Dengue Fever Infection Problem

Stefano Galvagno
Politecnico di Torino
s290191
s290191@studenti.polito.it

***Abstract*—This report describes a possible approach for the Dengue fever weekly infection prediction. The main steps of the process will be illustrated and discussed in order to show an overview and a possible solution for the problem.**

I. Problem Overview

This competition deals with the prediction of weekly Dengue infections in two cities located in Central America: San Juan (Porto Rico) and Iquitos (Peru). Dengue fever is a tropical infectious disease caused by the Dengue virus, transmitted by mosquitoes of the genus Aedes. Consequently, characteristics of the environment are fundamental to determine the proliferation of this viral vector. The dataset counts 1.569 entries and it is characterized by numerical, categorical and temporal attributes. In particular it is subdivided into two subsets:

- Development set: 1205 entries, 20 features, including the total amount of infections used to train the model.
- Evaluation set: 364 entries, 19 features, the data which has to be considered for the prediction.

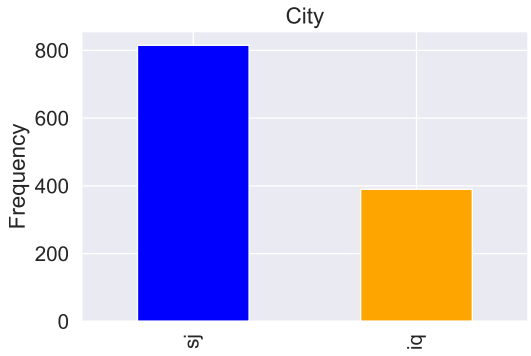
Most of attributes are characterized by *numeric measures* such as the temperature (both in Kelvin and in Celsius), humidity and the amount of precipitations. These data are gathered from different sources: a weather station, PERSIANN satellite and NCEP Climate Forecast System Reanalysis and describe the natural environmental conditions in which the spread takes place. In most of the cases they are normally distributed (except in the case of NCEP_precip_kg_per_m2, NCEP_humidity_percent and precip_mm). Furthermore, there is some *temporal information* about the date, the week and the year. To conclude, the *city*: San Juan and Iquitos, two cities located in different countries and climate areas; the former at the Tropic (inland), the latter at the Equator (island).

II. Proposed approach

A. Data preprocessing

The pillar of the whole process is the distinction in terms of analysis for the two cities. As we can see in Fig. 1 there are many more records from San Juan than from Iquitos. Furthermore, inspecting the distributions of features in the two locations we can observe not negligible differences. Fig. 2 shows a relevant difference in terms of total cases (Negative binomial shape for both cities): leaving out the frequencies, San Juan suffered from many more critical infections. For instance, considering Iquitos, 30 cases can be considered a threshold for eliminating outliers but in the case of San Juan the same value represents one of the peaks. In addition, Fig. 5 shows some evident differences both in terms of mean and variance for the same feature in different cities. These aspects justify the choice of two distinct models for two cities. In other words, the approach consists in a couple of regressors to evaluate separately data from each city.

Fig. 1. Records for each city



Missing values: only numerical attributes present some missing data. Since all the records are part of an historical collection that covers a period of almost twenty years, it is reasonable to impute all these NaN values through a forward fill. Actually, a better alternative in our case is linear imputation, which showed an improvement in the public score of 0,1 in terms of MAE. *The outlier detection and removal* is implemented by means of the IQR (Inter Quartile Range) approach, considering as bounds the first and third quartile. Through a manual inspection of outliers, we can note that these values are not errors but actually rare observations. Consequently we will not exclude all of them, but just the ones about total cases.

The only additional feature is the month that has been obtained from week start date attribute. As previously mentioned, the temporal aspect in this project is crucial. Fig. 3 displays the overall trend: both the locations present some seasonal fluctuations. Particularly, Fig. 4 clearly shows that the second half of the year is the most critical one in both the cases. As expected, since the two cities are located in different climate areas, the global maximum and minimum are reached in different months but anyway we can observe a common trend, with a two months delay for Iquitos. However, creating

