Capstone Proposal

By Hugo Lemos

Domain Background

Dengue fever is a mosquito-borne disease that occurs in tropical and sub-tropical parts of the world. In mild cases, symptoms are similar to the flu: fever, rash, and muscle and joint pain. In severe cases, dengue fever can cause severe bleeding, low blood pressure, and even death. Because it is carried by mosquitoes, the transmission dynamics of dengue are related to climate variables such as temperature and precipitation. Although the relationship to climate is complex, a growing number of scientists argue that climate change is likely to produce distributional shifts that will have significant public health implications worldwide.

An understanding of the relationship between climate and dengue dynamics can improve research initiatives and resource allocation to help fight life-threatening pandemics.

Problem Statement

Given the data of two cities, San Juan and Iquitos, each city spanning 5 and 3 years respectively, the objective is to learn from the data and predict the number of dengue cases each week, in each location, based on environmental variables describing changes in temperature, precipitation, vegetation, and more.

Datasets and Inputs

The Training data (features + labels) is provided by the drivendata competition and each entry in the features file has the following data:

City and date indicators:

- `city` – City abbreviations: `sj` for San Juan and `iq` for Iquitos
- `week_start_date` – Date given in yyyy-mm-dd format

NOAA's GHCN daily climate data weather station measurements:

- `station_max_temp_c` – Maximum temperature
- `station_min_temp_c` – Minimum temperature
- `station_avg_temp_c` – Average temperature
- `station_precip_mm` – Total precipitation
- `station_diur_temp_rng_c` – Diurnal temperature range

PERSIANN satellite precipitation measurements (0.25x0.25 degree scale):

- `precipitation_amt_mm` – Total precipitation

NOAA's NCEP Climate Forecast System Reanalysis measurements (0.5x0.5 degree scale):

- `reanalysis_sat_precip_amt_mm` – Total precipitation
- `reanalysis_dew_point_temp_k` – Mean dew point temperature

- reanalysis_air_temp_k – Mean air temperature
- reanalysis_relative_humidity_percent – Mean relative humidity
- reanalysis_specific_humidity_g_per_kg – Mean specific humidity
- reanalysis_precip_amt_kg_per_m2 – Total precipitation
- reanalysis_max_air_temp_k – Maximum air temperature
- reanalysis_min_air_temp_k – Minimum air temperature
- reanalysis_avg_temp_k – Average air temperature
- reanalysis_tdtr_k – Diurnal temperature range

Satellite vegetation - Normalized difference vegetation index (NDVI) - NOAA's CDR Normalized Difference Vegetation Index(0.5x0.5 degree scale) measurements

- ndvi_se – Pixel southeast of city centroid
- ndvi_sw – Pixel southwest of city centroid
- ndvi_ne – Pixel northeast of city centroid
- ndvi_nw – Pixel northwest of city centroid

Solution Statement & Evaluation Metrics

The solution will be to train a regression model through supervised learning. This model will analyse the Time series data provided for both cities, learning to extrapolate the number of cases for future dates - i.e. to predict the *total_cases* per (Year, Week, city).

For testing the model the competition host provides a test set, which is a pure future hold-out, meaning the test data are sequential and non-overlapping with any of the training data. Then we submit our test labels that will be evaluated by the competition using as evaluation metric the Mean absolute error (MAE). The absolute error is calculated for each label in the submission and then averaged across the labels.

Benchmark Model

The competition already provided a benchmark model. The benchmark model hypothesizes that the spread of dengue may follow different patterns between the two cities, therefore the dataset was divided and trained two separate models for each city.

As for the regression model it was chosen the Negative Binomial Distribution, one reason to choose this was because the variance of the labels values is greater than the mean of the labels, and as for the data it was pre-processed in order to fill the NaN values with the previous values and selected the four features that shown more correlation with the labels.

Finally as for training as a timeseries model , it was used a strict-future holdout set when splitting the training set and the test set, by keeping around three quarters of the original data for training and the rest to test. After trained, the model was use to predict the total cases for a test set provided by the competition and the results submitted. The submission score was 25.8173.

Project Design

The project work order will be the following:

1. First, understand the data.
    - Run a statistical description of the data, this means gets for each feature, the mean, the quartiles, the standard deviation, and the minimum and maximum values.

2. Analyse feature relevance
    - Through the use of Scatter Matrices and correlation heatmaps I am going to check the correlation between features and check the correlation between features and labels. The objective is to discard irrelevant (or even not less relevant) features.
3. Data Preprocessing:
    - Fill NaN values with the most recent value
    - Feature scaling using Normalisation, ensuring that each feature is treated equally when applying supervised learners
    - Outlier detection using the Turkey method.
    - Principal component analysis (PCA) to reduce dimensionality if it makes sense.
    - Transform data to be passed as an input to a time series algorithm. E.g. to use the sliding window method it's necessary to prepare the input over a time span of *n* time occurrences, example use the previous week and the current week as an Input.
4. Training models
    - Few possibilities on this, either in a perspective of what model will be used either how a training set and test set will be managed.
    - Regarding the models, I am planning to try Random Decision Trees, the Ensemble AdaBoost and Long Short-Term Memory (LSTM).
    - About how the training set and test set will be managed, one option that I'll try is to work on the size of the time window used to generate the training data, e.g. to predict the total_cases considering data of the last *n* weeks (several values for n will be tried). Other technique that will be tried is the Walk Forward Validation.
5. Finally: Train, Test, Analyse, Rethink, Rebuild, Train, Text …
    - Repeat this cycle as many times as it's productive to do so. During this iterative cycle new techniques could be tried, many lessons will be learned.