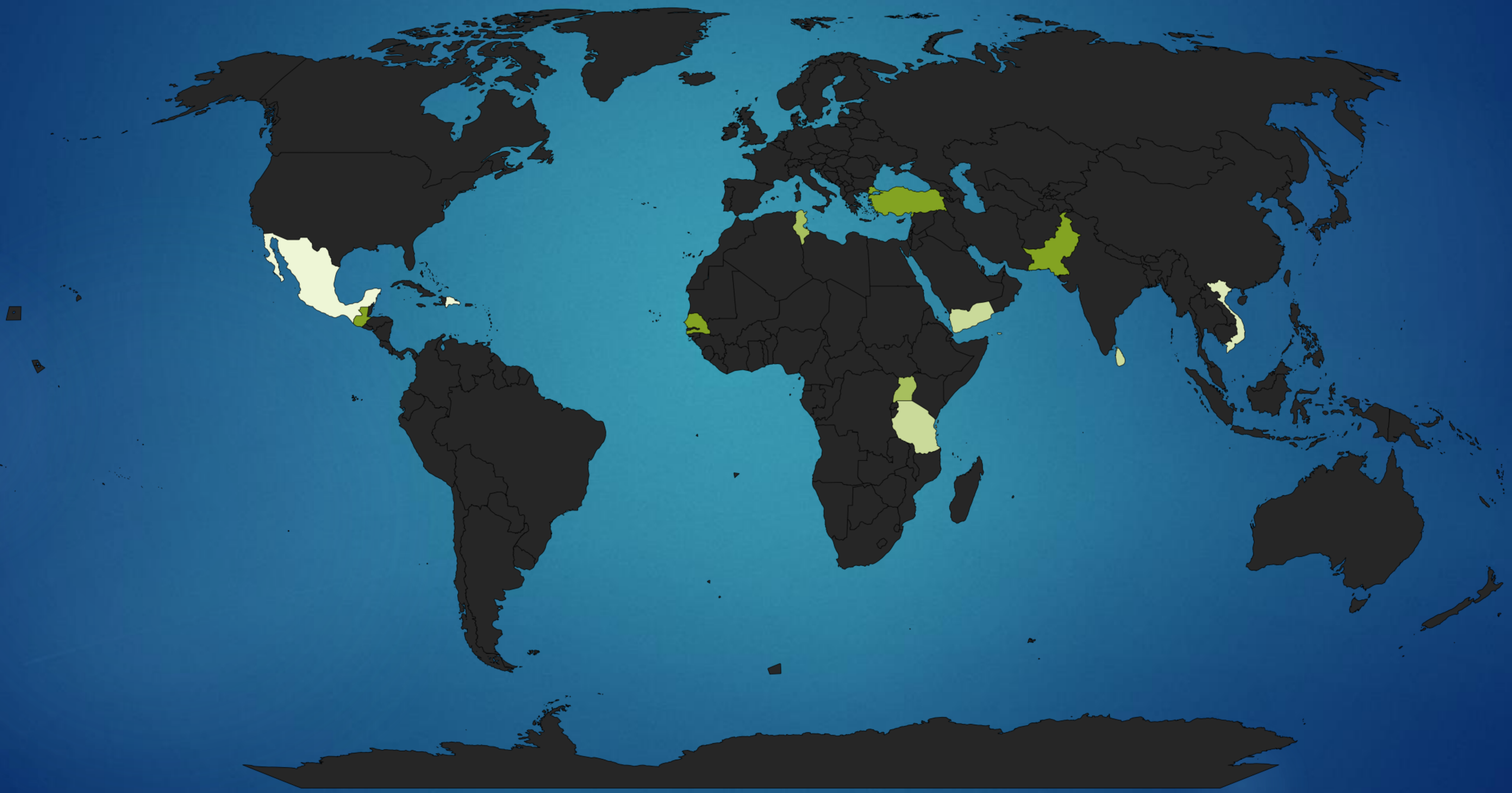
DISASTER AFTERMATH ANALYSIS: COUNTRY GROUP C

SUKANYA GHOSH
CHETANA MANE
NIRMANI AMARASINGHE
MANISH BABU SATHISH BABU
AADITYA KAUL
ISHITA TRIPATHI

Group 18 – Country C

Severity



PRE- PROCESSING

Data Cleansing Process

- Cleaning the special characters
- Removing duplicate rows.
- Checking the “Nan” values . Standard omittable limit of “Nan” values <5% . All dataset have been through this filter.
- Converting all event into uppercase

Observation of Data

- We have observed the following different cases in our Group C dataset, and we have invoked different pre-processing tactics to handle the observed cases as below :

CASE 1 – Monitory losses are too high (example : Mexico, Turkey). The outliers were set at the max of column

CASE 2 - Monitory losses columns have no values (Pakistan, Uganda)

CASE 3 - Mexico has event in Mexican which we converted to English

CASE 4 – Trend of DataCards and death follows for two type of relations into country a and b

CASE 5 – Normal cases for all above cases

Continued....

PRE- PROCESSING

Data Transformation

- ❑ Merging columns where the type of loss is conceptually same e.g. Losses…Local & Losses…USD have been merged as “Monetory.LossUSD” as both the columns indicates the monitory . The “Mode” conversion rate for last 1 month has been considered to calculate the Monetory.LossUSD.
- ❑ Merged Houses destroyed and houses damaged into houses ruined
- ❑ Merged directly affected and indirectly affected into affected

Feature Addition (for each country)

- ❑ Added features as Event_Severity score based on the ratio of deaths to data cards in a dataset. It then adjusts this score for specific by setting upper and lower bounds. Finally, it scales the severity score and returns it.

  Case 1 : If the calculated severity is greater than 1, it's set to 1. If the calculated severity is less than 0.5, it's set to 0.5.

  Case 2 : If the calculated severity is greater than 2, it's set to 2.If the calculated severity is less than 1, it's set to 1.
- ❑ Event_Factor(factoring the event for modelling).

Translation (Mexico)

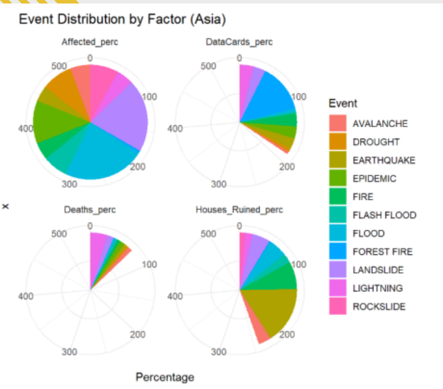
- ❑ Using dictionary to translate the name of the “Events” for Mexico to English which were written in Spanish. Example : “HAMBRUNA”=“FAMINE”

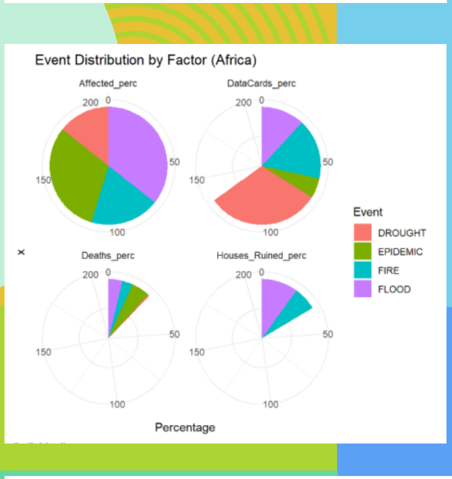
Visualizing Preprocessed Data

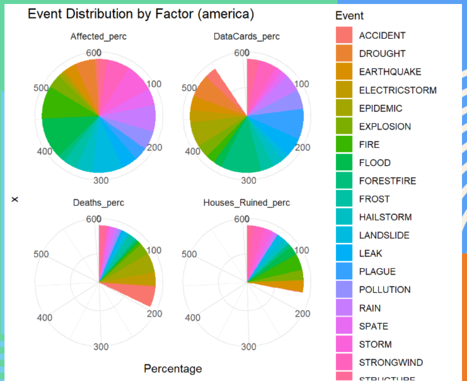
- ❑ Combined the Cleaned Dataset per Continent and performed the Visualization of the continent wise cleaned dataset. Plotted the visualization of the cleaned dataset of every country.

