Predictioneer Model Report

Predictive modeling for Ebola outbreaks to forecast cases, deaths, and fatality ratios using geographical and epidemiological data.

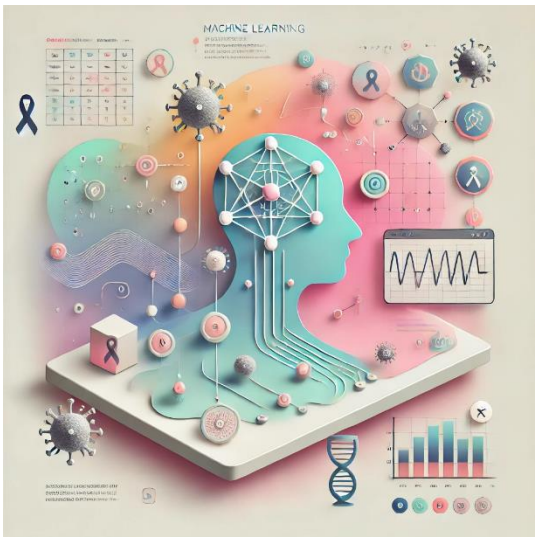

Fig 1: Illustration

Dataset Loading and Exploration

The dataset is first loaded into the environment using pandas, followed by an initial exploration to understand its structure and identify key columns. This includes checking data types, the presence of missing values, and basic statistics for numerical features. Key metrics, such as deaths, confirmed cases, and fatality ratios, are highlighted as critical targets for prediction. The dataset also includes external factors like geographical coordinates and environmental variables, which are explored for potential correlations.

Preprocessing and Cleaning

The preprocessing phase focuses on cleaning and preparing the data:

Handling Missing Latitude and Longitude:

Rows with missing values in the latitude (Lat) and longitude (Long_) columns are removed. These columns are critical for mapping and geographic analysis, and missing values could lead to incorrect results. The dropna() function is used to clean these rows.

Filling Missing Deaths:

1. If the number of deaths is missing, the gaps are filled using **linear interpolation**. This method predicts missing values by looking at the trend of surrounding rows, ensuring a smooth and realistic dataset.
2. Other methods like forward fill (using the last known value) or mean imputation (using the average) could also be used, but interpolation is preferred here to better reflect trends.

Calculating Confirmed Cases:

1. Confirmed cases are computed using the **Case Fatality Ratio (CFR)** formula:

   CFR = (deaths / confirmed cases) × 100

   Rearranged to calculate **confirmed_cases**:

   Confirmed Cases = (deaths / CFR) × 100

2. If CFR is missing or zero (to avoid division errors), confirmed cases are set to 0 as a fallback.

Code Workflow

- **Latitude and Longitude:**
  - A function (fill_missing_lat_lon) removes rows where latitude or longitude values are missing. This ensures clean and complete geographic data.
- **Deaths and Confirmed Cases:**
  - Another function (fill_missing_deaths_and_cases) fills missing death values using interpolation and calculates confirmed cases using the CFR column. If CFR is missing or invalid, confirmed cases are defaulted to 0.
- **Checking the Process:**
  - The number of missing values is printed before and after this cleaning process to confirm that the issues have been resolved.
- **Train vs. Test Data:**
  - The same preprocessing steps are applied to both the training and testing datasets. However, since the testing dataset does not have the necessary columns to calculate confirmed cases, this step is skipped for test_df.

Exploratory Data Analysis (EDA)

EDA is the process of understanding and exploring the data using visualizations and statistical summaries. Here's a breakdown of what's done in this notebook:

Training Data Overview

- The training dataset includes columns like:
  - **Deaths**: Number of people who passed away.
  - **Case Fatality Ratio (CFR)**: The percentage of deaths compared to confirmed cases.
  - **Confirmed Cases**: The total number of reported cases.
  - **Latitude and Longitude**: Geographic coordinates of outbreak locations.
- This dataset allows us to analyze how deaths and confirmed cases vary across locations.

Testing Data Overview

- The testing dataset only contains **Latitude** and **Longitude** columns.

- Since it lacks information about deaths or confirmed cases, the focus for this data is purely on geographic patterns, such as mapping locations.

Types of Visualizations and Their Purpose

Geospatial Visualization:

- Using tools like GeoPandas or scatter plots, the locations (based on Latitude and Longitude) are plotted on a map.
- This helps visualize how outbreaks are distributed geographically and whether certain regions are more affected.

