Milestone 2 – Supervised Learning

Maciej Nalepa
maciej.nalepa@alu.uclm.es

Piotr Maliszewski
piotr.maliszewski@alu.uclm.es

Gökay Iseri
gokay.iseri@alu.uclm.es

Başar Milli
basar.milli@alu.uclm.es

Abstract

The purpose of this work is to train a regressor model on the environmental data collected by various U.S. Federal Government Agencies from two cities (San Juan, Puerto Rico and Iquitos, Peru) to gain a better understanding of the Denge Spread Phenomena. These data are from a competition of the site DrivenData. The overall objective is to use supervised learning techniques to predict the number of cases on the basis of the collected data and create a model that will successfully predict them. Training data will be used to acquire a model and test data will be used to generate submissions and evaluate score among other competitors.

CCS Concepts

• **Computing methodologies** → **Machine learning**.

Keywords

machine learning, supervised

ACM Reference Format:
Maciej Nalepa, Piotr Maliszewski, Gökay Iseri, and Başar Milli. 2021. Milestone 2 – Supervised Learning. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In this project we explore supervised Learning methods. We were working on DrivenData DengAI: Predicting Disease Spread competition dataset. The whole dataset was used to achieve possibly best prediction results. However, there is an option to limit the data to the subset that was initially assigned to our working group.

1.1 Project files

The code and source files are available at https://github.com/GummyBear/ESI_MLTechniques.

2 Data Preprocessing

The data was preprocessed by filling NaN values using backward fill method, later scaled with a standard scaler. For experimenting PCA was applied to generate alternative version of the dataset.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

city	mean	var
sj	34.18	2640
iq	7.57	116

Table 1: Labels mean and variance values.

Distribution of the labels which is the *total_cases* value is highly unbalanced (1). Its mean is much lower than the variance. To potentially improve performance of linear models a logarithmic transformation has been applied to these labels in order to acquire a distribution closer to a standard distribution (2).

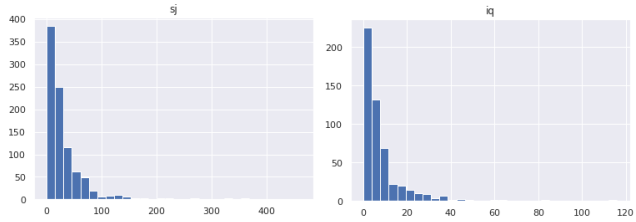

Figure 1: Distribution of the *total_cases* before transformation.

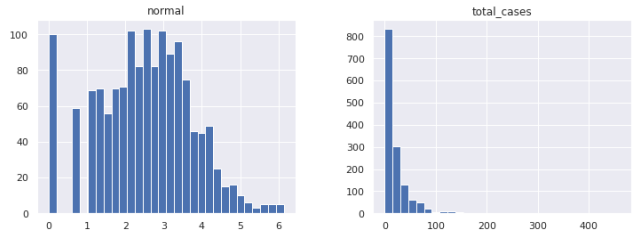

Figure 2: Distribution of the *total_cases* with logarithmic transformation (left-hand side).

Basic analysis steps have been performed to select best features and clean the dataset from highly correlated variables. Variables with absolute correlation higher than 0.8 were paired and marked for removal. The removed variable from each pair was chosen by its importance which is the correlation value with the label (4).

Importance analysis was performed also on the PCA transformed dataset. The mean importance of the cleaned features was 0.145, while the PCA features scored around 0.062. This can result in the models trained with PCA as input to perform worse than default ones.

