**DengAI – Prediction of Dengue Fever Occurences**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Contents** | **Page** |
| **1** | **Problem Description** | **3** |
| **2** | **Literature** | **3** |
| **3** | **Data cleansing** | **7** |
| **4** | **Data visualization** | **12** |
| **5** | **Application of predictive techniques** | **14** |
| **6** | **Goodness of fit evaluation** | **19** |
| **7** | **Conclusion** | **24** |

1. **Problem Description:**

The goal of the project is to predict the total\_cases label for each (city, year, weekofyear) in the test set. There are two cities, San Juan and Iquitos, with test data for each city spanning 5 and 3 years respectively. You will make one submission that contains predictions for both cities. The data for each city have been concatenated along with a city column indicating the source: sj for San Juan and iq for Iquitos. The test set is a pure future hold-out, meaning the test data are sequential and non-overlapping with any of the training data. Throughout, missing values have been filled as NaNs.

## The features in this dataset

You are provided the following set of information on a (year, weekofyear) timescale: (Where appropriate, units are provided as a \_unit suffix on the feature name.)

### City and date indicators

* city – City abbreviations: sj for San Juan and iq for Iquitos
* week\_start\_date – Date given in yyyy-mm-dd format

### NOAA's GHCN [daily climate data](https://www.ncdc.noaa.gov/oa/climate/ghcn-daily/) weather station measurements

* station\_max\_temp\_c – Maximum temperature
* station\_min\_temp\_c – Minimum temperature
* station\_avg\_temp\_c – Average temperature
* station\_precip\_mm – Total precipitation
* station\_diur\_temp\_rng\_c – Diurnal temperature range

### PERSIANN [satellite precipitation measurements](http://www.ncdc.noaa.gov/cdr/operationalcdrs.html) (0.25x0.25 degree scale)

* precipitation\_amt\_mm – Total precipitation

### NOAA's NCEP [Climate Forecast System Reanalysis](http://rda.ucar.edu/datasets/ds093.0/#metadata/detailed.html?_do=y)measurements (0.5x0.5 degree scale)

* reanalysis\_sat\_precip\_amt\_mm – Total precipitation
* reanalysis\_dew\_point\_temp\_k – Mean dew point temperature
* reanalysis\_air\_temp\_k – Mean air temperature
* reanalysis\_relative\_humidity\_percent – Mean relative humidity
* reanalysis\_specific\_humidity\_g\_per\_kg – Mean specific humidity
* reanalysis\_precip\_amt\_kg\_per\_m2 – Total precipitation
* reanalysis\_max\_air\_temp\_k – Maximum air temperature
* reanalysis\_min\_air\_temp\_k – Minimum air temperature
* reanalysis\_avg\_temp\_k – Average air temperature
* reanalysis\_tdtr\_k – Diurnal temperature range

### Satellite vegetation - Normalized difference vegetation index (NDVI) - NOAA's [CDR Normalized Difference Vegetation Index](https://www.ncdc.noaa.gov/cdr)(0.5x0.5 degree scale) measurements

* ndvi\_se – 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

1. **Literature:**

Literature source 1:

Regression with time series data from Forecasting: principles and practice by Rob J Hyndman and George Athana­sopou­los

Link: <https://www.otexts.org/fpp/4/8>

Inference 1:

The problem at hand is estimation of sales over a period, so, it involves data measured over a time scale and hence a time series linear model is applied to the data set to make the sales forecast. Since the number of factors affecting the sales is more than one the technique applied here is time series linear model with multiple explanatory variables.

A common feature of time series data is a trend. Using regression we can model and forecast the trend in time series data by including *t*=1,…,*T*,t=1,…,T, as a predictor variable:

*yt*=*β*0+*β*1*t*+*εt*

Literature source 2:

Stats book from Statsoft.com

Link: <http://www.statsoft.com/Textbook/Multiple-Regression#cresidual>

Inference 2:

When applying linear model to a data set, it is important to check whether the model is accurate and suitable to the data that is being used for analysis. To measure this accuracy, there are so many statistical measures that can be used. The measure applied include least squares, residual variance, correlation coefficient, residuals and r squared.

1. AICc is defined as follows:

https://www.jmp.com/support/help/images/FLM_B_Statistical_Details_65.png

where k is the number of estimated parameters in the model and n is the number of observations in the data set. This value can be used to compare various models for the same data set to determine the best-fitting model. The model having the smallest value, as discussed in Akaike (1974), is usually the preferred model.

1. BIC is defined as follows:

https://www.jmp.com/support/help/images/FLM_B_Statistical_Details_66.png

where k is the number of estimated parameters in the model and n is the number of

observations in the data set. When comparing the BIC values for two models, the model

with the smaller BIC value is considered better.

In general, BIC penalizes models with more parameters more than AICc does. For this

reason, it leads to choosing more parsimonious models, that is, models with fewer

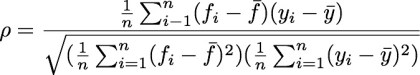
parameters, than does AICc.

