Title: Comparison of Linear & Nonlinear Models Using COVID-19 Time Series Dataset

---

Objective :-

The objective of this task is to:

- Compare **Generalized Linear Models (GLM)** and **Nonlinear Regression** techniques using real-world data.
- Analyze which model better captures the underlying patterns in the data.
- Justify the need for nonlinear models by understanding the limitations of linear assumptions.

Dataset Used :- Source: Johns Hopkins University – Center for Systems Science and Engineering (JHU CSSE)

- **Dataset:** Time Series of Confirmed COVID-19 Cases (Jan 2020 onward)
- **Link:** Click here
- **Selected Country:** India
- **Features:**
  - Date: Timeline of reported data
  - Cases: Cumulative COVID-19 cases
  - Days: Number of days since the first reported case in India

---

Background: Linear vs. Nonlinear Models :-

- **Linear Models (GLM)**
  - Assumes a **straight-line** relationship between independent and dependent variables.
  - Effective when the data shows a **constant rate of change**.
  - GLM (Gaussian family) is a generalization of linear regression that can be extended to other distributions.
- **Nonlinear Models**
  - Models that allow for **curvature** or **complex relationships**.
  - Useful for **growth curves**, **exponential patterns**, and **saturation behavior**.
  - In this task, a **logistic function** was used to represent the **S-curve** nature of pandemic growth.

Methodology :-

- **1. Data Preprocessing**
  - Filtered India-specific data.
  - Converted cumulative case counts into numeric arrays.
  - Calculated Days as a feature for modeling.
- **2. Applied GLM**
  - Model:

$Y=\beta0+\beta1\cdot Days$

  - Fit using `statsmodels.GLM()` with Gaussian distribution.
  - Prediction: `GLM_Pred`
- **3. Applied Nonlinear Regression (Logistic)**
  - Fit using scipy.optimize.curve_fit()
  - Prediction: Nonlinear_Pred

Evaluation Metrics

Metric	GLM (Linear)	Nonlinear (Logistic)
RMSE	268,920.32	✅ **48,762.58**
R^2 Score	0.8987	✅ **0.9963**
Residual Pattern	Systematic	✅ Random

Prediction Trend	Underestimates saturation	✅ Accurately models curve

Interpretation:

- **GLM RMSE** is much higher, indicating **greater error**.
- **R^2 Score** is much closer to **1.0** for the logistic model, suggesting a better fit.
- Residuals for GLM **increase over time** (underfitting).
- Residuals for nonlinear model are **scattered and close to zero**, indicating accuracy.

Visual Analysis

1. Actual vs. Predicted Cases

- **GLM** follows the early phase of growth but fails to account for saturation.
- **Logistic Model** captures the typical **epidemic S-curve** — initial slow growth, exponential phase, and final plateau.

2. Residual Plots

- GLM residuals are **not randomly distributed**, indicating poor fit.
- Nonlinear model residuals are **evenly spread**, suggesting a well-fit model.

Limitations of Linear Models

- Cannot capture **non-linear behaviors**, such as exponential growth or logistic saturation.
- Often **underfit** when dealing with real-world phenomena like:
  - Epidemics
  - Population growth
  - Financial trends
- Predicts **unbounded values**, which is **not realistic** for many practical applications.

When to Use Nonlinear Models

- When data exhibits **non-constant change**, especially:
  - **Exponential or logistic growth**
  - **Periodic behavior** (e.g., sine waves, weather cycles)
  - **Threshold effects** (e.g., drug dosage vs. effect)
- Logistic models are widely used in:
  - **Epidemiology**
  - **Ecological modeling**
  - **Marketing and adoption curves**

Conclusion

Through this task, we conclude that:

- **Nonlinear models** (especially logistic models) are **superior** for modeling real-world phenomena like the COVID-19 pandemic.
- While **linear models** are simple and easy to interpret, they are **not suitable** for datasets that show **complex or bounded growth**.
- **Model evaluation using residuals and prediction accuracy** strongly supports the use of nonlinear regression for epidemic modeling.
- A sound understanding of data characteristics is essential before selecting the modeling technique.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.optimize import curve_fit
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, r2_score

