Dengue Outbreak Prediction Model

Charushree MB

Team : Sri Surya Teja Navuluri , Yash Pawar

Business framing: "*DataInsight Labs* is a trusted technology partner on a mission to deliver AI and ML services that help businesses and organizations to personalize customer experience, automate and optimize processes, and leverage accurate predictions and forecasts. Our key expertise are AI & Big Data, Data Science, Predictive Analytics, Machine Learning & Deep Learning. *DataInsights Labs* provides AI consulting and development services to scale up every client's business. What started as a startup by a small group of passionate engineers and business managers is now a full-scale company and has clients from more than 20 domains.
Founded In: 2010
Employees: 500+ employees
Location: Wisconsin, US
Clients: United Healthcare, IBM, CVS, World Health Organization (WHO)

Problem Statement: The World Health Organization (WHO) has conducted "*Global strategy for dengue prevention and control*" campaign in 2012 which aimed to reduce the incidence and impact of dengue through a combination of measures; improving surveillance and monitoring systems, predicting disease spread, and increasing public awareness and community engagement. *DataInsight Labs* partnered with WHO as a part of the campaign to build a Machine learning model to predict the disease spread in two cities; San Juan(sj),Puerto Rico and Iquitos(iq),Peru. If there is a rise in cases, they can start campaign for prevention. We have partnered with National Centers for Environmental Information (NCEI), National Center for Atmospheric Research (NCAR) to provide us with the daily climate data, satellite precipitation measurements, climate forecast system reanalysis and satellite vegetation - normalized difference vegetation index (NDVI) for both the cities. These organizations are ready to provide the same in the future if we can help the cause. The main goal of this project is to build a model using the *train set* to predict [total dengue cases] for each [city, year ,week of year] in the *test set*. The key performance metric is *Mean Absolute Error (MAE)* and the objective is to build a reliable prediction model with minimizing the Mean Absolute Error.

1. Dataset

The Dengue train set contains 24 features that capture city name, Year, Week of the year and various components of climate such as temperature, vegetation index, precipitation measurements for 1456 observations and a target variable, "*total_cases*" which indicates the total number of dengue cases. Test set contains all 24 features and 416 observations. We are running our model to predict "*total_cases"* on the test set.

1.1 Data Description:

Since we are predicting the occurrence of Dengue based on climatological data, it is of paramount importance to understand climatic conditions of the two cities. San Juan has a tropical climate, with average temperatures ranging from the mid-70s Fahrenheit in the winter to the mid-80s Fahrenheit in the summer. The country experiences high humidity and receives a lot of rainfall, especially during the rainy season which runs from April to November. The climate in Iquitos is tropical, with high temperatures and

humidity throughout the year. Average temperatures range from around 77 degrees Fahrenheit during the cooler months to around 86 degrees Fahrenheit during the hottest months. The city experiences a rainy season from November to April, with heavy rainfall. We have a small data set with 936 observations for sj and 520 observations for iq. We have the data of San Juan from week 18 ,1990 to week 17, 2008 and of Iquitos from week 26, 2000 to week 25, 2010. Many of the temperature data are fairly correlated as expected. But we do not see any strong correlation between any of the features and toal_cases as shown in the correlation Heatmap.

