Table of Contents

Introduction

Disease control is an important aspect of a society to keep its inhabitants healthy. Every year many organizations spend a considerable amount of money and time to research about each disease and how to prevent them. In-order to do this, they require proper data (analyzed data) to help achieve a common goal. We as data scientists should provide our analysis on the available data and as machine learning students, we should be able to predict the disease spread based on the data we have. This is precisely what we have done in our project. We have analyzed the data set obtained from drivendata.org (https://www.drivendata.org/competitions/44/dengai-predicting-disease-spread/), done the necessary pre-processing on it and then have used multiple models to predict the total cases for any given condition.

Problem Description

We have obtained the data set for predicting total number of Dengue cases that might occur at any given time for any condition provided that the conditions are within the range of features considered for training the model.

Dataset Information

Total number of instances : 1456

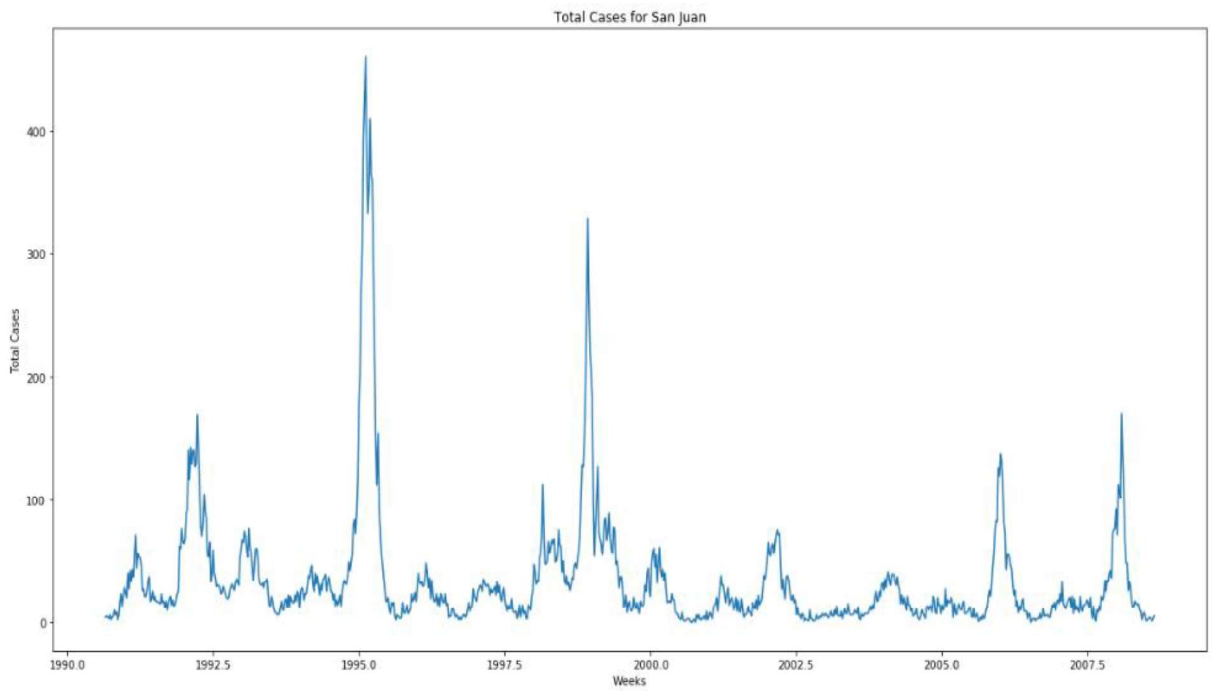
Total number of features : 24

1. city – City abbreviations: sj for San Juan and iq for Iquitos
2. year
3. weekofyear
4. week_start_date
5. station_max_temp_c – Maximum temperature
6. station_min_temp_c – Minimum temperature
7. station_avg_temp_c – Average temperature
8. station_precip_mm – Total precipitation
9. station_diur_temp_rng_c – Diurnal temperature range
10. precipitation_amt_mm – Total precipitation
11. reanalysis_sat_precip_amt_mm – Total precipitation
12. reanalysis_dew_point_temp_k – Mean dew point temperature
13. reanalysis_air_temp_k – Mean air temperature
14. reanalysis_relative_humidity_percent – Mean relative humidity
15. reanalysis_specific_humidity_g_per_kg – Mean specific humidity
16. reanalysis_precip_amt_kg_per_m2 – Total precipitation
17. reanalysis_max_air_temp_k – Maximum air temperature
18. reanalysis_min_air_temp_k – Minimum air temperature
19. reanalysis_avg_temp_k – Average air temperature
20. reanalysis_tdtr_k – Diurnal temperature range
21. ndvi_se – Pixel southeast of city centroid
22. ndvi_sw – Pixel southwest of city centroid
23. ndvi_ne – Pixel northeast of city centroid
24. ndvi_nw – Pixel northwest of city centroid

