Final Report

Problem

There are many more drugs with potential therapeutic effects that could ever be clinically tested. So there are significant efforts in using machine learning to reduce the set of promising drugs to a more promising subset clinically test. To help in that effort I wrote an algorithm that predicts if a drug will inhibit an enzyme essential to metabolism.

The Therapeutic Data Commons (TDC) is a nonprofit that "is an open-science platform with AI/ML-ready datasets and learning tasks for therapeutics, spanning the discovery and development of safe and effective medicines" provided this dataset.

The problem I worked on was predicting if a drug would inhibit an enzyme essential to metabolism named CYP2C19.

According to TDC "the CYP P450 genes are essential in the breakdown (metabolism) of various molecules and chemicals within cells. A drug that can inhibit these enzymes would mean poor metabolism to this drug and other drugs, which could lead to drug-drug interactions and adverse effects. Specifically, the CYP2C19 gene provides instructions for making an enzyme called the endoplasmic reticulum, which is involved in protein processing and transport."

The dataset looked like this:

Drug	Y
Clc1ccccc1-c1nc(-c2ccccc2)n[nH]1	1
COc1ccccc1C(c1nnnn1C(C)(C)C)N1CCN(Cc2ccncc2)CC1	1
CCC(c1nnnn1CC1CCCO1)N(CCN1CCOCC1)Cc1cc2cc(C)cc...	0
Br.N=c1n(CCN2CCOCC2)c2ccccc2n1CC(=O)c1ccc(Cl)c...	1
COc1ccc(/C(O)=C2/C(=O)C(=O)N(CCCC(=O)O)C2c2ccc...	0

The "Drug" column is the "SMILE" string representing the chemical structure molecule and the "Y" column is the label of if that drug will inhibit the enzyme.

There were 12,665 labeled drugs where ~1/3 were positive samples. The SMILES have a variable size with an average of 46.4 (+- 20.8) characters, a minimum size of 2 (a Nitrogen and Oxygen), and a max of 340 characters. Because the data was variable sized and topological most out-of-the-box algorithms would not work. Because most machine learning libraries are designed to work with vectors the first problem to solve was to find a way to featurize molecules.

Approach

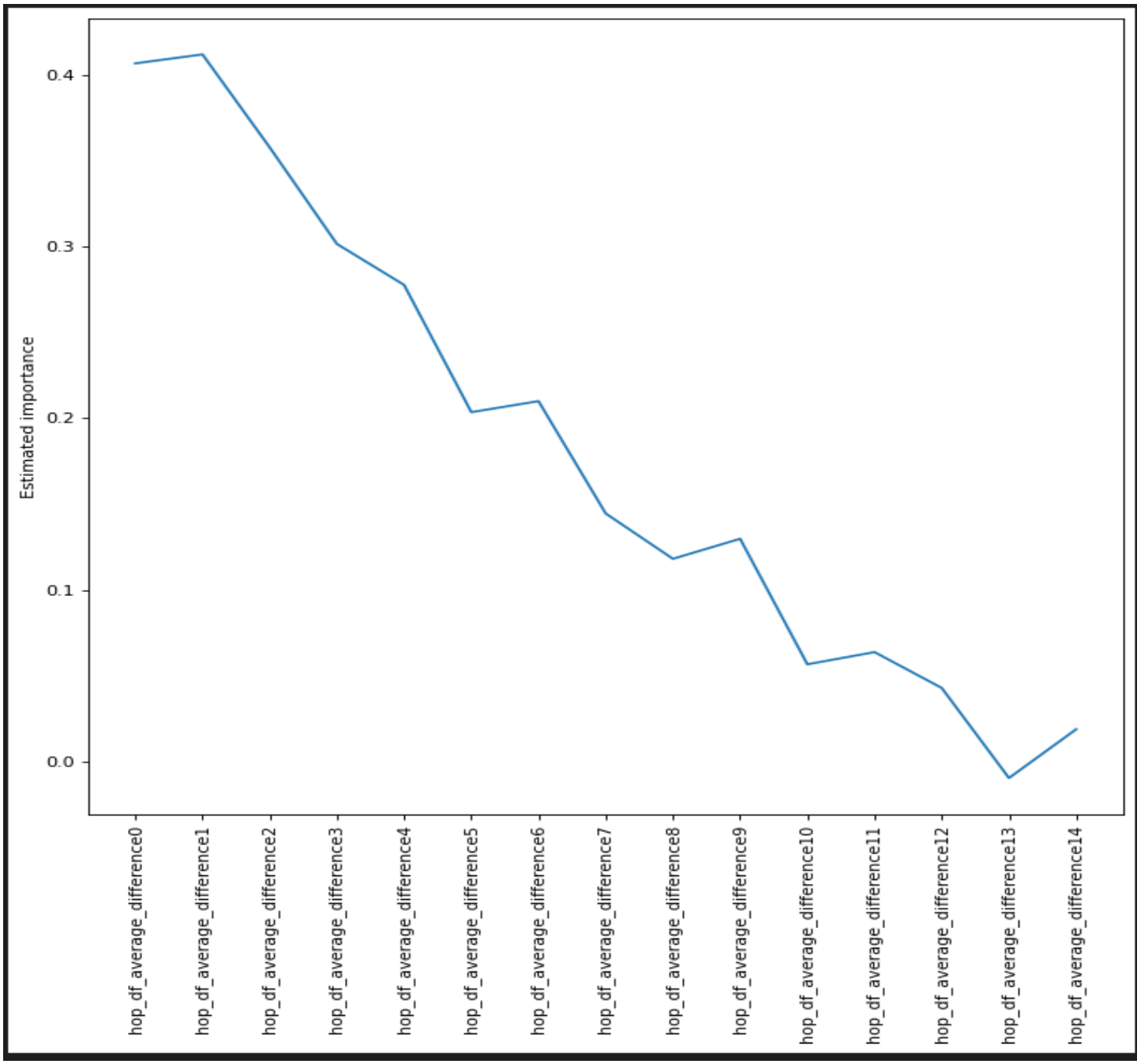
Since I do not have a background in chemistry the first thing I did was reach out to some people with a chemistry background. After talking with Anna and Kyle, I decided to featurize the molecule functional groups. Because those are the components that do the underlying chemical interaction.

