Based on the World Risk Index, calculated by the United Nations University Institute for Environment and Human Security (UNU-EHS) and included in the 2016 World Risk Report (WRR 2016) published by Alliance Development Works/Bündnis Entwicklung Hilft (BEH). The World Risk Index developed by UNU-EHS and BEH determines how likely 173 countries are to suffer a disaster as a result of natural hazards such as earthquakes, volcanic eruptions, storms, floods, droughts, and sea level rise. Our source files have risk score of 173 countries for the year 2011 to 2016. We intent to find a risk category of each country and group the countries into different cluster like low risk, medium risk and high risk.. based on vulnerability of each country using Kmeans clustering algorithm. Finally load data into azure sql and make some interactive visualizations.

<https://en.wikipedia.org/wiki/List_of_countries_by_natural_disaster_risk>

List of natural disasters by death toll:

A web data set collected from the following path includes death rates by disaster event for every country between 1992 and 2020. This data set covers all the disaster events that occurred during that time period. In this data set, our focus is to scrape data from the web, clean them, store them in Azure SQL, and perform some visualizations to show the trends, casualty by country and event, and statistics of each event.

<https://en.wikipedia.org/wiki/List_of_natural_disasters_by_death_toll>

ISO country list:

The purpose of this data set is to add more details, such as country code, country 3 digit alpha code, and latitude and longitude, to the existing data sets that we scraped from the internet. The file is in CSV format and is located in the following path.

Data Selection and Preprocessing:

Data selection and preprocessing is the very important step in every datamining problem. It includes data cleaning and feature selection process. The programming language that we have used at this phase is python and its libraries such as pandas, numpy, pyodbc, sklearn, matplotlib and plotly.

World risk and death rate cases:

The data that we obtained from the various sources are like unstructured and semi structure formats. So we think that MongoDB is the right database to stage our unstructured and semi structure data, then we apply following data cleansing and pre-processing steps to clean our data to be loaded to the respective database. finally, we have loaded our pre-processed data into Azure sql in cloud environment. Since it’s a team project and the volume of the data is also high, we have chosen Azure SQL database is to store our data.

The web data can be scraped using beautiful soup library from python. The data we got from the web is semi structured data.

We loaded the semi structure data into MongoDB.

Created a dataframe from the data that we retrieved from MongoDB for pre-processing. The dataframe contains the details about vulnerable countries with its risk scores by risk category.

Identified and remove unwanted rows and columns from the dataframe and renamed every column into proper naming convention.

Removed unwanted characters like ‘%’ from risk\_score attributes and converted them into float data time as machine leaning model will accept only numerical values.

Derived a column Risk\_Score from previous risk score columns.

We formatted our data to satisfy the first normal form. And finally, joined with country data set to get few attributes like Country\_Code, longitude, and latitude.

Death rates:

We carries out the same steps as above from step 1 to step 3, additionally, we applied standard country names and event types to all the countries. This dataframe contains information about death rates by disaster wise for the entire world.

Basically, this dataset is not complied with first normal form, hence we applied all the required transformation logic to make our data to satisfy the first normal form.

To plot the world map using plotly, the data file should contain the information of country code, latitude, and longitude of each country, hence we joined actual dataframe with country dataframe. To assign risk category for each country, we had to join with risk\_country dataframe.

Finally, pre-processed data will be loaded into Azure Sql server table.

**2.2 Database Management**

**Azure SQL Server:**