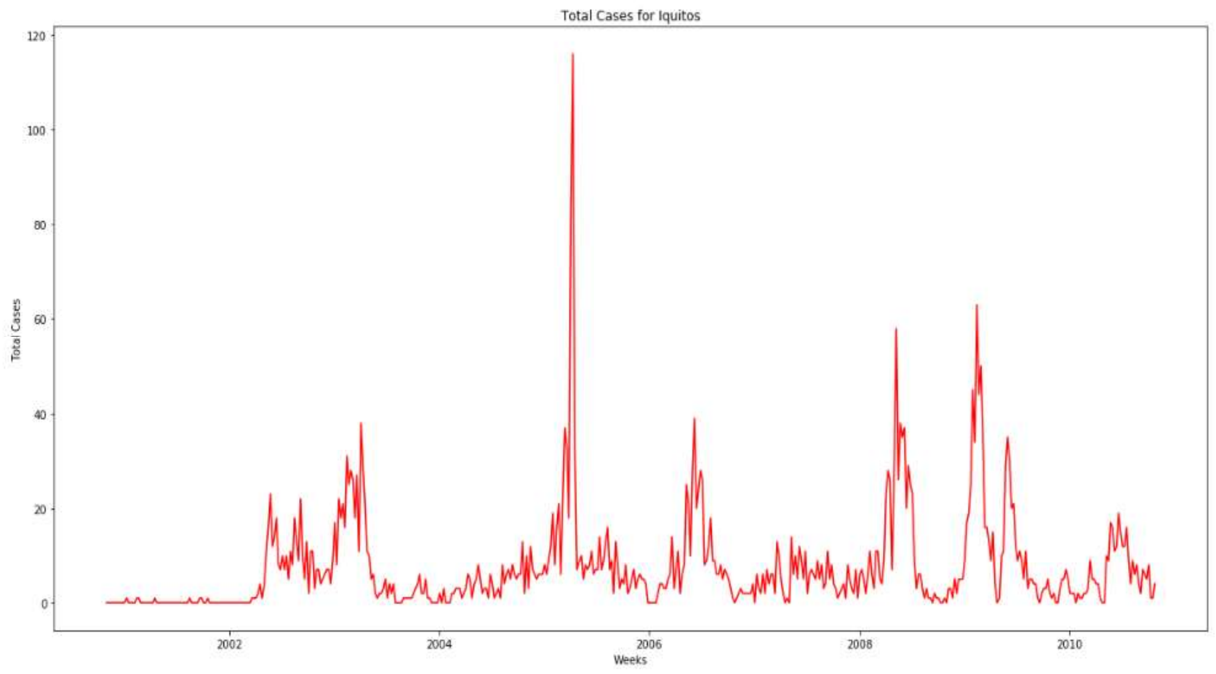
Target

25. total_cases

Visualization of Data



Data spread indicating total number of cases of Dengue outbreak over the duration from 1990 to 2008 for San Juan



Data spread indicating total number of cases of Dengue outbreak over the duration from 1990 to 2008 for Iquitos