1. R squared and Correlation:

In the context of predictive models (usually linear regression), where y is the true outcome, and f is the model’s prediction, the definition that I see most often is:

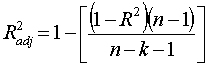
4471BBA8-E9DB-4D30-A9AE-A74F8C773247.jpg

In words, R2 is a measure of how much of the variance in y is explained by the model, f. Under “general conditions”, R2 is also the square of the correlation (correlation written as a “p” or “rho”) between the actual and predicted outcomes:



1. Adjusted R squared:

Adjusted R-Squared, R-Squared Adjusted -  A version of R-Squared that has been adjusted for the number of predictors in the model.  R-Squared tends to overestimate the strength of the association especially if the model has more than one independent variable.



1. Residual standard error:

Residual standard deviation: the standard deviation of the residuals (residuals = differences between observed and predicted values). It is calculated as follows:

Residual standard deviation

where **Y** are the observed values for the dependent variable, Y bar - Average of observed values is the average of the observed values and **Y**est are predicted values for the dependent variable. The residual standard deviation is sometimes called the *Standard error.*

Literature source 3:

Xgboost is short for eXtreme Gradient Boosting

Link: <https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html>

Its name stands for **eXtreme Gradient Boosting**, it was developed by Tianqi Chen and now is part of a wider collection of open-source libraries developed by the Distributed Machine Learning Community (DMLC). XGBoost is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed. Specifically, it was engineered to exploit every bit of memory and hardware resources for tree boosting algorithms.

First, let’s clarify the concept of boosting. This is an ensemble method that seeks to create a strong classifier (model) based on “weak” classifiers. In this context, weak and strong refer to a measure of how correlated are the learners to the actual target variable. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model. If you want to dig into boosting a bit more, check out information about a popular implementation called AdaBoost (Adaptive Boosting) here.

Now, gradient boosting also comprises an ensemble method that sequentially adds predictors and corrects previous models. However, instead of assigning different weights to the classifiers after every iteration, this method fits the new model to new residuals of the previous prediction and then minimizes the loss when adding the latest prediction. So, in the end, you are updating your model using gradient descent and hence the name, gradient boosting. This is supported for both regression and classification problems. XGBoost specifically, implements this algorithm for decision tree boosting with an additional custom regularization term in the objective function.

Literature source 3:

Ensemble methods and Random Forest Regression Model

Link: <https://www.analyticsvidhya.com/blog/2015/08/introduction-ensemble-learning/>

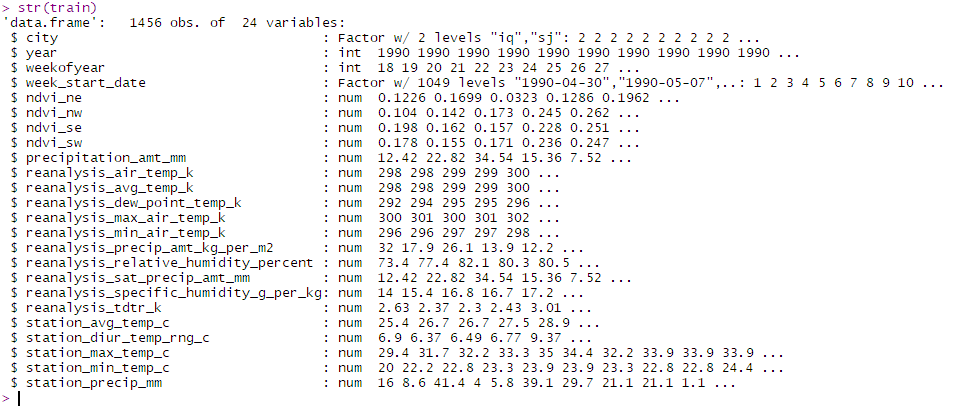
Ensemble methods are methods where multiple models are combined to improvise on the stability and predictive power of the model. It helps greatly to increase the accuracy of the resulting model with reduced bias and variance. Random Forest is one of the ensemble techniques used in Machine Learning. Random forest algorithm can be used for both Regression as well as classification. In Random Forest, multiple decision trees are constructed and combined to predict the value. Below are the high-level steps for Random Forest algorithm.

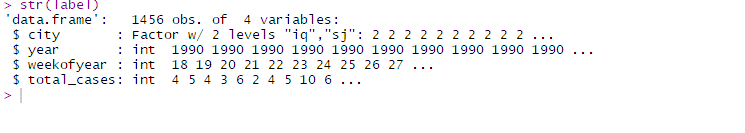
* Given dataset is subset into N number of samples with replacement
* Decision tree is constructed for each of the sample. M number of features is selected for node splitting in the trees. Algorithm considers M random features from available list of features to split the node for lowest Mean Square Error or Mean Absolute Error values. This value is kept constant for the entire model.
* Trees are allowed to grow fully with no pruning.
* In case of regression, mean or median of values in the terminal node of each tree is given as predicted value. In case of classification, mode of trees is given as predicted class.

1. **Data Cleansing:**

**Data Extraction:**

Data extracted to a data frame and summarized to for an understanding of the data and its features. This is done for both the training data and the labels provided.



