non-seasonal values more than seasonal values, that's why in SARIMA the result values are not much similar to accurate values.

Opportunities:
Further enhance the prediction accuracy, in the dataset more important parameters can be included. Distribution of age, level of healthcare facilities available, population density, individual and community movements etc parameters need to be included for better accuracy rate in prediction. Other time series models can be used in this Covid-19 dataset.

Appendix

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import itertools
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller

import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import datetime as dt

import warnings
warnings.filterwarnings('ignore')


# # Read Dataset

# In[2]:

df=pd.read_csv('covid_dataset.csv',parse_dates=['Day'])
df.info()

# # Checking Null

# In[3]:

df.isna().sum()

df = df.dropna()
df

# # Dataset Characteristics and Exploratory Data Analysis

# In[4]:


df.describe()
```

```
# In[8]:


df.sum(axis = 0, skipna = True)


# In[5]:


#Density estimation of values using distplot
plt.figure(1 , figsize = (20 , 6))

# manually add the column name of distribution plot
feature_list = ['Lab Test','Confirmed case', "Death Case"]
pos = 1
for i in feature_list:
    plt.subplot(1 , 3 , pos)
    plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
    sns.distplot(df[i], bins=20, kde = True)
    pos = pos + 1
plt.show()


# In[57]:


df['Confirmed case'].plot(legend=True,label='Confirm Case',kind ="hist")
plt.xlabel("Number of Confirmation")
plt.ylabel("Frequency")

# In[58]:


plt.figure(figsize = (10, 6))
s = sns.heatmap(df.corr(),
                annot = True,
                cmap = 'RdBu',
                vmin = -1,
                vmax = 1)
s.set_yticklabels(s.get_yticklabels(), rotation = 0, fontsize = 12)
s.set_xticklabels(s.get_xticklabels(), rotation = 90, fontsize = 12)
plt.title('Correlation Heatmap')
plt.show()
```

```
# In[59]:


plt.rcParams['figure.figsize'] = (10,6)
df['Confirmed case'][:272].plot(legend=True,label='apr-2020 to dec-
2020')
df['Confirmed case'][272:367].plot(legend=True,label='jan-2021 to apr-
2021')
df['Confirmed case'][367:].plot(legend=True,label='apr-2021 to dec-
2021')
plt.xlabel("Number of Days")
plt.ylabel("Number of Confirmation")


# # Regression Model

# In[60]:


# Regression Model

x = df.iloc[:, [0,1,3]]
y = df.iloc[:, [2]]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

d_x_train = x_train.drop(['Day'], axis=1)

d_x_test = x_test.drop(['Day'], axis=1)

lr = LinearRegression()
lr.fit(np.array(d_x_train).reshape(d_x_train.shape),np.array(y_train).re
shape(y_train.shape))

y_pred = lr.predict(np.array(d_x_test).reshape(d_x_test.shape))

result = pd.DataFrame()

result['Days'] = x_test['Day']
result['Actual'] = y_test['Confirmed case']


temp = pd.DataFrame()
temp = pd.DataFrame(y_pred)

result['Predict'] = temp[0].values
```

```
# In[61]:


result = result.sort_values(by = ['Days'])
result


# In[62]:


# Actual vs Prediction Graph
plt.figure(figsize = (10, 6))
plt.plot(result['Days'],result['Actual'])
plt.plot(result['Days'],result['Predict'])
plt.legend(['Actual','Predict'], ncol=2, loc='upper left')
plt.xlabel("Date")
plt.ylabel("Number of Confirmation")
plt.show()


# In[63]:

lr.coef_

# In[64]:


lr.intercept_


# # Score

# In[65]:


print("RMSE is: ", mean_squared_error(y_test,y_pred))

print("R2 score: ",lr.score(d_x_test,y_test))


# In[66]:


# lr.predict(np.array([[12,3]]))


# # Time Series Model
```

```
# In[67]:


df = df.rename({'Lab Test': 'Lab_Test', 'Confirmed case':
'Confirmed_Case', 'Death Case': 'Death_Case'}, axis=1)

df = df.drop(['Lab_Test', 'Death_Case'], axis=1)

df=df.groupby('Day').sum()
# df.head(10)

df=df.resample(rule='W').sum()
df

# In[68]:

plt.rcParams['figure.figsize'] = (10,6)
df.plot();

