CS4622 - Machine Learning

Final Report

Project Name : DengAI

Competition Team Name : cseuom-in14-group16

Group Members :

- **Viraj Gamage (140173T)**
- **Gathika Ratnayaka (140528M)**
- **Thejan Rupasinghe (140536K)**
- **Menuka Warushavithana (140650E)**

GitHub Repo Link :

https://github.com/ThejanRupasinghe/DengAI

Final Submission source : http://bit.ly/multi-model

Introduction

This project is based on the competition DengAI: Predicting Disease Spread in DrivenData.

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

It can be seen that the transmission of Dengue fever is related to various climate variables such as temperature and precipitation. Although the relationship between transmission of Dengue with climate variables is complex, number of experts argue that climate change is likely to produce a great effect on the spread of Dengue fever.

The problem we are trying to solve is, to predict the number of dengue cases (number of patients who are caught with Dengue) each week (in each location) based on environmental variables describing changes in temperature, precipitation, vegetation etc.

The competition provides a data set containing around 1450 entries, of climate conditions and reported dengue cases in two cities; San Juan and Iquitos. The goal is to predict the *total_cases* label for each *(city, year, weekofyear)* in the test set.

Methodology

First we had to preprocess the available training and testing data. We have splitted the dataset based on city. To fill the the missing values, we have taken several approaches. One approach was to fill the missing values by considering highly correlated redundant features (for example: *station_avg_temp_c* and *avg_air_temp_k*). After experimentation, we recognized that filling the missing values with mean value of the differences would improve the results rather than using options like forward fill, backward fill or ignoring the missing values. As normalization method, we have transformed each value into a z score. The reason behind normalization is to make training data less sensitive to the scale.

The next requirement has been to do feature engineering. Rather than choosing only the features which are highly correlated with the total number of cases, we have plotted the graphs for each feature value including total cases. There we could identify hidden patterns between features as well. One such pattern we identified was features like dew point temperature, specific humidity, min/max/average temperature peak every 52 weeks or around 52 weeks similar to the total number of cases in San Juan. Though that is the case, direct correlation was not shown as the quantitative change of these features are not proportional to the total number of cases. Therefore, in feature selection we considered correlation as well as hidden patterns between features. We have created new features using moving window average (window size = 52) for features like precipitation to capture the trend over that period of time.

After all, we had to select a suitable machine learning algorithm and evaluate the results. For evaluation, we have splitted the dataset as training and cross validation data set randomly. First, we have trained are tested on several machine learning models; linear regression, support vector regression (SVR), gradient boosting and random forest regressor. We got better results from gradient boosting and random forest regression for San-Juan. SVR and gradient boosting performed better for the city Iquitos. We selected those models for further processing. Next we have used GridSearchCV to tune the hyper parameters of each of those selected models. Then we have ended up with gradient boosting model for San-Juan and SVR with rbf (radial bias function) kernel for Iquitos.

Results and Analysis

Results obtained using different regression models are shown in Table 1.

Model	MAE for San Juan	MAE for Iquitos
Linear Regression	26.987	6.539
Support Vector Regression (kernel=’linear’)	22.792	5.686
Support Vector Regression (kernel=’rbf’)	21.810	5.617
Gradient Boosting	19.491	5.726
K-Nearest Neighbour Regression	26.482	6.521
Random Forest Regression	19.800	6.385

Table 1

Table 1 shows the mean absolute error for the cross validation set we have splitted. We have not used a neural network model because the available data set is less than 1500 records. We can observe that the linear regression fails. One possible reason might be that the correlation between each individual feature and the total_cases were less. Therefore we can say that the relationship between those features and total_cases are non linear. Another possible reason might be the correlation between selected features. We can observe that the error in Nearest neighbour regression algorithm is also high because of the high dimensionality. Gradient Boosting and Random Forest Regression models perform ahead of other models possibly due to ability to learn non-linear relationship and robustness to outliers. Surprisingly Support Vector Regression with 'linear' kernel performed better for Iquitos. That observation might be because of the Support Vector Regression models does not change its model parameters unless the threshold is met. By observing the above results, we proceeded with Random Forest Regressor and Gradient Boosting Models for San-Juan and Support Vector Regression and Gradient Boosting model for Iquitos and tested the results with hyper parameter tuning using GridSearchCV. The winner for San-Juan was Gradient Boosting model with mean absolute error of 16.1. For iquitos, best results were achieved by further tuned Support Vector Regression model having 'rbf' kernel, with the mean absolute error of 4.8. When we have submitted the predictions to the competition using those models, we ended up with a mean absolute error of 19.2 and the rank was 67 at the moment (out of 3700 competitors).

Conclusion

In this section, we will summarize the important points we need to state on this project. First, we could identify the importance of the preprocessing step. After experimentation, we could find the best way to fill missing values is to fill with the mean of the column when there are no redundant features. It outperformed other methods like forward fill, backward fill and interpolation methods and it resulted an improvement close to 1 with respect to mean absolute error. When there are redundant features, it is possible to make use of it and improve accuracy of the prediction further (in our case, we could reduce mean absolute error by around 0.5).

When it comes to feature selection, first we have just analyzed the correlation matrix containing correlation between each feature including the total cases. At first, we have taken just four features such that those features have less correlation with each other but high correlation with total cases. As the highest correlation between a feature and total_cases is around 0.25, the model failed to perform better. Then we referred to plotted graphs and discovered that there is a common pattern in occurances of peaks in certain features. That is another new experience we have found. Another discovery was that taking the trend for features like precipitation could improve results in this kind of scenarios. The reason is that if we closely observe there is a trend for dengue cases over period. If we consider trend of another feature, that introduced feature could help to identify the trend of dengue cases. When selecting a model, there is common saying that decision trees (specially gradient boosting) performs better. But we could identify that SVR with rbf kernel performed slightly better (0.4 improvement in mean absolute error) for iquitos. So it is always better to try few models rather than relying on one model.