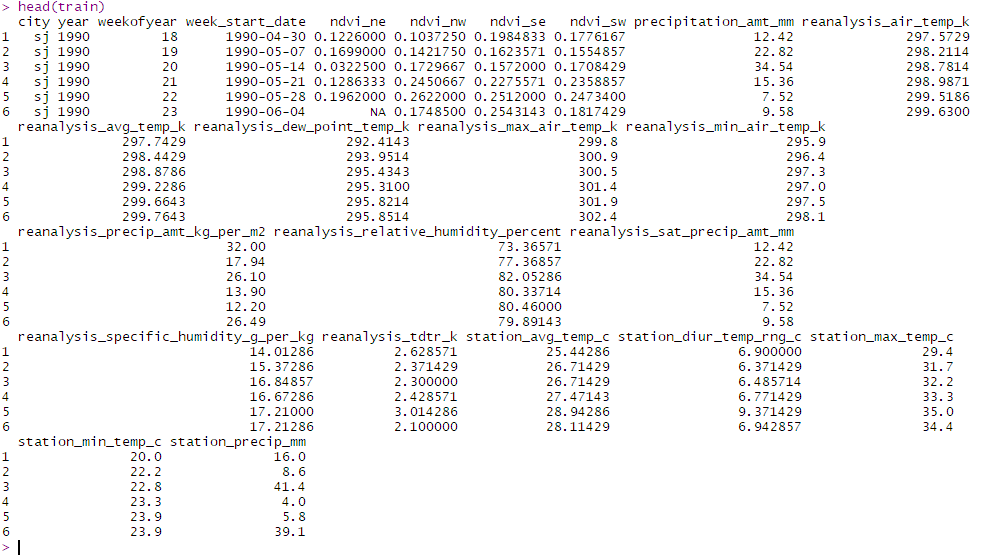


**Datatype Checking:**

Data imported from CSV file does not always have the appropriate datatypes for columns. Checking and changing the column datatypes is a vital step before proceeding with any data analysis. In this case, no observable data type conversions are required.

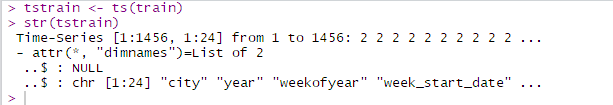
**Sorting Data:**

Since its time series data, data is sort as per date of observation of disease spreading.



**Create Time series sales data**

Convert the dataset to time series since the goal of the project is to predict occurrences of the disease over a time period and the data is a record of occurrences over a time period.

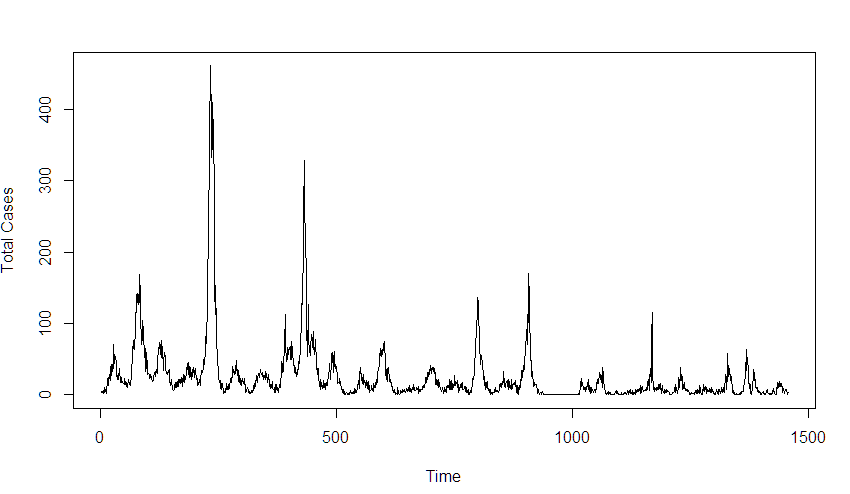


1. **Data Visualization**

Visualizing the data gives clear indication of patterns and abnormalities of data. It helps to understand the trend of data and to decide on the approach to analyze further.

**Time Series plot:**

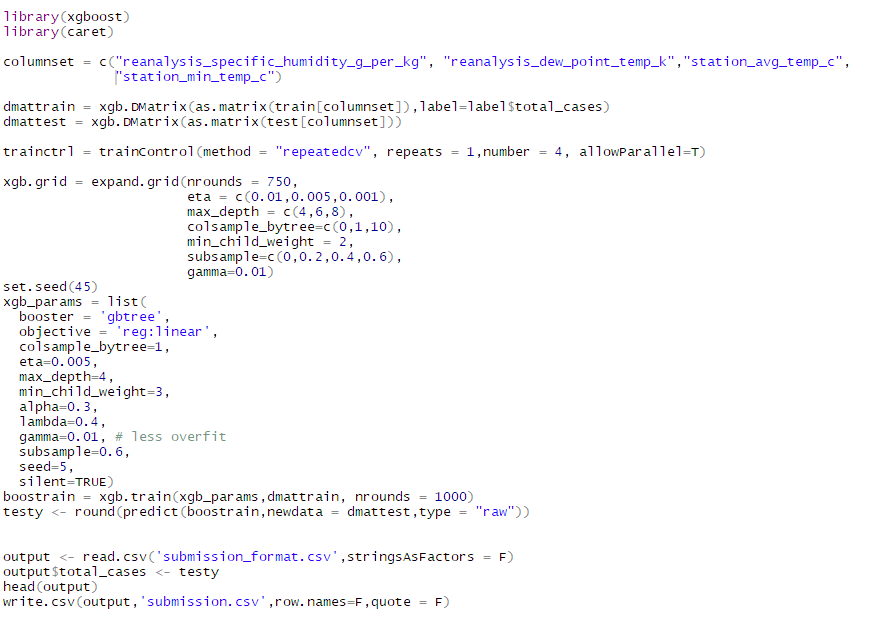


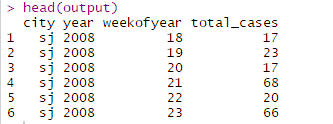


From the above plots, it can be inferred, there is no observable trend or seasonality.

