Final Project

DengAI: Predicting Disease Spread

Name: Young, Brent

DrivenData Name: bdy3176

DrivenData Public Score: 22.6058

Predict 413 Section #: 55

Quarter: Winter 2018

Introduction

Problem

The purpose of the final project is to analyze environmental data collected by various U.S. Federal Government agencies using time series and forecasting methods to predict the number of dengue fever cases (total_cases) reported each week (year, week of year) in San Juan, Puerto Rico (sj) and Iquitos, Peru (iq) in the test dataset. However, forecasting the number of dengue cases each week in each location can be difficult due to environmental variables such as changes in temperature, precipitation, vegetation, and more. As a result, these variables, along with total_cases from dengue_labels_train.csv and dengue_features_test have been consolidated into one dataset called dengue_features_train_combined.

Significance

The problem is significant/interesting because more than one-third of the world's population live in areas that are at risk for dengue infection. Additionally, the dengue virus is a leading cause of illness and death in the tropics and subtropics and more than 400 million are infected yearly. As a result, accurate dengue predictions can help public health workers and people around the world by taking steps to reduce the impacts of these life-threatening epidemics. For instance, departments in the U.S. Federal government such as Department of Health and Human Services, Department of Defense, etc. can use this data to ensure that they have enough health practitioners and resources available to combat the virus. As a result, this improves public health, decreases the number of future cases, reduces spreading of the virus, improves research initiatives, and ultimately helps save lives. Lastly, the fact that DrivenData has unique access to environmental data makes this problem even more interesting.

Data Exploration

Description of Training and Test Datasets

The training dataset is from week 18 of 1990 (4/30/1990) to week 17 of 2008 (4/22/2008) for San Juan (sj) and week 26 of 2000 (7/1/2000) to week 25 of 2010 (6/25/2010) for Iquitos (iq). The test dataset spans 5 years for San Juan and 3 years for Iquitos and is considered a true hold-out set, meaning that the test dataset is sequential and non-overlapping with any of the training data. For instance, the test dataset is from week 18 of 2008 (4/29/2008) to week 17 of 2013 (4/23/2013) for San Juan (sj) and week 26 of 2010 (7/2/2010) to week 26 of 2013 (6/25/2013) for Iquitos (iq).

Structure and Size of dengue_features_train_combined

The structure of the training dataset for San Juan has 936 rows and 26 variables, while the training dataset for Iquitos has 520 rows and 26 variables. The first variable, INDEX, is an integer variable that I manually added for imputation purposes and will be ignored. City is a factor variable and represents the two cities, year and weekofyear are integer variables, while week_start_date is a factor variable and represents the date. Total_cases is an integer variable

and represents the total number of dengue fever cases reported for that particular week. The rest of the variables are numeric and represent environmental variables such as changes in temperature, precipitation, and vegetation.

Figure 2: Descriptive Statistics of Total Cases for SJ and IQ

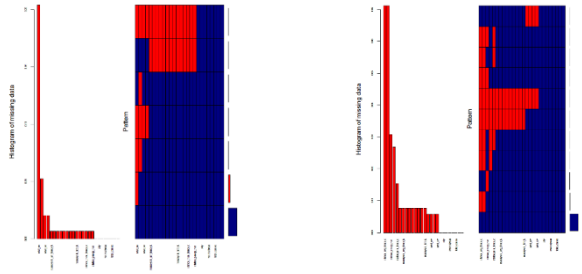
San Juan - Total Cases					
Min	1st Quartile	Median	Mean	3rd Quartile	Max
0	9	19	34.18	37	461

Iquitos - Total Cases					
Min	1st Quartile	Median	Mean	3rd Quartile	Max
0	1	5	7.565	9	116

Observations: Figure 2 shows summary statistics of total cases for each city. The data shows that the mean number of dengue fever cases reported each week in San Juan is 34.18, the median is 19, minimum is 0, and max is 461. In Iquitos, the mean number of dengue fever cases reported each week is 7.565, the median is 5, minimum is 0, and max is 116. There also appears to be some outliers and spikes in certain weeks for both cities. We will talk more about this in our data visualization section.

