Machine-learning informed gapfilling

Behzad D. Karkaria[1] Kiran R. Patil[1]

[1]MRC Tox

*To whom correspondence should be addressed;

E-mail:

Abstract

Genome scale metabolic model reconstructions are relatively poor at predicting species viability in specified media. Methods such as gap-filling, require experimental data to identify which metabolic reactions should be added to achieve species growth. However, this therefore requires *in vitro* screening of each species in different media environments. We have developed a machine learning pipeline that can predict the viability of a microbial species in a given defined media. We propose that this pipeline can be used to perform automatic gap-filling, to improve the output of existing automated metabolic reconstruction frameworks.

1 Introduction

Genome-scale metabolic models (GSMMs) are mathematical representations of an organisms metabolism. GSMMs are automatically reconstructed, using tools such as CarveMe an gapseq. These tools use associatons between genomes and gene annotation databases to construct a biochemical network, represented as a stoichiometry matrix. However, these models require curation informed by organism-media viability experiments. If an organism is viable in a defined media, the model has may need reactions added to ensure the model yields the same phenotype as the experiment.

2 Results

GSMM reconstruction is... Our dataset is composed of 92 species of gut bacteria, grown in 10 different media (Figure 1.) Figure 1a shows the media compositions. All media use glucose as a carbon source. All strains were inoculated at 0.01 OD, if the OD after 48hrs of culture was less than 0.1, we defined this strain as inviable in the media. Figure 1b shows the viability of each species in each media.

Describe dataset

Raw GSMMs are poor predictors of growth

Automated GSMM reconstruction are poor predictiors of growth. To illustrate this, we reconstructed all 92 models for species shown in Figure 1B. We refer to these uncurated, automatically constructed GSMMs as raw GSMMs. For each raw GSMM-media combination, we set the media concentrations and optimize to find the growth rate (μ). Where $\mu > 0$ we classifiy viabile growth, where $\mu < 0$ we classifiy inviabile growth. The performance of these raw GSMMs on the dataset are shown in Figure 2a. The confusion matrix shown in and performance metrics show that these raw GSMMs suffer from frequent false negatives (Figure 2b). This can be due to poor assembly quality and incomplete genome annotation datasets.

Integrating metabolism and media for viability classification

While the raw GSMMs are poor predictors of growth, we hypothesised that features of the raw GSMMs, paried with features of the defined media could be leveraged by machine learning techniques to predict species-media vialility.

As illustrted by Figure 2, raw GSMMs are incomplete and possibly incorrect representations of an organism. However, we expected this fuzzy representation to useful for predicting media-species viability. Here we demonstrate machine learning approaches that integrate information about the organism metabolism and the media environment to classify strain viability.