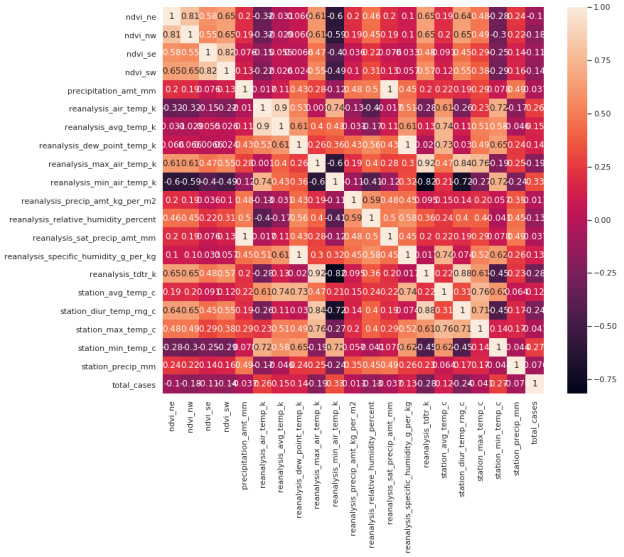

Figure 3: Data correlation matrix. Note that some variables have correlation reaching 1.

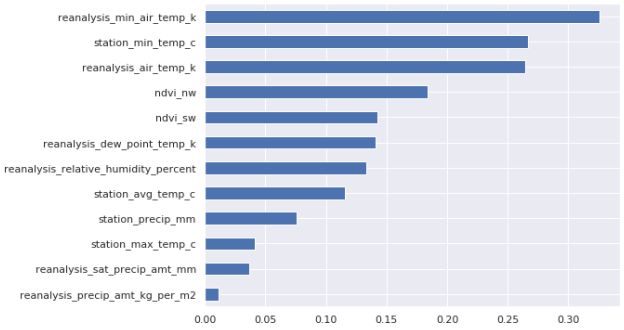

Figure 4: Cleaned features importance values.

3 Models

The goal is to find most robust regression model that will successfully predict the values on the unlabeled test data. We propose an approach of training 2 separate models for the city of San Juan and Iquitos as they show much different characteristics. What is more, we split the data for training and testing with the ratio of 15% for testing. Which is supposed to help determine a better model for prediction against the target unlabeled data.

Each training can be easily configured to run on different data in the notebook. Possible changes are the city, transformed or non-transformed labels and PCA transformed features. There is also an option to drop a set amount of least important features.

Following regression models have been selected to predict the values of our data:

- k-Nearest Neighbors – which serves as a baseline model
- Decision Tree
- Random Forest
- Linear Support Vector

We are also using the boosting models which are:

- Ada Boost
- Gradient Boosting

The metric score used was Mean Absolute Error. For obtaining a quality score of each model a fold on the training set was applied and maximum MAE was returned as the score of the model.

3.1 kNN

KNN regressor can use different weights during training. We tested its performance with default weight functions *uniform* and *distance* settings. For each branching system was tested checking consecutively next values of *n_neighbour* which is the number of neighbours to use in KNN algorithm. Performance of this model was our baseline which we try to improve.

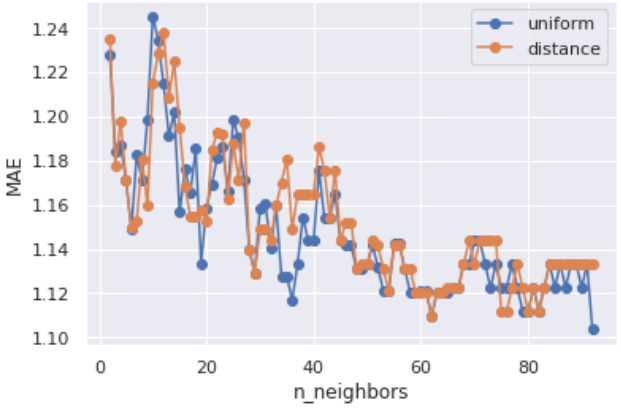

Figure 5: MAE of kNN. (low MAE values are the result of using transformed labels)

3.2 Decision Tree

The most important parameter for a Decision Tree regressor is the maximal depth of the tree. We tested its performance in the range between 2 and 40. Greater numbers haven't been decreasing the error.

This model can be an improvement over kNN.

3.3 Random Forest

We have been checking various values of *max_estimators* parameter and various *max_depths*. Values of max_estimators have been checked in the range from 2 to 7, while max_depth varied from 2 up to 20.