1. **Application of predictive techniques**

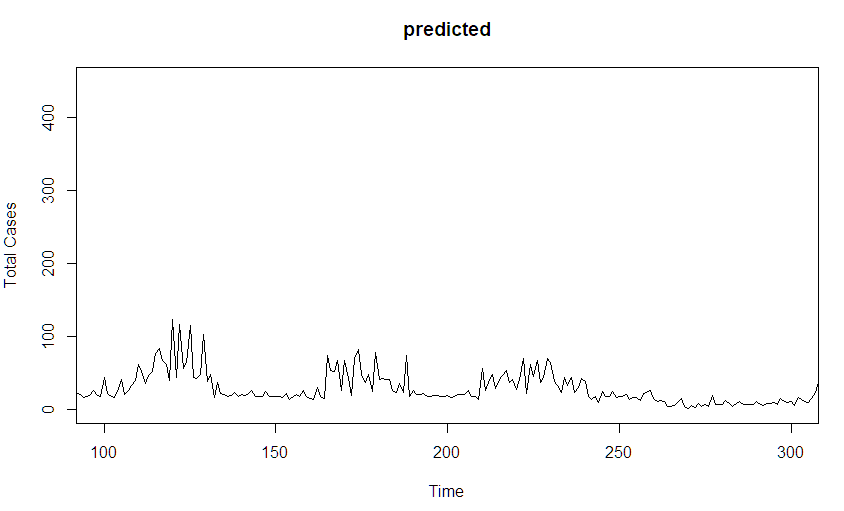
Applying boosting algorithms to the data set to predict the occurrence of the disease.





Plot of the predicted values





1. **Goodness of Fit Evaluation**

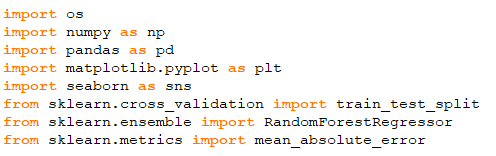
The model with xgboost algorithm implementation on evaluation has a mean absolute error of 29.2524

Model 2: Random Forest Regression using Python

As another model, Random Forest Regression is used to model the given dataset. Below is the description of data cleansing, visualization and modelling steps.

**Importing Libraries:**

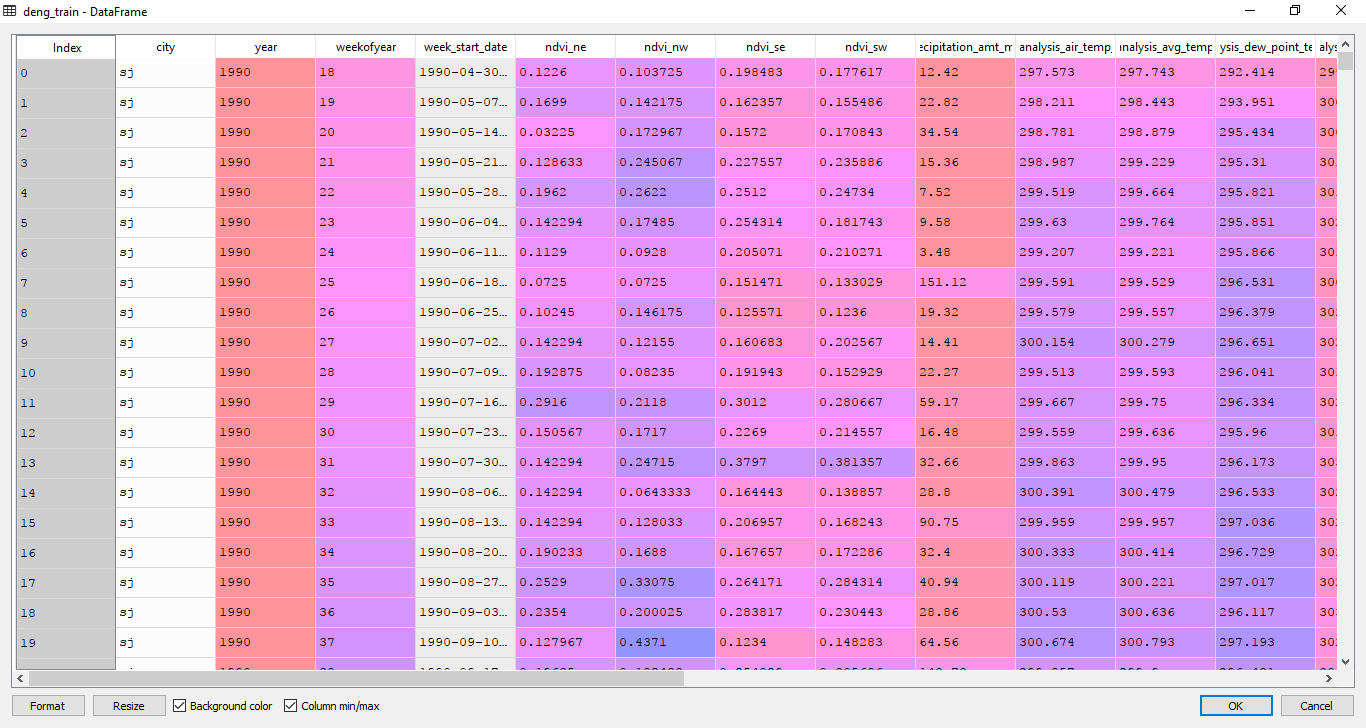
Before proceeding with the steps, it is necessary to import all the relevant libraries to do the job. In this case, Pandas and Numpy are used for data cleaning, seaborn and matplotlib are used for data visualization and sklearn for machine learning model.