Pre-processing

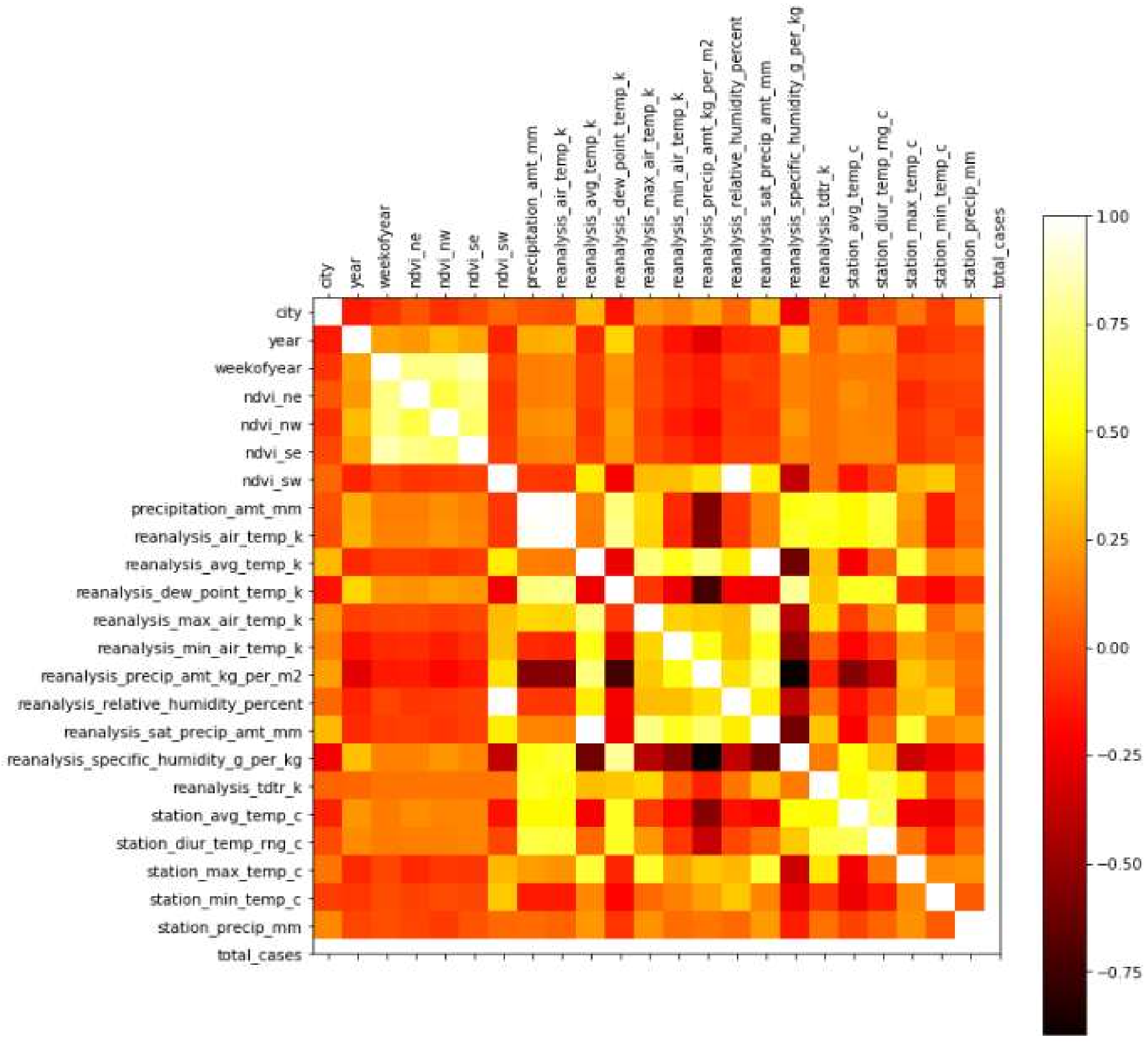
The data set contained many features related to temperature but they were in two different scales i.e Kelvin and Celsius. We first standardized all of these by converting Celsius scale feature to Kelvin by adding 273.15 for each value.

Then it was observed that a number of instances had blank values for certain features. This was solved by first replacing the null values with NaN, then substituting these NaN values with the previous instance value for that feature as the conditions do not vary a lot from one instance to another.

We then merged the training file with label file to obtain all features with the target attribute in a single dataframe by considering city, year and weekofyear available in both files.