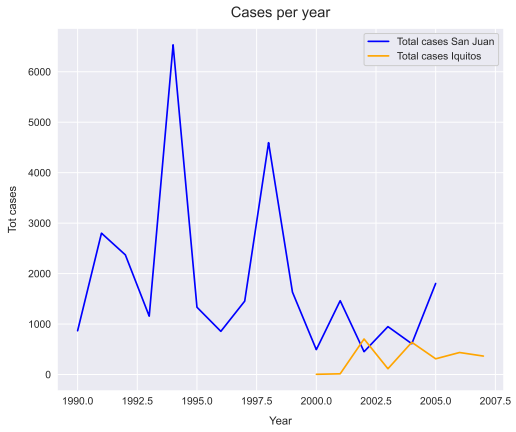


Fig. 2. Cases per year

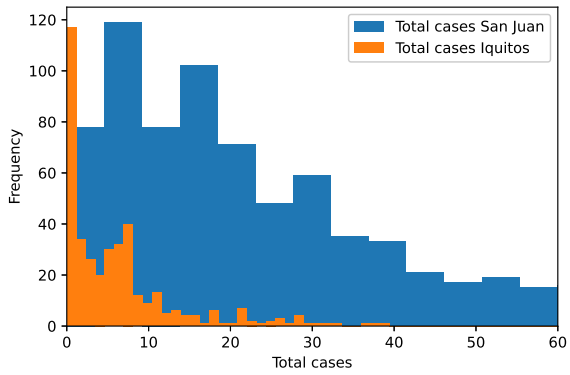


Fig. 3. Total cases

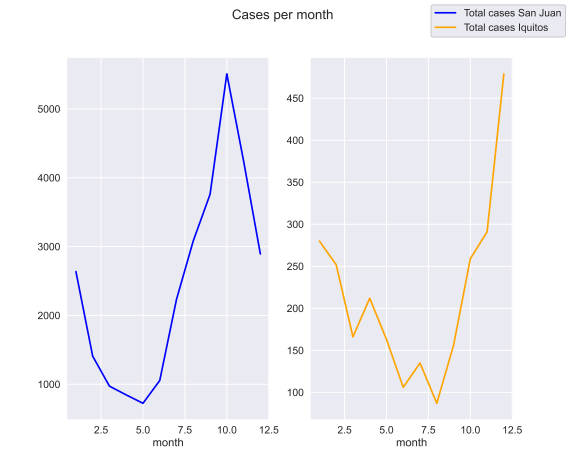


Fig. 4. Cases grouped by month

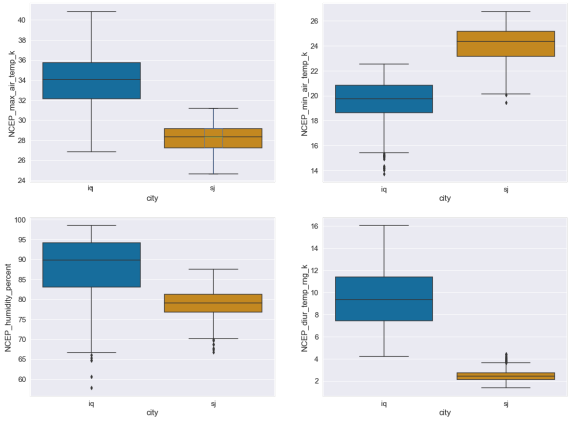


Fig. 5. Most relevant differences between SJ and IQ

a new feature denoting the semester proved not to be useful to improve the quality of the model.

Concerning *dimensionality reduction*, we can disregard some features that are linearly correlated. In particular we can exclude the ones that present a Pearson correlation greater than 0.9 in terms of absolute value.

Usually, models benefit from a scaling operation; in our case the best option proved to be MinMaxScaling, that allowed to improve the performance with respect to StandardScaling. Moreover, temperatures expressed in Kelvin have been converted in Celsius in order to obtain a coherent unity of measure. No duplicate records are present.

B. Model selection

The tested models:

- Random forest: this algorithm is interpretable so we can benefit from the feature importances in order to draw some conclusions (i.e. in our case about overfitting due to year). In addition, like most of the ensemble models, it is one of the best performers.
- AdaBoost Regressor: adaptive boosting is frequently used to improve the random forest performances. In particular it is based on a combination of weak learners, forest of stumps (decision trees with just 1 level). Differently from random forest, trees have different voting importances and the order in which they are built matters (this means that the errors committed by one of them will affect the way the following are made). Moreover this model works by putting more weight on wrongly predicted instances and less on those already handled well.
- KNeighbors Regressor: standard KNeighbors model where the predicted value is the average of the k nearest points in the training set.
- Ridge: common regularization model that aims to assign values close to zero to those coefficients associated with features that are not relevant in the regression.
- SVR: a commonly used model that exploits the division by means of hyperplanes to make predictions. One of the best performers.