**Data Cleansing:**

Features and Labels of training dataset lies in two different data files. It is extracted, and dataset of the columns are checked for appropriate data types and both files are joined as single training dataset using python Pandas Library.





**Handling NA Values in the Dataset:**

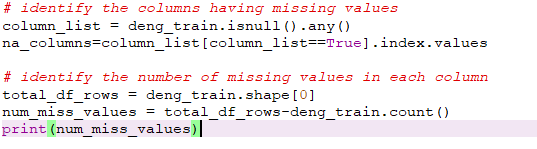
NA Values and Outliers can potentially affect the coefficients determined by the model. Lot of NA values and Outliers make our model unreliable and produce biased results. Below are few ways by which NA values can be handled.

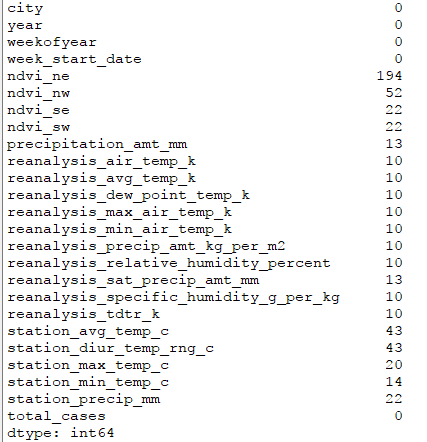
• Remove the rows that has NAs values exceeding the threshold, say 20% of values in a row.

• If the column is normally distributed, use mean of the column to replace the NA value

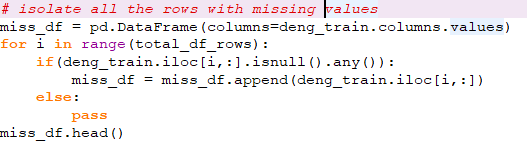
• If the column has high correlation with other column, conduct a regression between two to determine the possible missing values.

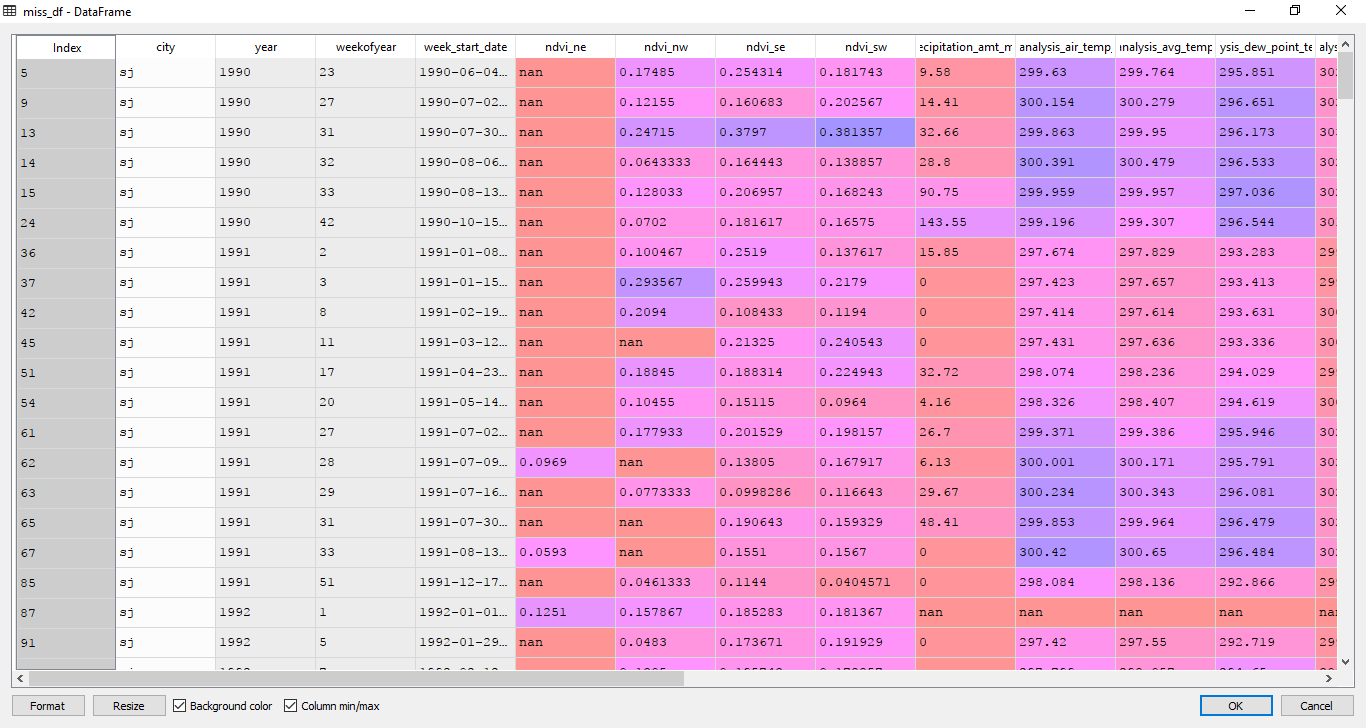
In this Dataset, several NA values are observed in the training set. Number of NA values in each column is identified as below:



