DengAI: Predicting Dengue Disease in Iquitos, Peru & San Juan, Puerto Rico Using Data Science

by Paul Ycay

CKME 136 - Data Analytics Capstone
Ryerson University

April 15, 2020

Background on Dengue Disease

- Dengue is the fastest spreading mosquito controlled disease worldwide, flourishing in poorurban areas subtropical, tropical climates
- The *Aedes Aegypti* species of mosquitoes are largely responsible for transmitting the virus, which causes symptoms of joint pain and high fever. Up to 50-100 million cases have been estimated in 100 endemic countries, spreading heavily within Latin America and Southeast Asia as of 2019
- In light of the ongoing COVID-19 outbreak, studying the relationships between meteorological factors and reported cases during an epidemic will help warn the general public in taking necessary precautions of future outbreaks.

Description of our Dataset

- This project will use Dengue data taken from the competition DengAI: Predicting Disease Spread, hosted by Driven Data. The data includes climatic information on San Juan, Peru & Iquitos, Puerto Rico between 1990-2010 (training set).
- Using varying meteorological data provided by the National Centers for Environmental Information (NOAA), the goal of this project was to predict the number of cases in the test set, which spans between 2008-2013 in San Juan and 2010-2013 in Iquitos.

Description of our Dataset

VARIABLE NAME (dengue_labels_train)	DESCRIPTION
city	Iquitos, Peru & San Juan, Puerto Rico
year	Year
weekofyear	Week of the corresponding year
week_start_date	Timeframe in DD-MM-YYYY
station_max_temp_c	Maximum temperature (°C): taken from National Centers for Environmental Information (NOAA) Global Historical Climatology Network (GHCN)
station_min_temp_c	Minimum temperature (°C): taken from National Centers for Environmental Information (NOAA) Global Historical Climatology Network (GHCN)
station_avg_temp_c	Average temperature (°C): taken from National Centers for Environmental Information (NOAA) Global Historical Climatology Network (GHCN)
station_precip_mm	Total precipitation (mm): taken from National Centers for Environmental Information (NOAA) Global Historical Climatology Network (GHCN)
station_diur_temp_rng_c	Diurnal temperature range (°C): taken from National Centers for Environmental Information (NOAA) Global Historical Climatology Network (GHCN)
precipitation_amt_mm	Total precipitation (mm)
reanalysis_sat_precip_amt_mm	Total precipitation (mm) : NOAA's National Centers for Environmental Prediction
reanalysis_dew_point_temp_k	Mean dew point temperature in Kelvin (K)
reanalysis_air_temp_k	Mean air temperature in Kelvin (K)
reanalysis_relative_humidity_percent	Mean relative humidity (ratio of the amount of water vapor actually present in the air to the greatest amount possible at the same temperature)
reanalysis_specific_humidity_g_per_kg	Mean specific humidity (mass g of water vapour in a unit mass kg of moist air)
reanalysis_precip_amt_kg_per_m2	Total precipitation (in kg /square meter)
reanalysis_max_air_temp_k	Max air temp in Kelvin (K)
reanalysis_min_air_temp_k	Min air temp in Kelvin (K)
reanalysis_avg_temp_k	Average air temp in Kelvin (K)
reanalysis_tdtr_k	Diurnal temperature range in Kelvin (K)
ndvi_se	NOAA's CDR Normalized Difference Vegetation Index. 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
total_cases	Total # of cases in timeframe

Description of our Dataset

Normalized Difference Vegetation Indices (NDVI):
Relationship between plants & Indices

- NDVI is a measure of plant health based on how the plant reflects light at certain frequencies; i.e. a calculation of vegetation health
- This value ranges from -1 to 1.
- Negative values correspond to dead plants or inanimate objects; healthy plants have positive indices