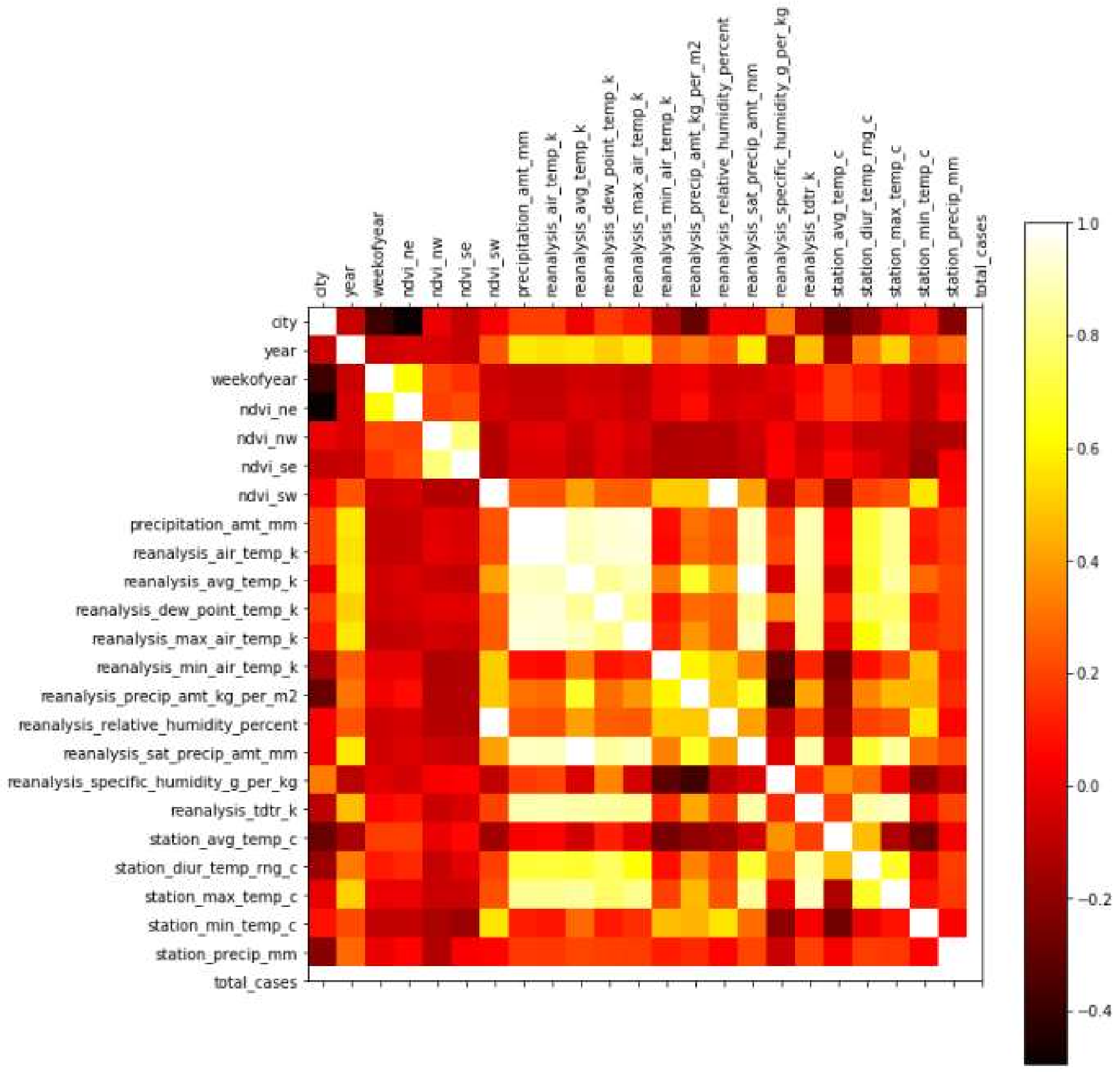
The merged data set contains the information for two different cities. They are San Juan(sj) and Iquitos(iq). Since these are different cities from different locations having different climatic conditions, the data needed to be analyzed separately, we divided the data based on city column available to us. All further pre-processing was done separately for each divided data set.

Co-relation plot was obtained for each city. Co-relation highlights how strong one feature is dependent or related to other features. This gives an insight into highly co-related features which can be eliminated(except one among them) before training the model. The co-relation plots for both cities are as shown below.



