DengAI: Predicting Disease Spread

Matthew Brotherton, Tyler Fulghum

Tennessee TECH

Introduction

Neglected tropical diseases, or NTD's, cause endemics in regions where health services and the standard of living are deficient. One such NTD, dengue fever, is of particular concern. To combat this debilitating disease, we predicted future dengue cases in Iquitos, Peru and San Juan, Puerto Rico, based on a prior decades' worth of data that includes case numbers on a weekly basis, as well as climate variables such as temperature, precipitation amount, humidity, and dew point, to name a few. From these data points, we extracted the most relevant variables, and then we extrapolated by creating predictive models from the supplied training and testing data.

Methodology

Multiple Linear Regression

Our goal was to perform a regression on the number of weekly dengue cases using the aforementioned variables. The scoring metric, mean absolute error with respect to the total_cases target, judged the goodness of fit; a low MAE was the primary objective. Each model had its respective hyperparameters, and these were extensively tuned via individual grid searches that tested the tens of thousands of possible parameter value combinations.

Data Preprocessing

Before we could start modeling, we needed each predictor to have as much useful information as possible. Unfortunately, many of our variables had missing data. In some instances, a variable was missing up to 13% of its possible records. On top of this, our predictive models would not accept NA's or NaN's. To remedy this, all relevant variable NA's were filled either via mean or forward fill, which propagates the last valid record. By doing this, we accepted that results will be skewed.
Because dengue has a reported incubation of 4-10 days, we shifted the total_cases target variable by both one and two weeks, which emulates a week-long incubation period, as well as potential transmission time between hosts.

Resources

DrivenData. (n.d.). DengAI: Predicting Disease Spread. Retrieved April 9, 2021, from https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/page/80/

https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/data/

https://community.drivendata.org/c/dengue-competition/18

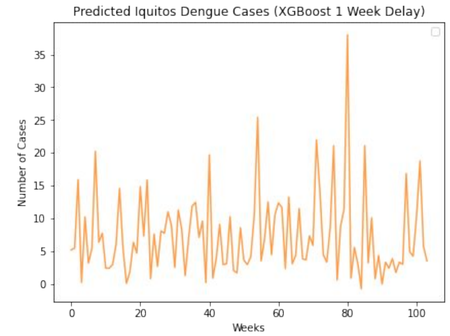
Regression Results

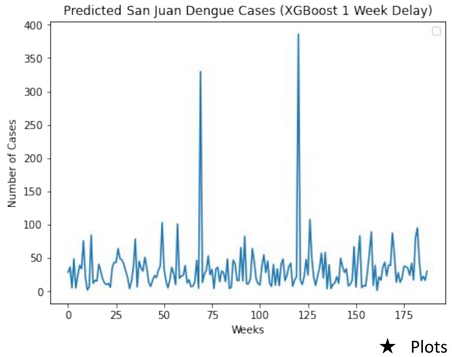
Random Forest

Iquitos MAE = 5.47

San Juan MAE = 18.11

Submission MAE = 29.61



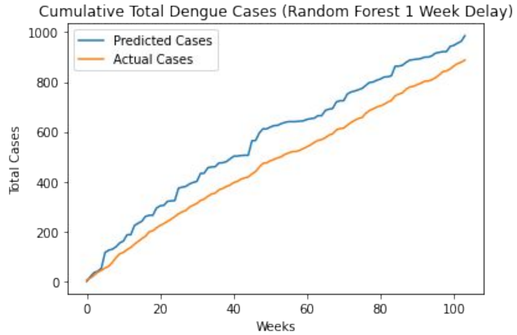


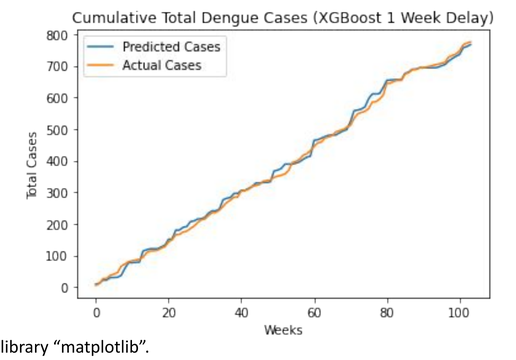
Extreme Gradient Boosting

Iquitos MAE = 5.49

San Juan MAE = 13.19

Submission MAE = 32.53





★ Plots generated by Python library "matplotlib".

Multi Layer Perceptron

Based on prior experience with neural networks, we were confident this model would generalize well, but obtaining the optimal parameters and architecture are both very involved processes.

Iquitos MAE = 6.81

San Juan MAE = 24.03

Submission MAE = 3713.21

★ We are unsure why the Sub MAE is so high; it will need more testing in the future.

Gradient Boosting

We had heard of this regression technique in a previous course, but we never got to implement it ourselves, so we felt we had to try it on this project. However, we underestimated the amount of tuning required to produce competent results.

Iquitos MAE = 5.19

San Juan MAE = 16.62

Submission MAE = 35.96

Regression Techniques

★ Each model used its respective grid searches' chosen hyperparameter values. The results were then predicted against the supplied test data, and an MAE score was calculated. The San Juan results were always substantially worse than Iquitos, as the former dataset is almost twice as large. For submission, the two prediction sets were merged, and then they were compared to DrivenData's withheld test data. This comparison yielded the final MAE score for the competition.

Random Forest

- Used Python library "sklearn.ensemble.RandomForestRegressor" to instantiate a Random Forest model for each city.
- Our first ensemble technique, and it provided the best results with minimal effort in tuning the model.

Extreme Gradient Boosting

- Used "xgboost" to instantiate an XGBoost model for each city.
- The hyperparameter search space is very large. Finding the best model settings took magnitudes longer than the Random Forest search. This was our last modeling technique attempted.

Multi Layer Perceptron

- Used "sklearn.neural_network.MLPRegressor" to instantiate an MLP model for each city.
- Difficult to assure the network architecture and the selected hyperparameter values were optimal. There are many choices that are very time consuming to randomly search and test for viability.

Gradient Boosting

- Used "sklearn.ensemble.GradientBoostingRegressor" to instantiate a Gradient Boosting model for each city.
- Our second ensemble technique applied, and it provided decent results with extensive tweaking. Obtaining an applicable model took many hours of trial and error.

Conclusion

All of our techniques were successful in generating a regression model. However, we found that the Multi Layer Perceptron was the least effective of the four. Perhaps with more time spent on honing the network architecture, this method could produce much better results. In general, our models have a hard time predicting outbreaks in particular. Nevertheless, the Random Forest and XGBoost models produced the lowest MAE scores. Lastly, the strict submission format requirements made submitting and receiving feedback on a model much more complicated than necessary.

Future Improvements

Under the data constraints set by the competition, we believe we could decrease our MAE score by having more time to work on the project. The official competition end date is Nov 1, 2021. However, our submission is due by May 3, 2021. Additionally, we have researched many other variables such as total sunlight hours, population density, and land to water area for each city. If utilized, these could influence the number of dengue cases and help us further improve our MAE score. Unfortunately, the competition prohibits the use of external data.