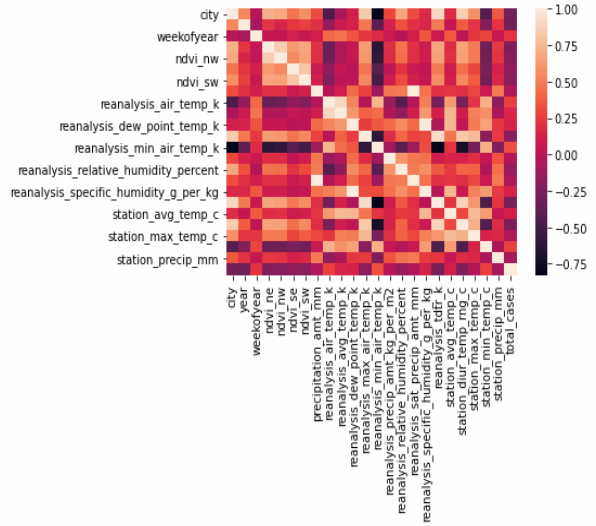

Figure 1.1.1 – Correlation Heatmap

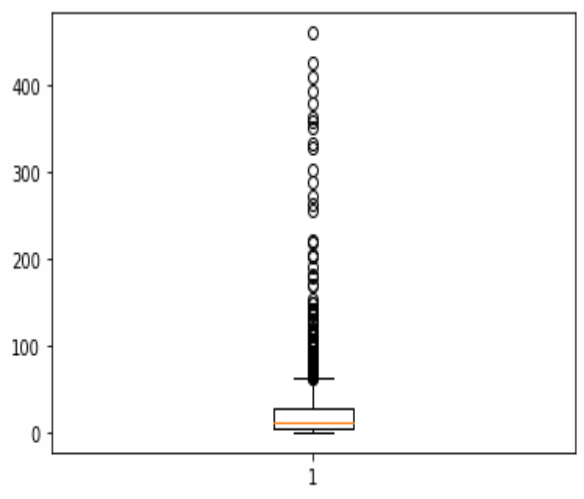

Figure 1.1.2- Boxplot of total_cases

Average Dengue cases are around 24/week, ranging between 0 cases/week to maximum of 461/week. We see a significant number of outliers with large values in total cases. These are important data for WHO as they likely predict the dengue outbreaks. For this reason, Mean Squared Error (MSE) would have been the better performance metric. However, we are continuing with the requirements of WHO.

1.2 Data Cleaning and Data Transformation:

Before proceeding with building the model, it is important to clean our data and perform necessary transformations to ensure better predictions. For this purpose, we have dropped the feature '*week_start_date'* and assigned binary values to city names: '0' to sj and '1' to iq. Since this is a time-series data, we cannot simply eliminate the null values in our data. One simple way to fill these null values is use latest data to update these null values. Since we can't build a model without those values, we'll take a simple approach and just fill those values with the most recent value that we saw up to that point. In addition, we have engineered *'month'* feature based on week of the year. Figure 1.2 is the graph of total cases plotted along years for both the cities. We can clearly see that the peaks and crest of total cases does not align for both the cities. Both the cities are different hemispheres and distances from the equator. There are multiple seasonal factors affecting the total cases in both the cities. To better the model, we have transformed the months for Iquitos by 4 months. This will help the peaks and crests to sync up between the cities.

In addition, we have created a new feature *'Veg'*, which is the average of all the four vegetation indices. This new feature will be higher if there is a lot of vegetation. Also, all the temperatures given to us are not in the same units. Here we chose to convert all the temperatures to Kelvin. There are different units for different input variables. To reduce the importance of conversion, we need to scale the features for our model. This will ensure that no single feature is given more importance than others.

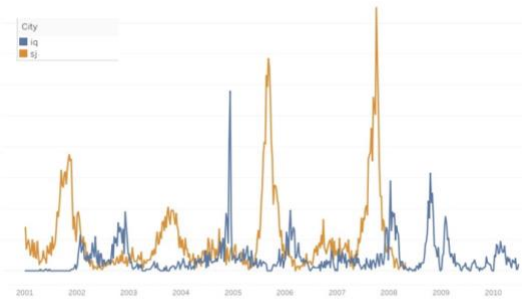

Figure 1.2- Timeseries of *total_cases*

2. Implementation

The dataset is split into train and test for the training and testing the model respectively. We used the popular *statsmodel* [1] and *sklearn*[2] libraries as they provide powerful functions that facilitate easy implementation of various models. In addition, we have used *seaborn* [3] and *matplotlib* [4] libraries for visualization purposes.