Co-relation plot for Iquitos data set

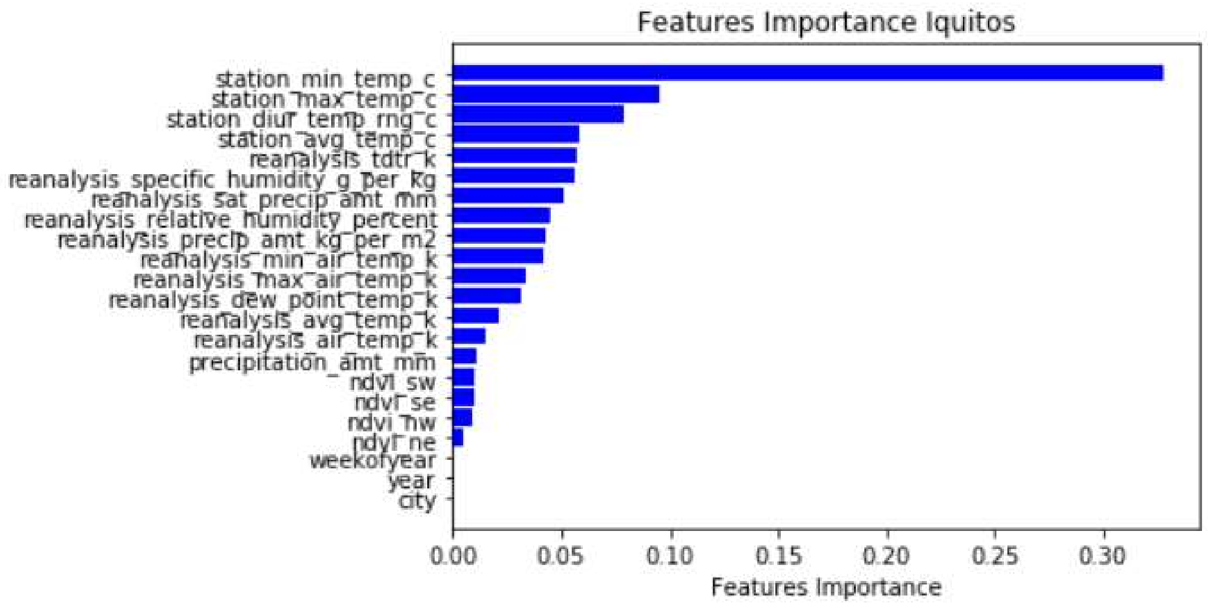
From the above co-relation plot, it is observed that features like *ndvi_ne, ndvi_nw, ndvi_se* are highly co-related and contribute in a similar manner for training the model. Hence we eliminated these features from our training data set for Iquitos data.



Co-relation plot for San Juan data set

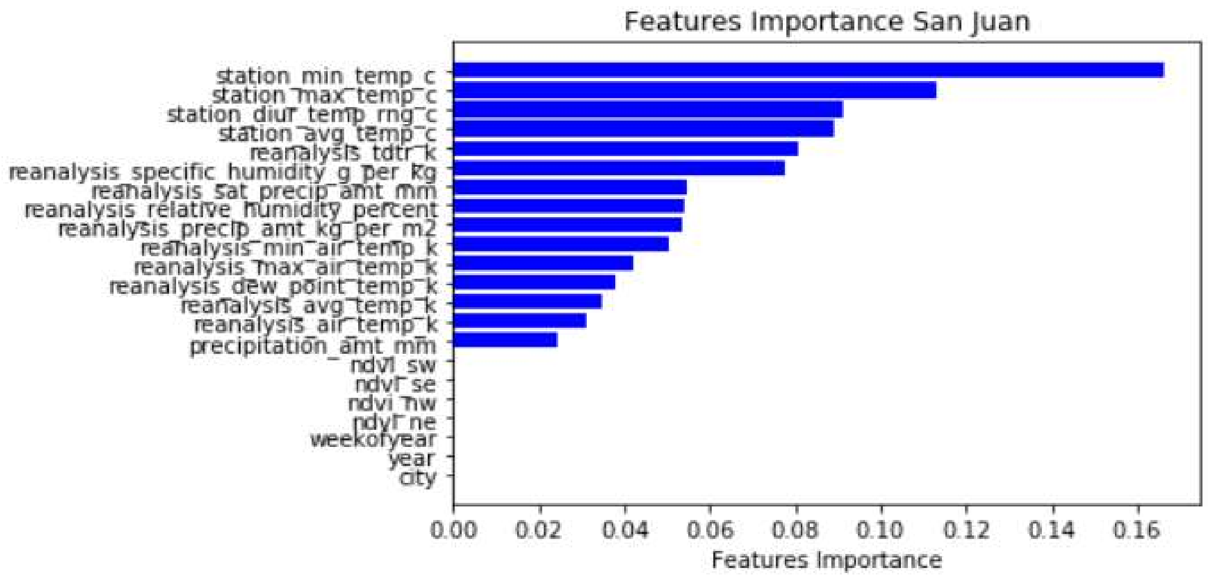
From the above co-relation plot, it is observed that features like *reanalysis_tdtr_k, reanalysis_sat_precip_amt_mm, reanalysis_dew_point_temp_k , reanalysis_air_temp_k , reanalysis_max_air_temp_k* are highly co-related and contribute in a similar manner for training the model. Hence we eliminated these features from our training data set for San Juan data.