Data Preparation (Processing of the Data, Cleaning, and Feature Creation)

Figure 3: Missing Values Analysis for San Juan and Iquitos on Training Dataset & MICE Imputation (Method= PMM or 'Predictive Mean Matching')



Observations: In order to make it easier to come up with predictions for each city, I separated the training dataset into two distinct training datasets and time series: sj and iq (e.g., sj= DE[1:936,] and iq= DE[937:1456,], which includes total_cases and all variables from dengue_labels_train and dengue_features_train. I also copied the test data from dengue_features_test and pasted it into the dengue_features_train_combined so that we are working from one file. Overall, there are 522 NaN's in the training dataset and 119 NaN's on the test dataset. This sums to 641 NaN's (excluding total_cases from test dataset). Figure 3 shows a histogram of missing values as a percentage for all the variables for San Juan (left) and Iquitos (right) for the training dataset using the VIM package. For San Juan (training dataset), the results show that there are 362 NaN's. Ndvi_ne has the most missing data (20%), while ndvi_nw (5%) has the second most missing data. Overall, 18 of the 20 environmental variables for San Juan have missing data, although the percentage of missing data for each variable is quite small. For Inquitos (training dataset), the results show that there are 160 NaN's. Station_avg_temp_c and station_diur_temp_rng_c have the most missing data (7%). Overall, 18 of the 20 environmental variables for Iquitos have missing data, although the percentage of missing data for each variable is quite small as well. To address the missing values, I split the

data into two dataframes (numeric and categorical) and then used the MICE package (predictive mean matching) to fill in the NaN's on both the train and test dataset and merged the dataframes back together using INDEX. I also had to manually fill in reanalysis_sat_precip_amt_mm and precipitation_amt_mm in Excel to 0 since the imputation was not filling in those variables. After conducting the imputation, my summary statistics showed that there were no more NA's. This is important since we can now use the environmental variables as regressors in our future models for both cities.

Data Visualization

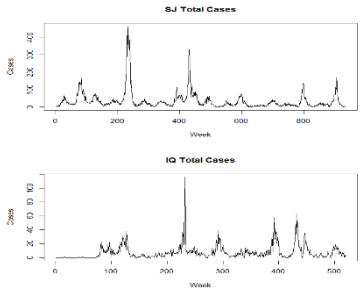


Figure 4

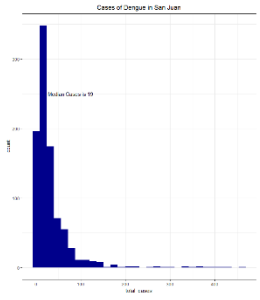


Figure 5

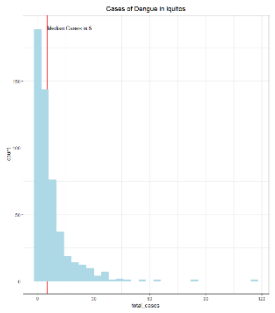


Figure 6

Observations: Figure 4 shows a timeplot of weekly data for both San Juan (top) and Iquitos (bottom). The timeplot shows that the weekly data for both cities are very volatile (random spikes and fluctuations) and show no trend. This could be due to environmental factors such as temperature, humidity, and contagiousness of the disease. Figure 5 shows a histogram of cases of dengue in San Juan. The results show that majority of cases are near the median of 19, there are a lot of cases that have 0, and there are outliers beyond 100 cases. Figure 6 shows a histogram of cases of dengue in Iquitos. The shape of the histogram is similar to San Juan, but there are a smaller number of cases. Majority of the cases are near the median of 5, there are also a lot of cases that have 0, and there are outliers beyond 35 cases.