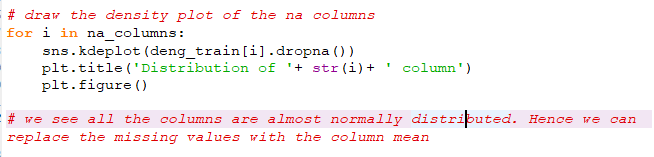


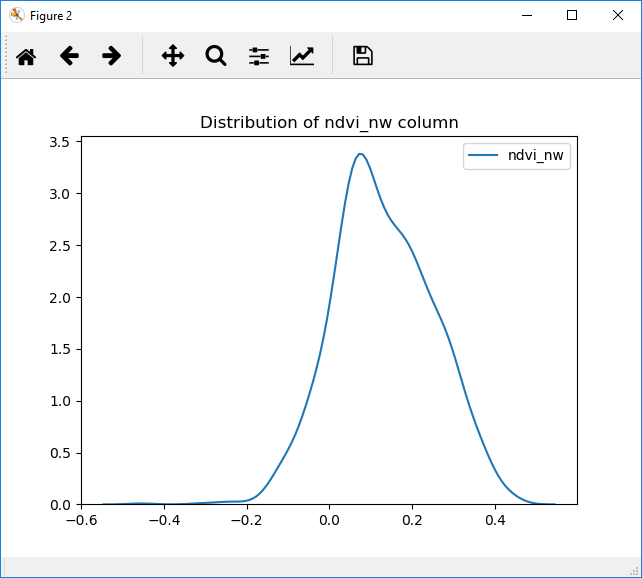
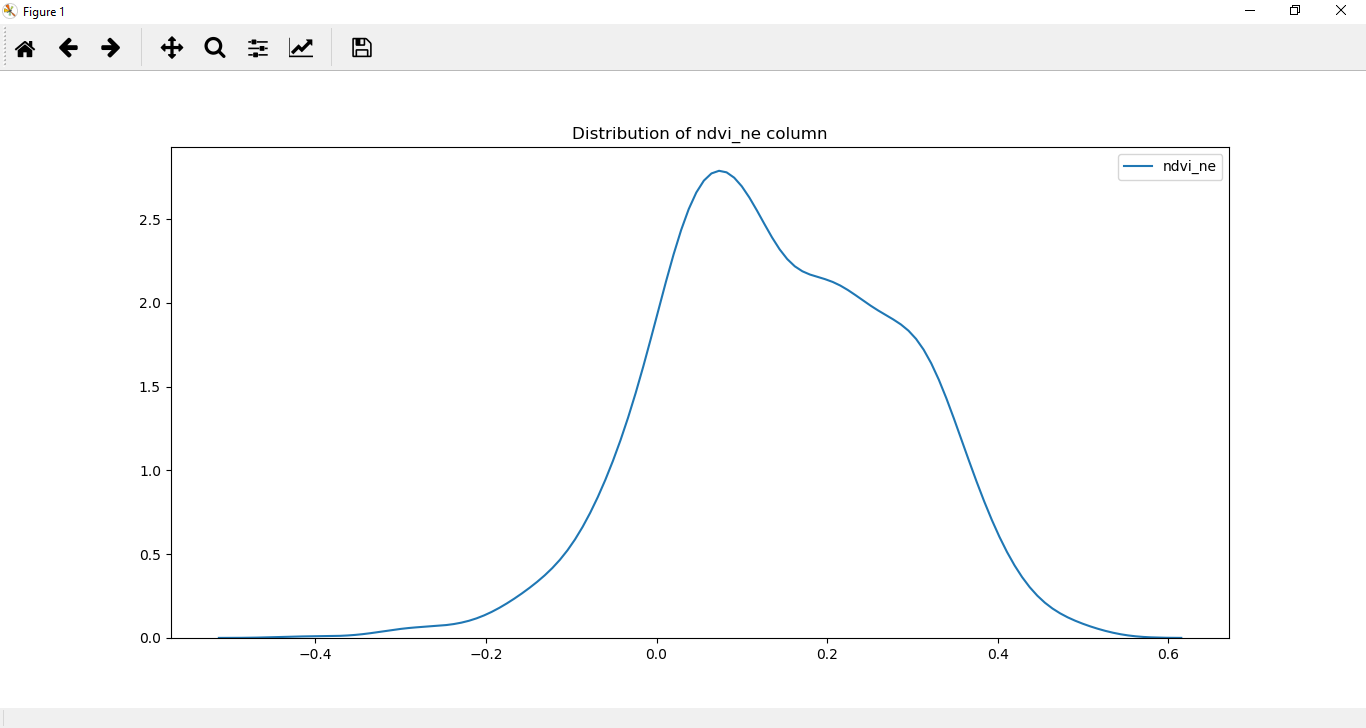
Before missing values are replaced with other reliable values, all the rows in the training dataset with NA values are isolated to separate dataframe for future verification.