As previously mentioned, exploiting the Random forest interpretability we observe that the year plays a fundamental

role in the prediction Fig. 6. However, this feature is the main

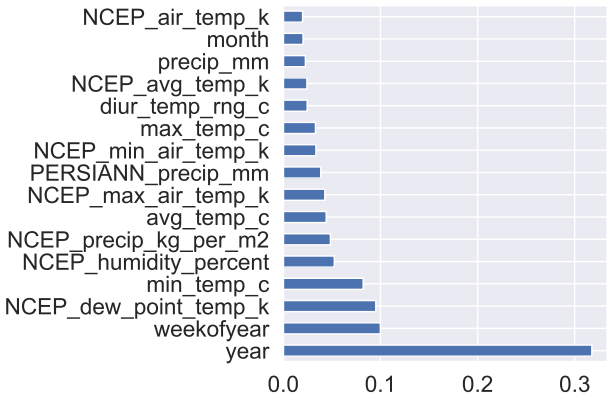


Fig. 6. Feature importances

reason of overfitting (since the evaluation set includes more recent data as well). Aware of this aspect, we will maintain the year since it improves both private and public results. Clearly, excluding the year, the gap between local and public results is smaller.

C. Hyperparameters tuning

From now on, despite the fact that the following approach is **not correct** in a general case, we will consider only the predictive model for Iquitos. This choice is justified by an inspection of the evaluation set that proved to contain only data from the Peruvian city. In general, we do not look at the test set, unless for checking if the sample has a similar distribution with respect to the one used during training (a posteriori, to investigate a possible reason of poor generalization of the model). Rather, we would perform a distinct fine tuning and merge the predictions from both the models. With this premise, the evaluation task is performed by running a grid search on a 80/20 train/test split on the filtered development set, looking for the best combination of parameters reported in Table I. The fine-tuning process involves a cross-validation approach with five folds.

III. Results

The results achieved by the Random forest and the AdaBoost are similar in terms of MAE (3.132 / 3.367) and outperform KNN, Ridge and SVR (4,662 / 4,854 / 4,284). However, the best model in terms of ability to generalize proved to be KNN, obtaining the highest public result with 5,610. The best configuration for KNN regressor is 'metric': 'manhattan', 'n_neighbors': 20, 'weights': 'distance'.

IV. Discussion

The proposed approach achieves satisfactory results with a reasonable time complexity. The main issue of this competition is dealing with overfitting since the training set is quite small. Even applying pre-pruning strategies on Random Forest the problem persists. In this sense KNN proved to be robust to noise and a good regressor. In order to boost the results, some alternative approaches are possible:

- Applying time series forecasting models like ARIMA,
- Recurrent Neural Networks
- A combination of ARIMA and one of the models presented so far

Moreover, analysing more in details the sources from which data are gathered could be effective if some of those prove to be, on average, less accurate because of several reasons (e.g. not using state of the art technologies to record temperature). Similarly, a more in depth study of the distribution of the features could lead to discovery hidden interesting patterns. To conclude, the choice of building a separate model for each city proved to be extremely effective: each model built on the unique data set generates a MAE that is at least double than the one that we obtained.

TABLE I
HYPERPARAMETERS

Model	Parameter	Values
Random Forest	max_depth, max_features, min_samples_split, min_samples_leaf, bootstrap, n_estimators	[2, 4, 6, 8, None], [1, 3, 10], [2, 4, 10], [1, 3, 10], [True, False], [100, 250, 500]
AdaBoost	n_estimators, learning_rate	[500,1000,2000], [.001,0.01,.1],
Kneighbors	metric, n_neighbors, weights	['manhattan', 'chebyshev', 'euclidean'], [5, 10, 20], ['uniform', 'distance'],
Ridge	normalize, alpha, tol, solver	[True, False], [1, 0.1, 0.05], [1e-3, 1e-6], ["auto", "svd", "cholesky", "lsqr", "sparse_cg", "sag", "saga"]
SVR	kernel, max_iter, tol	[poly, rbf, sigmoid], [5000, 10000], [1e-3, 1e-4]

TABLE II
FINE TUNING

Model	Best Config	MAE
Random Forest	bootstrap: False, max_depth: 6, max_features: 10, min_samples_leaf: 3, min_samples_split: 2, n_estimators: 500	3.132
AdaBoost	learning_rate: 0.001, n_estimators: 500	3.367
Kneighbors	metric: manhattan n_neoighbors: 20 weights: distance	4.614
Ridge	alpha: 1, normalize: True, solver: sag, tol: 0.001	4.854
SVR	kernel: rbf, max_iter: 5000, tol: 0.001	4.284