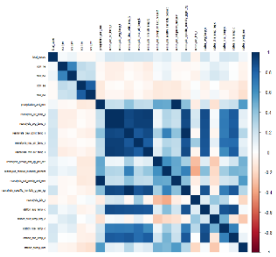


Figure 7

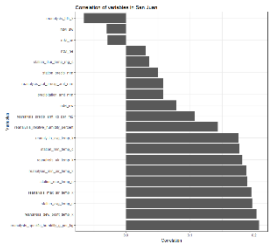


Figure 8

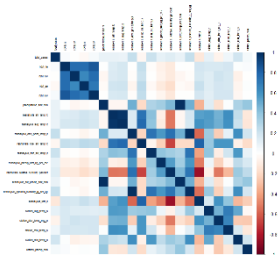


Figure 9

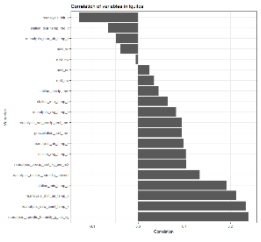


Figure 10

Observations: Figures 7 and 8 shows a correlation matrix and a correlation barplot of the environmental variables that were included in the dataset (along with total_cases) for San Juan. This gives us an idea of the most promising predictor variables based on the predictors that are most correlated with total_cases. The following variables are the most correlated with total_cases (near 0.20; blue shade): reanalysis_max_air_temp_k, reanalysis_specific_humidity_g_per_kg, reanalysis_dew_point_temp_k, station_avg_temp_c,

station_max_temp_c, station_avg_temp_c, reanalysis_min_air_temp_k, reanalysis_air_temp_k, station_min_temp_c, and reanalysis_avg_temp_k. Figures 9 and 10 shows a correlation matrix and a correlation barplot of the environmental variables that were included in the dataset (along with total_cases) for Iquitos. The following variables are the most correlated with total_cases (near .20; blue shade): reanalysis_min_air_temp_k, reanalysis_specific_humidity_g_per_kg, reanalysis_dew_point_temp_k, and station_min_temp_c. Overall, the results show that most of the temperature variables are strongly correlated with each other, while the vegetation index variables have weak correlations with the other variables. Furthermore, the results show that that there aren't any obvious strong correlations for total_cases for both cities. However, temperature/climate variables seem to be more positively correlated with total_cases compared to vegetation index variables (weak correlations). Interestingly, reanalysis_specific_humidity_g_per_kg and reanalysis_dew_point_temp_k were the most strongly correlated with total_cases for both cities, which make sense given that mosquitos thrive in wet and humid areas. Furthermore, as minimum, maximum, and average temperatures rise, the total_cases of dengue fever rises as well. This makes sense given that mosquito season is usually associated with hot weather. It is also interesting to note that the precipitation variables have very weak positive correlation to total_cases for both cities. Perhaps this is due to the fact that mosquitos need "standing water" to thrive and as a result, there is a "lag" in which total_cases will increase after rainfall.

Types of Models, Reviews of Literature, & Formulation of Models

Types of Models

Given that the EDA revealed volatility, weekly data, signs of contagiousness of dengue, and variables that are moderately correlated with total_cases. I used ETS (auto), auto.arima, and neural network (auto) to build my models. I decided to use ETS because weights decay exponentially as the observations get older (e.g., more recent observation, the higher associated weight). This can be useful since dengue is very contagious and the most recent values could be the most valuable. Next, I decided to use auto.arima with regressors given the environmental data/variables that were provided. Lastly, I will use a Neural Network model with regressors because it has the ability to "learn" from past data, has the ability to incorporate the environmental data/variables that were provided, and is robust to outliers, which is perfect in this situation given the volatility, random fluctuations/spikes in the data and contagiousness of dengue. For instance, lagged values of the timeseries can be used as inputs (e.g., the outputs of nodes in one layer are inputs to the next layer and the inputs to each node are then combined using a weighted linear combination. The results are then modified by a nonlinear function before being output). *See "Formulation of Models" section for more details on each model.*

