Spring 2019 metabolite Dataset Report

Kaustubh Lall

April 2019

1 Introduction and Goal

The goal of this project is to find the best features to classify the Metabolite OAT1 vs. OAT3 dataset using random forest and decision tree classifiers. The scores to be maximized are classification accuracy, based on 10-fold cross validation using stratified binary sampling (to fix class imbalance) and AUC score for a test-train split.

The approach used is to find the features with the best AUC scores for a given number k of features drawn from the set $\{2, 3, 4, 5, 6, 7, 8\}$. The best features, with respective feature importance will be reported. To do this, all possible combinations of features are tried, and those that correspond to best results are reported. All code and results can be found here:
GitHub (https://github.com/KaustubhLall/Drug-Research).

Best results obtained:

Features	AUC-AVG	AUC-DT	AUC-RFW	CA-AVG	CA-DT	CA-RFW
2	79%	75%	87%	68%	71%	70%
3	82%	81%	89%	71%	76%	78%
4	84%	82%	90%	74%	76%	78%
5	83%	82%	90%	74%	78%	79%
6	84%	82%	90%	74%	77%	78%
7	85%	83%	93%	78%	78%	80%
8	85%	83%	91%	77%	77%	82%

2 Data - Cleaning and Distribution

We had 25 ”real” feature columns after cleaning with 57 data points distributed 22-35 $OAT1$ and $OAT3$ respectively.

The data had extraneous columns : Vivo/Vitro, Metabolite/Drug, SLC, Significance (KO)/Fold Change (In Vitro), CID, Biochemical Name which could be removed. Specifically, I kept SLC in another list because this contained labels. Other than that these labels do not have biological meaning to separate out classes.

Many of the variables have high correlation with each other. To remove redundant variables, I removed all mutual pairs if they were correlated over 85% such that only the non-redundant ones were left. The correlation analysis gives us the following results for large positive correlations (> 0.87):

Correlated Features	Pearson Corr	p-value
7__24/(nof_Atoms-nof_Fragments)	1.000000	0.000000e+00
5__7/(molArea-nof_Atoms)	0.993586	1.246462e-130
5__24/(molArea-nof_Fragments)	0.993586	1.246462e-130
1__24/(molVolume-nof_Fragments)	0.993554	1.745201e-130
1__7/(molVolume-nof_Atoms)	0.993554	1.745201e-130
1__5/(molVolume-molArea)	0.991247	1.752824e-121
1__16/(molVolume-a_heavy)	0.982288	8.547405e-101
19__23/(C_R0-nof_RotB)	0.977995	1.899211e-94
16__24/(a_heavy-nof_Fragments)	0.970694	4.309652e-86
7__16/(nof_Atoms-a_heavy)	0.970694	4.309652e-86
5__16/(molArea-a_heavy)	0.968668	3.794397e-84
7__20/(nof_Atoms-C_sp3)	0.964032	3.858290e-80
20__24/(C_sp3-nof_Fragments)	0.964032	3.858290e-80
1__20/(molVolume-C_sp3)	0.938138	1.661728e-64
5__20/(molArea-C_sp3)	0.936010	1.542450e-63
13__16/(Complexity-a_heavy)	0.927815	4.219761e-60
4__9/(molPSA-nof_HBA)	0.912658	1.066100e-54
8__17/(nof_Chirals-C_R2)	0.897164	4.146746e-50
16__20/(a_heavy-C_sp3)	0.896385	6.741688e-50
8__18/(nof_Chirals-C_R1)	0.895264	1.348274e-49
4__10/(molPSA-nof_HBD)	0.880536	6.194669e-46

Discard: 1, 2, 14, 5, 12, 18, 20, 23 (molWeight, molVolume, molPSA, molCharge-total, Complexity, C-R2, C-R0, negCharge/Volume).

2.1 Division into Test and Training

To combat 90-48 class imbalance of $OAT1$ vs $OAT3$, I did the following split: 47 instances of $OAT3$ were used in the training set, and 47 instances of $OAT1$ were chosen from 90. This way, the training data has perfect class balance.

The issue is now the test set. To "balance", there are 44 instances of $OAT3$, repeated from training set (randomly selected) and 44 of $OAT1$.

3 Classifiers and Hyper-parameters

3.1 Decision Tree

Decision trees are very prone to over-fitting, which is intrinsically an issue in a small data-set like ours. Thus, we need to perform pruning. Ideally, I would use AB-pruning methods, however sklearn doesn't provide any AB-pruning implementations. However, we can limit the total depth of the tree and restrict leaf nodes to have a minimum number of samples. The parameters I chose were:

1. $max\_depth = 5$ This choice was purely heuristic, it makes sense to have a depth no more than 10% of data.