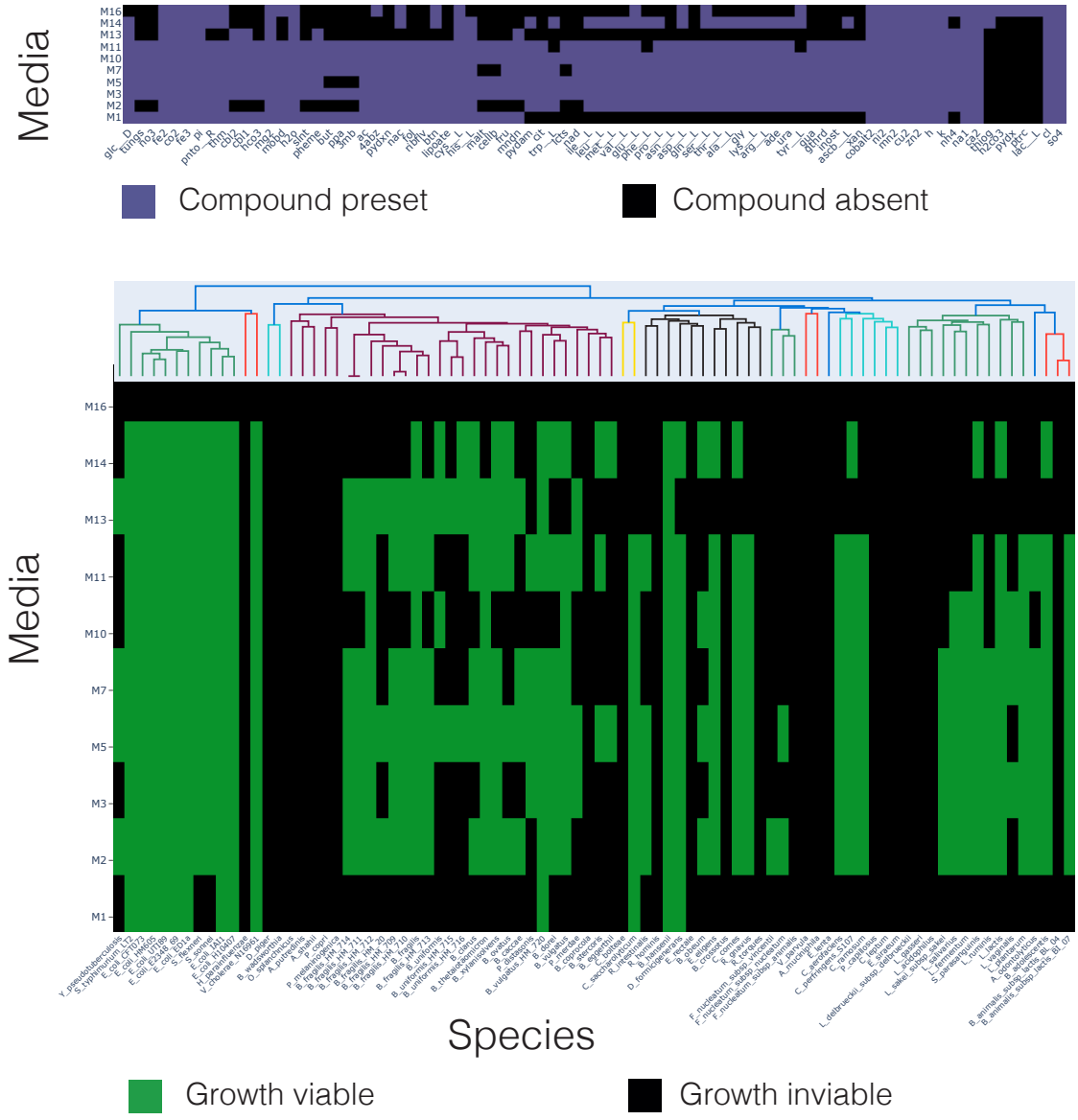
Media
M16
M14
M13
M11
M10
M7
M5
M3
M2
M1
Compound preset
Compound absent
Media
Species
Growth viable
Growth inviable

Figure 1:

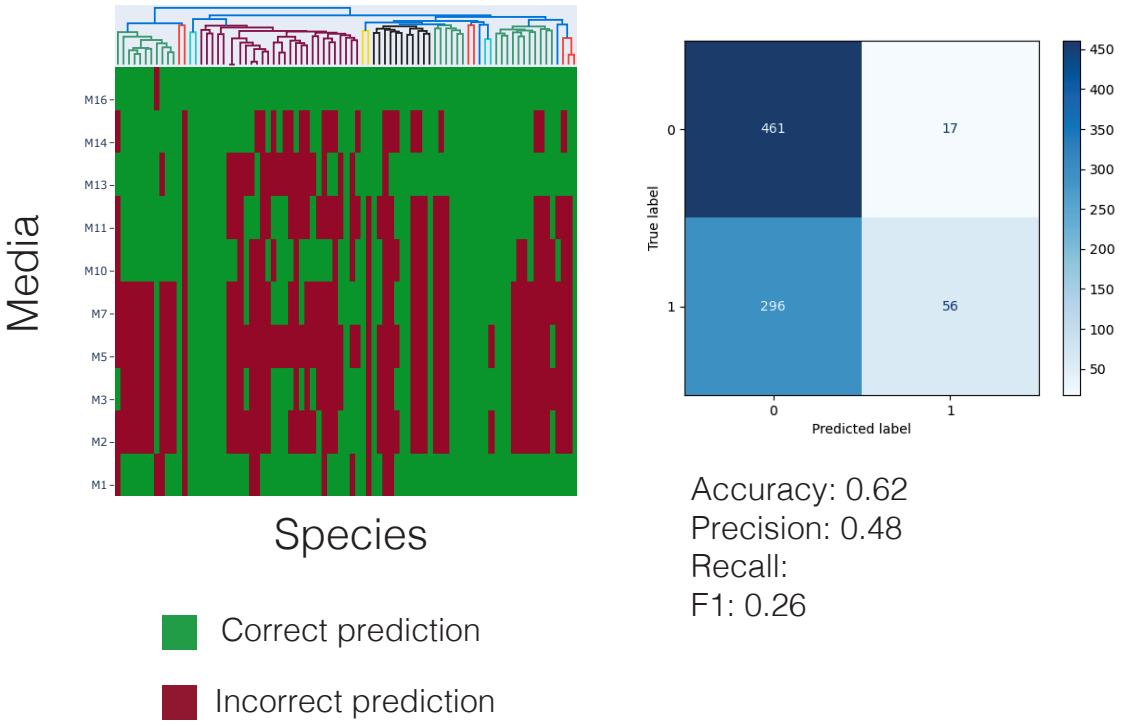

Figure 2: Performance of raw GSMMs as classifiers of growth

We reprsent each organism as a binary set of features indicating presence or absenece of a metabolic reaction in the raw GSMM (X_{Model}). We refer to this as a metabolic fingerprint. Similarly, each defined media is represented as a binary presence or absesnce of a compound (X_{Media}). Binary lables indicate growth for each media-species combination (y_g), as shown in Figure 1.

First, we generate a test set consisting of 20% of the whole dataset. The strains reserved for the test set were selected by traversing the x-axis shown in Figure 2, and selecting every nth strain. Validation sets were generated by randomly sampling from 20% of the training data (x strains). The training set therefore consists of x strains.

We decided to compare Random Forest, Gradient Boosting, support vector machines (SVM) and decision trees. We performed a gridsearch hyperparameter optimisation for each model. For each set of parameters, fitting was repeated 10-times to account for the stochastic behaviour of the models and the variation in sampling of the validation set. Optimal hyperparameters were chosen by the