Reviews of Literature *(see last page for references)*

There were many peer reviewed journals in the NU library database that used ETS, ARIMA, or Neural Network to forecast disease spread. For instance, in the *Malaria Research and Treatment (2017)*, Hussien, Eissa, and Awadalla use both ETS and ARMA models to forecast

malaria incidences in Sudan. Second, in the *Malaria Journal* (2009), Jacques, Stéphane, Loic, Lassane, Nadine, Ousmane, Jean, and Ogobar use ARIMA to forecast malaria incidence in Sudanese savannah area, Mali. Third, in the *Advanced Materials Research* (2012), Jin, Zuo, Ma, Guan, and Tan use neural network to forecast the prevalence of plant disease. Fourth, in the *Ophthalmology* (2005), Kaiserman, Rosner, and Pe'er use neural network to forecast the 5-year mortality from choroidal melanoma. Lastly, in the *BMC Infectious Diseases* (2011), Liu, Q; Liu, X; Jiang, and Yang use ARIMA to forecast the incidence of hemorrhagic fever with renal syndrome in China.

Formulation of Models

In order to test our predictions for our models, we will first split the data into a train and test set using the training set for each city. For instance, the training set for San Juan will include the first 675 observations and the test set will include the next 260 observations to mimic the period from week 18 of 2008 (4/29/2008) to week 17 of 2013 (4/23/2013). For Iquitos, we will also do the same thing. For example, the training set for Iquitos will include observations 937 to 1299 and the test set will include the next 156 observations to mimic the period from week 26 of 2010 (7/2/2010) to week 26 of 2013 (6/25/2013). We will then forecast each city using ETS, auto.arima, and neural network and then compare the results using performance/accuracy metrics. The model that has the best scores will be used for final submission.

San Juan

***ETS Model*:** Using the ETS model selection function, ETS produced a A,Ad,N model, which indicates additive errors, additive damped trend, and no seasonality. Figure 11 (top left), shows that ETS does not do a good job predicting total_cases of dengue (forecasted count in dark blue). The forecasted results are relatively flat and does not pick up the spikes very well. The forecast also shows very wide prediction intervals.

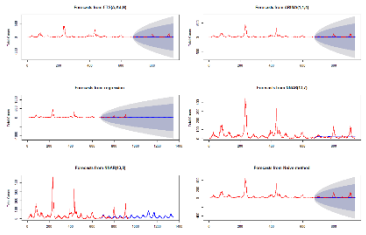


Figure 11

Auto.arima with Regressors: Using the auto.arima model selection function, auto.arima produced an ARIMA(1,1,4) which means 1 order of autoregressive, 1 degrees of first differencing, and 4 orders of the moving average part. Figure 12 shows a tsdisplay of the residuals, ACF of Residuals, and p-values for Ljung-Box test. The Ljung-Box test (p-value = 0.1719) confirmed that the model contained white noise. The following regressors are included in the model: reanalysis_max_air_temp_k, reanalysis_specific_humidity_g_per_kg, reanalysis_dew_point_temp_k, station_avg_temp_c, station_max_temp_c, station_avg_temp_c, reanalysis_min_air_temp_k, reanalysis_air_temp_k, station_min_temp_c, and reanalysis_avg_temp_k. I chose these based on the variables that were most correlated with total_cases. Figure 11 (middle left), shows that auto.arima with regressors does a slightly better job predicting total_cases of dengue than ETS and auto.arima without regressors. For instance, although the predictions are still somewhat

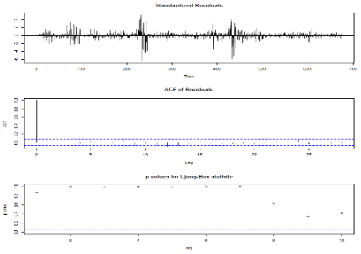


Figure 12

flat, it does a slightly better job picking up smaller spikes, whereas the auto.arima without regressors [ARIMA(1,1,4)] (top right) is relatively flat, similar to the ETS model. The forecast also shows wide prediction intervals, but is slightly narrower than auto.arima without regressors and ETS model.

