**Forecasting Disease Spread**

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. Using environmental data collected by various U.S. Federal Government agencies—from the Centers for Disease Control and Prevention to the National Oceanic and Atmospheric Administration in the U.S. Department of Commerce, this project is trying to predict the number of dengue fever cases reported each week in San Juan, Puerto Rico and Iquitos, Peru.

**Explore Data Analysis**

The training dataset is a relatively small dataset containing only around 1400 rows. We have initially 24 features as follows:

|  |  |
| --- | --- |
| Column Name | Type |
| city | String |
| year | Integer |
| week\_start\_date | Timestamp |
| station\_max\_temp\_c | Float |
| station\_avg\_temp\_c | Float |
| station\_precip\_mm | Float |
| station\_min\_temp\_c | Float |
| station\_diur\_temp\_rng\_c | Float |
| precipitation\_amt\_mm | Float |
| reanalysis\_sat\_precip\_amt\_mm | Float |
| reanalysis\_dew\_point\_temp\_k | Float |
| reanalysis\_air\_temp\_k | Float |
| reanalysis\_relative\_humidity\_percent | Float |
| reanalysis\_specific\_humidity\_g\_per\_kg | Float |
| reanalysis\_precip\_amt\_kg\_per\_m2 | Float |
| reanalysis\_max\_air\_temp\_k | Float |
| reanalysis\_min\_air\_temp\_k | Float |
| reanalysis\_avg\_temp\_k | Float |
| reanalysis\_tdtr\_k | Float |
| ndvi\_ne | Float |
| ndvi\_nw | Float |
| ndvi\_se | Float |
| ndvi\_sw | Float |

More details about the meaning of features can be found at <https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/82/#features_list>

The first thing is to check how many missing values we have. The result is as follows: A graph with numbers and text

Description automatically generated

From the plot we can see that most features have very few missing values. Ndvi\_ne column has around 13% missing values which mainly sources from San Juan. We can remove this feature for further step of feature engineering.

The distribution of target value is right skewed as the following graph shows. This means that we can do a log transformation for the target value. If we plot the total cases with time goes, we can see that San Juan has a longer timeline compared with Iquitos (2:1 for the size of data).

A graph of a number of cases

Description automatically generated A graph showing a number of cases

Description automatically generated

Since we just have two locations, it is worth investigating the features based on the locations. First, we checked the distribution of features.

**San Juan:**

A graph of a graph

Description automatically generated with medium confidence

**Iquitos:**

A graph of a number of blue lines

Description automatically generated with medium confidence

As we observed, a few features are skewed for both regions as we observed, and we can use log transformation to transform the features. The decision is as follows:

|  |  |
| --- | --- |
| **Right-skewed features** | **Left-skewed features** |
| precipitation\_amt\_mm | reanalysis\_dew\_point\_temp\_k |
| reanalysis\_precip\_amt\_kg\_per\_m2 | reanalysis\_min\_air\_temp\_k |
| reanalysis\_sat\_precip\_amt\_mm | reanalysis\_relative\_humidity\_percent |
| station\_precip\_mm | station\_min\_temp\_c |

For right-skewed features, we still use log transformation while for left-skewed features, we use square transformation. One interesting finding is that precipitation features are always right skewed.

Some insights can also be found from correlation plots for two cities.

**San Juan:**

A screenshot of a computer screen

Description automatically generated

**Iquitos:**

A screenshot of a computer

Description automatically generated

A black text on a white background

Description automatically generated

Features have relatively weak correlation with total cases (strongest is just 32.5%). However, climate variables have much stronger correlation with each other such as reanalysis\_dew\_point\_temp\_k and reanalysis\_specific\_humidity\_g\_per\_kg. More details would be discussed in the part of feature engineer.

We also have many plots for distribution between features and target values. And all plots are in the EDA notebook, for conciseness, I will not show all but give some observations here.

A graph of different colored dots

Description automatically generated with medium confidence

A graph of a tree

Description automatically generated

* For each category of features, we can see that two cities have relatively different patterns for the distribution between features and target values. This means that we potentially can model two cities respectively.
* We can see few features have strong trend with time going so that the column of year is not necessary in the modelling.

**Feature Engineer & Selection**

Feature engineering and selection is always one of the most important parts before we put some real models for the data. EDA has already given us enough insights about the data and now we need to use some techniques to extra valuable information from the features.

**Feature transformation:**

|  |  |
| --- | --- |
| **Log Transformation** | **Square Transformation** |
| precipitation\_amt\_mm | reanalysis\_dew\_point\_temp\_k |
| reanalysis\_precip\_amt\_kg\_per\_m2 | reanalysis\_min\_air\_temp\_k |
| reanalysis\_sat\_precip\_amt\_mm | reanalysis\_relative\_humidity\_percent |
| station\_precip\_mm | station\_min\_temp\_c |

Also, the city columns should be encoded to 1 as San Juan and 0 as Iquitos.

**Feature Removal:**

|  |  |
| --- | --- |
| **Features** | **Reason** |
| reanalysis\_avg\_temp\_k | reanalysis\_avg\_temp\_k and reanalysis\_air\_temp\_k is literally similar in meaning and indeed they are highly correlated. |
| reanalysis\_sat\_precip\_amt\_mm | reanalysis\_sat\_precip\_amt\_mm is almost the same (values) as precipitation\_amt\_mm. |
| ndvi\_ne | 13% missing |
| year | Few features have strong trend as time goes |
| weekofyear | Not useful compared with month |
| week\_start\_date | Month is more useful for the exact date |

**Feature Generation:**

|  |  |  |
| --- | --- | --- |
| **Features** | **Methodology** | **Reason** |
| month | Extract month from week\_start\_date | Strong indicators |
| station\_precip\_mm\_rolling\_month\_mean | Rolling mean (4 weeks, around one month) of station\_precip\_mm | Precipitation might be more useful for a longer-period window |
| station\_precip\_mm\_diff | Difference between this week precipitation and last week’s | Difference of precipitation reflect the change of climate |
| reanalysis\_temp\_humid\_index | Interaction between reanalysis\_relative\_humidity\_percent and reanalysis\_tdtr\_k | Hot and wet weather trends to increasing total cases of dengue fever |
| ndvi\_max\_precip | Interaction between max(ndvi features) and precipitation\_amt\_mm | More vegetation combining with more precipitation may lead to increasing total cases of dengue fever |

**Model**

Since the size of dataset is relatively small, data is divided into 90% of training data and 10% for validation data and I prefer to focus on tree-based models. The baseline models would be linear regression and random forest. Without any hyperparameter tuning, linear regression has a MAE of 12.82 for validation dataset and random forest has a MAE of 13.

For more complicated tree-based models, I chose LightGBM and XGBoost. I applied time series split cross validation to reduce the variance of model performance due to the limited size of dataset. Instead of traditional K-fold cross validation, time series split cross validation also avoids the issue of data leakage.

A graph with blue and red bars

Description automatically generated

For the hyperparameter tuning, I rely on Optuna framework, which is state-of-the-art algorithms to choose the best parameters and easy to visualize the process of tuning. The final result is that both XGBoost and LightGBM outperforms the baseline model and XGBoost has the best MAE of 11.9.

**Feature Importance Analysis**

For the final part of the project, we should focus on how the model value all the features.

**LightGBM**

A graph of blue and pink colored lines

Description automatically generated with medium confidence

**XGBoost**

A graph of data on a white background

Description automatically generated

XGBoost did not select our generated features as most important features while LightGBM picked our new-generated feature temp\_humidity\_index (interaction between temperature and humidity) as one of the most important features.