Machine Learning

Alain Aziz and Joana Pires

1 Introduction

Liquid chromatography is a method to detect drugs in human tissues and bio-fluids, each drug can be identified by its so-called Retention Time (RT). The retention time depends critically on the chemical properties of the drug and the exact configuration of the chromatography within a particular laboratory. We are provided with two datasets, the training dataset, containing the SMILES notation of each compound, the compound name, the Lab, the RT, the mol encoding and the 1024 ECFP values. The test set has all these columns except the RTs. In a seperate file, we have the CDDDs for each compound.

In this report, we will go through the ways we processed the data, the different machine learning models that allow us to predict this Retention Time and the results.

2 Processing

First we created the train and test data frames using the .csv files provided. The function has boolean parameters that lets the user determine if the sets contain ECFPs or/and the CDDDs in our datasets. Then, the "Lab" parameter was encoded using dummies encoding.

The constants and highly correlated (90%) parameters from the train set were removed and those same parameters were also removed in the train set. For the fingerprints parameters we removed some [ECFP: 54, 262, 432, 491, 555, 593, 892, 913] and for CDDDs [58, 149, 158, 161, 292, 297, 307, 350, 374, 439, 441, 458, 459, 494, 507]. Some columns that didn't seem relevant to predict the RT were removed from the data frame, like the Compound, SMILES and mol parameters (since it only indicates the memory address of the rdkit compound).

Then the training set and the test set were checked for missing values. The fingerprints didn't have missing values but the cddd had some and those were replaced by the mean of the columns they were in.

Additionally, the input from the train and the test were standardized. For some predictions such as the ones returned by the artificial network, we standardized the output as well.

Finally, after the prediction, all negative RTs were set to zero since the RT cannot be negative.

3 Models

To predict the retention time, one can use supervised models (Linear Regression, L1 or L2 Regulation, K-Nearest Neighbors, Polynomial Regression, Gradient Descend, XGBoost, Random Forests or Artificial Network) To tune the hyper-parameters different cross validation functions were used such as LassoCV and RidgeCV but we mostly used GridSearchCV. All the models were tuned using cross validation and some hyperparameters were added in order to avoid over-fitting.

Models	RMSE
Artificial neurons (CDDD)	0.23
XGB (CDDD)	0.42
Random Forest	0.97
Ridge regulation (L2)	1.27
Polynomial with regulation	1.27
Gradient Descend	1.29
Linear regression (CDDD)	1.33
Lasso regulation (L1)	3.45
KNN with K=10	5.71
Linear regression (ECFP)	29.68

Table 1: RMSE comparison of models.

3.1 Linear Method

We started with the most simple method, a linear regression. Even though we do not expect the best results from this method, it is still important in case there is a linear correlation between the parameters and the RT. We then

added regulation, we tried ridge regulation and lasso regulation. As expected Linear regression does not give very good results. However, the results are better if CDDD is used instead of the fingerprints.

3.2 Some other methods

We try many methods to predict the Retention Time but most of them weren't very effective. Most models had a root mean square error around 1.3 for the test error.

The K-Nearest Neighbors method is an improvement compare to the linear methode using ECFP but isn't when it comes to the CDDD. Ridge regulation is the one that is run by the polynomial with regulation function since it has the better score compared to lasso regulation. Polynomial regression uses Ridge regulation and gave us a linear curve, the best degree chosen by the Gridsearch being 1. For the Gradient Descent we got very close results to the polynomial methods.

Even after optimisation those models didn't seem to improve and therefore we decided to try some tree based models and artificial networks.

3.3 Best Models

We first started with a Neural Network. After trying to optimise the function with different learning rates, epochs, number of layers, activation function and dropout rates, it was clear that it was not a very good method to predict RTs using fingerprints. However, using CDDDs, it turned out to be the best method we have. The number of neurons can create an overfit if it's set too high. To avoid this, we added some features to our code, for example we added L1 regulation and used early stopping with a low patience.

We used tree based models such as Random Forest that has a good RMSE compared to non Tree Based models for the ECFP and CDDD datasets. It did however lag behind in its category when compare with XGBoost model.

The XGB model runs faster and better that random forest. However, it cannot surpass the optimsed neural network using CDDD. XGB is the second best model no matter which datasets we use (ECFP or CDDD).

4 Conclusions

In Figure 1, the sorted predicted Retention Times are plotted for each model. We can immediately see that the predictions are roughly similar. Some models however, like Knn and Lasso regulation, are a bit off for the larger RTs. We can also see that the predictions of the linear model are zero for the smaller RTs, this is due to the RTs predictions going below zero and getting corrected by the post processing function, setting all negative RTs to zero.

Using neural networks with CDDD gives the best results by far, followed by XGBoost using CDDD as well. The RMSE for these models are very small. At below 0.5, we can safely say that these methods are very reliable. Especially since the variance for the neural network is high at 13,13.

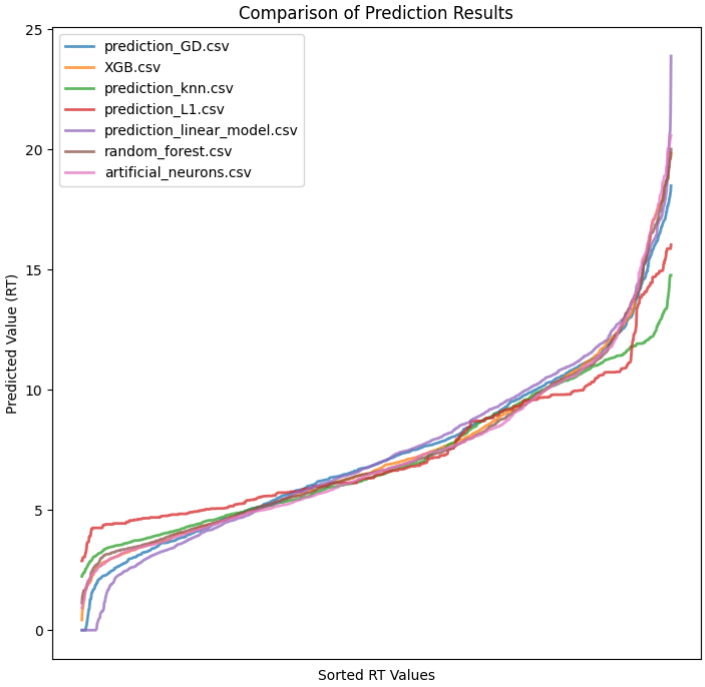

Figure 1: Comparision of prediction results