Neural Network with Regressors: Using the nnetar model selection function, nnetar produced a NNAR (13,8) model, which indicates 13 lagged inputs and 8 nodes of hidden layer. The following regressors are included in the model: weekofyear, reanalysis_specific_humidity_g_per_kg, and reanalysis_min_air_temp_k. I chose these variables because mosquitos thrive in hot and humid conditions and adding weekofyear helps factor in the contagiousness of the dengue disease. These variables also had high correlations with total_cases. Figure 11 (bottom left), shows that the neural network model with regressors does the best job picking up the spikes compared to ETS, ARIMA, and neural network without regressors [NNAR(13,7)] (middle right).

Iquitos

***ETS Model*:** Using the ETS model selection function, ETS produced a A,N,N model, which indicates additive errors, no trend, and no seasonality. Figure 13 (top left), shows that ETS does not do a good job predicting total_cases of dengue (forecasted count in dark blue). The forecasted results are relatively flat and does not pick up the spikes very well. The forecast also shows very wide prediction intervals, similar to what we saw in San Juan.

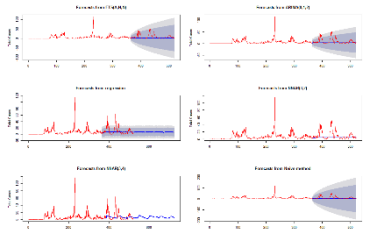


Figure 13

Auto.arima with Regressors: Using the auto.arima model selection function, auto.arima produced an ARIMA(3,0,0) which means 3 orders of autoregressive, 0 degrees of first differencing, and 0 orders of the moving average part. Figure 14 shows a tsdisplay of the residuals, ACF of Residuals, and p-values for Ljung-Box test. The Ljung-Box test (p-value = 0.3334) confirmed that the model contained white noise. The following regressors are included in the model: reanalysis_min_air_temp_k, reanalysis_specific_humidity_g_per_kg, reanalysis_dew_point_temp_k, and station_min_temp_c. I chose these based on the variables that were most correlated with total_cases. Figure 13 (middle left), shows that auto.arima with regressors does a slightly better job predicting total_cases of dengue than ETS and auto.arima without regressors [(ARIMA(0,1,2)] (top right). For instance, although the predictions are still somewhat flat, it does a better job picking up smaller spikes and forecasting the first spike, whereas the auto.arima without regressors (top right) is relatively flat, similar to the ETS model. The forecast also shows narrower prediction intervals that pick up the spikes compared to the wide prediction intervals in the ETS model and arima without regressors.

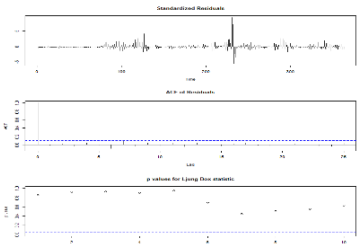


Figure 14

Neural Network with Regressors: Using the nnetar model selection function, nnetar produced a NNAR (3,4) model, which indicates 3 lagged inputs and 4 nodes of hidden layer. The following regressors are included in the model: weekofyear, reanalysis_specific_humidity_g_per_kg, and

reanalysis_min_air_temp_k. I chose these variables because mosquitos thrive in hot and humid conditions and adding weekofyear helps factor in the contagiousness of the dengue disease. These variables also had high correlations with total_cases, similar to what we saw in San Juan. Figure 14 (bottom left), shows that the neural network model with regressors does the best job picking up the spikes compared to ETS, ARIMA, and neural network without regressors [NNAR(3,2)] (middle right).