This model can be an improvement over kNN.

3.4 Linear SVR

Another tested regressor is Linear SVR which was optimized by tuning its regularization parameter C. The value was analyzed in the logarithmic space in the range between $10^{-5}; 10^{0.5}$.

Generally this model turned out to be one of the least effective.

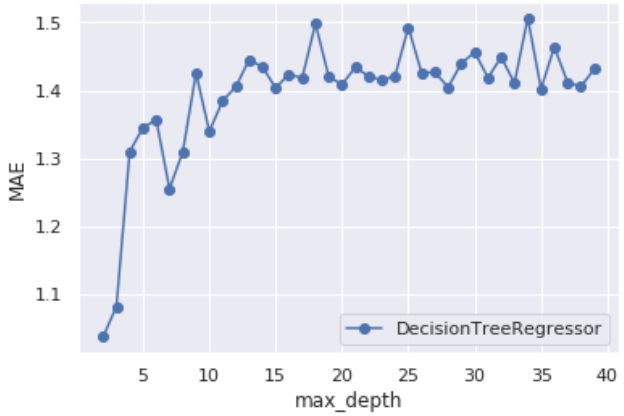

Figure 6: MAE of Decision Tree.

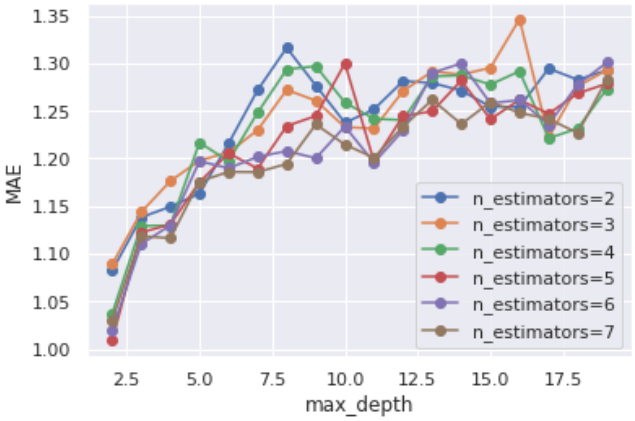

Figure 7: MAE of Random Forest.

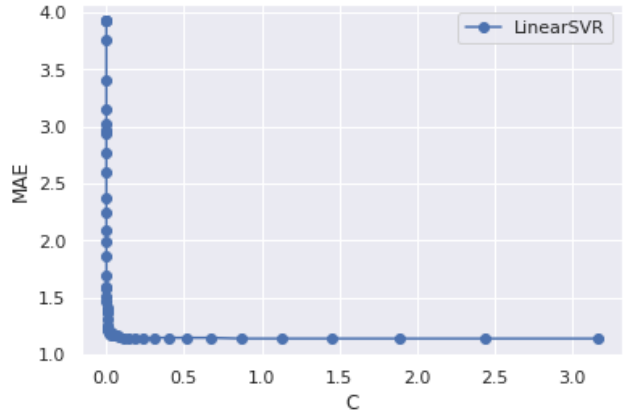

Figure 8: MAE of Linear SVR.

3.5 Visualisation

Each model has its visualisation, which is a plot showing the predicted and actual values.

Figure 9: Example prediction visualisation on a test set.

4 Optimization

4.1 Boosting

Ada Boost with decision tree of depth 3 and Gradient Boosting are the two boosting regressors chosen for further experimenting. Feature importance were measured on these two models and compared with default Decision Tree and Random Forest.

Models with boosting have the distribution of importance more evenly distributed in comparison to the other two.

