**Team Name: Error404!**

**Team Members:**

**Team Leader:** Manmohan Zanwar **Member 2:** Yash Rane **Member 3:** Kedar Desai **Member 4:** Harshal Suryavanshi

**From:** *Sanjay Ghodawat University, Kolhapur*

**Introduction:** Ebola is a deadly disease where quick action saves lives. This project uses machine learning to predict cases, deaths, and fatality rates in regions with missing data, helping health authorities respond faster and allocate resources effectively.

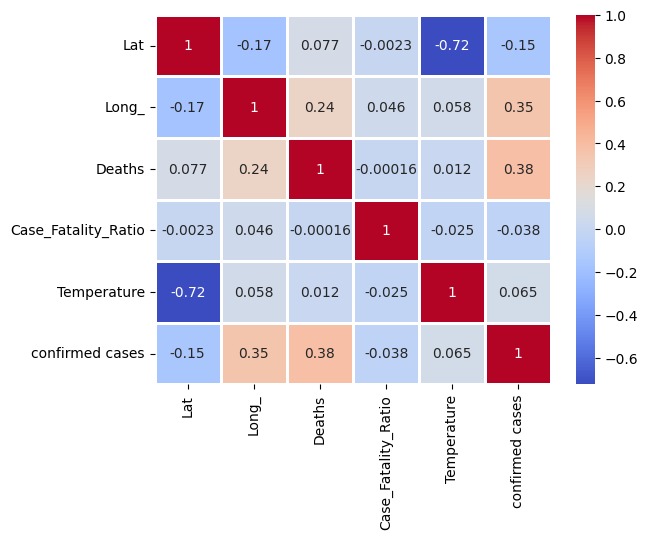
**Problem Statement:** The task is to develop a machine learning model to predict deaths, confirmed cases, and Case Fatality Ratio (CFR) for various regions affected by Ebola. The model will utilize available data, including:

* CFR for some regions,
* Reported deaths (partially available),
* Geographical coordinates (latitude and longitude).

The objective is to forecast these metrics for unreported regions, aiding in resource allocation, timely intervention, and containment efforts during outbreaks.

**Approach:**

* **Data Preparation**: Analyze available data (deaths, CFR, geographical info) and handle missing values.
* **Feature Engineering**: Create meaningful features to capture patterns in the data.
* **Model Selection**: Train machine learning models (e.g., Random Forest, XGBoost) to predict deaths, cases, and CFR.
* **Validation**: Evaluate model accuracy using metrics like RMSE and fine-tune for performance.
* **Prediction**: Forecast metrics for unreported regions and analyze trends.
* **Generalization**: Test the model on unseen data and adapt it for new regions or diseases.
* **Visualization**: Present results using maps and charts to highlight actionable insights.

**Correlation Heatmap**

RMSE:

1. Predicted deaths in trained dataset- 48.48
2. Predicted confirmed cases in test dataset- 0.94
3. Predicted deaths in test dataset- 058

**Key Sections**

**1. Setting Things Up**

We start by importing the tools we need:

* numpy: Handles all the math stuff.
* pandas: Makes working with tables and datasets a breeze.
* matplotlib & seaborn: Create cool charts and graphs.
* sklearn: Helps with splitting data and preparing for machine learning tasks.
* requests: Talks to external APIs to fetch data.
* tqdm: Shows a progress bar for loops (because who doesn’t like progress bars?).

**2. Loading and Exploring the Data**

* Data Source: The file train\_data.csv is loaded into a pandas dataframe.
* First Steps:
  + Preview the dataset with df.head() to see what we’re working with.
  + Check for any missing data using df.isnull().sum().
  + Get the dataset's size with df.shape.

**3. Cleaning the Data**

* Dealing with Missing Data:
  + Drop rows where the Lat (latitude) , Long (longitude) column is empty.
  + Remove rows with missing values in the Case\_Fatality\_Ratio column.
* Finding Errors:
  + Spot rows with weird values, like a Case\_Fatality\_Ratio of 600, and handle them accordingly.

**4. Adding Data from APIs**

This notebook includes code (currently commented out) to pull in data from the OpenWeatherMap API:

1. Temperature:
   * The get\_temperature() function takes latitude, longitude, and an API key to fetch temperature data.
   * It handles errors and appends the temperature to the dataset.
2. Key Notes:
   * You’ll need to add your own API key to make it work.
   * Make sure the API endpoint and key are valid before running.

**5. Saving the Cleaned Data**

* Once the data is cleaned and enriched, it can be saved as a new CSV file named cleandata\_temp.csv.
* Note: The save operation is currently commented out.

**Workflow in a Nutshell**

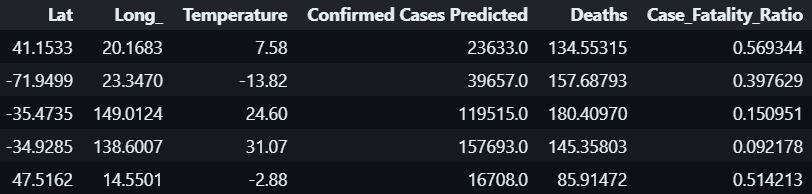
1. Load Data:
   * Start with the raw CSV file.
   * Take a first look and get familiar with the structure.
2. Clean Data:
   * Remove rows with critical missing values.
   * Deal with any errors or outliers.
3. Enhance Data:
   * Use APIs to add temperature and air quality data.
4. Save Data:
   * Prepare a clean, enriched dataset for analysis or further use.

**Tools You’ll Need**

* Python Libraries: numpy, pandas, matplotlib, seaborn, sklearn, requests, tqdm.

External API: OpenWeatherMap for temperature data.

**Output**:  
Train dataset Prediction  
  


Test data Prediction  
****

**Conclusion**:Through this project, I leveraged machine learning to predict Ebola cases, deaths, and CFR for regions with missing data. These predictions offer valuable insights to help allocate resources effectively and plan timely interventions. The approach is designed to support decision-making during outbreaks and can be adapted for other diseases or regions, making it a powerful tool for improving public health responses.