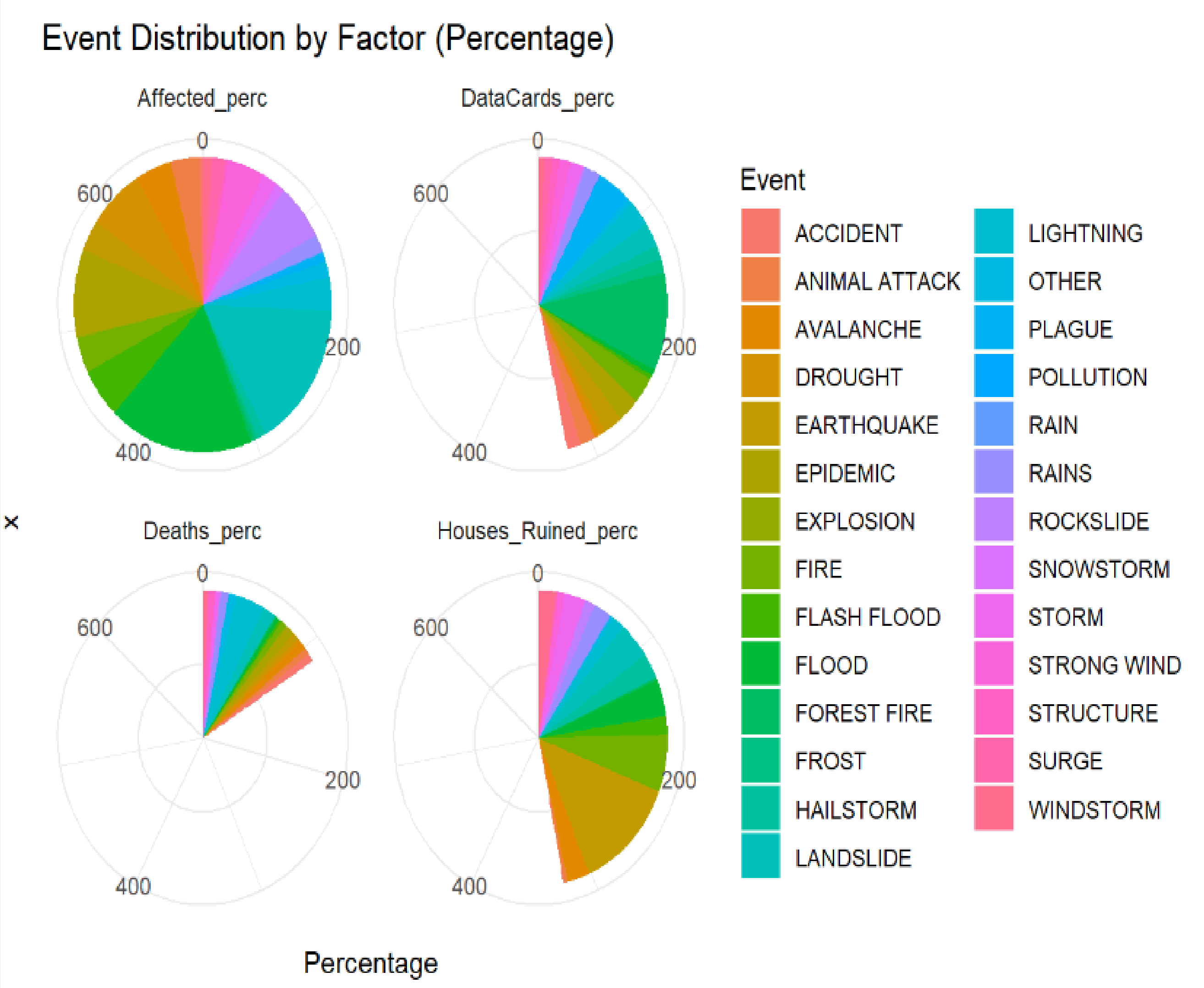
Pre- processing Visualization

Cleaned data visualization from each continent to all data combined



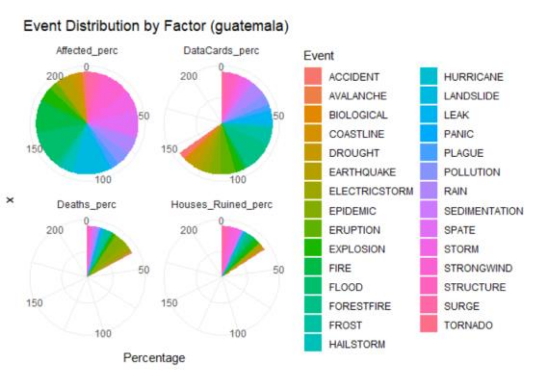


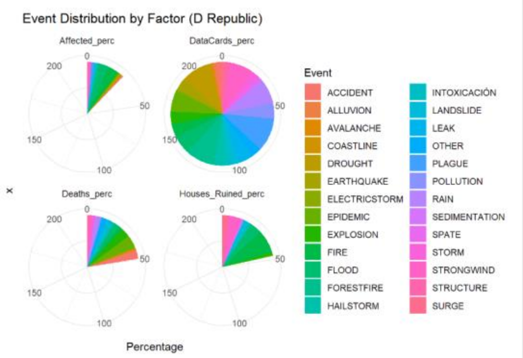


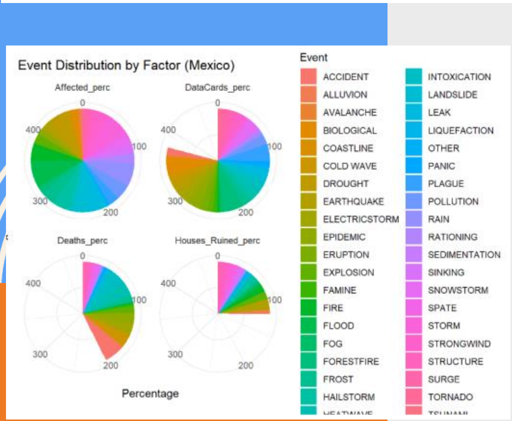


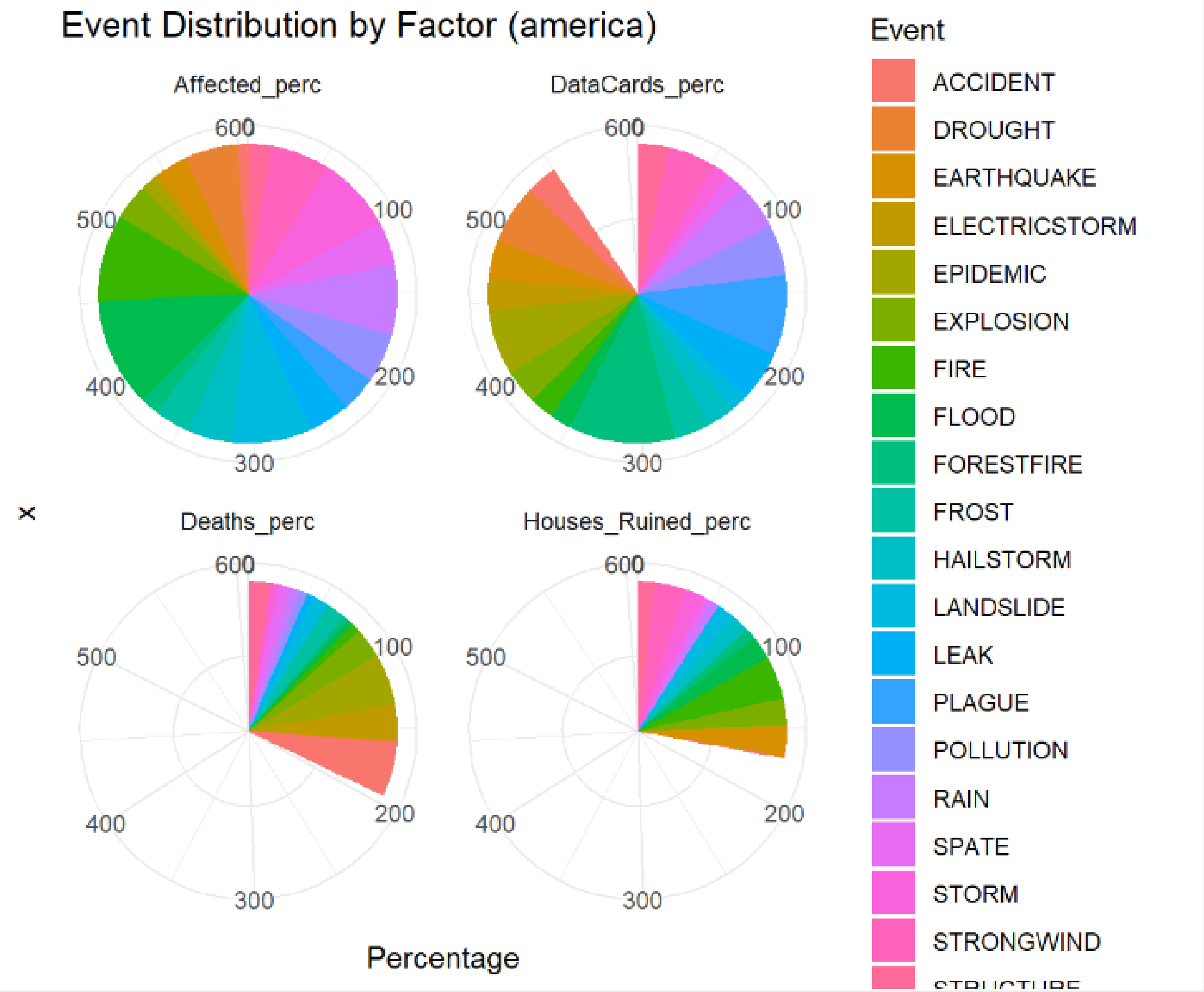
Pre- processing Visualization

Cleaned data visualization from each country to certain continent . For example : Guatemala ,Mexico & Dominical Republic data is feeding to America data . Below are visualization of Mexico and America

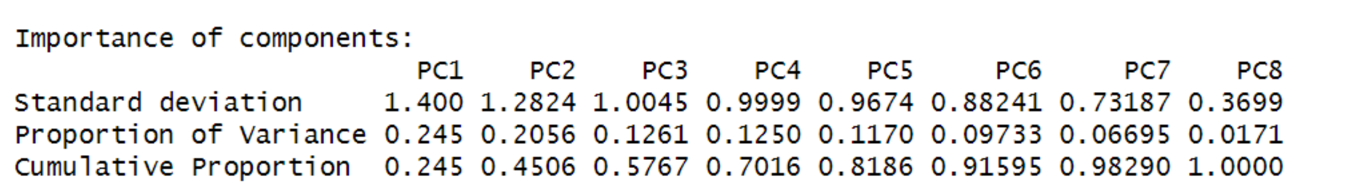




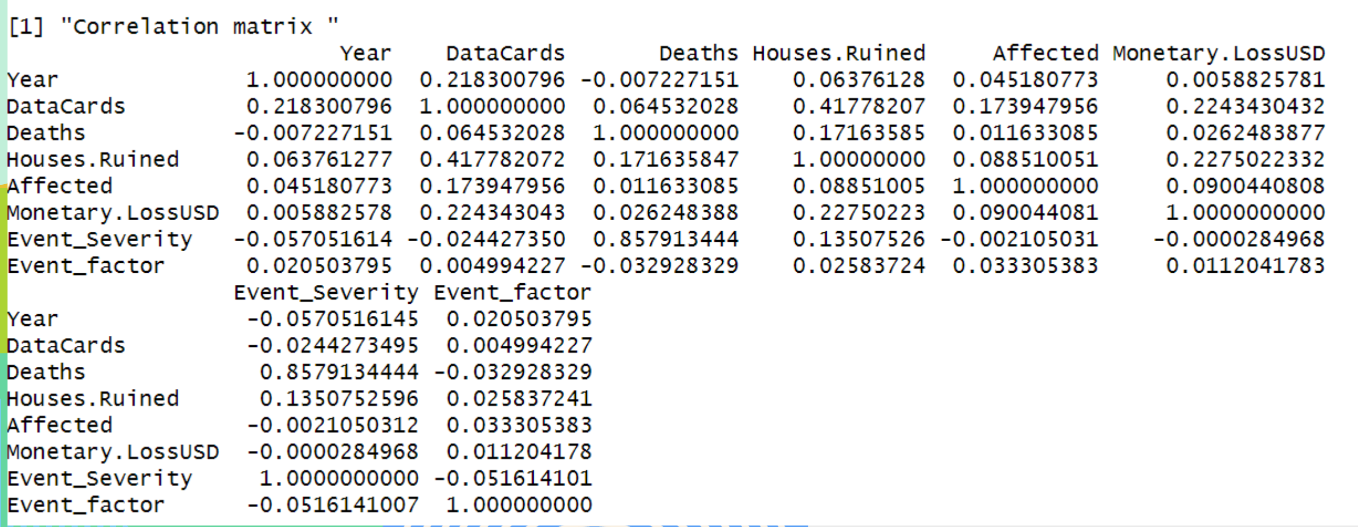




PCA , Correlation, Multicollinearity Checks (continent)



```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7    PC8
Standard deviation      1.400 1.2824 1.0045 0.9999 0.9674 0.88241 0.73187 0.3699
Proportion of Variance 0.245 0.2056 0.1261 0.1250 0.1170 0.09733 0.06695 0.0171
Cumulative Proportion  0.245 0.4506 0.5767 0.7016 0.8186 0.91595 0.98290 1.0000
```



```
[1] "Correlation matrix "
                        Year    DataCards       Deaths Houses.Ruined     Affected Monetary.LossUSD
Year             1.000000000  0.218300796 -0.007227151    0.06376128  0.045180773     0.0058825781
DataCards        0.218300796  1.000000000  0.064532028    0.41778207  0.173947956     0.2243430432
Deaths          -0.007227151  0.064532028  1.000000000    0.17163585  0.011633085     0.0262483877
Houses.Ruined    0.063761277  0.417782072  0.171635847    1.00000000  0.088510051     0.2275022332
Affected         0.045180773  0.173947956  0.011633085    0.08851005  1.000000000     0.0900440808
Monetary.LossUSD 0.005882578  0.224343043  0.026248388    0.22750223  0.090044081     1.0000000000
Event_Severity  -0.057051614 -0.024427350  0.857913444    0.13507526 -0.002105031    -0.0000284968
Event_factor     0.020503795  0.004994227 -0.032928329    0.02583724  0.033305383     0.0112041783
                Event_Severity Event_factor
Year             -0.0570516145  0.020503795
DataCards        -0.0244273495  0.004994227
Deaths            0.8579134444 -0.032928329
Houses.Ruined     0.1350752596  0.025837241
Affected         -0.0021050312  0.033305383
Monetary.LossUSD -0.0000284968  0.011204178
Event_Severity    1.0000000000 -0.051614101
Event_factor     -0.0516141007  1.000000000
```

Principal Component Analysis :

We have checked the PCA for each column and Till PC7 of our numeric data explains 98% of data and PC8 is the only numeric relation for event (Event_Factor).

Correlation & Multicollinearity Check:

Considering 'Death' as the dependent variable , we have calculated the Variance Inflation Factor to understand the correlation between Death and other attributes of the dataset and also if Multicollinearity exists .