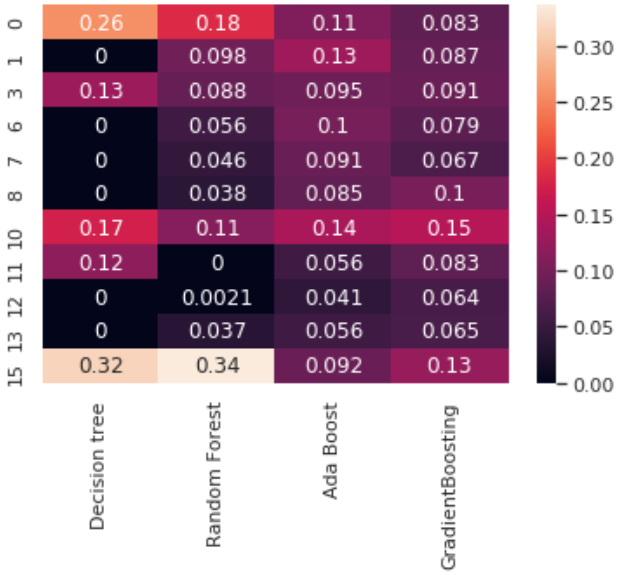

Figure 10: Boosting: feature importance (PCA features).

4.2 Parameter Search

We have performed a Grid Search and Random Search to optimize Random Forest Regressor model. The best discovered models were

added to the collection of all models fitted to the data for further analysis.

For the model collection we calculate MAE on the whole dataset, MAE on test dataset, R2 score on the whole dataset and a product of the first two absolute errors called MAE2.

	MAE	MAE_test	R2	MAE2
GradientBoostingRegressor(n_estimator...	0.543020	0.804695	0.454254	0.436966
RandomForestRegressor(max_depth=8...	0.673341	0.675275	0.142603	0.454691
RandomForestRegressor(max_depth=1...	0.662709	0.686931	0.225230	0.455236
AdaBoostRegressor(base_estima...	0.787064	0.833759	0.026295	0.656221
RandomForestRegressor(criterion='...	0.801126	0.828439	-0.045385	0.663684
DecisionTreeRegressor(max_depth=2...	0.808950	0.821335	-0.061606	0.664419
KNeighborsRegressor(n_neighbors...	0.810604	0.839612	-0.062980	0.680593
LinearSVR(C=0.518,los...	0.810321	0.843329	-0.083514	0.683368

Figure 11: Model evaluation and selection for submission.

4.3 Submission

In total 11 submissions were performed. Some turned out to be erroneous and had to be repeated. The submission files are available in the `submission` directory and they have their corresponding metrics file in the `metrics` directory.

Best score submission does not have a corresponding metric because it was generated during testing in the work in progress phase of our project. The result could not be improved with presented approach.

A screenshot confirming obtained results is available as `submission/screenshot.png`.

submission	score
best	28.9808
PCA + normal	29.1370
default	29.3486
normal	29.4904

Table 2: Submission scores.

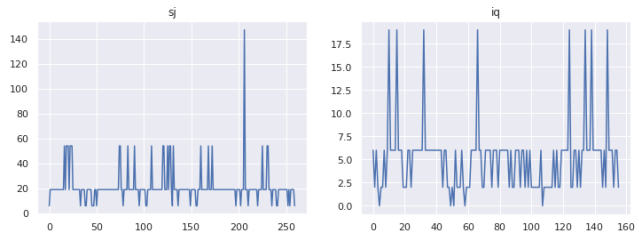

Figure 12: Example of final submission predictions for the cities of San Juan and Iquitos.

5 Refinement

The project could extend the optimization part, possibly a deeper analysis of some parameters in the presented models can be achieved. For example, Gradient Boosting was using a constant value of *n estimators*.

Also some others regression models could be implemented that can be an improvement over the currently best scoring solutions.

The problem could be treated as a time-series forecasting problem, a recurrent model may perform better on this dataset.

6 Summary

We have presented our approach to the supervised learning regression problem of the disease spread prediction.

Multiple regression models were tested and optimized for acquiring the most accurate parameters possible in reasonable time. Manual optimization, boosting and automatic search were used to generate a collection of models from which the best was picked.

Although the presented models have failed to best the drivendata benchmark model, they have bested the performance of our baseline kNN model, which was usually scoring at the bottom end results in our metrics.

Acknowledgments

This work is part of UCLM project of the Machine Learning Techniques subject. Group PMG.