Final Task for Machine Learning.

Eduardo Sánchez López
José Alejandro Libreros Montaño

Github: https://github.com/Edusanc95/MachineLearning
DrivenData username: Edusan

Introduction

We take as a base the past tasks that are uploaded in our github repository, please check it for past results and how we got to this state.

In this task we tried to obtain better results in the DrivenData competition by using different algorithms and changing some of our strategies. Some of the most important results are in the document annexed to this one called "results".

We used a local MAE calculator to check how close we are to the real prediction given by the DrivenData website, taking 60% of the original training data as training and the other 40% as test, which is not ideal since it has a high degree of randomization.

KNN

First we went back to the feature selection part, we thought that maybe adding new features to our predictors will help, for that matter we checked the correlation between the features and the target and selected the most correlated (or negative correlated) ones. It was not long that we observed that adding features basing on the correlation between the features and the total cases ups the error by little. Just making a KNN predictor with the features that are relevant in the decision tree was giving the best results.

Random Forests I

In this step we made an error, which was making a Cross-Validation test for the Random Forest. The depth that it gave us was 2 for both Iquitos and San Juan. After some local tests we believe that Random Forests with low depth are not a good model for this particular case, mainly because is too generic and the error increases too much. This specially important in San Juan, which has a lot of non-regular spikes. Iquitos data is more prone to overfitting exactly because the reverse of that statement, so maybe a low depth is a good option still. We will come back to this topic.

Bagging and Boosting

In Bamboo.py, it has been used the sklearn BaggingRegressor, with KNN as base estimator.

It has been used also the GradientBoostingRegressor, with the MAE criterion and the deep given by deep decision tree, according to each city, and a random state.

With BaggingRegressor, it was obtained a score 25.9952 and with GradientBoostingRegressor, it was obtained 26.8582 error percentage.

Naive Bayes

We knew from the start that this method wasn't going to obtain us a better result, but since we wanted to know how it works we tried anyway.

Gaussian Naive Bayes didn't work correctly since our data is not in a normal distribution and gave us a really bad result.

For science we submitted a result with BernoulliNB and we got a 35.0625, which makes sense because of how this method works, since it only estimates if an event occured or not, it’s not good for regression. In this case, it only predicted 3 and 6 for total_cases, which is obviously wrong.

Random Forests II

After studying this other methods we came back to this method, but instead of making a Cross-Validation test and believe blindy what it says, we used out own logic and criteria for the parameters.

We thought that Iquitos had a more balanced and regular data once looking at the graphical representation, also because it’s the dataset that got more overfitting in the past tests. Also we got the reference of the CV tests for the Decision Tree. With that reasoning we started with 3 depth for Iquitos and 5 for San Juan.

The best result we got so far was with RF with max_depth 3 in Iquitos and max_depth 5 in San Juan with a 25.2572. Note we didn't take into account station_avg_temp_c for Iquitos in this case.

We came back to the feature selection done in task 5, mainly adding "dew_point" to Iquitos, and making that the RF chooses automatically the depth and with 10000 trees. We obtained a 26.2212, a much worse error.

We imagine this happened because for this problem having too many trees and not a specific depth.