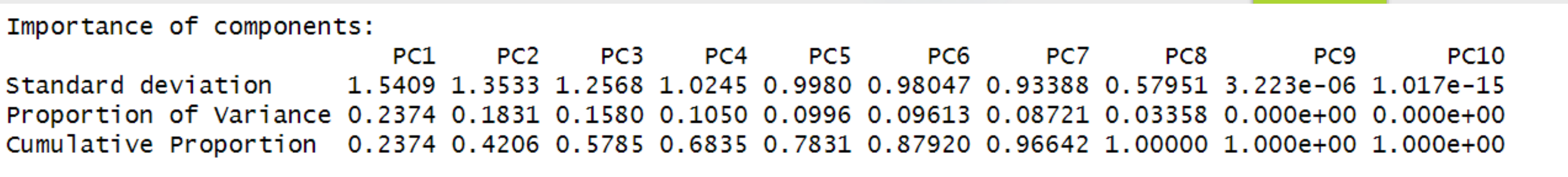
Example : Checks on America suggest no multicollinearity and higher correlation on Event Severity

Split:

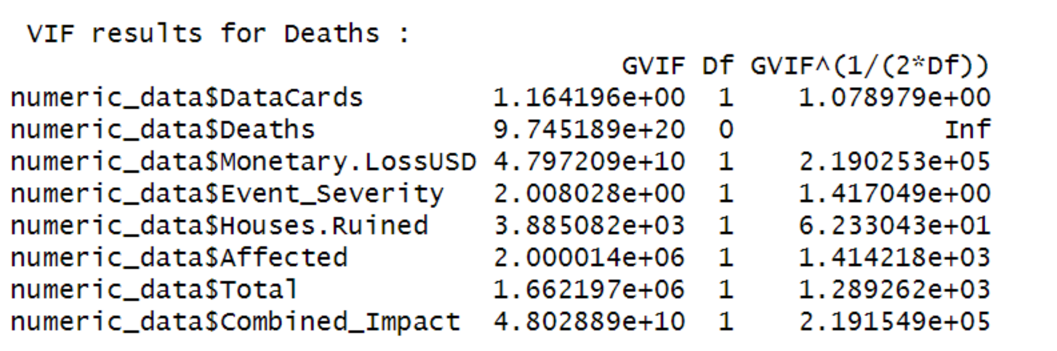
We split our data into 70 % training and 30 % testing.

PCA , Correlation, Multicollinearity Checks for (countryc)

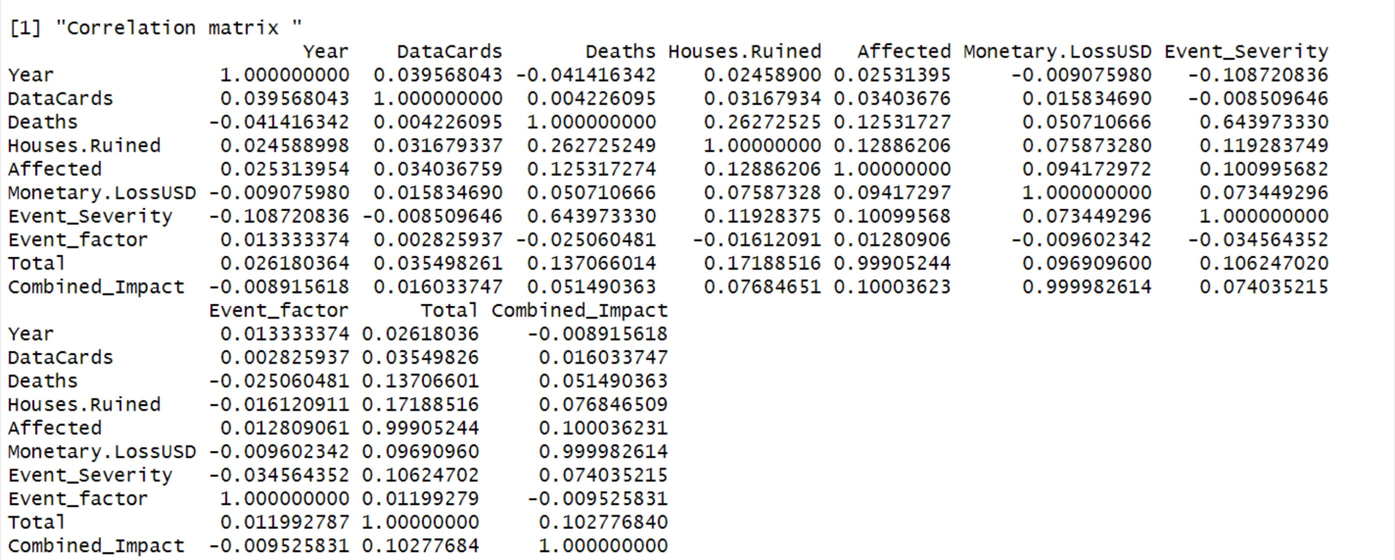
The two new feature Total and combined_Impact were introduced after cluster 1 which we have discussed after : this regression was performed after Clustering for each continent.



```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7     PC8       PC9      PC10
Standard deviation     1.5409 1.3533 1.2568 1.0245 0.9980 0.98047 0.93388 0.57951 3.223e-06 1.017e-15
Proportion of Variance 0.2374 0.1831 0.1580 0.1050 0.0996 0.09613 0.08721 0.03358 0.000e+00 0.000e+00
Cumulative Proportion  0.2374 0.4206 0.5785 0.6835 0.7831 0.87920 0.96642 1.00000 1.000e+00 1.000e+00
```



```
 VIF results for Deaths :
                                         GVIF Df GVIF^(1/(2*Df))
numeric_data$DataCards           1.164196e+00  1    1.078979e+00
numeric_data$Deaths              9.745189e+20  0             Inf
numeric_data$Monetary.LossUSD    4.797209e+10  1    2.190253e+05
numeric_data$Event_Severity      2.008028e+00  1    1.417049e+00
numeric_data$Houses.Ruined       3.885082e+03  1    6.233043e+01
numeric_data$Affected            2.000014e+06  1    1.414218e+03
numeric_data$Total               1.662197e+06  1    1.289262e+03
numeric_data$Combined_Impact     4.802889e+10  1    2.191549e+05
```



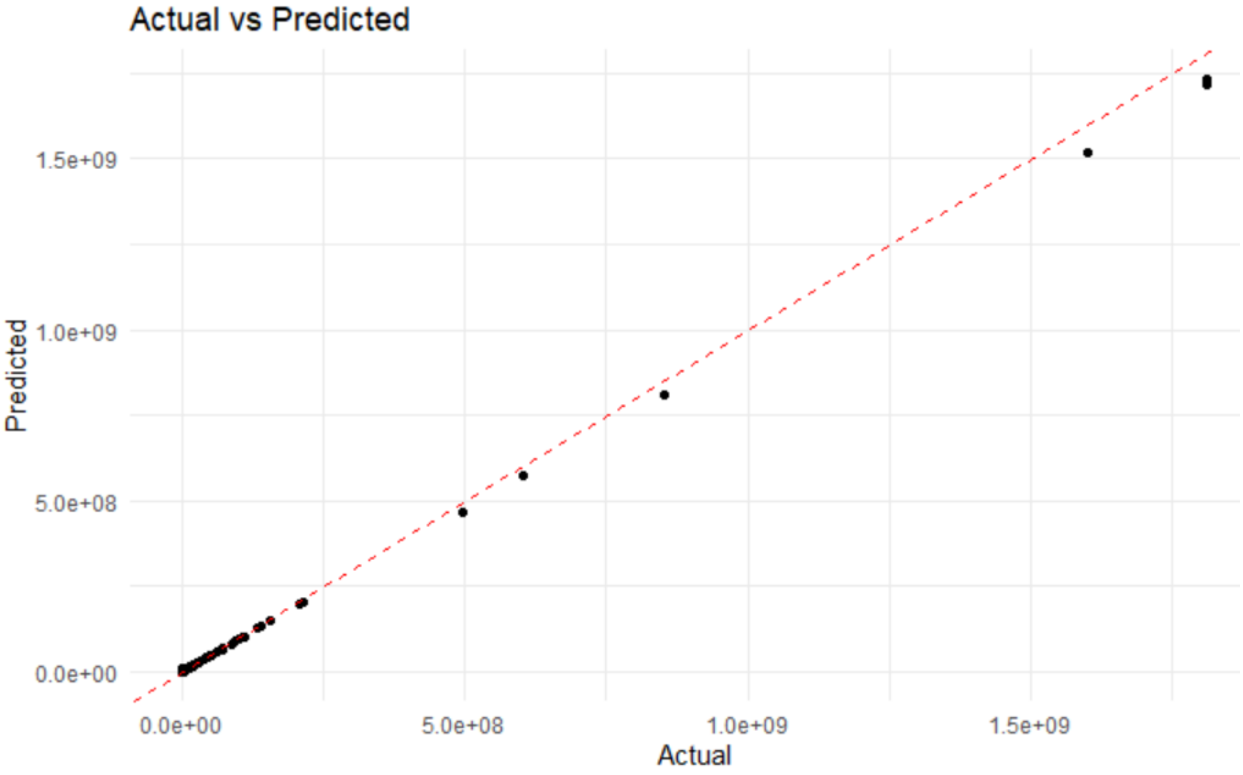
```
[1] "Correlation matrix "
                       Year    DataCards       Deaths Houses.Ruined   Affected Monetary.LossUSD Event_Severity
Year            1.000000000  0.039568043 -0.041416342    0.02458900 0.02531395     -0.009075980   -0.108720836
DataCards       0.039568043  1.000000000  0.004226095    0.03167934 0.03403676      0.015834690   -0.008509646
Deaths         -0.041416342  0.004226095  1.000000000    0.26272525 0.12531727      0.050710666    0.643973330
Houses.Ruined   0.024588998  0.031679337  0.262725249    1.00000000 0.12886206      0.075873280    0.119283749
Affected        0.025313954  0.034036759  0.125317274    0.12886206 1.00000000      0.094172972    0.100995682
Monetary.LossUSD -0.009075980  0.015834690  0.050710666    0.07587328 0.09417297      1.000000000    0.073449296
Event_Severity -0.108720836 -0.008509646  0.643973330    0.11928375 0.10099568      0.073449296    1.000000000
Event_factor    0.013333374  0.002825937 -0.025060481   -0.01612091 0.01280906     -0.009602342   -0.034564352
Total           0.026180364  0.035498261  0.137066014    0.17188516 0.99905244      0.096909600    0.106247020
Combined_Impact -0.008915618  0.016033747  0.051490363    0.07684651 0.10003623      0.999982614    0.074035215
                Event_factor      Total Combined_Impact
Year             0.013333374 0.02618036    -0.008915618
DataCards        0.002825937 0.03549826     0.016033747
Deaths          -0.025060481 0.13706601     0.051490363
Houses.Ruined   -0.016120911 0.17188516     0.076846509
Affected         0.012809061 0.99905244     0.100036231
Monetary.LossUSD -0.009602342 0.09690960     0.999982614
Event_Severity  -0.034564352 0.10624702     0.074035215
Event_factor     1.000000000 0.01199279    -0.009525831
Total            0.011992787 1.00000000     0.102776840
Combined_Impact -0.009525831 0.10277684     1.000000000
```

Regression Models (Continent)

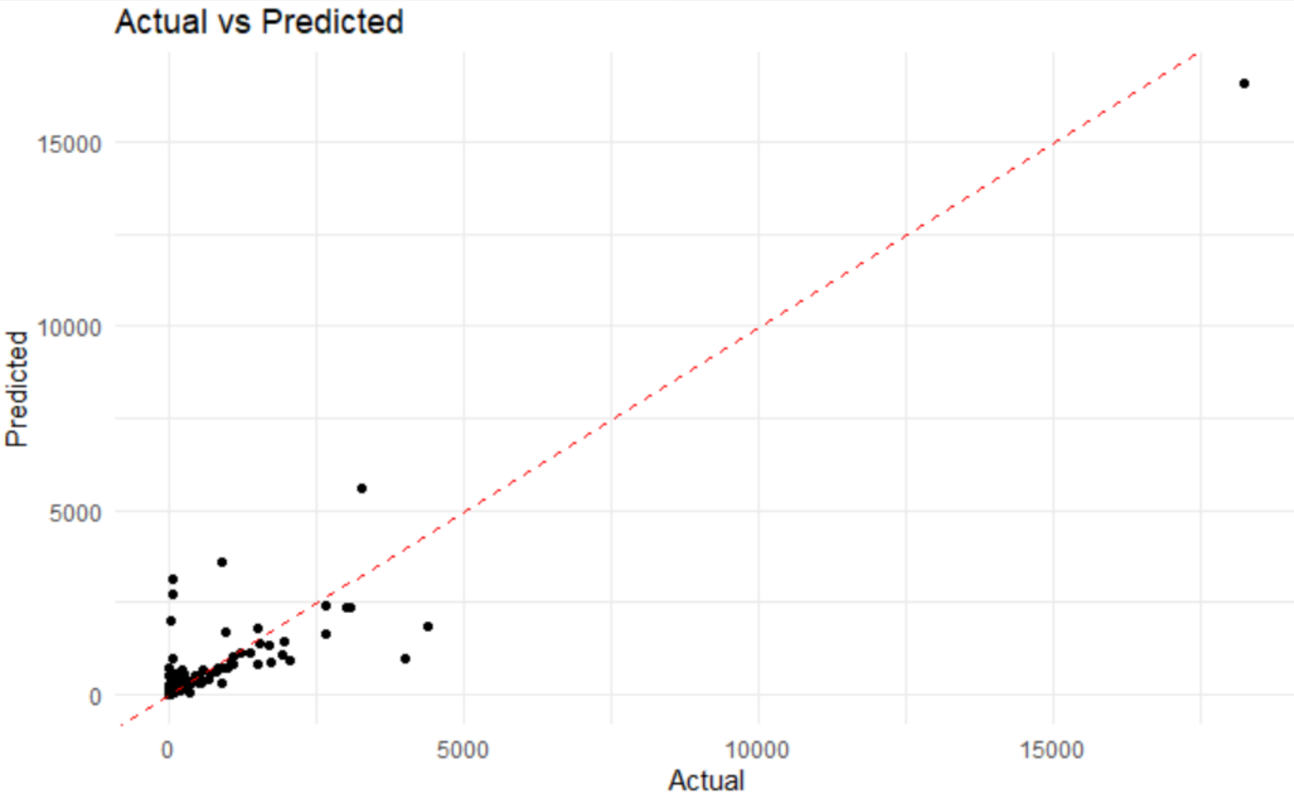
Regression Model Selection: We performed regression on each continent and then on each country and selected Elastic Net Regression Model & Random Forest Regression Model.

