Assessment Report

on

"COVID-19 Case Prediction"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

GROUP : 2

GROUP MEMBERS

AMIT YADAV (202401100300067)

AKASH KUMAR GUPTA (202401100300025)

ANSHIKA GAUTAM (202401100300054)

ANANYA SHARMA (202401100300038)

ARPIT DIXIT (202401100300067)

Section: A

Under the supervision of

"BIKKI SIR"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

The COVID-19 pandemic brought significant global challenges, particularly in healthcare planning and public policy. Predicting future case trends using historical data helps authorities prepare and respond effectively. This project utilizes machine learning regression techniques to forecast COVID-19 case counts based on time-series data.

---

2. Problem Statement

To create a time-series prediction model using historical COVID-19 data that forecasts future daily case counts. The model aims to identify patterns and project case trends, which is vital for timely interventions.

---

3. Objectives

- Collect and preprocess historical COVID-19 case data.
- Preprocess and transform the data into a format suitable for time-series modeling.
- Apply the ARIMA model for time-series forecasting.
- Evaluate and visualize model performance.

- Forecast future cases using trained models.

---

4. Methodology

- **Data Collection**: The dataset was uploaded in CSV format using Google Colab.
- **Data Preprocessing**:
    - Dates converted to datetime format.
    - Missing values filled with zero or interpolated.
    - Converted case data into a time-series format.
- **Model Building**:
    - Linear Regression: Used to model the overall trend
    - Selected the optimal ARIMA parameters (p, d, q) using plots like ACF and PACF or automated tools.
- **Model Evaluation**:
    - Accuracy and loss were tracked during ARIMA training.
    - Visual comparison of predicted and actual values was performed.

---

5. Data Preprocessing

The dataset was preprocessed as follows:

- **Converted string dates to datetime objects.**

- **Sorted values chronologically.**
- **Filled missing case values.**
- **Time-series was made stationary by differencing, if required.**
- **Data was resampled (if needed) to maintain regular intervals (daily).**

---

6. Model Implementation

Two models were implemented:

- **Linear Regression: Simple model predicting future values based on past trends.**
- **LSTM (Deep Learning): The ARIMA model is a statistical time-series forecasting model with three parameters:**
- **AR (AutoRegressive): uses past values.**
- **I (Integrated): uses differencing to make the data stationary.**
- **MA (Moving Average): uses past forecast errors.**
-

Training was performed in Google Colab using scikit-learn and TensorFlow.

---

7. Evaluation Metrics

- **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)** were used to measure accuracy.
- **Visual Accuracy**: Predictions plotted against actual data for trend comparison.
- **Forecast Plot**: Projected next 30 days' cases using the trained models.

---

8. Results and Analysis

- **Linear Regression** provided a smooth projection of future cases but lacked the ability to adapt to fluctuations.
- The ARIMA model accurately followed the trend of COVID-19 case growth.Forecasts provided insights into potential future trends over a 30-day period.
- Visualizations revealed that the LSTM model tracked the pandemic curve more accurately than linear regression.

---

9. Conclusion

The ARIMA model provided a simple yet effective way to forecast future COVID-19 cases using past data. While it does not capture highly non-linear patterns like deep learning models, it performs well for trend-following time-series. This project shows how statistical forecasting methods can support real-world planning and response during pandemics.