Data Approach: Data Preperation

- Import the 4 datasets. Merge the following datasets together: dengue_features_train & dengue_labels_train
- Common libraries used to explore the data was `dplyr`, `tidyr`, `readr`, as well as pre-loaded R functions.
- Check the format of variables and convert it appropriately (such as the date format)
- Identify dimensions of the data & determine NAs. We subset the data by city.
- Impute missing climate data (either median or most recent non-NA prior to it)

Data Approach: Exploratory Analysis

- We treat the training data as two seperate datasets based on city & make statistical assumptions on each dataset
- Using the pastecs library, we can easily generate our univariate & bivariate analysis in the form of a data-frame based on each city.
- The function stat_desc() will provide us basic statistics, such as the mean, median, mode, and any outliers of the data. It will also provide us advanced stats in a single data-frame
- We provide plots of our response variable total_cases, and how it functions overtime, as well as Time Series plots of relevant climate features.
- Correlation matrices can be used in determining which features have low influence on the target variable, total_cases
- Finally, we plot the response variable against the features in our data, which can tell us the appropriate machine learning algorithms to proceed with

Data Approach: Modeling

- From the hypotheses made about the data in the initial steps (correlation, p-values, plots), deploy techniques and algorithms to test the data.
- Test our data with multiple linear regression first, predicting `total_cases`.
- Apply Random Forest algorithm: used in both classification and regression.
- Random forest uses an ensemble of decision trees (randomized), where each tree determines a vote for prediction among the target variable; the algorithm picks the prediction with the most votes.
- Apply Support Vector Machine last; if regression is not suitable for this data due to non-linear relationships, SVM will be able to treat this

Data Approach: Validation

- Each algorithm will be approved on the testing set
- We pick the algorithm with the most accuracy and the least Mean Square Error
- This step will come towards the end of the project, after all tests have been made.
- We use the algorithm and its respective model to predict the total cases in the Test Set; we submit our predictions on the website. The website's scoring metric is based on Mean Absolute Error; used to calculate the amount of error in the predictions, and averages all of the absolute errors