- **Primary Aim - To explain the deaths and Monetary loss in USD of each country and the continent , separately.**
- **Performance & Accuracy observation , we have decided to train and test our data with both the regression model to provide a comparative idea.**

✓ **Elastic net model accuracy on MonetaryUsd(Country C) –** At Lambda : 4049919 with Accuracy : 99.99% .

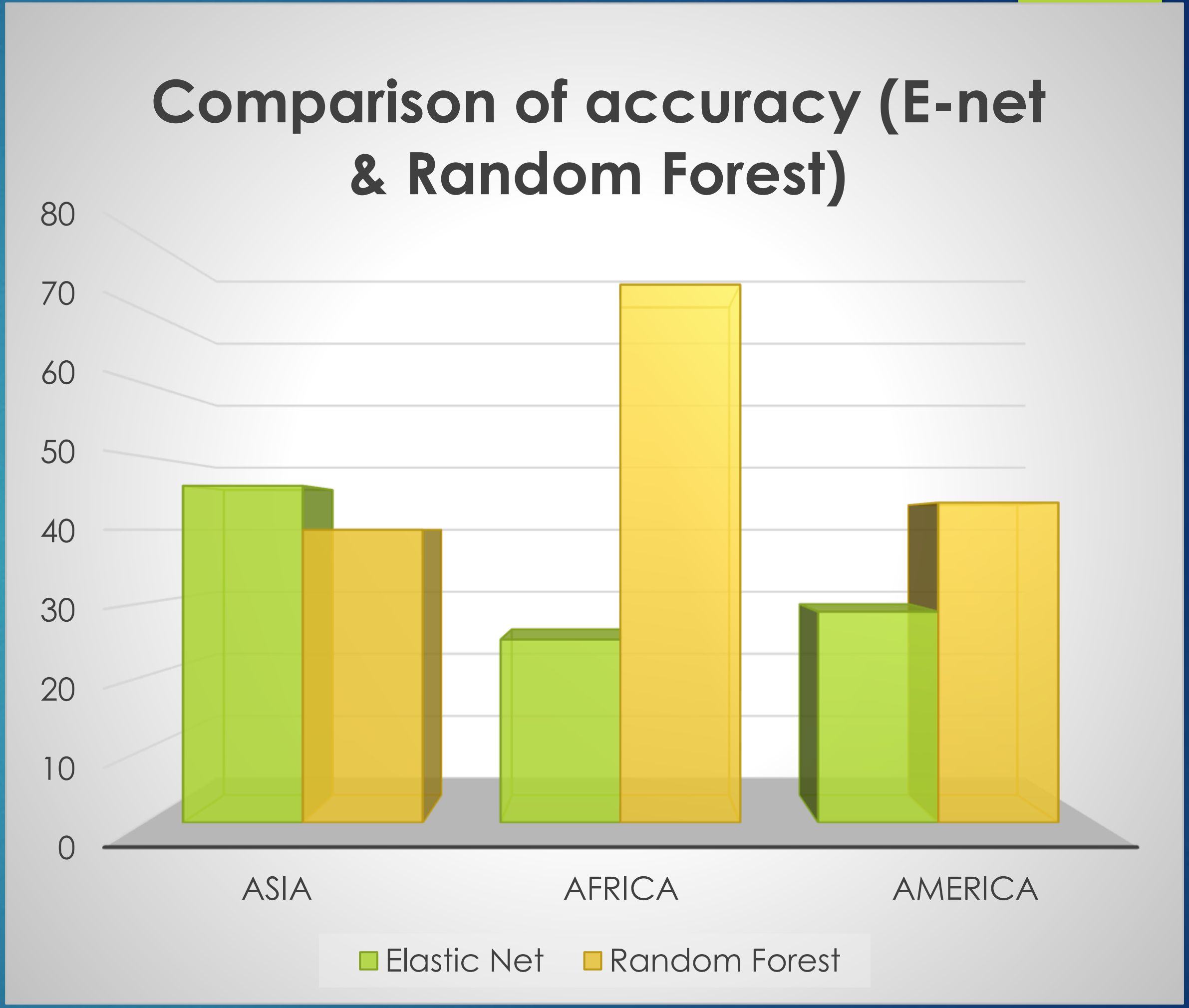


✓ **Random Forest model accuracy on Combined Data -** at 1000 tree with Accuracy : 86% to detect the inconsistency we performed the 2 models on every continent and further on every country



Predicting Deaths: On every Continent

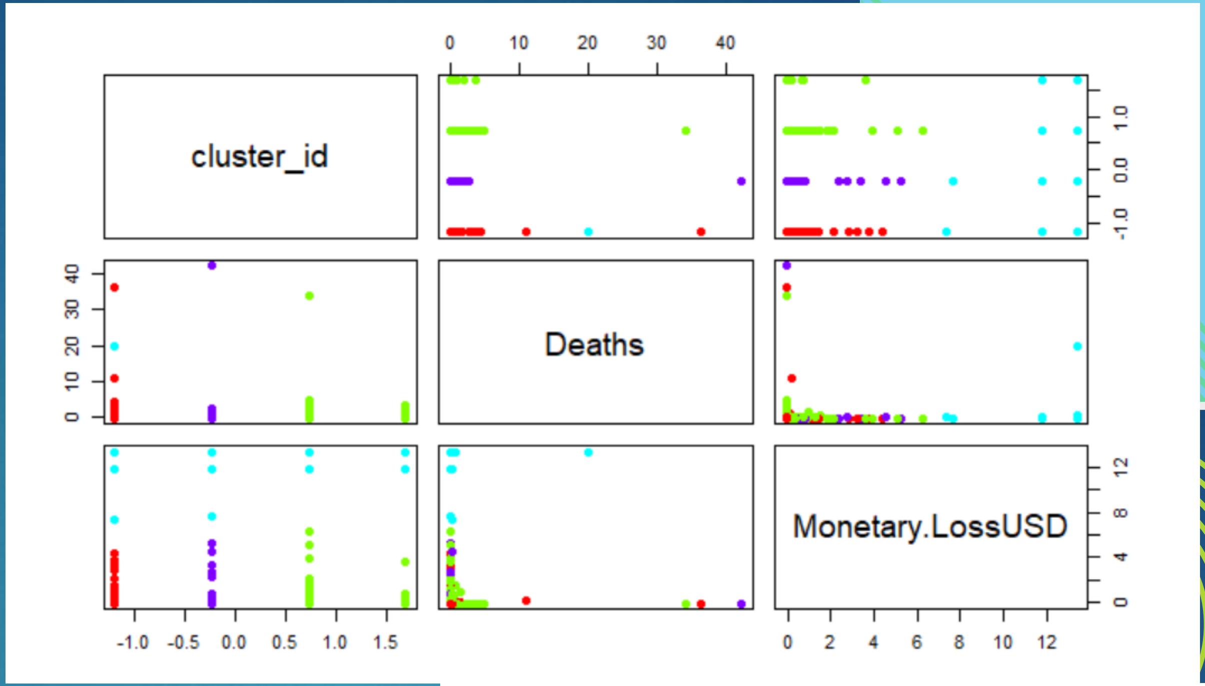
- Asia - E-net accuracy(46%) is better than Random Forest accuracy(44%). But for Pakistan & Sri Lanka is not responding well
- Africa – Random forest is performing better (Acc : 73.5%) than E-Net (Acc : 25%) . Tunisia and Senegal is not performing well
- America - Random forest is performing better (Acc : 43.7%) than E-Net (Acc : 28.8%). Dominican Republic is not performing well.

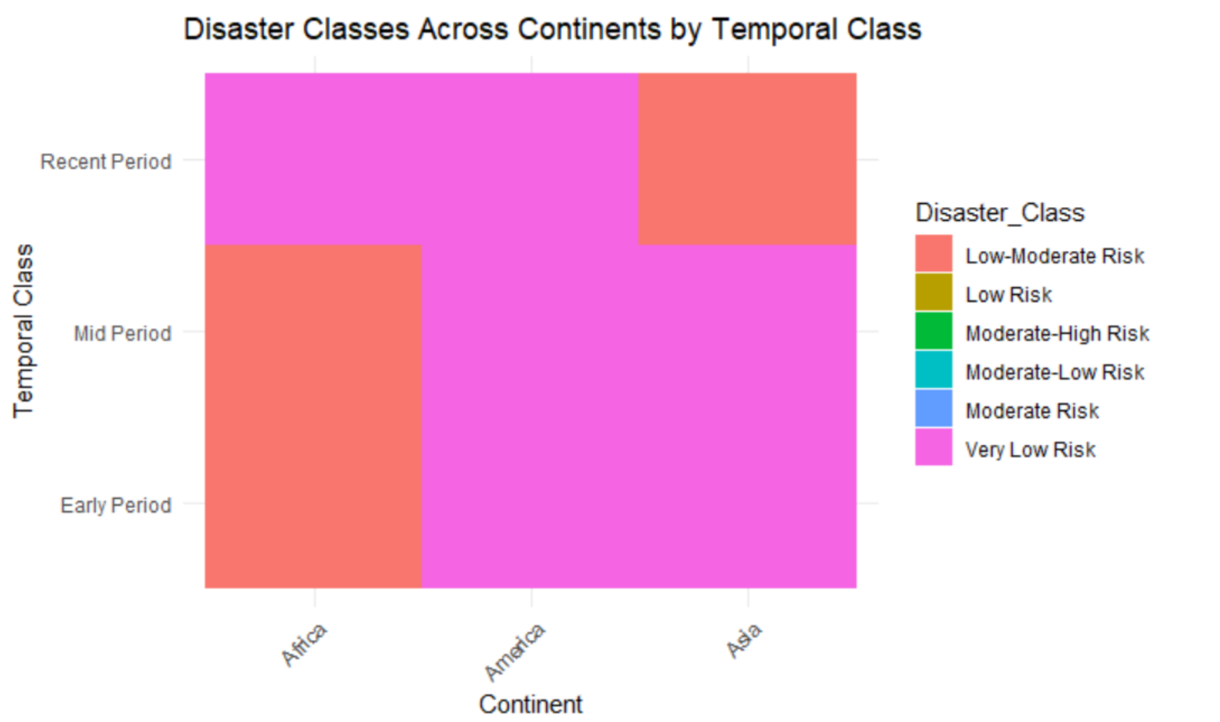


CLUSTER DISTRIBUTION

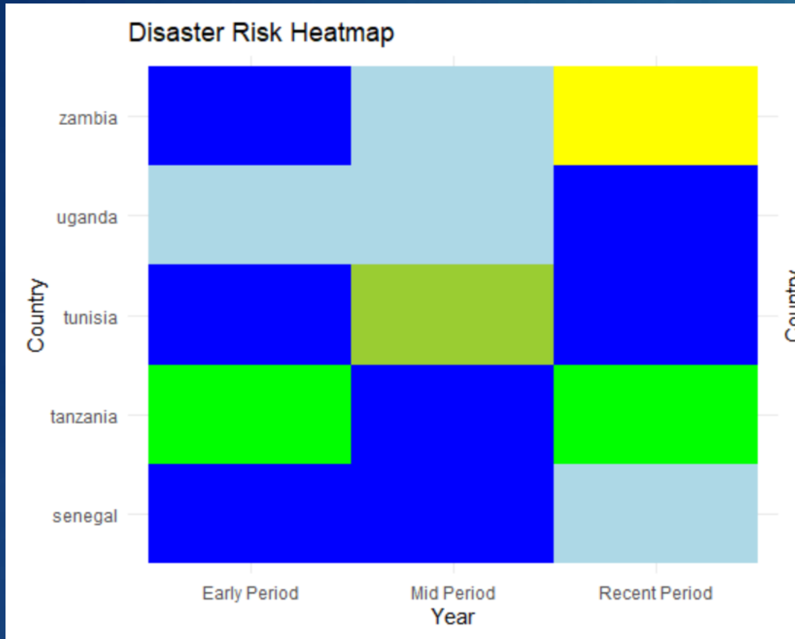
- K-medoids / kmean : We have considered the attributes ClusterID(), Monetary USD, and Deaths calculated K-medoid with 5 clusters which we have presented using the following pair plot :

