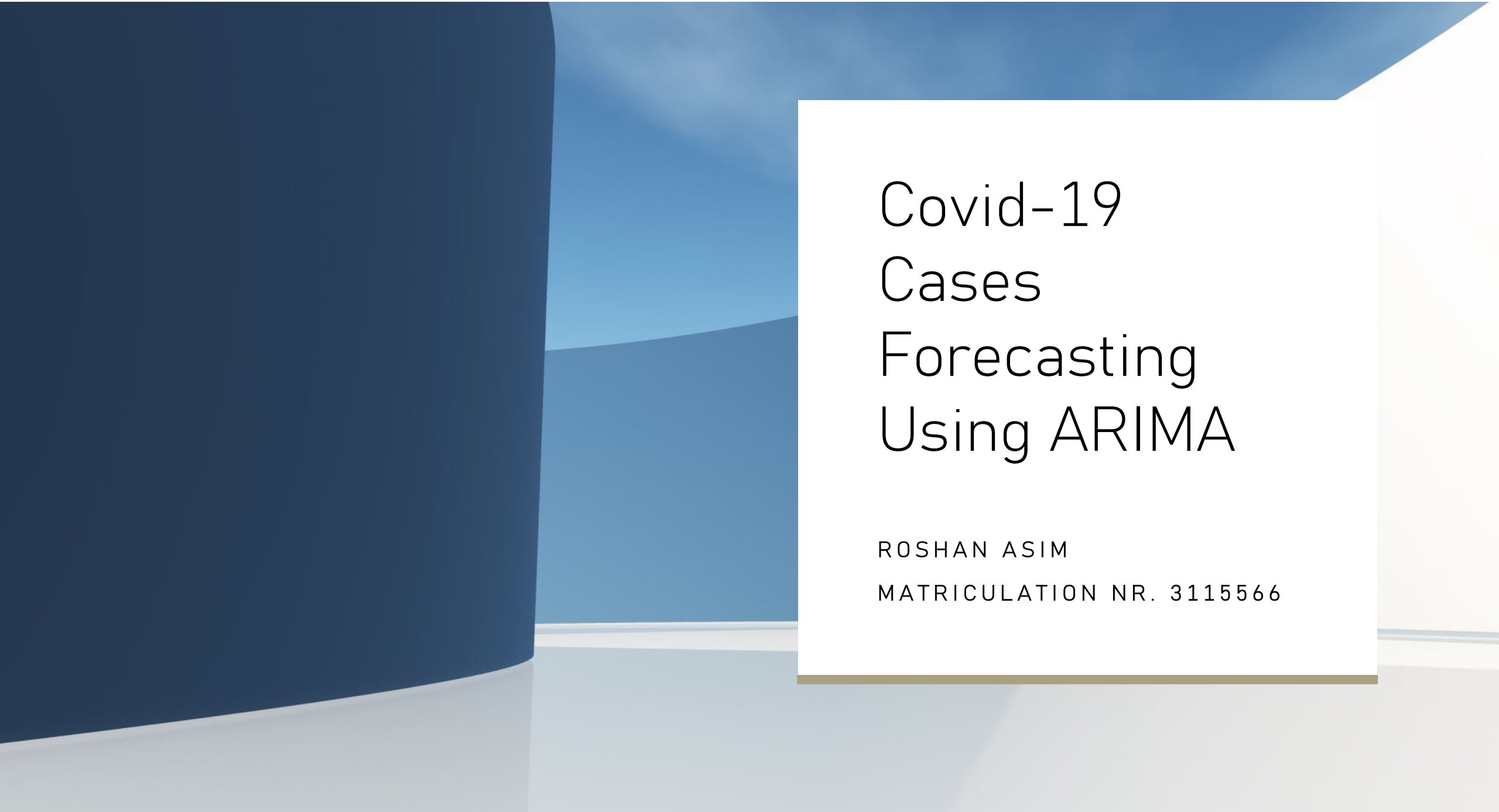


Covid-19 Cases Forecasting Using ARIMA

ROSHAN ASIM

MATRICULATION NR. 3115566

Problem Statement

Use time series analysis to model and forecast new cases of COVID-19 in the US using ARIMA.

Data Source

COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University

DataSet Preparation

- Melting the dataframes to convert the wide format to long format.
- Removing records for other countries except the United States.
- Merging confirmed, deaths, and recovered dataframes.
- Computing active cases by subtracting the recovered and deaths from the confirmed cases.
- Grouping data by date and country to aggregate the data.

Data Transformation

1

Compute new cases, new deaths, and new recoveries by taking the difference of the cumulative cases, deaths, and recoveries.

2

Merge the new cases, deaths, and recoveries data into the main dataframe.

3

Create a time series of the new cases and plot it.

4

Check for stationarity of the time series data using the Augmented Dickey-Fuller (ADF) test and plot the autocorrelation and partial autocorrelation functions.