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i|$$

Initial Analysis

Univariate & Bivariate Analysis of Iquitos, Peru

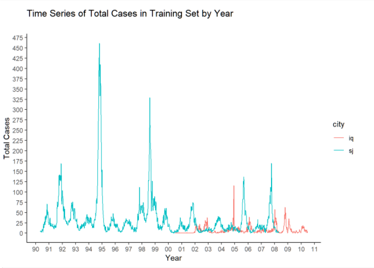
	nbr.val	nbr.null	nbr.na	min	max	range	sum	median	mean	SE.mean	CI.mean	var	std.dev	coef.var
ndvi_ne	516	0	0	0.06172857	0.5083571	0.4466285	136.3675	0.2653786	0.2642780	0.003577638	0.007028559	6.604538e-03	0.08126831	0.30751065
ndvi_nw	516	0	0	0.03586000	0.4544286	0.4185686	123.3429	0.2335928	0.2390366	0.003377975	0.006636305	5.887928e-03	0.07673283	0.32100869
ndvi_se	516	0	0	0.02988000	0.5383143	0.5084343	129.2624	0.2501286	0.2505086	0.003408345	0.006695970	5.994277e-03	0.07742272	0.30906215
ndvi_sw	516	0	0	0.06418333	0.5460167	0.4818334	137.9656	0.2624285	0.2673752	0.003794193	0.007453999	7.428284e-03	0.08618749	0.32234668
precipitation_amt_mm	516	4	0	0.00000000	210.8300000	210.8300000	33150.8000	60.4700000	64.2457364	1.550429800	3.045944900	1.240378e+03	35.21899490	0.54819190
reanalysis_air_temp_c	516	0	0	21.48571429	28.4871429	7.0014286	12755.2814	24.6728571	24.7195376	0.051550270	0.101274670	1.371234e+00	1.17099695	0.04737131
reanalysis_avg_temp_c	516	0	0	21.74285714	29.7785714	8.0357143	13407.2500	25.9714286	25.9830426	0.058641250	0.115205480	1.774419e+00	1.33207303	0.05126701
reanalysis_dew_point_temp_c	516	0	0	16.93857143	25.3000000	8.3614286	11528.9786	22.7021429	22.3429817	0.062390040	0.122570290	2.008539e+00	1.41722931	0.06343063
reanalysis_max_air_temp_c	516	0	0	26.85000000	40.8500000	14.0000000	17509.3000	33.9000000	33.9327519	0.104904860	0.206094090	5.678595e+00	2.38298030	0.07022656
reanalysis_min_air_temp_c	516	0	0	13.75000000	22.8500000	9.1000000	10173.8000	19.9000000	19.7166667	0.073212540	0.143831970	2.765799e+00	1.66306926	0.08434840
reanalysis_precip_amt_kg_per_m2	516	1	0	0.00000000	362.0300000	362.0300000	29726.6900	46.4400000	57.6098640	2.213742000	4.349076000	2.528738e+03	50.28655500	0.87288100
reanalysis_relative_humidity_percent	516	0	0	57.78714286	98.6100000	40.8228571	45737.7843	90.9171429	88.6391168	0.333862100	0.655899140	5.751537e+01	7.58388911	0.08555917
reanalysis_sat_precip_amt_mm	516	4	0	0.00000000	210.8300000	210.8300000	33150.8000	60.4700000	64.2457364	1.550429800	3.045944900	1.240378e+03	35.21899490	0.54819190
reanalysis_specific_humidity_g_per_kg	516	0	0	12.11142857	20.4614286	8.3500000	8821.5929	17.4285714	17.0961102	0.063646430	0.125038560	2.090248e+00	1.44576888	0.08456712
reanalysis_tdtr_c	516	0	0	-269.43571429	-257.1214286	12.3142857	-136194.7000	-264.1857143	-263.9432170	0.107790290	0.211762750	5.995273e+00	2.44852460	-0.00927671
station_avg_temp_c	516	0	0	21.40000000	30.8000000	9.4000000	14196.6905	27.6000000	27.5129660	0.040295890	0.079164530	8.378593e-01	0.91534655	0.03326964
station_diur_temp_rng_c	516	0	0	5.20000000	15.8000000	10.6000000	5417.3317	10.5266667	10.4987048	0.068831980	0.135226000	2.444726e+00	1.56356205	0.14892904
station_max_temp_c	516	0	0	30.10000000	42.2000000	12.1000000	17530.2000	34.0000000	33.9732558	0.059783450	0.117449430	1.844215e+00	1.35801892	0.03997318
station_min_temp_c	516	0	0	14.70000000	24.2000000	9.5000000	10939.0000	21.3500000	21.1996124	0.055332080	0.108704350	1.579806e+00	1.25690321	0.05928897
station_precip_mm	516	19	0	0.00000000	543.3000000	543.3000000	31972.9000	44.7500000	61.9629840	2.768783000	5.439498000	3.955737e+03	62.89465200	1.01503600
total_cases	516	95	0	0.00000000	116.0000000	116.0000000	3920.0000	5.0000000	7.5968992	0.475303200	0.933771600	1.165712e+02	10.79681320	1.42121320

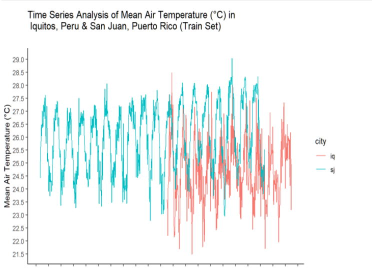
Initial Analysis

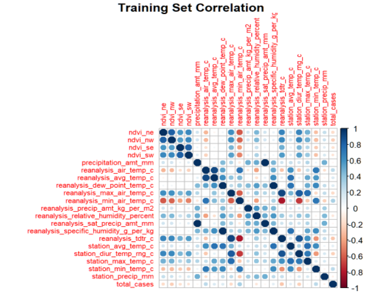
Univariate & Bivariate Analysis of San Juan, Puerto Rico