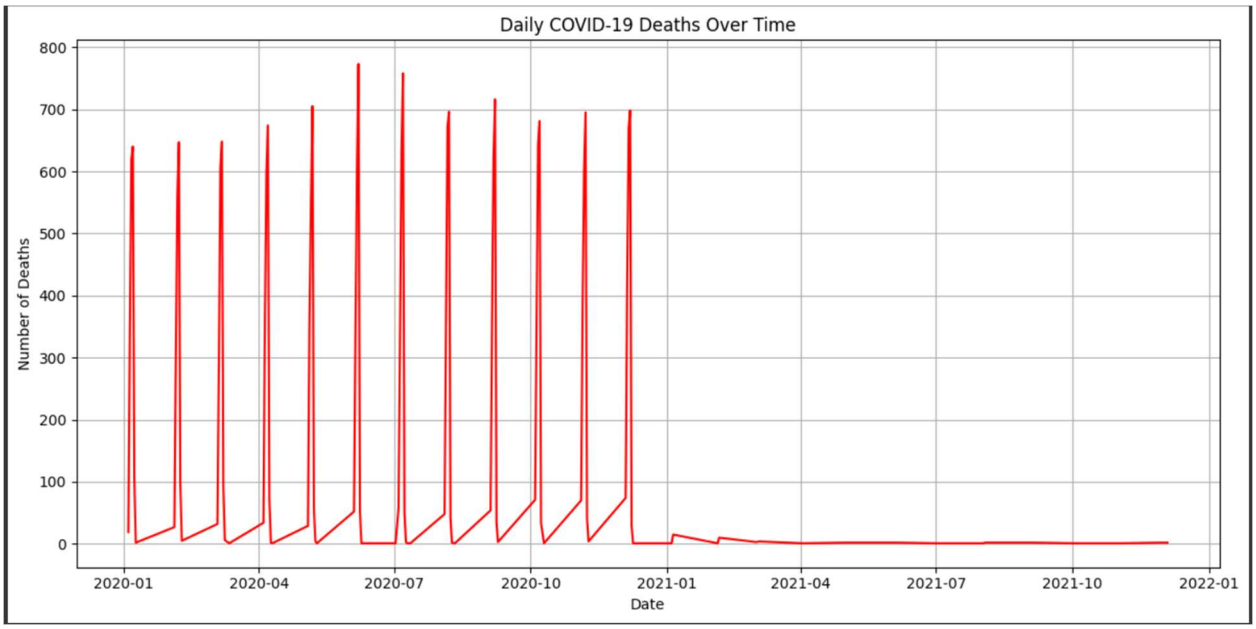
---

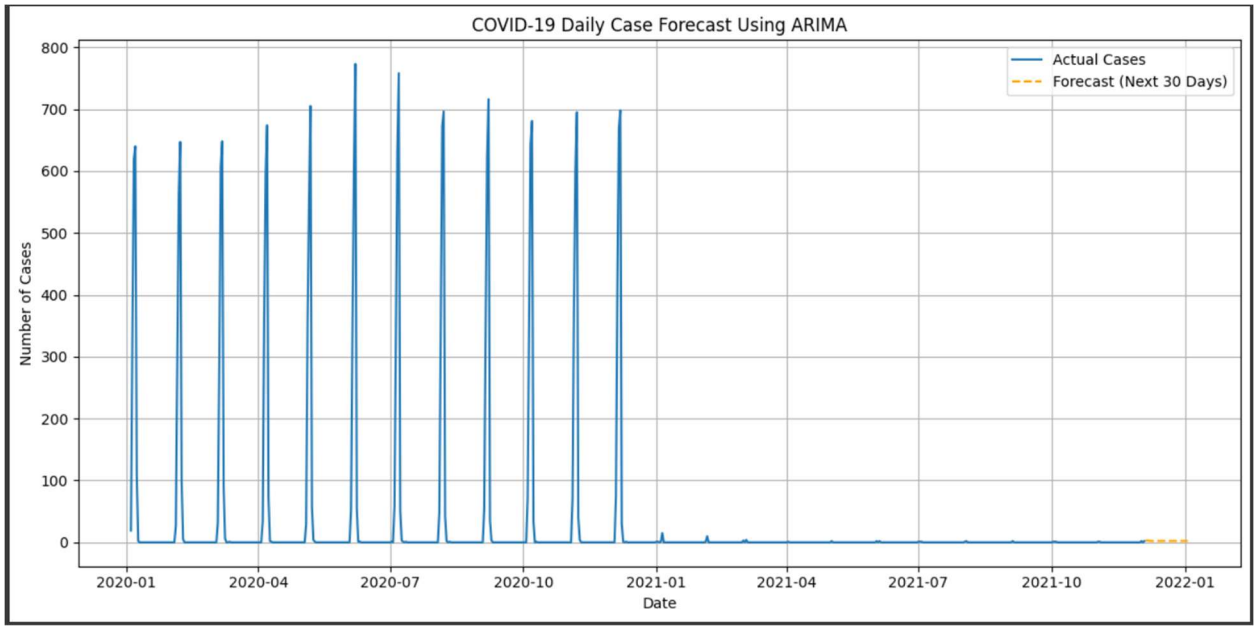
10. References

- TensorFlow documentation
- scikit-learn documentation
- pandas and numpy documentation
- Matplotlib and Seaborn for visualization
- Johns Hopkins CSSE COVID-19 Data Repository

```python
#Step 1: Import libraries
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
import warnings
warnings.filterwarnings("ignore")
```

```python
#Step 2
# Load the dataset
df = pd.read_csv("Covid Data.csv")

# Convert DATE_DIED to datetime and remove invalid entries
df["DATE_DIED"] = pd.to_datetime(df["DATE_DIED"], errors='coerce')
df = df.dropna(subset=["DATE_DIED"])

# Filter only confirmed COVID-19 cases (classification 1, 2, 3)
confirmed = df[df["CLASIFFICATION_FINAL"].isin([1, 2, 3])]

# Count deaths by date
daily_deaths = confirmed.groupby("DATE_DIED").size().reset_index(name="deaths")

# Plot deaths over time
plt.figure(figsize=(12, 6))
plt.plot(daily_deaths["DATE_DIED"], daily_deaths["deaths"], color='red')
plt.title("Daily COVID-19 Deaths Over Time")
plt.xlabel("Date")
plt.ylabel("Number of Deaths")
plt.grid(True)
plt.tight_layout()
plt.show()
```






```
# Step 3: Load and clean the data
df = pd.read_csv("Covid Data.csv")

# Convert date column
df["DATE_DIED"] = pd.to_datetime(df["DATE_DIED"], errors="coerce")
df = df.dropna(subset=["DATE_DIED"])

# Filter confirmed cases
confirmed = df[df["CLASIFFICATION_FINAL"].isin([1, 2, 3])]

# Group by date and count daily cases
daily_cases = confirmed.groupby("DATE_DIED").size().reset_index(name="cases")
daily_cases = daily_cases.set_index("DATE_DIED").asfreq("D", fill_value=0)
```

```
[ ]  # Step 4: Visualize actual vs forecasted data
     plt.figure(figsize=(12, 6))
     plt.plot(daily_cases.index, daily_cases["cases"], label="Actual Cases")
     plt.plot(forecast_dates, forecast, label="Forecast (Next 30 Days)", linestyle="--", color="orange")
     plt.title("COVID-19 Daily Case Forecast Using ARIMA")
     plt.xlabel("Date")
     plt.ylabel("Number of Cases")
     plt.grid(True)
     plt.legend()
     plt.tight_layout()
     plt.show()
```