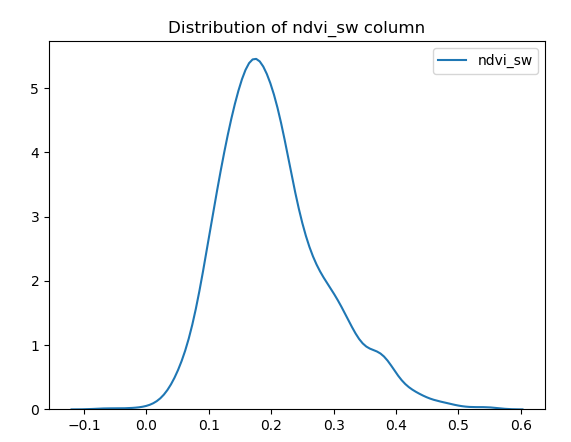
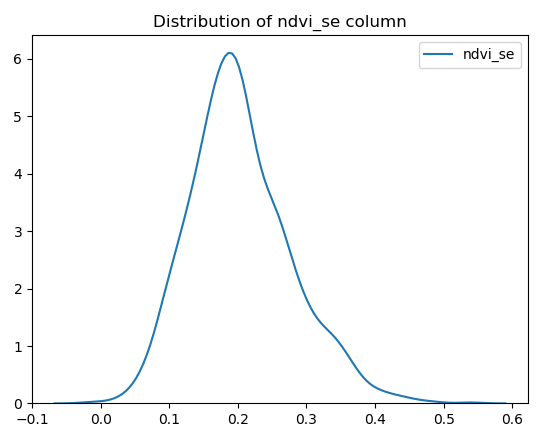




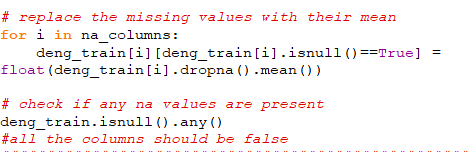
As mentioned earlier, if the column is normally distributed then missing values of that column can be replaced with the mean of the column. Normality of column is checked by plotting the density curve of each column. Below is the distribution of ‘ndve\_se’,’ndve\_nw’,’ndve\_ne’ and ‘ndve\_sw’ columns as the sample columns. Same plot is plotted for all the columns with NA values.







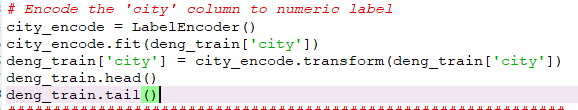
All the columns with NA values are normally distributed, hence NA values can be replaced with the mean of respective columns. Below code is used to do the same.

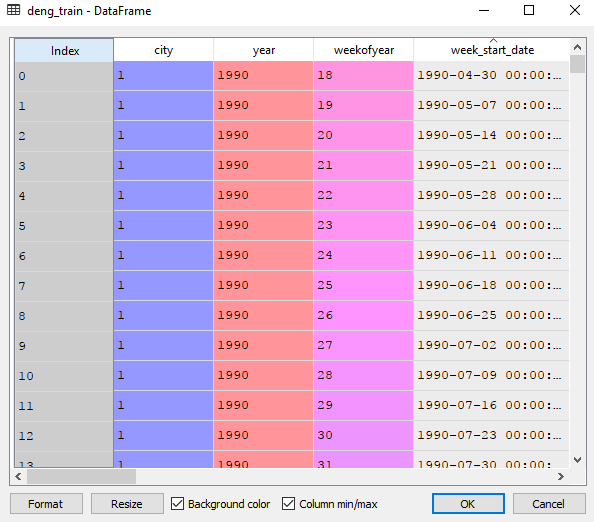


After executing the above code, all the NA values are now replaced with their respective column mean. Now the data is ready for next step which is correlation analysis.

**Label Encoding:**

‘City’ Column is categorical column. Hence it need to be encoded with numerical values before passing on the variable to the model. In python, ‘LabelEncoder’ class in sklearn.preprocessing library is used for this.

