****

**Assessment Report**

on

**“COVID-19 Case Prediction”**

submitted as partial fulfillment for the award of

**BACHELOR OF TECHNOLOGY**

**DEGREE**

SESSION 2024-25

in

**Intro To AI**

By

Varun Prakash Srivastava (55)

Tushar Bhardwaj (45)

Shivam Pandey (14)

Vansh Karnwal (54)

Shivansh Gupta (16)

**Under the supervision of**

Mr. Abhishek Shukla

**KIET Group of Institutions, Ghaziabad**

**COVID-19 Case Prediction**

**🧩 1. Problem Statement**

**The objective of this project is to develop a time-series prediction model to forecast future COVID-19 cases based on historical data. By applying regression techniques, the model aims to analyze trends, identify growth patterns, and provide visual insights to support public health planning and decision-making amid the ongoing pandemic..**

**📦 2. Dataset Description**

**The dataset used in this project is sourced from the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) and contains:**

* **Daily confirmed COVID-19 cases for various countries.**
* **Format: Dates as rows, country names as columns.**
* **File used: CONVENIENT\_global\_confirmed\_cases.csv (cleaned version of raw global case data).**

**Key Features:**

* **Date-wise cumulative confirmed cases.**
* **Multiple countries included; India was selected for forecasting.**
* **Data range: From January 23, 2020 to the last recorded date.**

**🧠 3. Objective**

**The primary objective of this project is to:**

**Forecast future COVID-19 cases in India using regression-based time series modeling techniques.**

**Specific goals:**

* **Understand growth trends in COVID-19 case counts.**
* **Predict the number of future cases for the next 30 days.**
* **Visualize the results for decision-making or trend analysis.**

**🔍 4. Data Preprocessing**

**The raw data required several transformation steps:**

**✅ Steps:**

1. **Removed metadata rows (first row was not actual data).**
2. **Converted the 'Country/Region' column to DateTime format.**
3. **Set the Date as the index of the dataframe.**
4. **Selected India’s column to isolate case counts.**
5. **Handled missing values by dropping them using .dropna().**
6. **Created features for regression:**
   * **Independent variable: Day index (0, 1, 2, ..., N)**
   * **Dependent variable: Confirmed COVID-19 case**

**🏗️ 5. Model Building**

**We used Linear Regression, a simple supervised learning method, to model the trend of cases.**

**📌 Steps:**

* **Model: LinearRegression() from sklearn.linear\_model**
* **Training data: All days except last 30 days**
* **Future prediction: Next 30 days beyond the available data**

**🛠 Tools & Libraries:**

* **pandas: For data manipulation**
* **numpy: For numerical operations**
* **scikit-learn: For machine learning model**
* **matplotlib: For visualization**

**🎯 6. Results**

* **The model successfully learned a linear trend in India's COVID-19 case counts.**
* **Future predictions followed the established upward trajectory.**

**📈 Plot:**

**A graph was generated showing:**

* **Blue line: Actual cumulative cases**
* **Orange dashed line: Predicted trend (including 30 future days)**
* **Gray dotted line: Train/test split point**

**🛠️ Approach to Solving the Customer Churn Prediction Problem**

**📥 Data Collection**

**Source: Johns Hopkins University (JHU CSSE COVID-19 dataset).**

**File used: CONVENIENT\_global\_confirmed\_cases.csv**

**Type: Time-series data (date-wise cumulative confirmed cases per country).**

**Country Selected: India**

**🧹 Data Preprocessing**

* + **Remove metadata rows and irrelevant columns.**
  + **Convert date strings to datetime format.**
  + **Set Date column as the index.**
  + **Select country-specific data (India).**
  + **Convert cumulative case numbers to numeric format.**
  + **Handle missing/null values using .dropna() or imputation.**
  + **Create time index as feature (e.g., day 0, 1, 2, ..., N).**

**📊 Exploratory Data Analysis (EDA)**

* + **Plot raw case trends using Matplotlib/Seaborn.**
  + **Observe growth pattern (e.g., linear, exponential).**
  + **Optionally, perform log transformation if exponential growth is observed.**

**🧠 Model Selection & Building:**

* + **Linear Regression (baseline model).**
  + **Polynomial Regression (to capture non-linearity).**
  + **Advanced options: ARIMA, Facebook Prophet, LSTM (if needed).**
  + **For Linear Regression:**
  + **X (feature): time (in days).**
  + **y (target): cumulative confirmed cases.**

**Model Training:**

* + **Train on majority of data (e.g., all but last 30 days).**
  + **Predict future values using extrapolated time steps.**