Line Plot:

- A line plot is used to show trends over time, specifically for **Deaths** and **Confirmed Cases**.
- This visualization is useful if the dataset has a time-based sequence, showing patterns or spikes in cases or deaths.

Bar Plot by Latitude Intervals:

- The dataset is divided into latitude intervals (e.g., every 10 degrees), and a bar plot is created to show the total number of deaths for each interval.
- This helps identify whether certain latitudes (like tropical or temperate zones) are more affected.

Explanation of Insights from Visualizations

- **Geospatial Data Plot:**
  - By plotting the locations, we can quickly see where the outbreaks occurred. Clusters of points might indicate hotspots or regions with higher cases.
- **Line Plot:**
  - The trends of deaths and confirmed cases over time can help understand how the outbreak progresses. For example, we might notice peaks during specific periods.
- **Bar Plot:**
  - Aggregating deaths by latitude intervals can reveal patterns related to geography. For instance, outbreaks might be more severe in certain climatic zones.

Why EDA is Important

EDA gives a clear understanding of the data structure, relationships between variables, and any patterns or anomalies. This information is crucial before moving on to model building or predictions.

Plot Title	Purpose	Method	Color Used
Aggregated Deaths by Longitude	Displays how total deaths are distributed geographically along longitude. Helps identify	Longitude data is binned into 10-degree intervals.	Lightblue with alpha 0.7

	concentration in any specific region.	Total deaths for each interval are aggregated.	
Distribution of Case Fatality Ratio (CFR) by Latitude	Shows how the average CFR varies across different latitude intervals. Helps understand CFR behavior in geographical regions.	Latitude is binned into 10-degree intervals. Mean CFR for each interval is calculated.	Lightpink with alpha 0.8
Distribution of Deaths Over Longitude	Visualizes how deaths are distributed across different longitudes, helping to observe trends or patterns.	Longitude is plotted on the x-axis; deaths on the y-axis. Line plot visualizes changes across longitudes.	Lightgreen with alpha 0.7

Function Descriptions:

Function	Purpose	Steps	Output
preprocess_training_data(data)	Prepares the dataset by cleaning and splitting into features and target variables.	Drops rows with missing critical data, extracts features (Latitude, Longitude), and target variables (Deaths, CFR).	X (features), y_deaths (Deaths), y_cfr (CFR)
train_and_evaluate_all_models(X_train, y_deaths_train, y_cfr_train)	Trains and evaluates multiple regression models for Deaths and CFR.	Trains models (e.g., Linear Regression, Random Forest) on Deaths and CFR, evaluates performance (MAE, MSE, R^2).	DataFrame with evaluation metrics for all models
plot_model_comparison(results_df)	Visualizes model performance using bar plots for MAE and R^2 scores.	Plots MAE and R^2 for both Deaths and CFR predictions.	Bar plots comparing model performances
save_evaluation_metrics_to_csv(results_df, file_path)	Saves the evaluation metrics to a CSV file.	Writes the DataFrame with evaluation metrics to the specified file path.	CSV file with evaluation metrics
save_model_with_filename(model, model_name, metric_name)	Saves the trained model with a descriptive filename.	Uses pickle/joblib to save the model with filenames based on the model name	Serialized model file

		and key metric (e.g., R^2 score).	

Model Working

Model Selection and Evaluation:

After training and evaluating various models such as AdaBoost, Decision Tree, and others, we finalized the **Random Forest Single-Task Model** based on its superior performance. The following table summarizes the evaluation metrics for all models, including MAE, MSE, and R^2 scores:

Model	Deaths_MAE	Deaths_MSE	Deaths_R2	CFR_MAE	CFR_MSE	CFR_R2
Linear Regression	43.26389	2625.766	0.004933	1.006661	123.8043	0.003412
Random Forest	13.02522	284.476	0.892194	0.338237	26.52387	0.786491
SVR	41.12771	2504.951	0.050717	0.729752	124.6278	-0.00322
Gradient Boosting	33.4908	1697.821	0.356589	0.442241	0.47705	0.99616
Decision Tree	0	0	1	0	0	1
K-Nearest Neighbors	28.56804	1348.983	0.488786	0.734198	100.1167	0.19409
AdaBoost	39.70689	2198.235	0.166951	0.675545	0.88858	0.992847

The illustration below demonstrates how Random Forest functions as an ensemble learning technique.