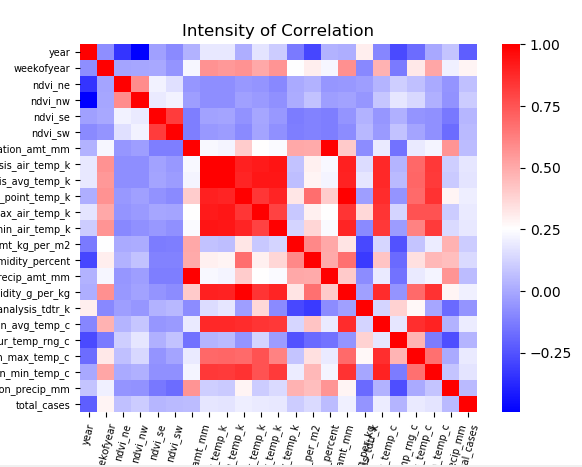




It can be observed that city column is now encoded to numerical values with 1 representing San Juan and 0 representing Iquitos.

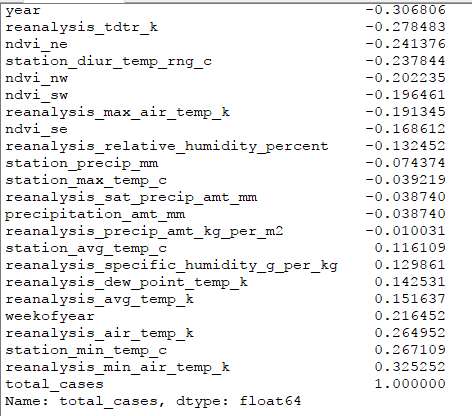
**Correlation Analysis**

Number of variables involved here is 25. Hence heatmap is used to better visualize the intensity of correlation between variables.



Actual correlation values of response term ‘total\_cases’ with other predictors is below.

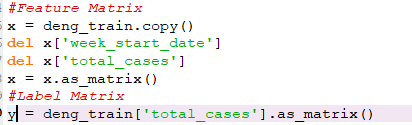




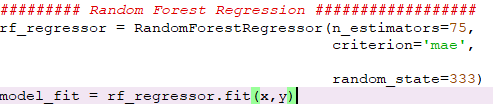
**Random Forest Regression Model**

It can be observed from the correlation heatmap and correlation values that response term has no strong correlation with any of the predictors. ‘Reanalysis\_min\_air\_temp\_k’ and ‘year’ are the two predictors with highest correlation with the response variable. This is one of the reason for choosing Random Forest Regression for this model. As it an ensemble model which follows bagging technique, model creates multiple decision trees and average of trees is chosen as the predicted value.

Training data is first prepared by removing unnecessary columns. In this case, ‘week\_start\_date’ is removed from the features and all other columns are retained. Also Training data is split into feature matrix and label matrix to facilitate the model construction.



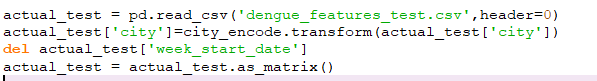
Below is the construct of the model. ‘RandomForestRegressor’ class from sklearn.ensemble library is used for constructing Random forest regression.



Here, n\_estimators and criterion are hyperparameters which are set to ‘75’ and ‘mae’ respectively. It indicates the model will create 75 decision trees with each node in each tree split for best ‘Mean Absolute Error’(MAE) value. So, while splitting nodes, model will select the predictor that gives least MAE value. Random state is used to seed the model. Model is fitted and trained with training dataset.

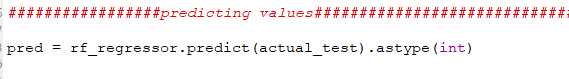
**Predicting the test data**

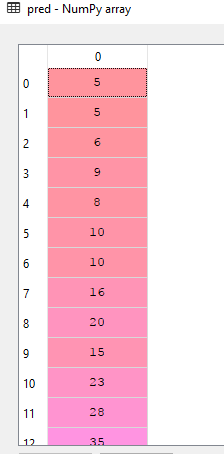
Now the model is fitted and ready for predicting the test data. Before test data is supplied to the model, it needs to be preprocessed with have the right number of features as training data and appropriate data types.



Now the preprocessed test data is supplied to model to predict the actual values.

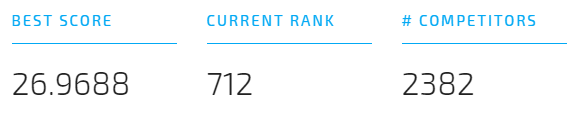






**MAE Score**

On submitting the model to the competition site, ‘Mean Absolute Error’ value for the test dataset using the random forest regression model is turned out to be 26.968.



**Conclusion**

From the two models (xGBoost and Random Forest) that are constructed for the dataset, Random forest method performs better in terms of MAE value and can be used for predicted the future observations.

**Recommendations:**

Below are recommendations to improve the efficiency of the model,

* Number of estimators in the random forest can be changed and accuracy of the model can be tested
* Before selecting hyperparameters like n\_estimators, criterion,max\_features in the model, k-fold cross validation can be done to select the best fitting hyperparameters for the given dataset
* Other complex algorithms like Support Vector Machines Regression can be tried to model the data

Below are the recommendations to improve the features in the given dataset,

* Inclusion of demographic features like population count in the area, Age group of affected people will improve the correlation between predictors and response variable
* Too many temperature measurements are given, neither of which strongly correlated to response variable. So, it is better to include only few atmospheric variables like Average Temperature, Average humidity, Average Precipitation.
* DengAi is mainly caused during winter season by the mosquitoes that feeds on the long stagnating rain water. So, Average Rainfall in the area shall be included as one of the features.