2.1 Generalized Linear Models

Generalized Linear Regression models (GLMs) allow for flexible distributions of the outcomes (including categorical outcomes, counts, or skewed/heavy-tailed data), where the parameters of the distribution theta are "linked" to a linear combination of the feature vector.

$$x_i = (1, x_{i1}, \ldots, x_{ip}), \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} = x_i \beta:$$

$$Y_i \sim \text{SomeDistribution}(\theta_i), \text{SomeLinkFunction}(\theta_i) = x_i \beta$$

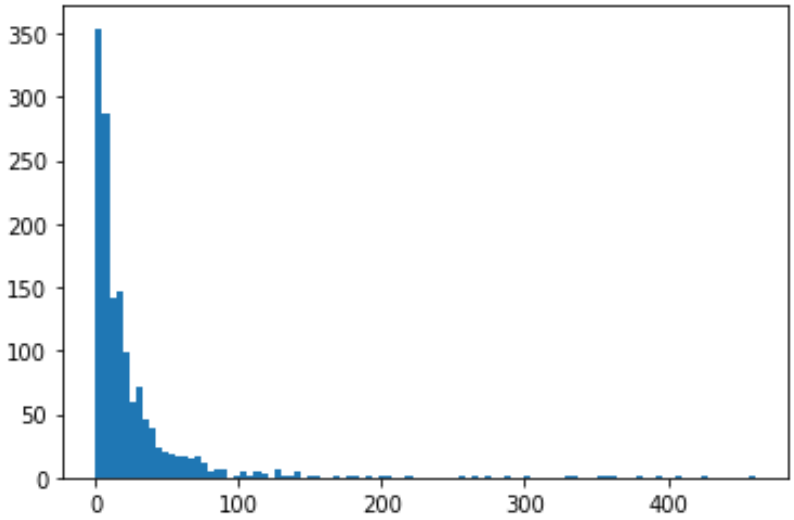

Figure 2.1- Distribution of total_cases

2.1.1 Gamma Distribution Model

The gamma distribution is appropriate to use in situations where the data are positive, continuous, and skewed to the right. The *total_cases* distribution looks skewed and heavily tailed data (Figure 2.1). Rather than assuming a normal model, we are assuming *total_ cases* is gamma distribution.

Mean Absolute Error (MAE): 19.387

Mean Absolute Error (MAE) with citywise split: 28.967

2.1.2 Negative Binomial Distribution Model

The negative binomial distribution is appropriate to use in situations where the data are count data (i.e., non-negative integers) as in our case. In addition, Variance > Mean suggests that we can assume the distribution to be Negative Binomial Distribution.

Mean Absolute Error (MAE): 16.979

Mean Absolute Error (MAE) with citywise split: 18.289

The liner models fail to effectively capture the deep interactions between the features. The climatological features in our dataset are deeply interconnected and the interactions between them are essential for better predictions.

2.2 Bootstrap Aggregating (Bagging) and Boosting

The fundamental concept behind ensemble by aggregation is a simple but powerful one- *the wisdom of crowds*: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. Bagging and Boosting are two examples of such ensembles.

2.2.1 Random Forest Model

Bootstrap Aggregating creates multiple bootstrap samples out of the original dataset so that each new bootstrap sample will act as another (almost) independent dataset. Then, we fit a model for each of these samples and finally aggregate by taking the average of their outputs. Random Forest is an extension of Bagging that also randomly selects subsets of features used in each data sample.

For the single dataset, we can clearly see that year, vegetation, week of the year, temperature have the highest feature importance. This is expected as number of cases are increasing with year and vegetation is an important factor for the spread of dengue.

