Alexandre Maillard
Mehmet Riza Arseven

BIO-322 Machine Learning for Bioengineers
23 December 2021

Weather Prediction in Pully

Will it rain tomorrow?

INTRODUCTION

In this mini-project, the goal was to predict if it will rain in Pully (a municipality in Switzerland) the next day or not. To realize the latter, a training data consisting of 528 predictors of weather measurements around different towns in Switzerland and the precipitation probability the next day in Pully was provided. After appropriate tuning of different models, predictions were made on the test data. They were later sent to the Kaggle competition to obtain a ranking on the leaderboard in terms of AUC (Area under ROC curve).

RESULTS AND THE PROCESS

For this project, a multitude of linear and non linear models were used. At first, only regression models were used but classifiers were proved to be more effective as they return a probability value between 0 and 1. To get the data from the classification outputs, a function called "probability_output_Multiclass" was coded. For the regressors, it was made sure that the prediction was a probability (between 1 and 0) via the function "enemy_of_out_of_bounds". Another important thing to mention was the failure to standardize the dataset. Even when the dataset did not contain any NaN values or any 0's that may result in a division by zero, multiple models crashed immediately (kNNRegressor and RidgeRegressor) when used with standardized data. A quite interesting finding was that the crashes did not occur when used with only a partial section of the data (less predictors). The latter finding however was not the case in the NeuralNetworkRegression as it predicted NaN for all standardized values. Below are the models used for predictions in this mini-project:

A-LINEAR MODELS

1-Linear Regression

It was the simplest regression model, takes mere seconds to run. But it wasn't very precise with a Kaggle AUC score of 91%.

2-Ridge Regression

It was same as linear regression but with a penalty lambda. The latter makes sure the model does not follow features with big values just because they are big. The model was tuned with cross-validation and the best hyper-parameter found was lambda =1e4. Kaggle AUC= 93%

3-Logistic Classification

Basically the classification version of ridge regression using the sigmoid function. The hyper-parameter lambda was 4.64e5. Kaggle AUC= 93%

B-NON-LINEAR MODELS

1-kNN Regression

Predicts the outcome looking the nearest neighbbours. The optimal hyper-parameter K was found to be K=40. Kaggle AUC= 91%

2-XGB Regression

A popular tree-based method. Learning rate :eta and max_depth was auto tuned with grid search cross validation. The optimal values found were eta=0.02236, max_depth=5. Lambda and number_rounds were hand tuned for reasons of performance: the optimal values found were num_rounds=1000 and lambda=1e1. This was the best model yet with a Kaggle AUC score of 94%.

3-Neural Network Classification

The dropout was hand-tuned with cross validation and its optimal value was found to be near 0.25. The epochs and the number of neurons in the hidden layers were hand-tuned and found to be n_hidden=4096 and epochs=1000. With higher number of neurons, it is generally expected that the model is more robust but it was chosen not to be extremely high because of performance issues. The optimizer chosen is ADAMW and the sigmoid function chosen is NNlib.σ.

CONCLUSION

Our best non-linear model passed slightly our best linear model by %1 AUC. However, it should be said that it is in fact surprising to see that the performance of the linear methods are particularly good. Given the cumbersome, time consuming nature of tuning strong non-linear models and given the performance of the linear models in this problem, our group assumes that with adequate feature engineering and appropriate correlation maps, this problem would be accurately and more rapidly solved by adapted linear models.