The last step of pre-processing was to obtain feature importance. This was performed using RandomForest Classifier. The feature importance plot is as shown below,



Feature importance plot for Iquitos

As seen from the abobe plot for Iquitos, features like ndvi_se, ndvi_sw, ndvi_nw, ndvi_ne, weekofyear, city, year, precipitation_amt_mm, reanalysis_air_temp_k, reanalysis_avg_temp_k contribute minimally. Hence these will be eliminated along with nthose obtained from co-relation plot to form the final training data set for Iquitos city.

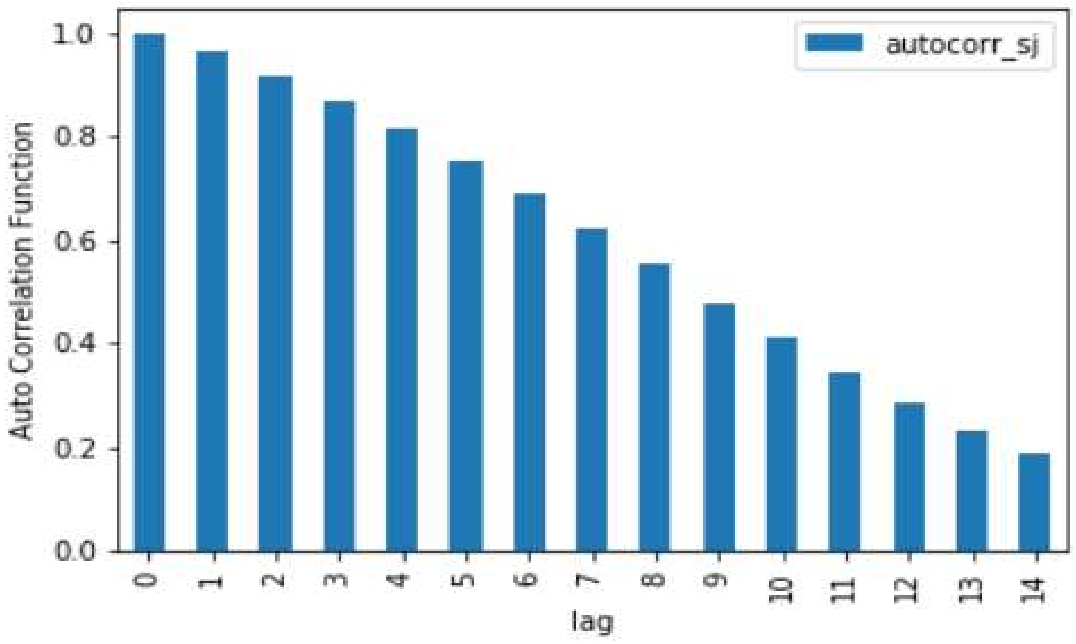


Feature importance plot for San Juan

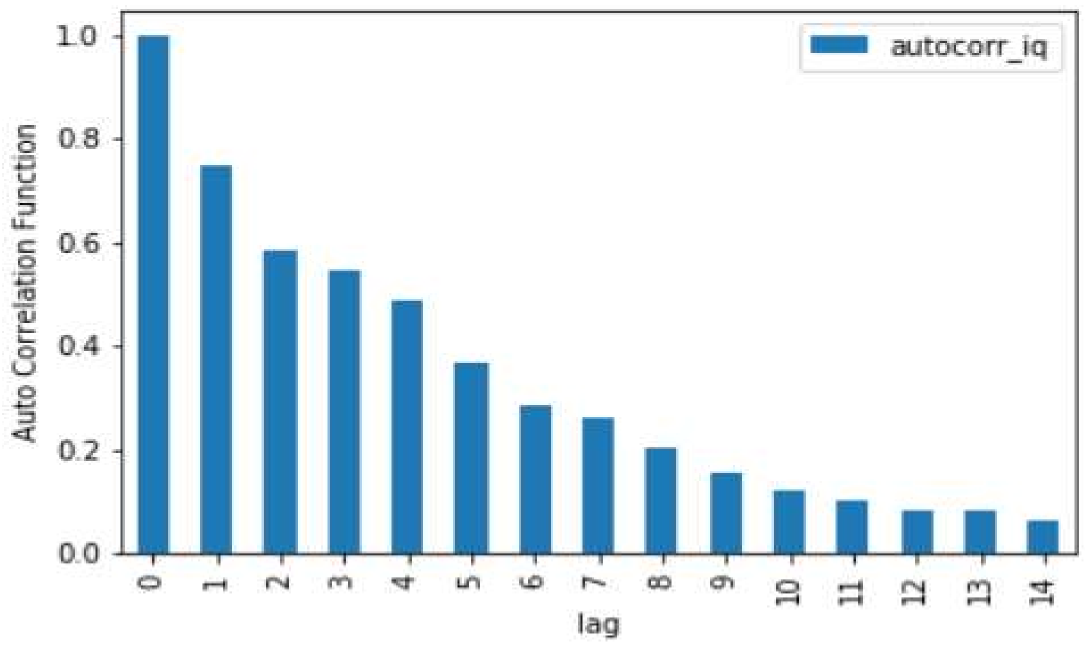
As seen from the abobe plot for San Juan, features like ndvi_se, ndvi_sw, ndvi_nw, ndvi_ne, weekofyear, city, year, precipitation_amt_mm contribute minimally. Hence these will be eliminated along with those obtained from co-relation plot to form the final training data set for San Juan city.

Feature Engineering

Auto-correlation plot was obtained for total cases with itself from previous 15 weeks. This is because, we have to consider the incubation period of the disease before we see the symptoms. This may be from one week to a few weeks. For this to occur, the conditions of previous weeks will be responsible as favorable conditions are needed for mosquitos to breed in large numbers. To facilitate all these factors, we have considered previous 15 weeks values of total cases to obtain the auto-correlation plot. These are as shown below,

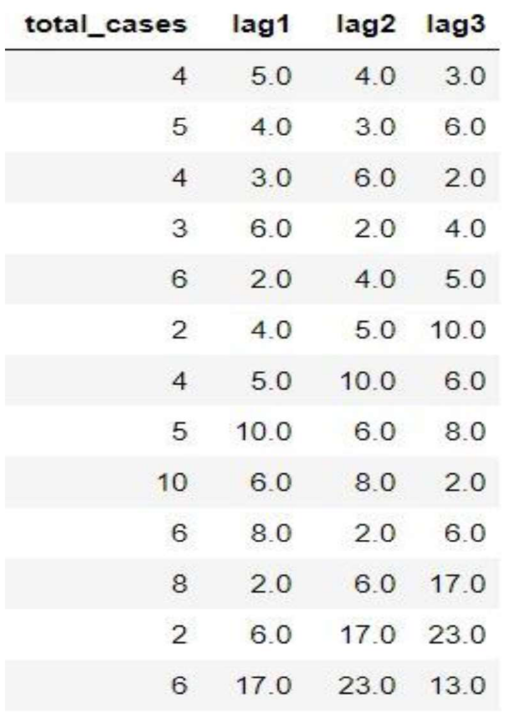