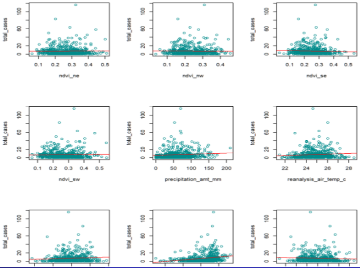
	nbr.val	nbr.null	nbr.na	min	max	range	sum	median	mean	SE.mean	CI.mean	var	std.dev	coef.var
ndvi_ne	930	0	0	-0.40625000	0.4934000	0.8996500	54.40049	0.0587750	0.05849515	0.003465400	0.006800921	1.116837e-02	0.10568051	1.80665429
ndvi_nw	930	0	0	-0.45610000	0.4371000	0.8932000	60.86540	0.0673875	0.06544667	0.003071183	0.006027260	8.771912e-03	0.09365849	1.43106568
ndvi_se	930	0	0	-0.01553333	0.3931286	0.4086619	165.14261	0.1767012	0.17757269	0.001869493	0.003668918	3.250353e-03	0.05701186	0.32106210
ndvi_sw	930	0	0	-0.06345714	0.3814200	0.4448771	155.00670	0.1677584	0.16667387	0.001831719	0.003594786	3.120329e-03	0.05585991	0.33514496
precipitation_amt_mm	930	238	0	0.00000000	390.6000000	390.6000000	32881.44000	20.6050000	35.35638700	1.461819000	2.868851000	1.987332e+03	44.57950200	1.26086100
reanalysis_air_temp_c	930	0	0	22.78857143	29.0500000	6.2614286	24192.69714	26.1042857	26.01365284	0.040544110	0.079568660	1.528757e+00	1.23642919	0.04753001
reanalysis_avg_temp_c	930	0	0	22.96428571	29.0142857	6.0500000	24298.03571	26.2285714	26.12692012	0.039960700	0.078423700	1.485077e+00	1.21863747	0.04664298
reanalysis_dew_point_temp_c	930	0	0	16.49285714	24.6457143	8.1528571	20422.35286	22.3142857	21.95951920	0.051480470	0.101031490	2.464722e+00	1.56994332	0.07149261
reanalysis_max_air_temp_c	930	0	0	24.65000000	31.1500000	6.5000000	26271.40000	28.3500000	28.24881720	0.041281830	0.081016450	1.584896e+00	1.25892666	0.04456564
reanalysis_min_air_temp_c	930	0	0	19.45000000	26.7500000	7.3000000	22461.20000	24.3500000	24.15182796	0.042455050	0.083318930	1.676261e+00	1.29470516	0.05360692
reanalysis_precip_amt_kg_per_m2	930	2	0	0.00000000	570.5000000	570.5000000	28332.84000	21.3000000	30.46541900	1.168290000	2.292793000	1.269358e+03	35.62805500	1.16945900
reanalysis_relative_humidity_percent	930	0	0	66.73571429	87.5757143	20.8400000	73068.40857	78.6678571	78.56818126	0.111145680	0.218125710	1.148863e+01	3.38948769	0.04314072
reanalysis_sat_precip_amt_mm	930	238	0	0.00000000	390.6000000	390.6000000	32881.44000	20.6050000	35.35638700	1.461819000	2.868851000	1.987332e+03	44.57950200	1.26086100
reanalysis_specific_humidity_g_per_kg	930	0	0	11.71571429	19.4400000	7.7242857	15393.74000	16.8457143	16.55240860	0.051184680	0.100451010	2.436481e+00	1.56092305	0.09430187
reanalysis_tdtr_c	930	0	0	-271.79285714	-268.7214286	3.0714286	-251689.37143	-270.6928571	-270.63373272	0.016359300	0.032105470	2.488928e-01	0.49889161	-0.00184342
station_avg_temp_c	930	0	0	22.84285714	30.0714286	7.2285714	25116.07143	27.2285714	27.00652842	0.046415200	0.091090800	2.003565e+00	1.41547346	0.05241227
station_diur_temp_rng_c	930	0	0	4.52857143	9.9142857	5.3857143	6284.35714	6.7571429	6.75737327	0.027413280	0.053799130	6.988838e-01	0.83599268	0.12371563
station_max_temp_c	930	0	0	26.70000000	35.6000000	8.9000000	29395.40000	31.7000000	31.60795699	0.056312380	0.110514210	2.949108e+00	1.71729665	0.05433115
station_min_temp_c	930	0	0	17.80000000	25.6000000	7.8000000	21018.60000	22.8000000	22.60064516	0.049392760	0.096934310	2.268869e+00	1.50627665	0.06664751
station_precip_mm	930	24	0	0.00000000	305.9000000	305.9000000	24910.50000	17.7500000	26.78548390	0.961631200	1.887221300	8.600032e+02	29.32581080	1.09483970
total_cases	930	4	0	0.00000000	461.0000000	461.0000000	31734.00000	19.0000000	34.12258100	1.688694000	3.314097000	2.652069e+03	51.49824200	1.50921300