Mean Absolute Error (MAE): 11.661

Mean Absolute Error (MAE) with citywise split: 12.823

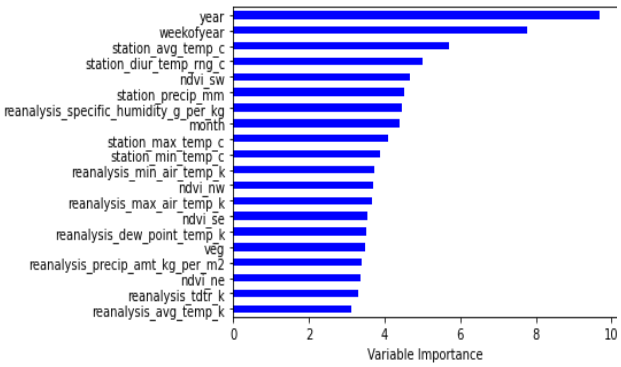

Figure 4.2.1 – Variable Importance

2.2.2 Gradient Boosting Model

Boosting is a technique that consists in fitting models sequentially, in an adaptive way: each model in the sequence is fitted giving more importance to observations in the dataset that were *badly handled* by the previous models in the sequence. Intuitively, each new model focuses its effort on residuals of the last model. Gradient boosting uses a gradient descent algorithm to minimize loss when adding models to the ensemble. Subsequent models are fit to *pseudo-residuals* instead of adjusting weights.

For the single dataset, we get similar important features as we got in the bagging. In bagging, *veg* is in the top five but in boosting it has very less significance. For the single dataset, we get similar important features as we got in the bagging. In bagging, veg is in top five but in boosting it has very less significance.

Mean Absolute Error (MAE): 11.208

Mean Absolute Error (MAE) with citywise split: 12.281

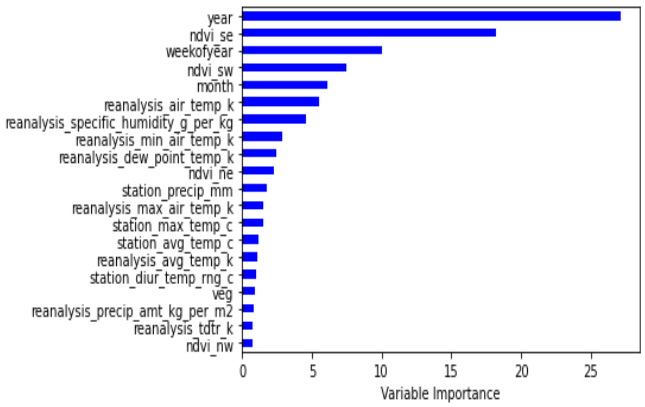

Figure 4.2.2- Variable Importance

3. Conclusion:

GLMs (Gamma and Negative binomial model) are linear models, and they fail to capture the deep interaction between the features. The linear models fail to effectively capture the deep interactions between the features. The climatological features in our dataset are deeply interconnected and the interactions between them are of paramount importance to make better predictions. This is the main reason why we see tremendous improvements when we use ensemble learners: Bagging and Boosting . *Random forests* arrive at their predictions by fitting trees to bootstrap replications of the dataset, just as in bagging. However, they additionally sample from the set of features to reduce the correlation of the predictors, thus it finesses the correlation problem of bagging while keeping the aggregation benefit. They are unreasonably effective, and it is not very difficult to fit them, they require no or minimal tuning, and they often produce very accurate predictions. In Gradient Boosting, boosting works successively. We start by fitting a model to the outcomes, but then the next model is fitted to the residuals from the first step (i.e., the part the first model couldn't explain). The total predictions are then the sum of the first plus the second predictions. In our analysis, Gradient boosting does improve MAE compared to bagging, but only slightly. This might also depend on the dataset, how the trees are considering the outliers. However, it is worth noticing that all the models perform better for single dataset compared to citywise split of dataset, further emphasizing the importance of deep interactions between the features. Thus, I would consider Random Forest and Gradient Boosting models to be superior to other benchmark models for this experiment.

References:

[1] "statsmodel," https://www.statsmodels.org/stable/index.html, online; accessed 20th December,2022.

[2] "sklearn library" https://scikit-learn.org/stable/modules/model_evaluation.html online; accessed 20th December 2022.

[3] "seaborn," https://seaborn.pydata.org/index.html ,online; accessed 20th December, 2022.

[4] "matplotlib" https://matplotlib.org/stable/gallery/style_sheets/style_sheets_reference.html , online;accessed 20th December, 2022