Because molecules are graphs (nodes and edges) one of the first things I did was watch a free Stanford class on Machine Learning on Graphs taught by Professor Jure Leskovec.

In one of the lectures, Leskovec introduced a classic algorithm named ColorRefinement that is typically used for embedding graphs and nodes by topology.

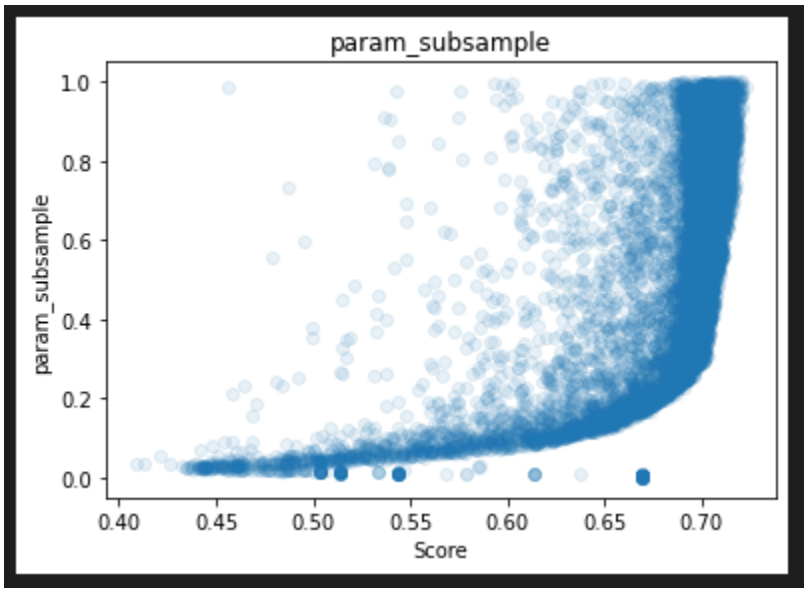
I realized that with a slight tweak, initializing nodes as a hash of node attributes instead of as a constant, I could embed subgraphs in a way that contains both attribute and topology information. This is appropriate because the smaller subgraphs are what do the chemical interaction. I wrote a Python implementation of a variation ColorRefinement with a tutorial here.

I trained an untuned model on each of the first 14 hop neighborhoods and stored the AUPRC score then compare those scores to when I permuted the targets. This gives an estimate of how the predictive power of each hop size.