Auto-correlation plot for San Juan



Auto-correlation plot for San Juan

From the above plots, one can observe that the conditions of previous 3 weeks contribute heavily for the total cases to increase in current week. To accommodate this factor, three lag factors have been introduced in the data set by shifting the total cases by one, two and three positions above respectively. An illustration is shown below,



total_cases	lag1	lag2	lag3
4	5.0	4.0	3.0
5	4.0	3.0	6.0
4	3.0	6.0	2.0
3	6.0	2.0	4.0
6	2.0	4.0	5.0
2	4.0	5.0	10.0
4	5.0	10.0	6.0
5	10.0	6.0	8.0
10	6.0	8.0	2.0
6	8.0	2.0	6.0
8	2.0	6.0	17.0
2	6.0	17.0	23.0
6	17.0	23.0	13.0

Model training and validation

Performance metric :

MAE: This measures the closeness predictions to the actual outcomes

Mean Absolute Error metric is chosen as this is a regression problem and not classification.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

$$Actual = y_i$$

$$Predicted = \hat{y}_i$$

Gradient Boosting

learning_rate : learning rate shrinks the contribution of each tree by learning_rate.

n_estimators : The number of boosting stages to perform.

max_depth : maximum depth of the individual regression estimators.

max_leaf_nodes : maximum number of leaf nodes to consider

Learning_rate	N_estimators	Max_depth	Max_leaf_nodes	MAE

0.1	100	3	None	Sj = 8.79
				Iq = 3.6

0.1	50	3	None	Sj = 8.49
				Iq = 3.73

0.1	100	4	3	Sj = 8.85
				Iq = 4.32

0.05	100	4	3	Sj = 8.79
				Iq = 4.29

0.05	100	4	16	Sj = 8.56
				Iq = 3.51

ADA Boosting

Base_estimator : The base estimator from which the boosted ensemble is built.