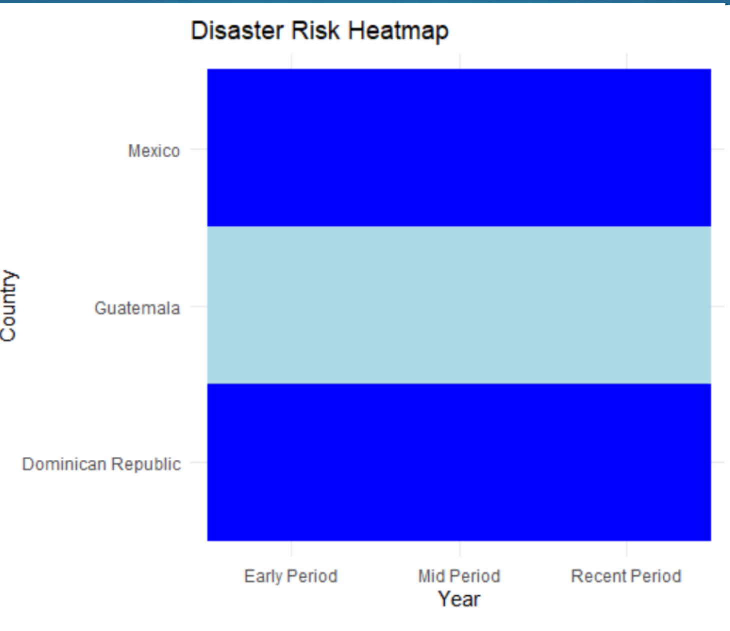
- Class Risk level –
A custom function classifies disaster risks based on severity, impact, and cluster characteristics, providing risk levels.

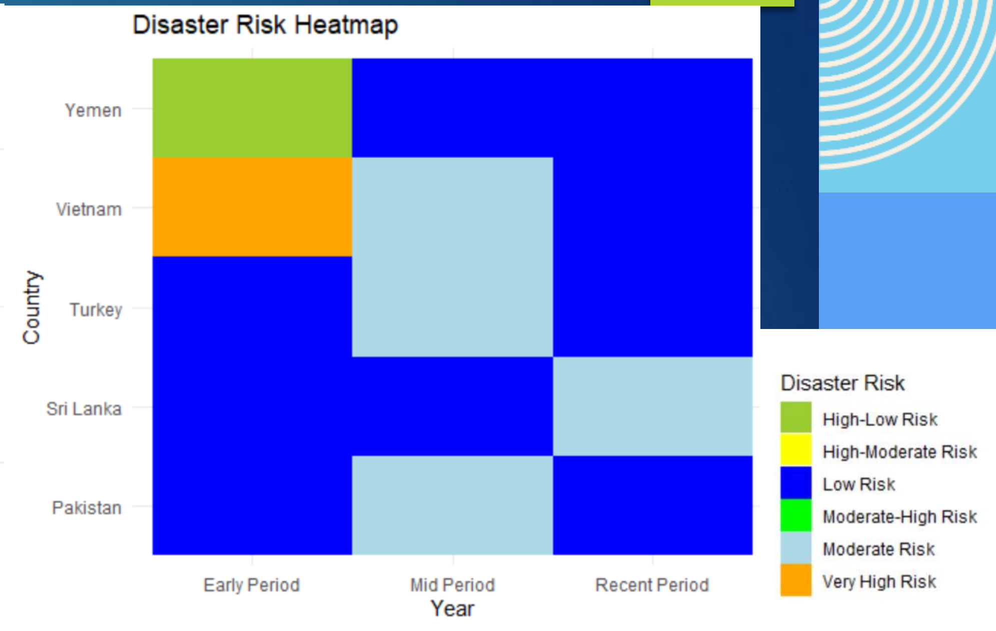


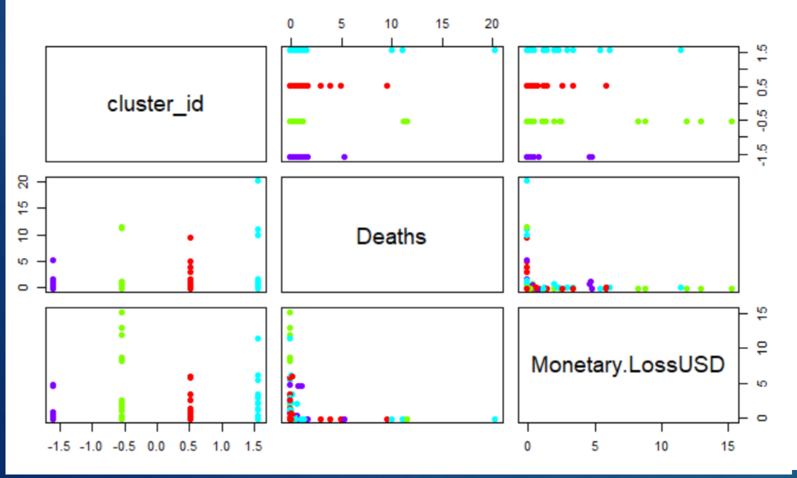


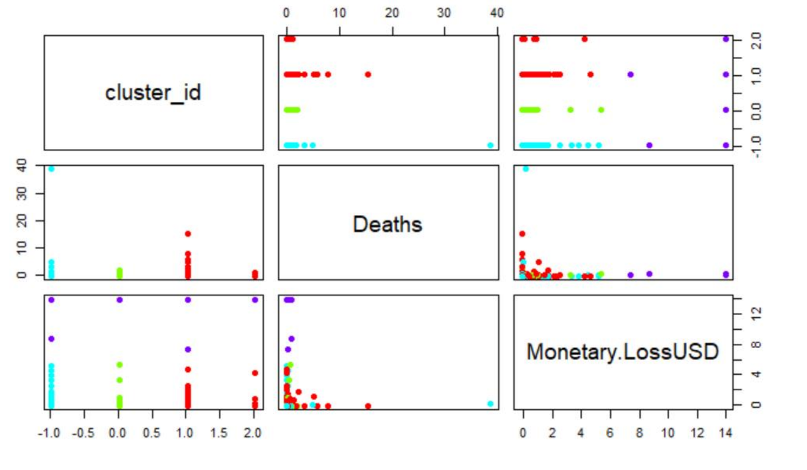
Classification on Risk level of country and Death and Loss in USD Distribution with respect to DataCards.

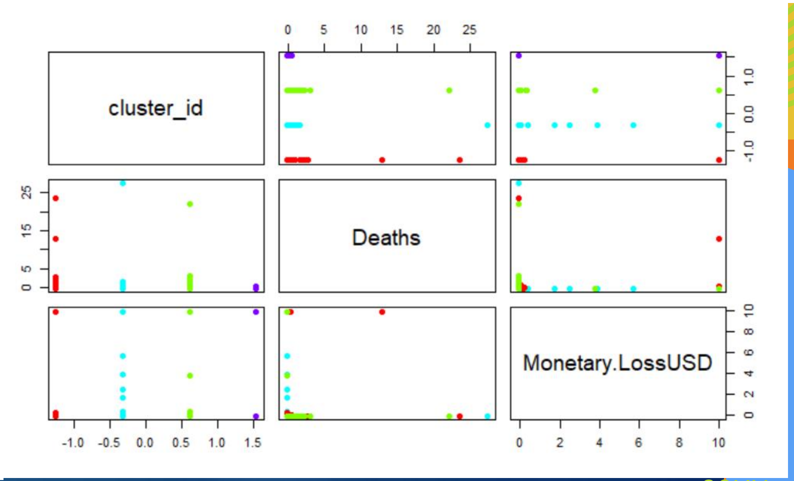


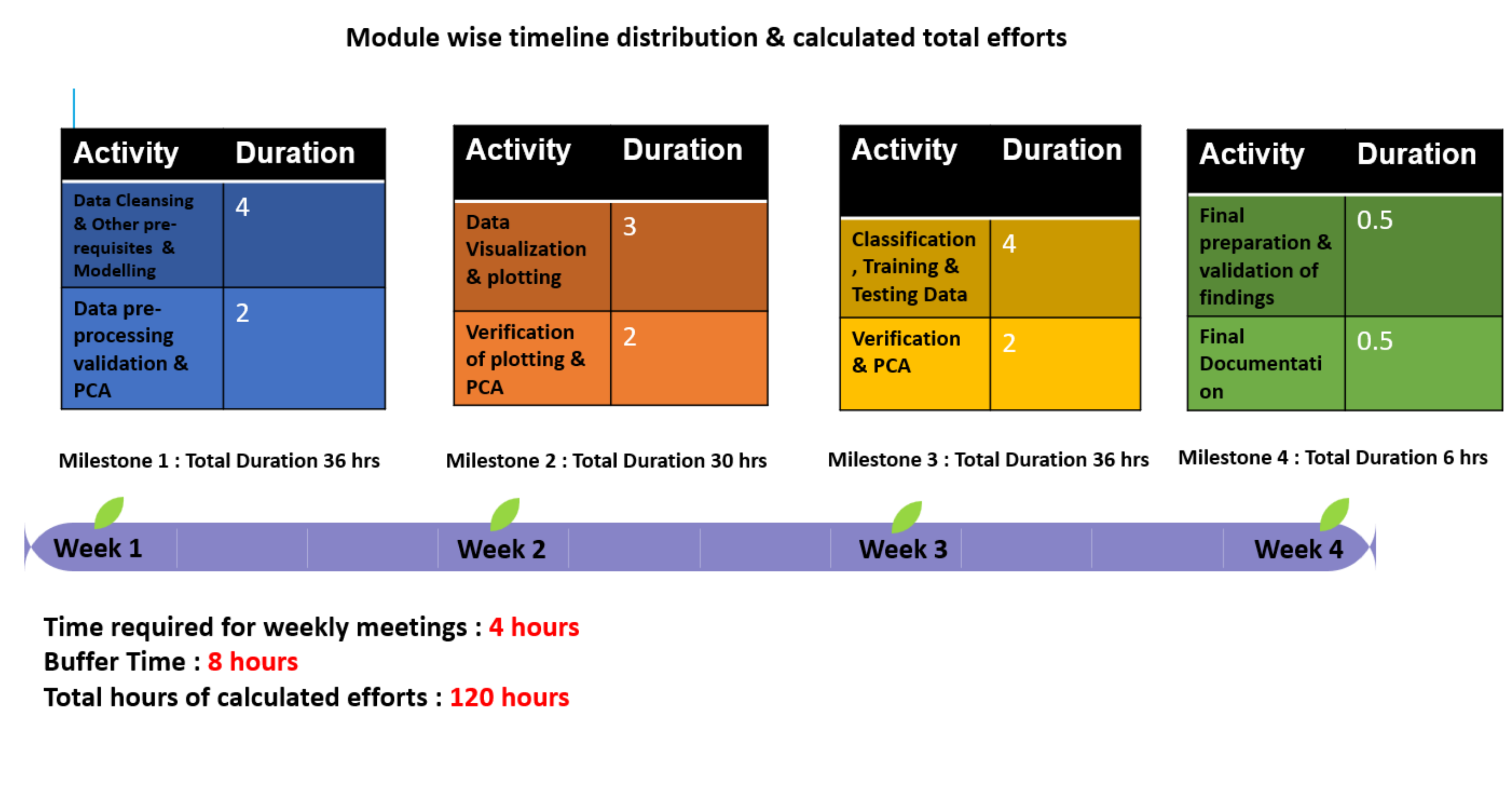












Module wise timeline distribution & calculated total efforts

Activity	Duration
Data Cleansing & Other pre-requisites & Modelling	4
Data pre-processing validation & PCA	2