Performance/Accuracy

San Juan

Model Name	Train/Test	ME	RMSE	MAE	MASE	AIC	AICc	BIC
ETS	Training	0.0020122	14.49581	8.651223	0.9993015	8019.142	8019.267	8046.23
	Test	14.460337	30.87331	16.386551	1.8928081			
Auto ARIMA	Training	0.0046429	14.32655	8.722667	1.007554	5514.4	5514.53	5541.48
	Test	13.325234	30.35769	15.944399	1.841735			
ARIMA w/ Regressors	Training	0.008011	14.20721	8.708994	1.005975	5521.1	5521.83	5588.8
	Test	11.494992	29.0032	15.190389	1.75464			
NNETAR	Training	-0.0051708	7.275002	5.402023	0.6239869			
	Test	5.3676529	27.648299	15.204791	1.7563032			
NNETAR w/ Regressors	Training	-0.0146085	6.339327	4.707927	0.5438119			
	Test	7.35164	26.983471	13.273606	1.5332323			
NAÏVE	Training	0.004451	14.77492	8.65727	1			
	Test	15.475096	31.36835	16.87739	1.949505			

Figure 15

Observations: Figure 15 shows the performance/accuracy metrics of the ETS, Auto.ARIMA, Auto.ARIMA w/regressors, NNETAR, NNETAR w/regressors and NAÏVE models for San Juan. The results show that in regards to the error statistics on the training dataset, NNETAR with regressors produced the best error statistics (lowest RMSE, MAE, MASE) compared to the other models. NNETAR without regressors produced the second best results, while the rest of the models produced similar error statistics. In regards to the test dataset, NNETAR w/regressors again produced the best error statistics (lowest RMSE, MAE, MASE) compared to the other models on the test set. NNETAR without regressors produced the second best results, while the rest of the models produced similar error statistics. All the models on the test dataset performed better than the Naïve model benchmark. Overall, NNETAR w/regressors is the clear winner. As a result, I decided to apply NNETAR w/regressors to San Juan for final submission. These results are not surprising based on our forecast plots (figure 11).

Iquitos

Model Name	Train/Test	ME	RMSE	MAE	MASE	AIC	AICc	BIC
ETS	Training	0.0110323	7.523666	3.395224	0.9276008	3610.775	3610.842	3622.458
	Test	7.593042	14.065382	8.256953	2.2558619			
Auto ARIMA	Training	0.0279883	7.103661	3.628815	0.9914197	2454.27	2454.34	2465.94
	Test	6.1476122	13.335817	7.715963	2.1080593			
ARIMA w/ Regressors	Training	0.0305529	6.882235	3.706159	1.012551	2447.48	2447.89	2478.64
	Test	3.6343268	12.232133	7.434517	2.031166			
NNETAR	Training	-0.0135819	5.356883	2.987637	0.8162451			
	Test	5.905008	13.202585	7.657755	2.0921564			
NNETAR w/ Regressors	Training	0.0084624	4.141932	2.443157	0.667489			
	Test	3.7403496	11.968018	7.153952	1.954514			
NAÏVE	Training	0.0027624	7.819617	3.660221	1			
	Test	8.8980892	14.810695	9.038217	2.469309			

Figure 16

Observations: Figure 16 shows the performance/accuracy metrics of the ETS, Auto.ARIMA, Auto.ARIMA w/regressors, NNETAR, NNETAR w/regressors and NAÏVE models for Iquitos. The results show that in regards to the error statistics on the training dataset, NNETAR with regressors produced the best error statistics (lowest RMSE, MAE, MASE) compared to the other models. NNETAR without regressors produced the second best results, while the rest of the models produced similar error statistics. In regards to the test dataset, NNETAR w/regressors again produced the best error statistics (lowest RMSE, MAE, MASE) compared to the other models on the test set. All the models on the test dataset performed better than the Naïve model benchmark. Overall, NNETAR w/regressors is the clear winner, similar to what we saw in San Juan. As a result, I decided to apply NNETAR w/regressors to Iquitos for final submission. These results are not surprising based on our forecast plots (figure 13).