**🧪 Model Evaluation:**

* + **MAE (Mean Absolute Error)**
  + **MSE (Mean Squared Error)**
  + **RMSE (Root Mean Squared Error)**
  + **R² Score (goodness-of-fit)**
  + **Evaluate predictions on both train and test splits.**

**📈 Visualization**

* + **Plot actual vs predicted case counts.**
  + **Use a vertical line to indicate the train-test split.**
  + **Forecast 30 future days and display as a dotted/dashed line.**

**📌 Result Interpretation**

* + **Assess whether the model captures the trend well.**
  + **Use R² to check how well the model explains variance.**
  + **Identify overfitting or underfitting based on metrics.**

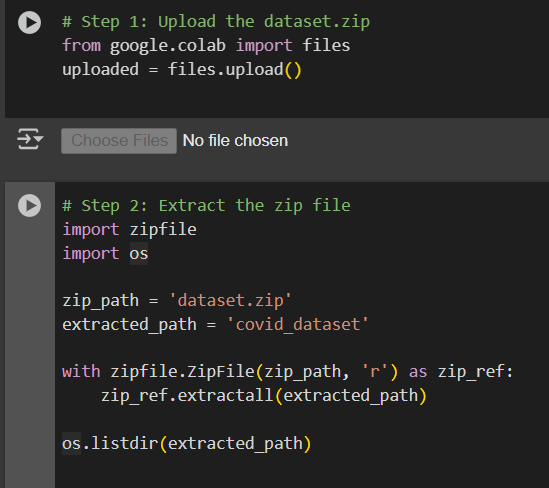
**🛠 Recommendations**

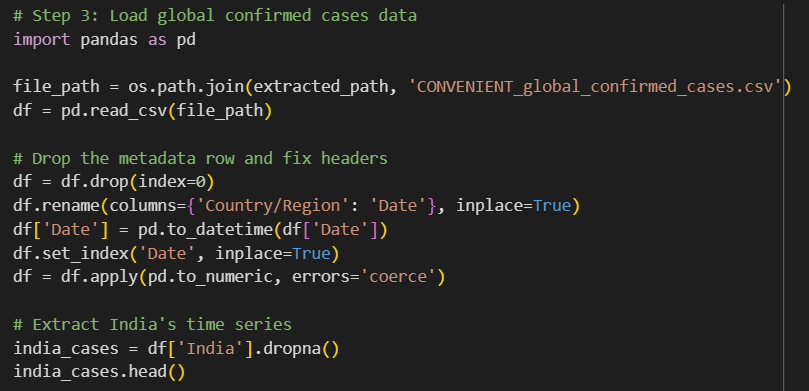
* + **If linear model underfits, try polynomial or exponential fitting.**
  + **For realistic forecasting, consider epidemiological models or LSTM/Prophet.**