Correlation, plots, & relationships of features against response variable









Multiple Linear Regression

- We split `train_df` into training & testing splits (80% & 20%, respectively)
- Fit linear model using all possible climate features as independent variables; most significant predictors in the model were the vegetation indices
- Multiple R^2 yielded a value of 0.1657 and an adjusted R^2 of 0.1519 - poor performance
- Tried to backward elimination & refitted
- The following variables were chosen as the final model: `ndvi_ne`, `ndvi_nw`, `ndvi_se`, `ndvi_sw`, `reanalysis_avg_temp_c`, `reanalysis_relative_humidity_percent`, `reanalysis_specific_humidity_g_per_kg`, `station_diur_temp_rng_c`, & `station_max_temp_c`.
- Similar performance as previous; MLR is not the best model for predicting on this dataset

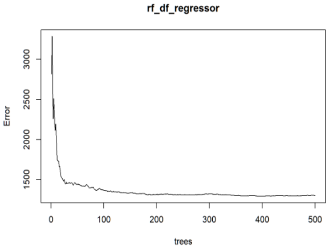
Multiple Linear Regression Summary Using Backward Elimination

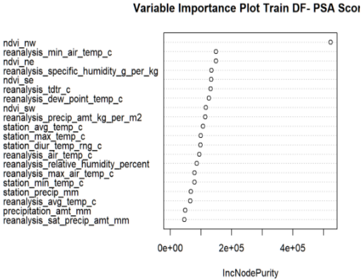
```
##
## Call:
## lm(formula = total_cases ~ ndvi_ne + ndvi_nw + ndvi_se + ndvi_sw +
##     reanalysis_avg_temp_c + reanalysis_relative_humidity_percent +
##     reanalysis_specific_humidity_g_per_kg + station_max_temp_c,
##     data = subset(df_training, select = -c(1:4, 26:28)))
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -72.14 -20.90  -6.11   7.41 384.05
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            437.9971    67.9928   6.442 1.73e-10 ***
## ndvi_ne                                 68.1498    17.8590   3.816 0.000143 ***
## ndvi_nw                               -114.5563    19.7607  -5.797 8.69e-09 ***
## ndvi_se                                -87.8794    30.9550  -2.839 0.004605 **
## ndvi_sw                                109.9257    29.4167   3.737 0.000195 ***
## reanalysis_avg_temp_c                  -20.0380     3.1770  -6.307 4.03e-10 ***
## reanalysis_relative_humidity_percent    -4.9165     0.5811  -8.461  < 2e-16 ***
## reanalysis_specific_humidity_g_per_kg   25.5800     2.8646   8.930  < 2e-16 ***
## station_max_temp_c                       2.7055     1.0814   2.502 0.012493 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.61 on 1157 degrees of freedom
## Multiple R-squared:  0.1598, Adjusted R-squared:  0.154
## F-statistic:  27.5 on 8 and 1157 DF,  p-value: < 2.2e-16
```

