System Design Specification document

Data sources

The data used was structured environmental data that was collected by several Federal Government agencies in the U.S. Some of these agencies include the **Center for Disease Control and Prevention** and the **National Oceanic and Atmospheric Administration**

Data loading

The data was made available in 3 *comma-separated values* (csv) files. The first file contained all the training data that was used to learn from the acquired data. This file is what enabled us to establish relationships and also develop a better understanding of our data. The second file contained the number of dengue fever cases recorded in the time this data was collected and finally, the last file contained the testing data.

We decided to use Pandas(an open-source library for the Python programming language that is optimized for data manipulation) because of its many features and ability to work with numerous file types. The data to be analyzed was in CSV format and pandas has a `pandas.read_csv()` method that reads these types of files into a pandas DataFrame, which is a data structure in a table-like form.

Data wrangling

Data wrangling involves cleaning up raw data to make it useable in analytical algorithms. The data wrangling approach we took on involved three major steps namely, Data exploration, Dealing with missing values, and Reshaping the data.

In data exploration, we identified columns and the data within each of these columns, this was made possible by the use of the pandas.DataFrame.describe() method which generates descriptive statistics about a DataFrame like central tendency, dispersion, missing values, and dataset distribution.

The datasets used for the analysis had a lot of missing values with `ndvi_ne` column missing up to 158 values. 20 other columns also missed values but at reduced severities. The approach used for dealing with missing values was Imputation which involves the replacement of missing data with substituted values. This process was made possible with the use of the scikit-learn

python library. The method used was `sklearn.impute.SimpleImputer()` which is an imputation transformer for completing missing values. The strategy used for replacing the missing numeric values was `mean` which calculates the mean along each column and then replaces each missing value along that column with the mean.

In reshaping of the data, we carried out label encoding for the `city` column once again using another method from scikit learn. Label encoding is a process of conversion of categorical text values into numerical data. The method used was `sklearn.preprocessing.LabelEncoder()` which encodes labels with values between `0` and `n_classes(the number of classes) - 1`.

Data Visualization

In data visualization, we attempt to get insights about the data by representing data in a graphical format. Several visual elements such as charts, tables, graphs, and maps can be used for this task.

We have not yet done visualization so…

Here, we shall talk about the skewdness of the data, nature of the data, and explain why we use any type of plot we use.