Milestone 1 : Total Duration 36 hrs

Activity	Duration
Data Visualization & plotting	3
Verification of plotting & PCA	2

Milestone 2 : Total Duration 30 hrs

Activity	Duration
Classification , Training & Testing Data	4
Verification & PCA	2

Milestone 3 : Total Duration 36 hrs

Activity	Duration
Final preparation & validation of findings	0.5
Final Documentation	0.5

Milestone 4 : Total Duration 6 hrs

Time required for weekly meetings : 4 hours

Buffer Time : 8 hours

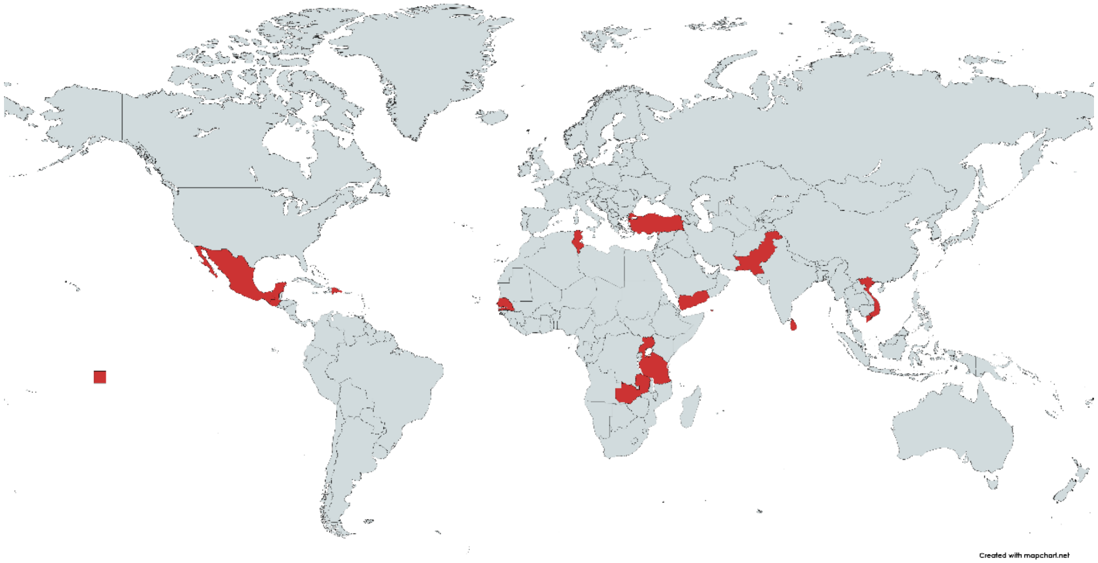
Total hours of calculated efforts : 120 hours

ACTION PLAN

Contents

Introduction

This coursework involves studying information about disasters from a website called https://www.desinventar.net/DesInventar/. This website keeps track of various disasters in approximately 90 countries worldwide. The United Nations Office for Disaster Risk Reduction is the primary supporter of this database, but other organizations have also contributed to its development. Our main task is to investigate how different types of disasters have affected various countries. In this assignment we are considering 13 countries: Dominican Republic, Guatemala, Mexico, Pakistan, Senegal, Sri Lanka, Tunisia, Turkey, Uganda, United Republic of Tanzania, Vietnam, Yemen, and Zambia



Overview

Data Cleansing:

- Cleaned special characters and removed duplicate rows.
- Ensured "Nan" values were within an omittable limit (<5%).
- Standardized event names to uppercase for consistency.

Data Observation & Pre-processing:

- Identified specific cases such as extreme monetary losses, missing values, and language inconsistencies across countries.
- Applied tailored pre-processing methods to handle these cases:
  - Addressed outliers in extreme monetary losses.
  - Handled missing loss values.
  - Translated event names for consistency.
  - Merged and transformed columns to unify similar data types.

Data Transformation:

- Engineered new features like severity scores based on death-to-data ratio, adjusting and scaling these scores for relevance.
- Merged and transformed columns to create more cohesive and informative datasets.

Visualizing Data/ Model / Cluster:

- Visualized cleaned datasets by continent and country-wise for enhanced insights.

Regression Modelling:

- Utilized Elastic Net and Random Forest Regression models for predicting deaths and monetary losses in USD across continents and countries.
- Focused on understanding model performance variations across different regions and countries.

Clustering Analysis:

- Employed K-Means and K-Medoids clustering techniques to categorize data based on temporal, severity, and impact attributes.
- Visualized the clusters for better understanding using heatmaps and pair plots.

Insights:

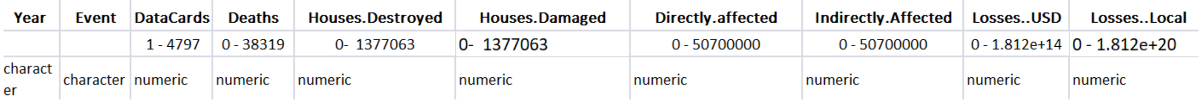
- Identified varying model performances across continents and countries.
- Highlighted the strengths and weaknesses of regression models in predicting disaster impacts.
- Utilized clustering techniques to categorize and visualize data attributes, aiding in data comprehension.

Overall, our analysis methodically processed the dataset, employed regression modeling, and used clustering techniques to extract meaningful insights about disaster impacts across various regions and their predictive factors.

Dataset Description

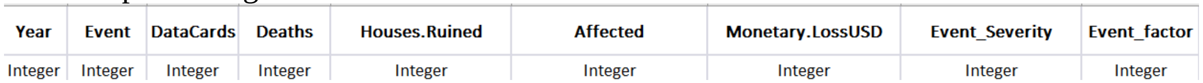
- The initial row delineates the column headers across all variables encompassing country, continent, and combined datasets. Each column signifies a distinct attribute within the dataset, specifying the diverse characteristics and information captured.
- The subsequent row illuminates the range of values encapsulated within each column, providing insights into the spread and distribution of numerical data. It delineates the minimum and maximum values or statistical indicators that outline the extent of the dataset's numerical composition.
- Following the range depiction, the subsequent row sheds light on the mode or characteristic manner in which data appears within each column. It signifies the predominant class, type, or mode of data exhibited within the dataset, offering insights into the most frequently occurring data category or structure within each attribute.

Raw Dataset



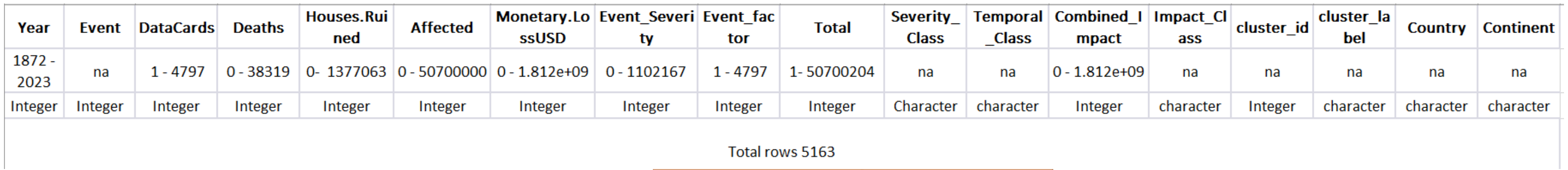
Year	Event	DataCards	Deaths	Houses.Destroyed	Houses.Damaged	Directly.affected	Indirectly.Affected	Losses..USD	Losses..Local
		1 - 4797	0 - 38319	0- 1377063	0- 1377063	0 - 50700000	0 - 50700000	0 - 1.812e+14	0 - 1.812e+20
character	character	numeric	numeric	numeric	numeric	numeric	numeric	numeric	numeric

After Preprocessing



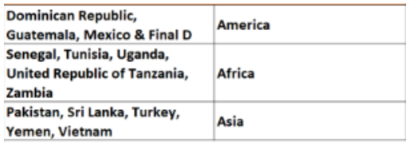
Year	Event	DataCards	Deaths	Houses.Ruined	Affected	Monetary.LossUSD	Event_Severity	Event_factor
Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer

Final dataset with cluster –



Year	Event	DataCards	Deaths	Houses.Ruined	Affected	Monetary.LossUSD	Event_Severity	Event_factor	Total	Severity_Class	Temporal_Class	Combined_Impact	Impact_Class	cluster_id	cluster_label	Country	Continent
1872 - 2023	na	1 - 4797	0 - 38319	0- 1377063	0 - 50700000	0 - 1.812e+09	0 - 1102167	1 - 4797	1- 50700204	na	na	0 - 1.812e+09	na	na	na	na	na
Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Integer	Character	character	Integer	character	Integer	character	character	character

Total rows 5163



Dominican Republic, Guatemala, Mexico & Final D	America
Senegal, Tunisia, Uganda, United Republic of Tanzania, Zambia	Africa
Pakistan, Sri Lanka, Turkey, Yemen, Vietnam	Asia

R Libraries

dplyr

Description: dplyr is a powerful package for data manipulation. It provides a set of functions, so we use select and mutate in our project.

tidyr

Description: tidyr focuses on reshaping and tidying data. It includes functions like **pivot_longer()** that help transform data between wide and long formats.

ggplot2

Description: ggplot2 is a data visualization package that implements the grammar of graphics. It allows for the creation of intricate and customizable plots using a layered approach.

car

Description: car stands for "Companion to Applied Regression." It includes various functions useful for regression analysis, diagnostics, and plotting for regression models. Like VIF.

glmnet

Description: glmnet is a package primarily used for fitting generalized linear models with elastic net regularization. It's helpful for handling high-dimensional data and performing variable selection.

caret

Description: caret (Classification And Regression Training) is a versatile package for training and testing machine learning models. It provides a unified interface for model training, parameter tuning, and evaluation.

cluster

Description: cluster is a package for clustering analysis. Functions like **clara()** can be used for clustering large datasets by creating subgroups based on similarity.

Data Preprocessing

Data pre-processing plays a pivotal role in readying raw data for comprehensive analysis and modeling. This report outlines the crucial steps involved in data pre-processing, elucidating their significance in enhancing data quality for next analytical procedures and modeling tasks.

Also, the systematic process employed in R for reading, cleaning, and preparing data sourced from multiple CSV files for next analysis. The focus lies on the steps undertaken to ensure data integrity, handle missing values, and eliminate duplicates to facilitate accurate analysis.

Data Cleaning

Reading CSV Files:

The code uses the read.csv function to ingest data from 13 distinct CSV files, notably "country.csv". The parameter header=TRUE signifies that the first row of each CSV file holds column names, while na. strings =c ("") is employed to denote empty strings ("") as missing values (NA).

Cleaning Column Names:

Upon reading each CSV file, the code undertakes column name clean-up:

- Use of the colnames () function for accessing and changing column names.
- Implementation of gsub ("^\s+|\s+$", "", colnames ()) with a regular expression (gsub ()) to drop leading and trailing spaces from column names.

Checking Missing Values:

The code initiates by printing the total count of missing values within each dataset (country) for analysis. The sum(is.na(data_frame)) function computes the aggregate number of missing values (NA) in each dataset.

Handling Missing Values:

Post identification of missing values, the code selectively removes rows holding missing values if the percentage of missing data is less than 5% for specific columns in each dataset. This process is executed using the na. omit () function.