N_estimators : The maximum number of estimators at which boosting is terminated.

Learning_rate : Learning rate shrinks the contribution of each tree by learning_rate.

Loss : The loss function to use when updating the weights after each boosting iteration.

Base_estimator	N_estimators	Learning_rate	Loss	MAE

DecisionTreeRegressor	50	1	linear	Sj = 10.26
				Iq = 5.21

DecisionTreeRegressor	25	1	linear	Sj = 9.1
				Iq = 3.64

DecisionTreeRegressor	25	1	square	Sj = 9.25
				Iq = 3.51

GradientBoostingRegressor	25	1	square	Sj = 8.62
				Iq = 4.14

GradientBoostingRegressor	25	0.5	square	Sj = 8.69
				Iq = 3.75

SVR

C : Penalty parameter C of the error term.

max_iter : Hard limit on iterations within solver

epsilon : It specifies the epsilon-tube within which no penalty is associated in the training loss function with points predicted within a distance epsilon from the actual value.

C	Max_iter	Epsilon	MAE

1	-1 (no limit)	0.1	Sj = 30.11
			Iq = 5.97

5.0	631	0.075	Sj = 29.07
			Iq = 5.79

10	631	0.075	Sj = 28.86
			Iq = 6.00

2	1000	0.055	Sj = 29.23
			Iq = 5.79

3	1000	0.3	Sj = 29.17
			Iq = 5.77

Logistic regression

Solver : Algorithm to use in the optimization problem.

Penalty : Used to specify the norm used in the penalization.

Max_iter : Hard limit on iterations within solver.

Solver	Penalty	Max_iter	MAE

Liblinear	L1	100	Sj = 17.4
			Iq = 5.00

Sag	L1	100	Sj = 17.87
			Iq = 5.91

Sag	L1	800	Sj = 17.47
			Iq = 4.36

lbfgs	L2	100	Sj = 17.44
			Iq = 4.14

liblinear	L2	100	Sj = 17.40
			Iq = 4.99

Random Forest

N_estimators : The number of trees in the forest.

Max_depth : The maximum depth of the tree.

Bootstrap : Whether bootstrap samples are used when building trees.

N_estimators	Max_depth	Bootstrap	MAE

10	None	True	Sj = 9.79
			Iq = 4.42

20	None	True	Sj = 8.32
			Iq = 3.67

20	None	False	Sj = 10.87
			Iq = 3.85

20	2	True	Sj = 11.39
			Iq = 4.50

20	7	True	Sj = 7.80
			Iq = 3.90

Deep Learning

Hidden_layer_sizes : The ith element represents the number of neurons in the ith hidden layer.

Activation : Activation function for the hidden layer.

Alpha : L2 penalty (regularization term) parameter.

Learning_rate : Learning rate schedule for weight updates.

Max_iter : Maximum number of iterations.

Hidden_layer_sizes	Activation	Alpha	Learning_rate	Max_iter	MAE
25,25, 25,25, 25,25, 25,25, 25,25	Relu	0.0001	constant	200	Sj = 11.05 Iq = 7.33
25,25,20,15,10,5	Relu	0.0001	constant	200	Sj = 9.62 Iq = 7.32
25, 25,25, 25,25	Relu	0.0001	Adaptive	500	Sj = 8.15 Iq = 5.68
15,15,15,10,10,10	Tanh	0.05	Adaptive	500	Sj = 29.44 Iq = 6.54
15,15,15,10,10,10	relu	0.5	Constant	500	Sj = 9.97 Iq = 6.70

Conclusion

We understood that Pre-Processing the data is extremely important to obtain accurate and desired results. It helps eliminate redundant information from the data set before training the model.

Among all the regression models we used, it was observed that Gradient Boosting technique resulted in best prediction and was also more accurate. This is because Gradient Boosting build trees sequentially one at a time where each new tree helps to correct the previously trained tree errors.