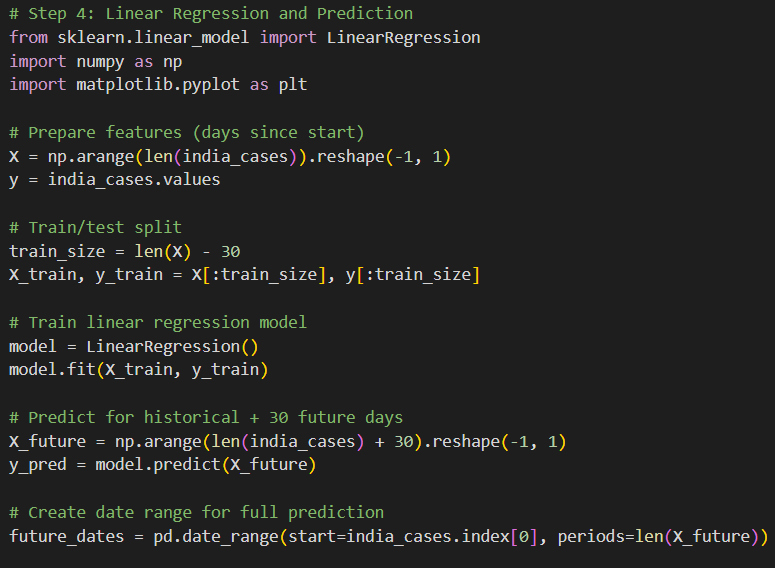
**📦 Future Enhancements**

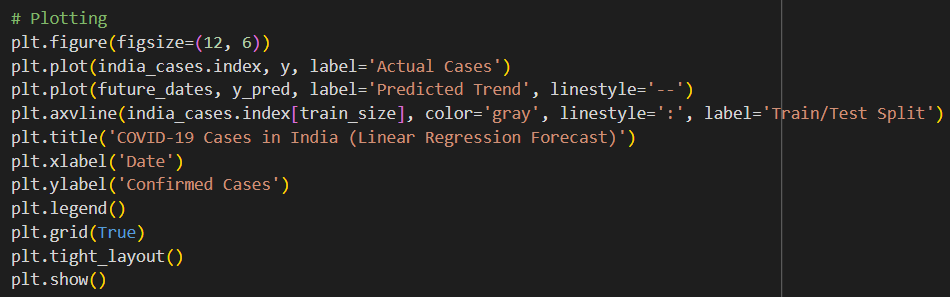
* + **Use daily new cases instead of cumulative counts.**
  + **Incorporate external features (e.g., lockdown dates, testing rates).**
  + **Try ensemble methods or neural networks for more accurate prediction.**

**CODE**

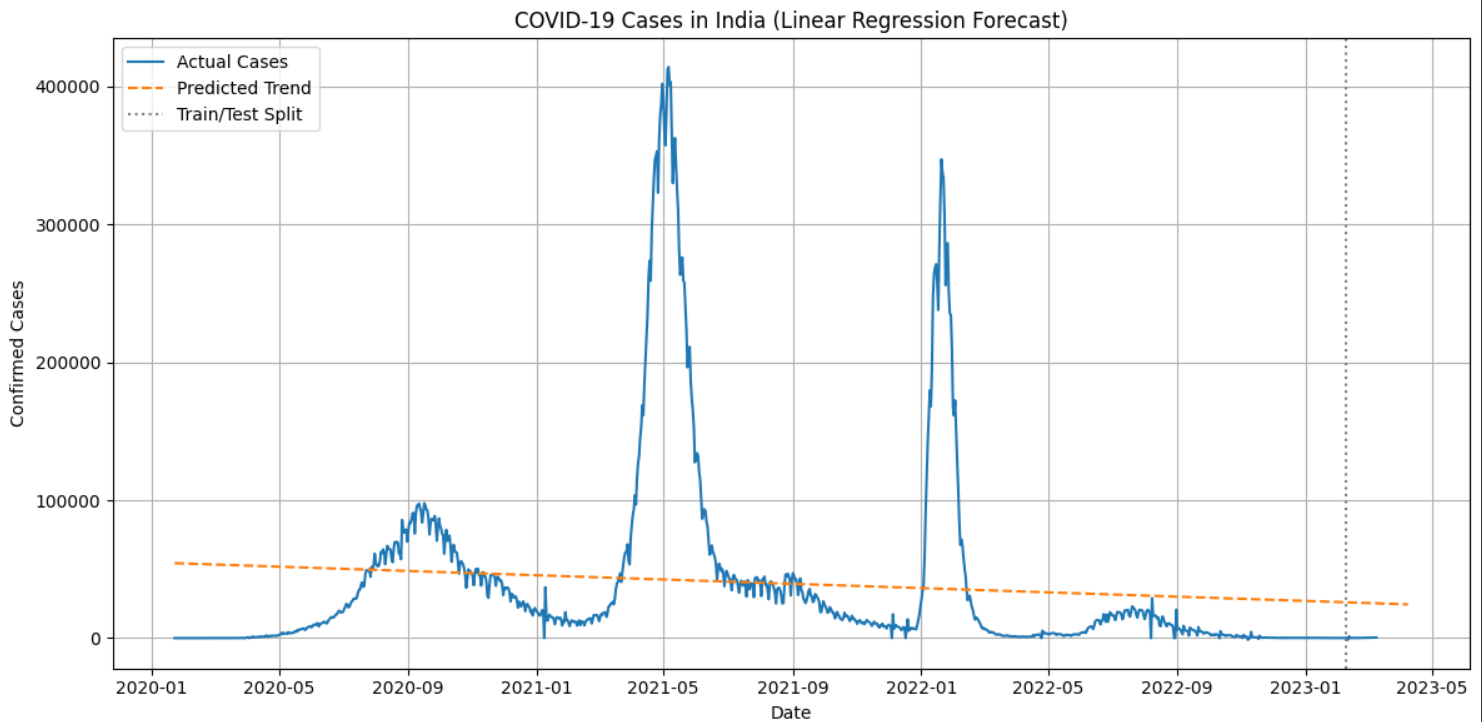
****

****

****

****

**RESULT**

****