# # Sesonal Decompose

# In[69]:

#
seasonal_decompose(df['Confirmed_Case'],model='additive',freq=16).plot()
;

seasonal_decompose(df['Confirmed_Case'],model='multiplicative',freq=24).
plot();


# # Stationarity Check

# In[70]:

afdtest = adfuller(df['Confirmed_Case'])

print('p-value of adfuller test is: ',afdtest[1])

if afdtest[1] <= 0.05:
    print('Result: Stationary')
else:
    print('Result: Not Stationary')
```

```
# # Train test Split

# In[71]:

train=df[:55]
test=df[55:]


# # Get P,D,Q values from iterations
# In[165]:

p = range(0,8)
q = range(0,8)
d = range(0,2)

pdq_combinations = list(itertools.product(p,d,q))

rmse = []
order1 = []

# In[166]:


count = 0
for pdq in pdq_combinations:
    count += 1
    print(count)
    pdqs = list(pdq)
    pdqs.append(24)
    pdqs = tuple(pdqs)
    try:
        model = sm.tsa.statespace.SARIMAX(train.Confirmed_Case,order =
pdq,seasonal_order = pdqs).fit()
        pred = model.predict(start = len(train), end=(len(df)-1))
        error = np.sqrt(mean_squared_error(test['Confirmed_Case'],pred))
        order1.append(pdq)
        rmse.append(error)

    except:
        continue

# In[167]:


result = pd.DataFrame(index=order1, data=rmse, columns=['RMSE'])
```

```
# In[168]:


result.sort_values(by='RMSE').head(20)
# # ACF and PACF Test

# In[72]:


fig = plt.figure(figsize=(12,8))
ax1 = fig.add_subplot(211)
fig = sm.graphics.tsa.plot_acf(df['Confirmed_Case'],lags=40,ax=ax1)
ax2 = fig.add_subplot(212)
fig = sm.graphics.tsa.plot_pacf(df['Confirmed_Case'],lags=40,ax=ax2)

# # Seasonal Arima Model building

# In[74]:


final_model =
sm.tsa.statespace.SARIMAX(train.Confirmed_Case,order=(0,1,5),seasonal_or
der=(0,1,5,24)).fit()

prediction = final_model.predict(start = len(df)-2, end=(len(df)+40))
prediction.head(10)


# In[75]:


df['Confirmed_Case'].plot(legend=True, label='Actual', figsize=(10,8))
prediction.plot(legend=True, label='Prediction')


# # New Prediction Regression

# In[100]:


p_data = pd.read_csv('covid_dataset_extra.csv')

# p_data.isna().sum()

p_data = p_data.dropna()
p_data
```

```
pre_data = p_data.drop(['Day'], axis=1)

x_pre = pre_data.iloc[:, [0,2]]
y_pre = pre_data.iloc[:, [1]]

y_pred_extra = lr.predict(np.array(x_pre).reshape(x_pre.shape))

mean_squared_error(y_pre,y_pred_extra)

result_pre = pd.DataFrame()

result_pre['Days'] = p_data['Day']
result_pre['Actual'] = y_pre['Confirmed case']


temp2 = pd.DataFrame()
temp2 = pd.DataFrame(y_pred_extra)

result_pre['Predict'] = temp2[0].values

result_pre


# In[101]:

# Actual vs Prediction Graph

plt.plot(result_pre['Days'],result_pre['Actual'])
plt.plot(result_pre['Days'],result_pre['Predict'])
plt.xticks(np.arange(len(result_pre['Days'])), result_pre['Days'],
rotation=90, horizontalalignment='right')
plt.legend(['Actual','Predict'], ncol=2, loc='upper left')
plt.xlabel("Date")
plt.ylabel("Number of Confirmation")
plt.show()

# # Comaparison Between Two Model

# In[78]:

result_pre["Days"] = pd.to_datetime(result_pre["Days"])
comparison = result_pre.groupby('Days').sum()
comparison = comparison.resample(rule='W').sum()

comparison = comparison.rename({'Predict': 'Regression'}, axis=1)

comparison['SARIMA'] = prediction[1:len(comparison)+1]

# In[79]:

comparison
```