Checking and Removing Duplicate Values:

The treatment of missing values, the code proceeds to inspect duplicate rows within each dataset. The sum(duplicated(data_frame)) function quantifies the total count of duplicated rows in each dataset. Subsequently, efforts are made to drop these duplicate entries to ensure uniqueness across all records.

Text modification:

- The text cleaning process begins by standardizing text case, converting all entries in the country$Event column to uppercase for consistency.
- Next, unique event descriptions are extracted to create a reference list aiding in the construction of a translation dictionary. An empty dictionary is initialized to map Mexican words in the country$Event column to their English translations. A defined function, translate_mexican_to_english, handles text conversion, whitespace removal, and translation lookup from the initialized dictionary. This function is then applied to each country$Event entry using sapply(), facilitating the translation of Mexican text to English based on the provided dictionary.

Data Transformation

Case 1: Data Transformation

- **Objective**:
  - The code aims to streamline the dataset's structure by consolidating information from six columns into three essential categories.
- **Actions Taken**:
  - Combined **Houses.Destroyed** and **Houses.Damaged** columns into **Houses.Ruined**.
  - Aggregated **Directly.affected** and **Indirectly.Affected** columns to form **Affected**.
  - Calculated **Monetary.LossUSD** by summing **Losses..USD** and a portion of **Losses..Local**, converting the resulting

    currency to an integer.
    - Removed columns 5 to 10 from the dataset to enhance clarity and focus on essential information.

Case 2: Data Transformation with Threshold Adjustment(turkey, mexico)

- **Objective**:
    - Similar to Case 1, this section deals with data transformation but incorporates handling missing values and setting a threshold.
- **Actions Taken**:
    - Replicated Case 1 actions and additionally handled missing values in **Monetary.LossUSD**.
    - Replaced missing values in **Monetary.LossUSD** using either the maximum non-missing value or a predefined threshold (**e9**) based on country-specific considerations.
    - Removed columns 5 to 10 from the dataset to maintain consistency in data representation.

Feature 1: add_Event_Severity Function

- **Objective**:
    - Introducing a function to derive a new metric, **Event_Severity**, showcasing the relationship between **Deaths** and **DataCards**.
- **Functionality**:
    - Created **add_Event_Severity** function to compute **Event_Severity** as the ratio of **Deaths** to **DataCards**.
    - Incorporated adjustments based on observed trends in specific countries (**a_countries** and **b_countries**), converting resulting ratios to integer values by rounding.

Feature 2: Event_factor Calculation.

- **Objective**:
    - Conversion of the **Event** column into a categorized integer representation.
- **Conversion Process**:
    - Transformed **Event** into an integer factor representation (**Event_factor**) to facilitate categorical analysis and classification.

The provided code executes a series of data transformations, combining columns, computing monetary values, handling missing data, and adjusting thresholds. Additionally, two features are introduced: a function to calculate event severity considering country-specific trends and a conversion of events into a categorized integer representation for analytical purposes. These actions collectively aim to streamline data representation and enhance analytical capabilities within the dataset.

Data Reduction:

Data Reduction through Principal Component Analysis (PCA) is a prevalent technique in data analysis and machine learning, transforming original variables into uncorrelated principal components. Here's a concise breakdown of the process:

1. **Exclusion of Non-Numeric Data:**

In the initial step of the data reduction process through Principal Component Analysis (PCA), the focus is on preparing the dataset for numerical analysis. Initially, a modified dataset, **"numeric_data"** is created by removing the non-numeric "Event" column. The exclusion of non-numeric data is a crucial pre-processing step in PCA because PCA is a technique that relies on numerical data to uncover patterns and relationships within the dataset. Also, for regression we need the data that is numeric in nature for better prediction. Including non-numeric variables could lead to complications during the modelling, as PCA operates on quantitative variables to calculate correlations and variances. This exclusion helps numerical analysis suitable for PCA.

2. **Principal Component Analysis (PCA):**

Before applying PCA, it is common practice to scale the data. Scaling involves centring the data by subtracting the mean of each variable, and then dividing by the standard deviation. This standardization ensures that variables are on a similar scale, preventing dominance by variables with larger magnitudes. PCA is executed on scaled data **("scaled_data")** using the **prcomp()** function. This function takes the scaled data as input and computes the principal components. The resulting object contains valuable information about the principal components, variance, and other essential details. Prior to extracting principal components, the data is centred and scaled. Here, **“pca_result”** holds valuable information about the principal components, variance, and other essential details.

3. **Summary and Results:**

A summary of PCA results highlights standard deviations, the variance explained by each principal part, the loadings matrix, and the transformed data. This summary encapsulates crucial information about the principal components, enabling us to gain deeper insights into the dataset's structure. Let's break down the key components of the summary:

a. **Standard Deviation:**

Reflects the variance captured by each principal part. Higher standard deviations show greater explanation of variance in the data.

b. **Proportion of Variance:**

Is the ratio of a component's squared standard deviation to the total sum of squared standard deviations of all components.

This proportion provides a quantitative measure of how much information each principal component contributes to the overall variance.

c. **Cumulative Proportion:**

Indicates the total variance proportion explained by the first N principal components. In our scenario, we aim to observe how many principal components are needed to capture a significant amount of variance. The cumulative proportion should ideally sum up to 1.00000.

Interpretation for Maximum Cases:

- We observe that PC1 explains around 30-40% of the total variance, and PC2 contributing 20-15% with subsequent principal components contributing in descending order.
- By the seventh PC, approximately 95% of the total variance is captured.
- PC8 adds minimal variance, bringing the cumulative proportion to 100%.

Practical Implications:

- <u>Dimensionality Reduction</u>: The analysis demonstrates that retaining the first six principal components captures a substantial portion (99.35%) of the total variance. This informed decision facilitates dimensionality reduction while maintaining a high level of information integrity.
- <u>Model Robustness</u>: The selection of primary principal components ensures model robustness, emphasizing the importance of these components in explaining the underlying variance within the dataset.

The **summary()** function helps us better grasp the results of Principal Component Analysis (PCA). It gives us valuable information to make smart choices about which principal components to keep for further analysis. By looking at things like standard deviations, proportion of variance, and cumulative proportion, we can decide which components best capture the important patterns in our data. This way, we choose the most meaningful pieces of information to work with, ensuring our analysis reflects the main characteristics of the dataset.

Dimensionality reduction can keep only the primary principal components that explain most variance. In this scenario, the first six components capture a large portion (99.35%) of variance, rendering the seventh component's removal inconsequential. It’s a sole relationship that pertains between "Event" and other components with numeric reference.

Data Splitting

Creating Data Partitions: Case 1

The process of partitioning data for analysis involves the use of the **createDataPartition()** function, primarily found in the caret package within R. This function runs by using the 'Year' column from the dataset related to the country's data. Its primary function is to segment the dataset into two distinct sets of indices, separating data for training and testing purposes.

- **p = 0.7:** This parameter signifies a division of 70% for training and reserves the remaining 30% for testing.
- **list = FALSE:** This specification ensures that the output of the function supplies a vector of indices rather than a list structure.

Splitting the Dataset:

Following the creation of data partitions, two distinct subsets are formed from the 'country_c' dataset:

- **training_data_country:** This subset is generated by extracting rows from 'country_c' based on the indices obtained through the **createDataPartition()** function. It serves the purpose of training various machine learning models.
- **testing_data_country:** Forming the data not included in the training set, this subset has the remaining part of the dataset. Typically, it is employed to assess the performance of the trained models.

Creating Data Partitions: Case 2 (Tunisia)

- Imbalanced distribution of years in dataset splitting can cause unequal representation in training and testing sets, potentially leading to skewed results and biased models.
- Uneven distribution across years might create disproportionate sample sizes in each set, affecting the model's performance and generalization.
- **Solution**:
  - *Stratified Splitting*: Implement a stratified splitting technique to support proportional representation of years in both training and testing sets. This method ensures a balanced distribution of years in each set, minimizing the risk of bias.
  - *Adjust Splitting Approaches*: Account for year imbalance by adjusting the splitting method. Consider alternative strategies that prioritize fair representation of all years in both sets, such as custom splitting algorithms and oversampling techniques.

- *Rationale*: By employing stratified splitting or changing traditional splitting methods, we aim to create more fair datasets for training and testing. This approach will mitigate biases introduced by imbalanced year distribution, promoting a more correct and reliable model performance across all years. Addressing the imbalanced year distribution in dataset splitting is crucial for developing robust models. Implementing stratified splitting and adjusting splitting approaches will foster fair representation of all years in training and testing sets, enhancing the reliability and performance of our models.

- Perform splitting similarly to Case 1

Data Visualization

The dataset includes information on events, data cards, deaths, houses ruined, and affected populations. The aim of this analysis was to assess the distribution of events concerning different factors, aiming to understand the relative impact of these factors within each event.

Methodology: The analysis process involved several steps:

1. **Data Preparation:** The dataset was processed to calculate the total value for each factor—DataCards, Deaths, Houses Ruined, and Affected—by summing their respective values.
2. **Percentage Calculation:** Percentages for each factor within every event were computed. This step allowed understanding the proportional contribution of each factor to the total within each event.
3. **Data Reshaping:** The data was transformed into a format suitable for visualization, helping the examination of each factor's distribution across different events.
4. **Visualization:** Using pie charts, the analysis visualized the distribution of events based on different factors. Each pie chart illustrated the proportionate contribution of factors within individual events.

Results:

The visualization highlighted a comprehensive view of how different factors—DataCards, Deaths, Houses Ruined, and Affected—contribute to various events within the dataset. This breakdown by factors enabled a clear understanding of their relative significance within different events.

Conclusion:

The analysis highlighted the varying impacts of different factors across events in the country dataset. By examining the proportional contribution of each factor within events, the analysis supplies insights into the relative importance of these factors in shaping the nature and severity of incidents.

Regression Modelling

Variance Inflation Factor

- A linear regression model was constructed with predictors including deaths, monetary loss in USD, event severity, houses ruined, and affected population.
- Variance Inflation Factor (VIF) analysis was employed using the **car** library in R to find multicollinearity among the predictor variables.

Findings:

- VIF scores were calculated for each predictor variable in the model.
- Higher VIF values suggest a stronger correlation between predictor variables, showing potential multicollinearity issues. And the low VIF Value suggest a weak correlation.

Interpretation:

- Elevated VIF values show substantial correlations among predictors, which can compromise the reliability and accuracy of the model.
- Addressing multicollinearity is crucial to ensure stable coefficient estimates and enhance predictive performance.

Recommendations:

- As we cannot removing highly correlated variables because they capture 50- 40 % of total variance and are crucial components
- We will Refine the model by reassessing variable selection and use regularization methods to mitigate multicollinearity effects.

Elastic- Net Regression model

Elastic Net regression is a hybrid model that combines the strengths of both Ridge and Lasso regression techniques. It's used in machine learning and statistics for regression analysis, especially when dealing with datasets that have multicollinearity (correlations between predictors) and a large number of predictors (high-dimensional data).

Here's a breakdown:

1. **Ridge Regression** introduces a penalty to the model by adding the squared size of coefficients to the cost function. It's effective at reducing multicollinearity by shrinking the coefficients, but it doesn't perform variable selection; it just regularizes.
2. **Lasso Regression** also introduces a penalty but uses the absolute values of coefficients. Lasso performs both

regularization and variable selection by shrinking the coefficients of less important features to exactly zero, effectively removing them from the model.

Elastic Net combines both Ridge and Lasso penalties in its cost function. It's controlled by two parameters: alpha (which determines the balance between Ridge and Lasso) and lambda (the strength of the regularization). This combination helps mitigate the limitations of each method alone, making Elastic Net more robust in selecting variables and handling multicollinearity.

It's particularly useful when dealing with datasets with many variables, some of which may be correlated. The Elastic Net's combined penalty term allows for feature selection while also handling correlated predictors better than Lasso alone.

In summary, Elastic Net regression is a versatile tool that balances between Ridge and Lasso regression, offering the advantages of both regularization methods and addressing their individual limitations.