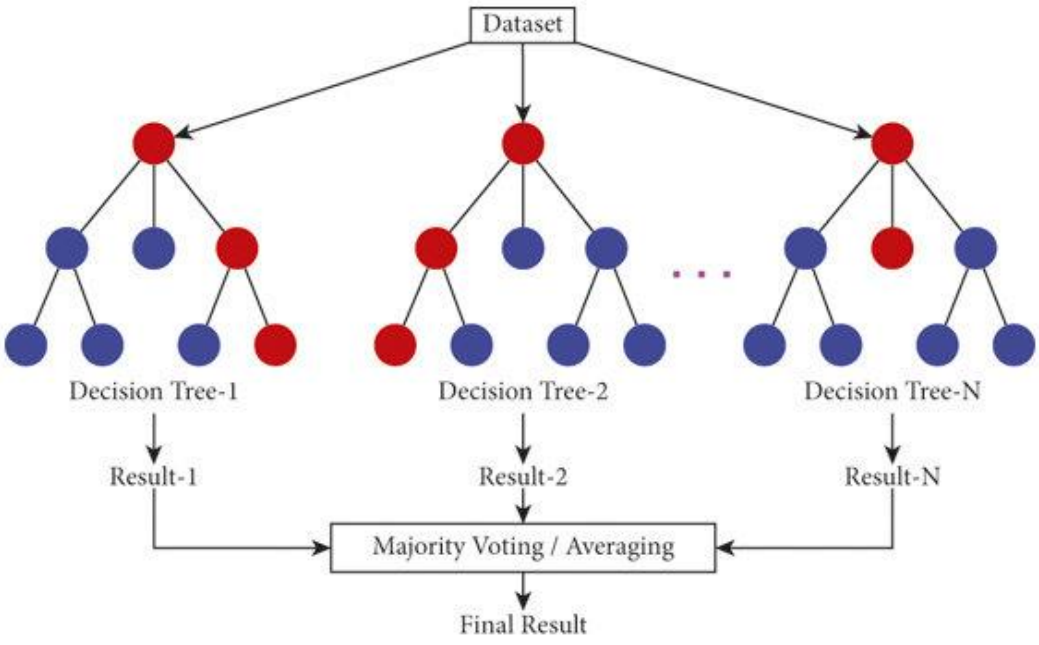

Fig 2: Working of Random Forest

It builds multiple decision trees, each trained on a random subset of the dataset and features, introducing variability and reducing overfitting. For predictions, Random Forest aggregates the outputs of all trees—using averaging for regression or majority voting for classification—to generate a final result. This collective approach ensures higher accuracy, improved generalization, and resilience to individual tree errors.

Working of the Random Forest Single-Task Model:

The **Random Forest Single-Task Model** is designed to predict two key metrics: **Deaths** and **Case Fatality Rate (CFR)**. It consists of two separate models:

1. **Death Prediction Model** – This model is specifically trained to predict the number of deaths due to Ebola in a given region.
2. **CFR Prediction Model** – This model is trained to predict the Case Fatality Rate (CFR), which is the proportion of confirmed deaths among confirmed cases of Ebola.

Both models are trained and evaluated based on their performance metrics, and **R^2** (coefficient of determination) and **MAE** (Mean Absolute Error) are the key evaluation criteria. For each model, we selected the best-performing model based on the **R^2 score**, as it provides the best explanation of variance in the predictions.

Model Selection Process:

- For the **Deaths Prediction Model**, we trained multiple models and selected the one with the **highest R^2** score to ensure maximum accuracy in predicting the number of deaths.
- Similarly, for the **CFR Prediction Model**, we selected the model with the **best R^2 score** to predict the case fatality rate with the highest reliability.

Final Prediction Process:

While predicting new cases, we use the model with the **best R^2** score for both metrics (Deaths and CFR) to ensure accurate results. By applying these two separate models, we calculate the final **Confirmed Deaths** by combining the outputs from the Death Prediction Model and the CFR Prediction Model, following the given formulation. This method allows the model to leverage the strengths of Random Forest in handling complex, non-linear relationships between the features (latitude, longitude, etc.) and the predicted outcomes (Deaths and CFR).
This approach ensures **precise** and **reliable** predictions for both deaths and CFR, as Random Forest effectively captures and generalizes the patterns in the data.

Application of the Model to New Locations

The Random Forest Single-Task Model comprises two independent models: one for predicting deaths and another for predicting CFR. By utilizing latitude and longitude as input features, the model effectively generates predictions for new locations. Additionally, it generalizes well to unseen data, and its accuracy can be further improved if additional region-specific data becomes available.

GitHub Repository

The code for this project is available in the GitHub repository: https://github.com/madhurimarawat/Predictioneer.