DrivenData - MAE	
Model Approach SJ/IQ	Public
ETS*	33.7572
Auto Arima*	33.7572
Auto Arima w/ Regressors*	34.0120
NNETAR*	26.3726
NNETAR w/Regressors*	**22.6058**

Figure 17, asterisk indicates that model was applied to both cities

Observations: Figure 17 shows the MAE score of NNETAR with regressors that was applied to both San Juan and Iquitos and submitted to DrivenData. The auto NNETAR model that was applied for final submission was NNAR (10,7) for San Juan and NNAR (5,4) for Iquitos with the following regressors: weekofyear, reanalysis_specific_humidity_g_per_kg, and reanalysis_min_air_temp_k. I also generated submissions for ETS, Auto.ARIMA, Auto.ARIMA w/regressors, and NNETAR for comparison purposes. The results show that NNETAR with regressors by far produced the best results. Overall, this is similar to what we saw in the performance/accuracy section for both cities.

Conclusion

Limitations

There are two primary limitations from the NNETAR model with regressors. First, is that the NNETAR model with regressors doesn't always predict the volatile spikes that we saw in the timeplots and often times had a mismatch with the actual results (as seen in our train/test data). Another limitation of the NNETAR model with regressors is that it is somewhat difficult to see what is going on behind the scenes. For instance, there is an element of "randomness" in the predictions produced by the model.

Future Work

In regards to future work, there are four areas that could help improve my models. First, it would be beneficial to obtain additional environmental/climate variables such as a "standing water variable" and healthcare response system variables and possibly incorporate those variables as additional regressors within NNETAR. Second, it could be helpful to explore adding lags into the model, which can be helpful for certain variables (e.g., precipitation). Third, it could be beneficial to explore different mixing of models using an ensemble approach since model diversity can help increase accuracy and performance. Lastly, it could be helpful to explore other modeling techniques (forecasting, regression, and machine learning). For instance, instead of just using ETS, auto.arima, and neural networks, using VAR, GARCH models, Poisson regression, Negative binomial regression, random forest, or xgboost could be interesting to explore.

Learnings

In the end, I learned six primary things from building these models. First, I learned how to forecast a dataset that contained weekly data and two different locations. Second, I learned that it's really important to conduct a thorough EDA and that a lot can be learned from it. Third, I learned that trying different modeling approaches is crucial and to never settle on a model due to gut instinct. We can never assume that one modeling approach will do better than the other. Fourth, I learned that adding regressors to neural networks can be very beneficial and choosing the "correct" regressors is crucial. Fifth, I learned that it is important to make sure you add a set.seed when running neural networks because the results tend to slightly change due to the randomness of the modeling technique. Lastly, I learned about dengue and how climate change can affect the spread of this contagious disease.

References

1. Hussien, H., Eissa, F., Awadalla, K. (2017). Statistical Methods for Predicting Malaria Incidences Using Data from Sudan. *Malaria Research and Treatment*, Vol. 2017. 9. [Peer Reviewed Journal].
2. Jacques,D., Stéphane,R.,Loic,F., Lassane,D., Nadine,D., Ousmane,T.,Jean, G., Ogobara, D. (April 2009). Modelling malaria incidence with environmental dependency in a locality of Sudanese savannah area, Mali. *Malaria Journal*, 8(1), 61. [Peer Reviewed Journal].
3. Jin, B., Zuo, Y.,Ma, X.,Guan, H.,Tan, F. (2012). The application on the forecast of plant disease based on an improved BP neural network. *Advanced Materials Research*, Vol. 433-440, 5469-5473. [Peer Reviewed Journal].
4. Kaiserman, I., Rosner, M., Pe'er, J. (2005). Forecasting the Prognosis of Choroidal Melanoma with an Artificial Neural Network. *Ophthalmology*, 112(9), 1608.e1-1608.e6. [Peer Reviewed Journal].
5. Liu,Q., Liu, X., Jiang, B.,Yang, W. (2011). Forecasting incidence of hemorrhagic fever with renal syndrome in China using ARIMA model. *BMC Infectious Diseases*, 11, 218-218. [Peer Reviewed Journal].