Random Forest

- Random forest creates multiple decision trees at a a time by taking select variables at random. It simultaneously develops multiple trees in combination and finally averages the error to bring out the best possible results. We use the package `randomForest` for our analysis
- Created 9 models in total using this algorithm: 3 models on the entire train set, 3 models for each set subsetted by city
- 1^{st} random forest model created 500 trees & selected 6 independent variables at random
- 2^{nd} extends on the first by using optimal number of trees giving least MSE. Use optimal tree number to prune the model
- 3^{rd} uses feature selection based on Node Purity. Higher Node Purity of that variable, the more useful it is in the model. Return 6 variables with the highest Node Purity, & call it into 3^{rd} model

Random Forest Summary on Entire Train Set

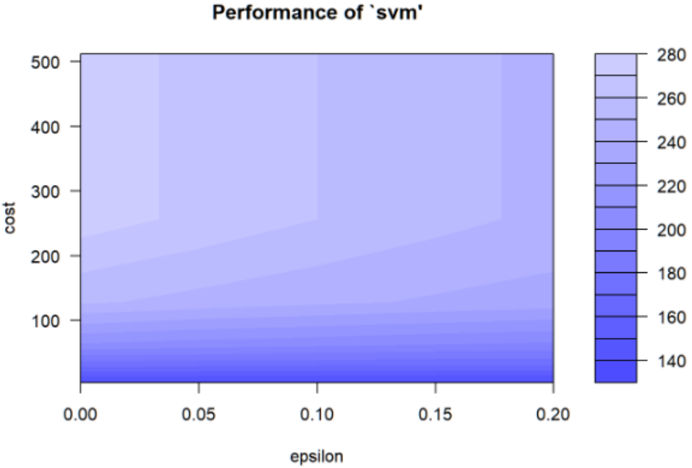




Support Vector Machine

- As SVM is non-parametric, it wont actually train the network.
- The SVM algorithm tries to plot all of the data in an *n*-dimensional hyper plane and applies the same logic on the test set, based on the reference created by the training set.
- The algorithm then tries to draw the boundary between the classes based on Support Vector machines.
- We use the package `e1071` to perform our analysis.
- We try the grid approach to build multiple models at a time, so that we can pick the best model from all the developed models.'
- 3 models were created: 1 for the entire train set, and 1 for each city subsets
- Out of the 3 models, Iquitos yielded the lowest *MSE*

Support Vector Machine - Parameter Tuning For Iquitos



Overview of Random Forest & SVM Models

Model Name	Algorithm	Description	MSE on train	% Variance covered on Train dataset	MSE on test
rf_df_regressor	Random Forest	Built on df_training dataset. We have used a basic Random forest model which has created 500 trees from the selection of 6 random independent variables.	1304.461	41.91	666.8971
rf_df_regressor2	Random Forest	After building the Random Forest algorithm, the algortithm tries to fit the model to the data, which may cause overfitting. As the model has generated 500 trees, it may generate a higher MSE. We developed the model based on the number of trees generating the least MSE	1260.36	43.87	620.1547
rf_df_regressor3	Random Forest	We have built this model with top 6 variables based on their Node Purity rating. We find it using the function VarImportance() . Node Purity indicates the ease of identifying the variable to a class or value. It's better to have high Node Purity.	1092.494	51.35	538.5882
rf_sj_regressor	Random Forest	Built on sj_training dataset. Similar to rf_df_regressor	1867.153	40.14	753.0785
rf_sj_regressor2	Random Forest	Built with trees pruned for minimum MSE	1938.588	37.85	737.931
rf_sj_regressor3	Random Forest	Built with the top 6 variables based on Node Purity	1464.458	53.05	790.4614
rf_iq_regressor	Random Forest	Built on iq_training dataset. Similar to rf_df_regressor	135.0008	-3.23	52.23506
rf_iq_regressor2	Random Forest	Built with trees pruned for minimum MSE	137.1541	-4.87	52.31801
rf_iq_regressor3	Random Forest	Built with top 5 Variables based on Node Purity	127.5276	2.49	60.39508
tunemodel_df	SVM	Built on df_training dataset with different Epsilon and Cost values; this has chosen the best possible model. Once the training completes, we can bring out the best tuned model and the parameters for it, such as Epsilon & Cost Once we pickup best model, we can use that for training	na	na	800.2505
tunemodel_sj	SVM	Similar to one above, but built on sj_training	na	na	856.6288
tunemodel_iq	SVM	Similar to one above, but built on iq_training	na	na	61.12607