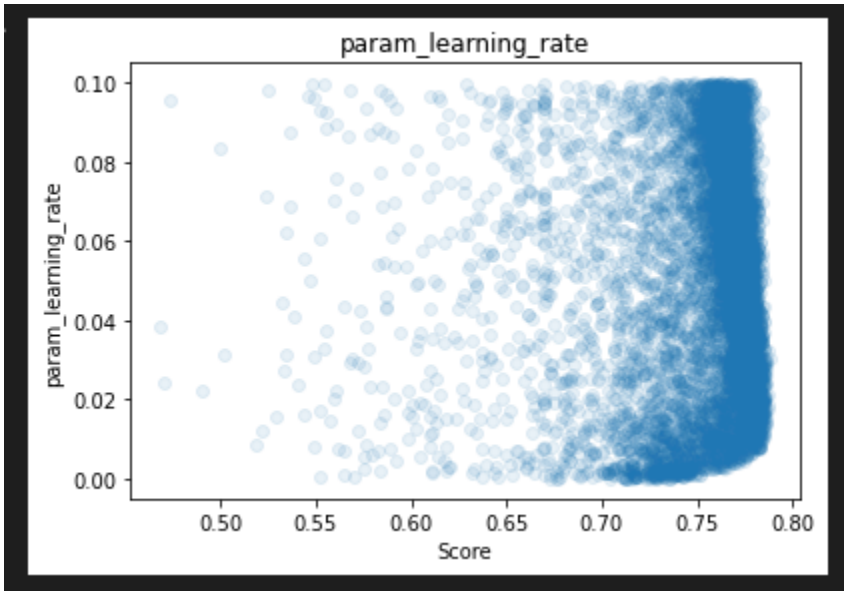


After looking at how much predictive power was lost at each step I decided to only train models on the first 4 hops DataFrames.

I next tested ~5,000 combinations of hyperparameters of the LGBMRegressor with RandomizedSearchCV(). Here are some scatter plots looking at the relationship between AUPRC and the hyperparameter.



The impact of the subsample parameter on the 3rd hop df.



Impact of the learning rate parameter on the 0 hop df.

I then took the best 4 models found in the random search and tested 100,000 random combinations of a weighted average.

index	weight_1	weight_2	weight_3	weight_4	AUPRC
31079	0.228070	0.508772	0.157895	0.105263	0.796028
4834	0.232394	0.507042	0.183099	0.077465	0.796014
93031	0.237113	0.505155	0.159794	0.097938	0.796001
62021	0.236994	0.502890	0.179191	0.080925	0.795998
11659	0.236842	0.506579	0.157895	0.098684	0.795997

The AUPRC column here refers to the average AURPC score on the train test split when you apply these weighted averages to each of the model's predictions. Fortunately, these weights converged and so I decided the final weights would be the average weights of the best 20 of 100,000 random weights.

index	model_0	model_1	model_2	model_3	target
1935	0.312784	0.208339	0.295249	0.286069	False
1936	0.33413	0.34766	0.449999	0.583438	True
1937	0.460933	0.445365	0.691044	0.297753	True
1938	0.426971	0.588611	0.686233	0.507866	True
1939	0.859865	0.984376	1.068336	0.679388	True

Each model's prediction on a CV-fold of the training data.

While all of these models are trying to predict the same thing their predictions are typically not that correlated. This is a good sign because it means that creating an ensemble of them will add more value than if they were strongly correlated.

After I determined the best models and weights I hardcoded them into Modeling/Final_weighted_model.ipynb and submitted that model to the Therapeutic Data Commons. The final weighted ensemble had an AUPRC of .767 +-.003 and is currently the best and most simple model of those submitted.

Next Steps

I refactor the Color Refinement algorithm more generalizable and push it to Pip. In particular, the algorithm computes initial colors as a hash of all node attributes and I think that it would be improved by allowing the end-user to decide what attributes to use in initializing node colors.

I should use the same embedding and weighted classifier model to submit models for the rest of the binary classification tasks at TDC.

I think there is a significant improvement to be made in hyperparameter tuning and in deciding how many colors and hops to embed the molecules. I did not use cross-validation to decide on 2000 colors and 4 hops. There is likely room for improvement by using more models and I suspect a more sophisticated technique such as Bayesian Model Averaging/Combination would lead to more accurate predictions.