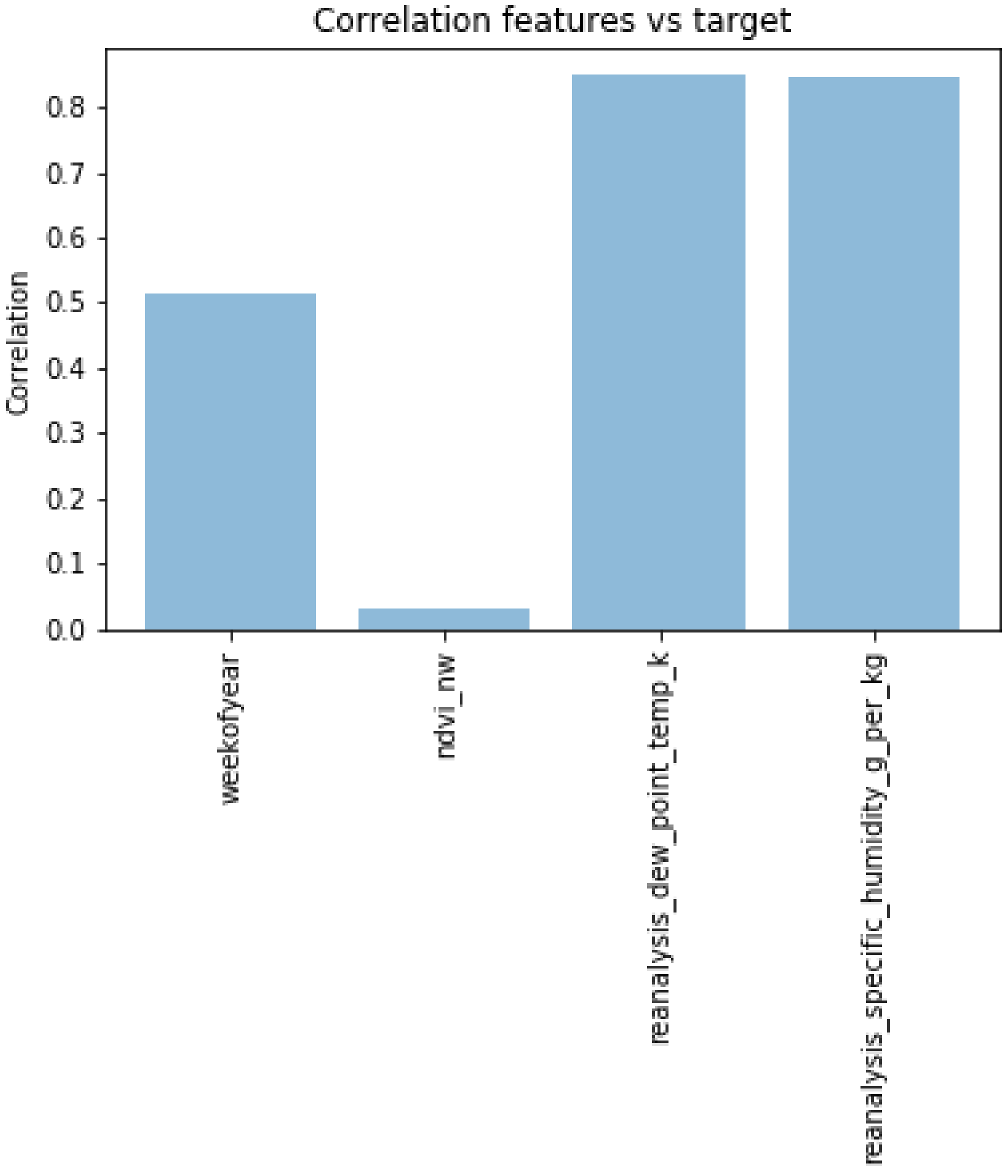
We came back to 3 for depth for Iq and 5 depth for SJ and also dropped up to 1000 estimators, for this we got a 25.3005, a very similar error to our first one. We upped the depth of Iquitos to 5 and got a 25.3413.

The best result we got in the entire competition was a 24.9375 that we got with this features and with San Juan depth as 10, and Iquitos depth as 3. From this we toyed with the depths, but came to the conclusion that those were the best depths.

From this we can now affirm our theory that the cases in Iquitos have less outliers and the "curve" is smoother and the overfitting is way worse than in San Juan.

Since we hit a roadblock with this theory, we decided to go back and check once again the feature selection. First we studied if it was possible to delete features in San Juan, in which we have 5, we started studying the least relevant feature in the Decision Tree test, which was “station_min_temp_c”.

For that we studied the correlation with the other selected features, if it was too correlated to any of those it will be an easy ditch, as we can see in this graph.



For our surprise this upped the error up to 25.6779, so we tried again keeping “station_min_temp_c” but deleting “reanalysis_dew_point_temp_k”. This ALSO increased the error up to 25.8245.

We then realised that we still had the sj depth as 10, which didn’t represent the current state for this feature selection, so after doing our own local MAE tests, we observed that 5 gave the best result.
Still, the error went up to 25.8654. This was specially surprising because with these features and depths, the local MAE error was the lowest we have seen so far, going as low as 19.89, while with other parameters we usually obtained around 25. Granted is somewhat random and the same test should be iterated multiple times, but we think it was worth noting.

Conclusions

The best result we got was 24.9375, obtained through Random Forests. The other techniques studied didn’t help as much, but that was because they are suited for other types of problems. With more time we wanted to give bagging and boosting more time and study other techniques. Also we think that we could have made a better job at selecting features, although the final results are not too bad.

In the previous task we talked about some of the improvements that we could make, doing the mean of the 3 closer rows instead of the mean of the whole feature and rounding it up or down the final number of cases instead of just rounding it down always. Well, the tests done with these changes didn’t provide better results, in fact it upped our error by 0.3~ in every single test.
A theory for this phenomenon is that doing the mean of the 3 closer values of the column makes the data distribution way more lineal than it should be and that our predictions have already higher values than it should be, so rounding them up doesn’t help with that.

Something very interesting was that using our local MAE estimator, we saw that the error was much higher in San Juan, usually Iquitos rounded the 6-10 mark, and San Juan the 25-30 mark, so if given more time we would study San Juan separately in more depth. Perhaps a good approach would be split San Juan in stations (with the feature *week_of_year*, like we divide the original dataset into Iquitos and San Juan with the feature *city*) so that way we can treat the data in a more granular way.