2. $min\_samples\_for\_leaf = 3$ This choice also makes sense. It should be odd, and small enough that it allows us to express the complicated relations in our small data-set. Too big, and we would not have a lot of diverse paths in our decision tree, and it could potentially cause some guesswork. $3, 5$ are the choices that make the most sense, because they ensure one class is majority and and still leave enough samples to express some complexity.

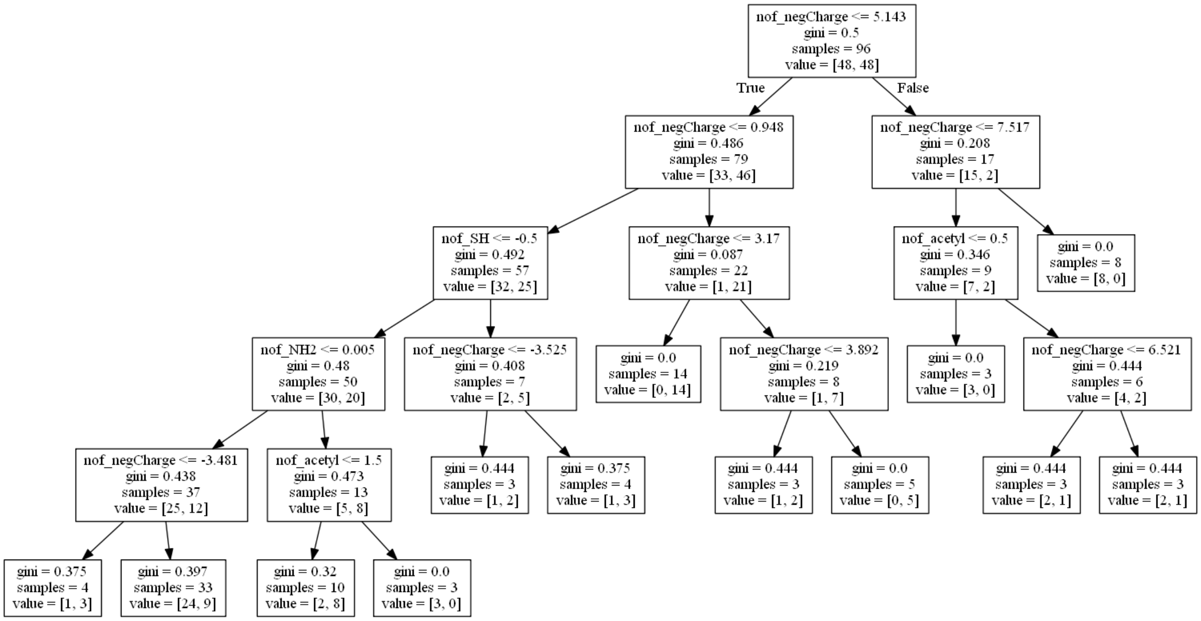
Note, I tried running this classifier sans pruning as well, which gave slightly better (1-2) classification best-case results for AUC scores.

3.2 Random Forest Walk

The best performing classifier of them all, RFW makes very good work of the data and helps reduce over-fitting. The only hyper-parameter is the number of estimators, which is set to the sklearn default of 100.

4 Results and Report

The best accuracy scores are summarized in the description. For a more detailed view, see the feature importance ipynb file. Below is a visual of the decision tree that gives the best classification accuracy:



The feature importance for the tree with those features are:

DT Imp.	RFW Imp.	Feature
0.000000	0.060412	nof_COOH
0.100568	0.288163	negCharge/Volume
0.018213	0.088354	posCharge/Volume
0.000000	0.037543	nof_posCharge
0.422495	0.163420	nof_HBA
0.458724	0.362107	nof_Chirals

Another interesting table is:

DT Imp.	RFW Imp.	Feature
0.061809	0.054556	nof_COOH
0.000000	0.022223	nof_SO3H
0.000000	0.035290	nof_PO4
0.282553	0.128427	nof_HBA
0.411740	0.423966	nof_Chirals
0.243898	0.335538	PSA/Area

Note how two features are not used by the decision tree, but are used by the random forest. These correspond to the best RFW CA.

5 Closing Notes

This problem is not typical of most machine learning problems. The combined issues of having a lot of features, a very small data-set makes the problem very difficult and inherently easy to get inconsistent results on. Over/under-fitting, variance in data in occur in a lot of ways. While data augmentation can help offset the issue, it may or may not introduce some kind of bias. The idea of the research was to simply find what set of features work best at describing this kind of medical data. A good test, one that I would like to do in the future is to use this metabolite dataset to find features for metabolites, which can be described with the same chemical features used. This could serve as a blind test to see if the approach is able to find features that generalize well.

Overfitting is a major concern for this problem. One way I found of dealing with it is to not select the top features, but rather to proceed with a metric that evaluates what groups of features occur together etc. to be able to determine what best describes molecules which exhibit properties similar to metabolite-transporter interactions. Also useful would be to remove features that are redundant (molArea and molWeight) or highly correlated, which will be an active area of interest for me in the next quarter.