Reasons for Employing Elastic Net Regression Despite Low Multicollinearity (Low VIF):

1. **Variable Importance:**
    - Elastic net (L1 & L2 regularization) shrinks coefficients to zero, aiding automatic feature selection, even with low multicollinearity.
2. **Handling Correlated Predictors:**
    - VIF might detect multicollinearity but doesn't fully resolve it. Elastic net effectively manages correlated predictors for a stable, generalizable model.
3. **Overfitting Control:**
    - Simultaneous L1 & L2 penalties in elastic net help control overfitting better than Lasso or Ridge, crucial with numerous predictors.
4. **Flexible Coefficient Shrinkage:**
    - In scenarios with mild correlation but not strong multicollinearity, elastic net balances variable selection and shrinkage effectively.
5. **Improved Prediction Performance:**
    - Even in low multicollinearity scenarios, elastic net regularization can enhance prediction by mitigating overfitting risks.

Contextual Justification:

- **Essential Correlated Variables:**
    - Crucial variables, capturing 40-50% of variance, are interconnected and pivotal to the model's integrity.

- **Refinement Over Removal:**
    - Dropping these key variables compromises the model's representativeness, needing nuanced variable selection.
- **Elastic Net's Role:**
    - Offers both variable selection and regularization despite low multicollinearity, vital for managing correlated predictors.
- **Balanced Approach:**
    - Elastic net balances selection and regularization to support key variable significance, minimizing overfitting risks while ensuring model integrity.
- **Strategic Utilization:**
    - Tailored approach keeps model robustness, emphasizing the importance of correlated variables without compromising predictive performance.

The decision to use Elastic Net despite low multicollinearity underscores the strategic need to manage key variables while preserving model integrity and predictive efficacy.

To implement an Elastic Net regression model using the **glmnet** package in R for predicting the 'DataCards' variable based on a set of features in a dataset. Here's a breakdown of the code:

1. **Data Preparation:**
    - **y <- training_data_country$DataCards**: Defines the response variable 'DataCards.'
    - **X <- subset(training_data_country, select = -c(Event))**: Creates a matrix 'X' having predictor variables, excluding the 'Event' column.
2. **Training and Testing Sets:**
    - **train_data <- training_data_country** and **test_data <- testing_data_country**: Divides the dataset into training and testing subsets.
3. **Creating Matrices:**
    - **X_train** and **y_train**: Matrices for training data, separating predictors and the response variable.
    - **X_test** and **y_test**: Matrices for testing data, separating predictors and the response variable.
4. **Standardizing Predictors:**
    - **X_train_std <- scale(X_train)** and **X_test_std <- scale(X_test)**: Standardizes the predictor variables

separately for training and testing datasets. This step ensures that all variables have a mean of zero and standard deviation of one.

5. **Fitting the Elastic Net Model:**
    - **enet_model_country <- cv.glmnet(X_train_std, y_train, alpha = 0.5)**: Fits the Elastic Net model using cross-validation (**cv.glmnet**) on the standardized training data. The **alpha = 0.5** parameter signifies a 50-50 combination of Lasso (L1) and Ridge (L2) penalties, indicative of the Elastic Net approach.
6. **Making Predictions:**
    - **y_pred <- predict(enet_model_country, newx = X_test_std)**: Generates predictions for the testing dataset using the trained Elastic Net model.
7. **Evaluating the Model:**
    - **mse <- mean((y_pred - y_test)^2)**: Calculates the Mean Squared Error (MSE) between the predicted 'DataCards' values and the actual values in the testing dataset.
    - **rsquared <- cor(y_pred, y_test)^2**: Computes the R-squared value, being the proportion of variance in the 'DataCards' variable that the model explains.
8. **Printing Evaluation Metrics:**
    - **print(paste("Mean Squared Error:", mse))**: Displays the computed Mean Squared Error.
    - **print(paste("R-squared:", rsquared))**: Displays the R-squared value as an assessment of the model's goodness of fit.

We visualizations to serve different purposes:

- The first plot (**plot(enet_model_country)**) helps in understanding the behavior of coefficients or the penalty profiles of the model across different levels of regularization. This is particularly helpful in Elastic Net models, which use a combination of Lasso and Ridge penalties.

- The second plot (**plot(y_test, y_pred, ...**) is a scatterplot that visually compares the model's predictions against the actual values. The diagonal red line (**abline (0, 1, col = "red")**) shows the ideal scenario where predictions perfectly match the actual values. Deviations from this line suggest how well the model performs – the closer the points are to the red line, the better the predictions align with the actual values.

Model visualization

- **Data Preparation**:
    - Calculates percentages for each factor relative to the total count, generating the **continent_perc** dataset.
    - Reshapes the data using **pivot_longer()** to create a clearer representation (**continent_long**) of factors and their respective percentages.
- **Threshold Establishment**:
    - Sets a threshold at 50% of the total to filter significant events, representing percentages higher than the mean.
- **Filtering and Visualization**:
    - Derives **filtered_continent_long** by grouping events, summing their percentages, and filtering for events surpassing the established threshold.
    - Merges **filtered_continent_long** with the initial dataset to retain only significant events, preparing a refined dataset for visualization.
- **Visualization**:
    - Utilizes **ggplot()** to generate a polar bar chart illustrating the distribution of significant events across different factors within each continent.
    - Facets in the chart provide a clear comparison of event distributions across factors, offering insights into the proportional significance of events within continents.
- **Outcome**:
    - This analysis visually represents the distribution of events across factors on a continental scale, providing insights into the prevalence and impact of various events within each continent.

Classification

Classification 1

The analysis undertook a comprehensive exploration of disaster incident data from multiple countries (Countries)to derive insights into severity and temporal categorization, impact classification, clustering analysis, visualization and cross-country analysis, and disaster risk classification based on DataCards.

The amalgamated data from various nations was juxtaposed to find patterns and risks associated with different factors influencing disasters.

Severity and Temporal Categorization:

*Insights***:** The severity of incidents was bifurcated into "High Severity" and "Low Severity" based on the mean of 'Event_Severity' column. Additionally, temporal categorization was executed through quartile divisions of the 'Year' column, delineating periods as "Early Period," "Mid Period," and "Recent Period."

*Purpose***:** Understanding severity and temporal patterns enriches the interpretation of model outcomes, providing context to predictions in different time frames and incident intensities.

Impact Classification Clustering Analysis:

*Impact Classification***:** The combined impact of incidents, including Deaths, Houses Ruined, Affected, and Monetary Loss in USD, is classified into quartiles as "Minimal Impact," "Moderate Impact," and "Severe Impact". This classification elucidates the gravity of incidents across various countries.

*Clustering Analysis(cluster_id)***:** Utilizing K-means clustering, incidents are categorized into four clusters: "Least DataCards," "Moderately Low DataCards," "Moderately High DataCards," and "Most DataCards". This stratification aids in understanding incident occurrences concerning the volume of DataCards.

*Purpose***:** These classifications facilitate a deeper understanding of incident gravity and occurrence patterns, adding another layer to the interpretation of the Elastic Net model.

Visualization and Cross-Country Analysis:

*Visual Representations***:** Stacked pie charts and heatmaps visually represent relationships between impact, severity, temporal factors, and clustered incident data across multiple countries.

*Insights***:** These visualizations offer a holistic view of incident trends and risk profiles across nations, aiding in the identification of patterns and anomalies.

Disaster Risk Classification:

*Function Utilization***:** A bespoke function classifies disaster risks based on severity, impact, and cluster characteristics, yielding risk levels such as "Extremely High Risk" = "red", "Very High Risk" = "orange", "High-Moderate Risk" = "yellow", "High-Low Risk" = "yellowgreen", "Moderate-High Risk" = "green", "Moderate Risk" = "lightblue", "Low Risk" = "blue", "High Risk" = "purple", "Clustered Risk" = "skyblue", "Moderately High DataCards Risk" = "pink", "Moderately Less DataCards Risk" = "violet", "Least

DataCards Risk" = "grey", and "Undefined Risk" = "white"

*Purpose***:** This classification further refines our understanding of incident risks, allowing for nuanced interpretation and actionable insights.

In conclusion, our model visualization and comprehensive analysis extend beyond traditional metrics, providing a multifaceted understanding of the Elastic Net model's performance and its implications in real-world scenarios. The incorporation of severity, impact, clustering, and cross-country analysis enriches the interpretability of our predictive model, fostering informed decision-making in disaster incident analysis.

Classification 2

In our exploration of disaster incident data, we utilized the Clara function to conduct clustering analysis, specifically focusing on a subset of variables: 'ClusterId,' 'Deaths,' 'Monetary.LossUSD,' This analysis aims to uncover underlying patterns and relationships within the data, providing valuable insights into distinct groups of incidents.

Data Preparation:

We start by selecting pertinent variables for clustering, including the temporal dimension ("ClusterId") and key impact indicators ('Deaths', 'Monetary.LossUSD'). Standardizing the data with the scale function ensures that each variable contributes equally to the clustering process, mitigating issues arising from different scales.

Clustering Technique: Clara Method

With the data appropriately prepared, we apply the Clara function to perform clustering. The k parameter is set to 4, indicating our intention to identify five distinct clusters within the data. The samples parameter influences the number of samples considered during the clustering process.

Visualizing Cluster Characteristics:

The pairs plot provides a visual representation of the relationships between different variables within each cluster. Patterns, trends, and potential correlations become apparent through the scatterplots. The legend facilitates the identification of distinct clusters within the plot.

This clustering approach helps us categorize incidents into five distinct groups, enabling a nuanced understanding of how these incidents relate to each other based on the selected variables. Further analyses and interpretations can be drawn based on the specific characteristics of each cluster, aiding in targeted decision-making and response strategies.

Key Observations:

Pattern Exploration: The pairs plot enables the exploration of patterns and relationships between impact indicators and the temporal dimension within each cluster.

Distinctive Clusters: Unique colors represent distinct clusters, signifying groups of incidents that share similar characteristics.

Visual Correlation: The arrangement of points in the pairs plot illustrates potential correlations or differences within and between clusters, contributing to a comprehensive understanding of incident dynamics.

This cluster analysis serves as a crucial step in unraveling the complexities of disaster incidents, facilitating subsequent in-depth investigations into the factors influencing each identified cluster.

Conclusions

1. **Varied Model Performance:**
   Continental Disparities: The performance of regression models varied significantly across continents, indicating diverse influencing factors in predicting deaths For instance, Asia showed different predictive patterns compared to Africa and Americas.
2. **Clustering Insights:**
   Temporal Shifts: in recent period : Asia disaster level has moved up and Africa disaster level has moved down as America has remained at the same level of disaster risk level.
   Frequency Disparities: As the frequency (DataCards) increases the Death and monetary loss is decreased suggesting that the low frequency of event has lower death. Monetary.LossUSD dependency for DataCards cluster is better distributed than death.
3. **Long-term Implications:**
   Policy Implications: These insights can inform disaster management policies.
4. **Future Projections:**
   Forecasting Possibilities: As trends are evident, we can project potential future risk levels based on historical patterns. This can aid in proactive disaster planning and resource allocation.

By delving into temporal trends, regional disparities, the influence of severity and impact, and aligning clustering insights with expected risk levels, you can derive comprehensive insights from the heatmap. These insights will be crucial for informed decision-making and strategizing disaster management approaches.

References

Title: "Data Mining Techniques for Analysis of Disaster Data"
Authors: R. Suthar, A. Singh
Published in: International Journal of Computer Science and Information Security
Year: 2015

Title: "Data Mining Techniques in Natural Disaster Analysis"
Authors: N. Jain, P. Gupta
Published in: International Journal of Computer Science and Mobile Computing
Year: 2014

Title: "Application of Data Mining Techniques in Disaster Prediction and Management"
Authors: S. Roy, S. Das, S. Bhattacharyya
Published in: Procedia Technology
Year: 2014

Title: "Data Mining Techniques in Disaster Data Management: A Review"
Authors: S. Selvi, K. Kavitha
Published in: International Journal of Computer Applications
Year: 2017

Title: "Data Mining for Disaster Management"
Authors: V. Kumar, A. Srivastava, S. Kumar
Published in: International Journal of Computer Applications
Year: 2016

Youtube: https://statquest.org/video-index/
Github : https://github.com/ishitaxtripathi/DisasterAnalysis