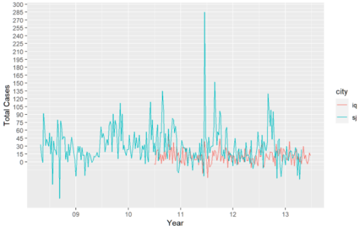
Overview of Random Forest & SVM Models

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (f_i - y_i)^2$$

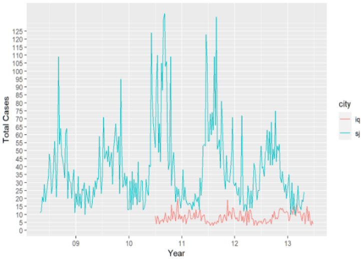
- N is the number of data points, f_i is the value returned by the model, and y_i is the actual value for data point i. We calculate the scores of each model above using the test set sample split. The lower the MSE, the better the score
- `rf_iq_regressor` (Random Forest Regressor 1 for Iquitos) is the best model for predicting in the Iquitos test set
- `rf_sj_regressor2` (Random Forest Regressor 2 for San Juan) is the best model for predicting in the San Juan test set
- `rf_df_regressor3` (Random Forest Regressor 3 for Entire set) is the best model for predicting the entire test set

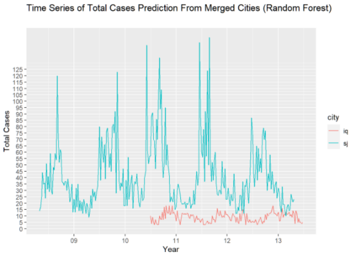
Plotting Predicted Cases

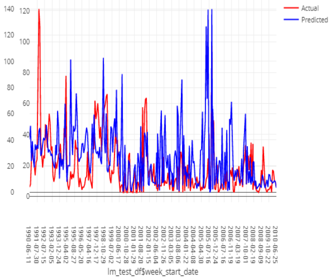
Time Series of Total Cases Prediction in Test Set (SVM)



Time Series of Total Cases Prediction in Test Set (Random Forest)




Time Series of Total Cases Prediction From Merged Cities (Random Forest)
Total Cases
Year
09
10
11
12
13
city
iq
sj


Actual
Predicted
lm_test_df$week_start_date

Final Statements & Further Results

DengAI: Predicting Disease Spread
HOSTED BY DRIVENDATA

Submissions

BEST	CURRENT RANK	# COMPETITORS	SUBS. MADE
27.4784	2346	8709	3 of 3

SUBMISSION RESTRICTIONS

- Placed in the top 27% among 8709 competitors
- Several ways to improve our models and achieve a higher score in the future
- Better treatment of variables; explore normalization & Principal Component Analysis (PCA)
- Artificial Neural Networks (ANN) may benefit this project as neural networks can identify the hidden patterns within the data
- As the data is time series, we can also explore ARIMA (Auto Regressive Integrated Moving Average) models as well

Final Statements & Further Results

- We were able to explore patterns in our data that derive from the literatures studied
- Analyzing climate patterns in areas with high infection rates will help us determine how these vector borne diseases and carriers behave in the future. By determining these patterns, humans are more equipped to prevent such outbreaks
- Likewise, the idea of analyzing social and physical patterns among humans during the COVID-19 pandemic (whether they are traveling, isolating, social distancing, etc...) will influence how the outbreak behaves in the future

The End