Application of ARIMA

```
# ARIMA models
model1 = ARIMA(data_ts, order=(1, 0, 1)).fit()
print(model1.summary())

model2 = ARIMA(data_ts, order=(2, 0, 2)).fit()
print(model2.summary())

model3 = ARIMA(data_ts, order=(1, 0, 2)).fit()
print(model3.summary())

model4 = ARIMA(data_ts, order=(1, 0, 3)).fit()
print(model4.summary())

model5 = ARIMA(data_ts, order=(2, 0, 3)).fit()
print(model5.summary())

model_final = ARIMA(data_ts, order=(5, 1, 2)).fit()
print(model_final.summary())

model6 = auto_arima(data_ts, seasonal=False)
model6.fit(data_ts)
print(model6.summary())
```

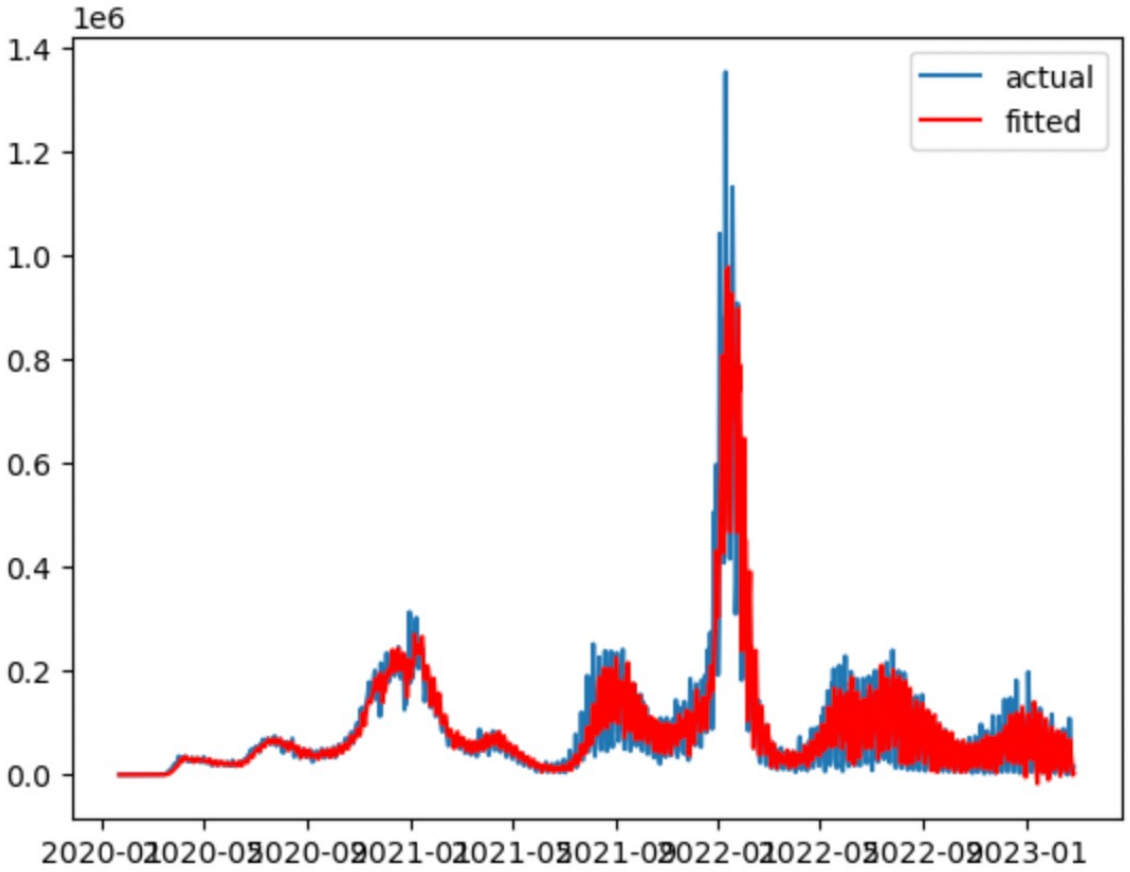
In [60]:

```

print(model1.aic, model1.bic, model1.hqic)
print(model2.aic, model2.bic, model2.hqic)
print(model3.aic, model3.bic, model3.hqic)
print(model4.aic, model4.bic, model4.hqic)
print(model5.aic, model5.bic, model5.hqic)
print(model_final.aic, model_final.bic, model_final.hqic)
```

```
28314.269589472206 28334.400086516318 28321.874068918318
28253.444126888986 28283.639872455155 28264.850846058154
28265.2307846389 28290.39390594404 28274.736383946543
28595.77885315666 28625.974598722827 28607.185572325827
28355.071869876396 28390.300239703593 28368.379708907094
27643.2047403306 27683.458670400705 27658.411690168632
```

Fitted Model



Forecast Future values from Model

In [62]:
```
# predict future values
predictions = model_final.forecast(steps=30)
plt.plot(data_ts, label='actual')
plt.plot(model_final.fittedvalues, color='red', label='fitted')
plt.plot(predictions, color='blue', label='forecast')
plt.legend()
plt.show()
```