# ----------------------------
# 1. Data Acquisition and Preparation
# ----------------------------

# Load the dataset
file_path = '/content/time_series_covid19_confirmed_global.csv'
data = pd.read_csv(file_path)

# Filter for India
country = 'India'
country_data = data[data['Country/Region'] == country].iloc[:, 4:].sum().reset_index()
country_data.columns = ['Date', 'Cases']

# Convert date and calculate days since start
country_data['Date'] = pd.to_datetime(country_data['Date'])
country_data['Days'] = (country_data['Date'] - country_data['Date'].min()).dt.days + 1

# ----------------------------
# 2. Generalized Linear Model (GLM)
# ----------------------------

X_glm = sm.add_constant(country_data['Days'])  # Intercept
glm_model = sm.GLM(country_data['Cases'], X_glm, family=sm.families.Gaussian()).fit()
country_data['GLM_Pred'] = glm_model.predict(X_glm)

# ----------------------------
# 3. Nonlinear Regression - Logistic Growth Model
# ----------------------------

def logistic(x, L, x0, k):
    return L / (1 + np.exp(-k * (x - x0)))

# Initial guess: L (max), x0 (mid), k (growth rate)
p0 = [max(country_data['Cases']), np.median(country_data['Days']), 0.1]
params, _ = curve_fit(logistic, country_data['Days'], country_data['Cases'], p0=p0)
country_data['Nonlinear_Pred'] = logistic(country_data['Days'], *params)

# ----------------------------
# 4. Evaluation Metrics
# ----------------------------

glm_rmse = np.sqrt(mean_squared_error(country_data['Cases'], country_data['GLM_Pred']))
glm_r2 = r2_score(country_data['Cases'], country_data['GLM_Pred'])

nonlinear_rmse = np.sqrt(mean_squared_error(country_data['Cases'], country_data['Nonlinear_Pred']))
nonlinear_r2 = r2_score(country_data['Cases'], country_data['Nonlinear_Pred'])

print("🔹 Model Evaluation Summary")
print(f"GLM - RMSE: {glm_rmse:.2f}, R² Score: {glm_r2:.4f}")
print(f"Nonlinear - RMSE: {nonlinear_rmse:.2f}, R² Score: {nonlinear_r2:.4f}")

# ----------------------------
# 5. Visualization
# ----------------------------

plt.figure(figsize=(12, 6))
sns.scatterplot(x='Days', y='Cases', data=country_data, label='Actual Cases', s=50)
sns.lineplot(x='Days', y='GLM_Pred', data=country_data, label='GLM Prediction', color='red')
sns.lineplot(x='Days', y='Nonlinear_Pred', data=country_data, label='Nonlinear Prediction (Logistic)', color='green', linestyle='--')
plt.title(f'COVID-19 Cases in {country}: Linear vs. Nonlinear Model')
plt.xlabel('Days Since First Case')
plt.ylabel('Cumulative Confirmed Cases')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# ----------------------------
# 6. Residual Analysis
# ----------------------------

country_data['GLM_Residuals'] = country_data['Cases'] - country_data['GLM_Pred']
country_data['Nonlinear_Residuals'] = country_data['Cases'] - country_data['Nonlinear_Pred']

plt.figure(figsize=(14, 5))
```

```
# GLM Residuals
plt.subplot(1, 2, 1)
sns.residplot(x='Days', y='GLM_Residuals', data=country_data, lowess=True, color='red')
plt.title('GLM Residuals')
plt.xlabel('Days Since First Case')
plt.ylabel('Residuals')
plt.grid(True)

# Nonlinear Residuals
plt.subplot(1, 2, 2)
sns.residplot(x='Days', y='Nonlinear_Residuals', data=country_data, lowess=True, color='green')
plt.title('Nonlinear Model Residuals')
plt.xlabel('Days Since First Case')
plt.ylabel('Residuals')
plt.grid(True)

plt.tight_layout()
plt.show()
```

```
<ipython-input-2-76b83123916c>:23: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `da
  country_data['Date'] = pd.to_datetime(country_data['Date'])
🔹 Model Evaluation Summary
GLM - RMSE: 4756887.58, R² Score: 0.9269
Nonlinear - RMSE: 1979805.07, R² Score: 0.9873
```