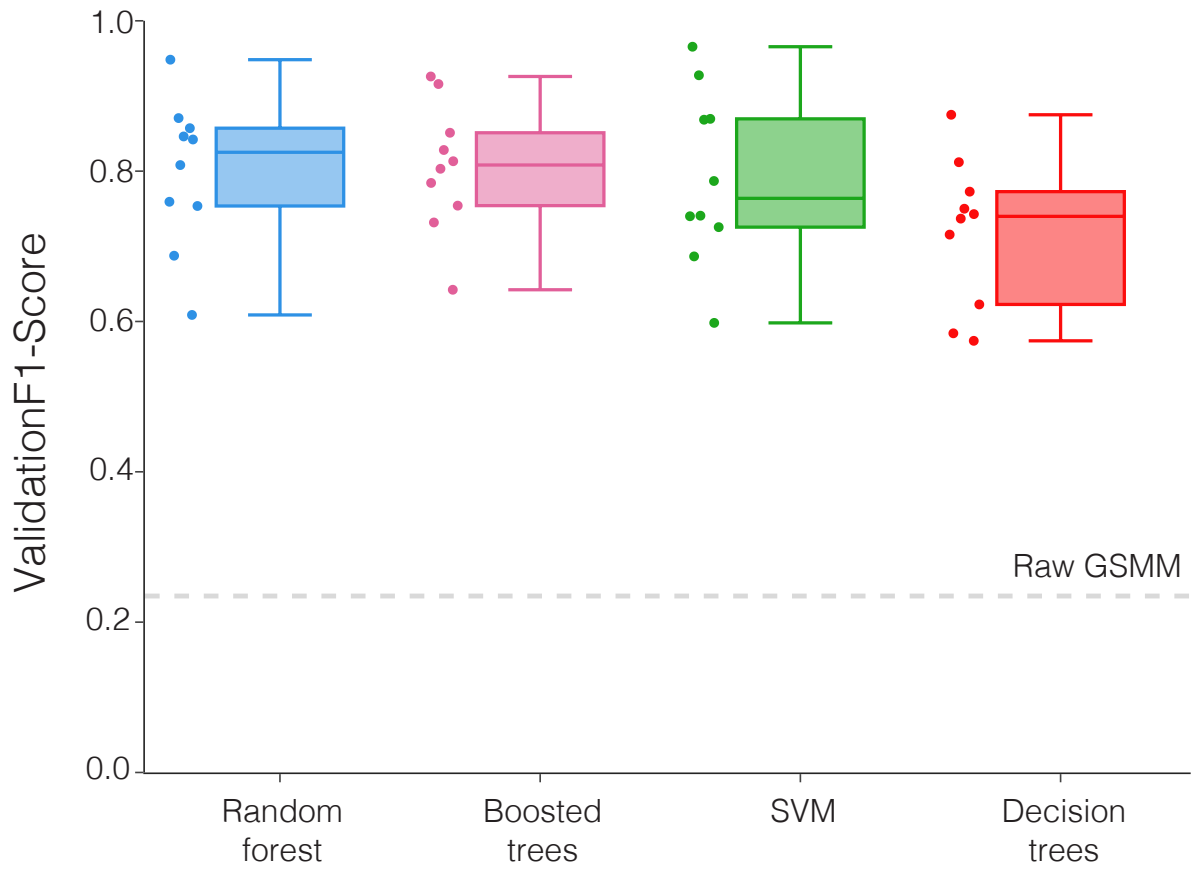

Figure 3: Comparison of trained classifier validation F1-scores

highest mean validation F1-score. We focus on the F1-score because we are more interested in correcly classifying strain viability than correctly classifying inviable strains. Figure 3 shows a comparison optimised models. We can see that random forest, gradient boosting and SVM classifiers all perform similarly. Table 2 shows the mean and standard deviations of each optimised model. Furthermore, we can see that the trained classifiers performance on the validation set greatly exceeds the performance of raw GSMMs on the entire dataset.

Model	**Mean Accuracy**	**Mean F1-Score**
Random Forest	0.82	0.80
xGBoost	0.82	0.80
SVM	0.81	0.79
Decision tree	0.76	0.72
Raw GSMM	0.62	0.23

Using the optimal hyperparameters identified for random forest (see Methods), we retrained a model using the entire training set, and assessed its performance on the test set. In Figure 4A we can see that the classifier performs poorly on *B. animalis subsp lactis BL 04*, *C. bolteae* and *E. eligens*. However, this classifier is a considerable improvement on the performance of raw GSMMs.

Automated gapfilling using random forest classifier

Having trained the classifier on a set of gut bacteria and demonstrated good performance on the test set, we wanted to see if this method was able to provide contextual information about the organism to inform gapfilling.

The KOMODO dataset contains information about organism-media pairings to grow difficult-to-culture microbes. We took a subset of this dataset for anaerobes that grow in a range of 6.0 - 7.5pH, loosely matching the data used to train the random forest classifier. This 94 datapoints, consisting of 49 different medias and 93 different organisms.

We reconstructed raw GSMMs of all organisms using CarveMe. We simulated the growth in the growth media defined by the KOMODO dataset. These are media that are known to be culture each organism. The raw GSMMs had a accuracy of zero, always predicting no growth.

For each organism, we used the trained random forest classifier to predict if the organism should grown on each of the 10 media used in training. If the classifier predicted growth, we performed gapfilling on the raw GSMM for that media. We then retested resulting 'polished' GSMMs that had been gapfilled as informed by our random forest classifier.

Figure 5 shows the performance of the gapfilled models on the KOMODO dataset. We can see that the performance of these models is still poor, however, we see that 7 polished models now correctly predict growth which the raw GSMMs did not. We believe this is a strong indication that our methods can provide improvements without the need for additional experiments. However, to unlock the true potential of these techinques, we require a larger and more diverse dataset of negative and

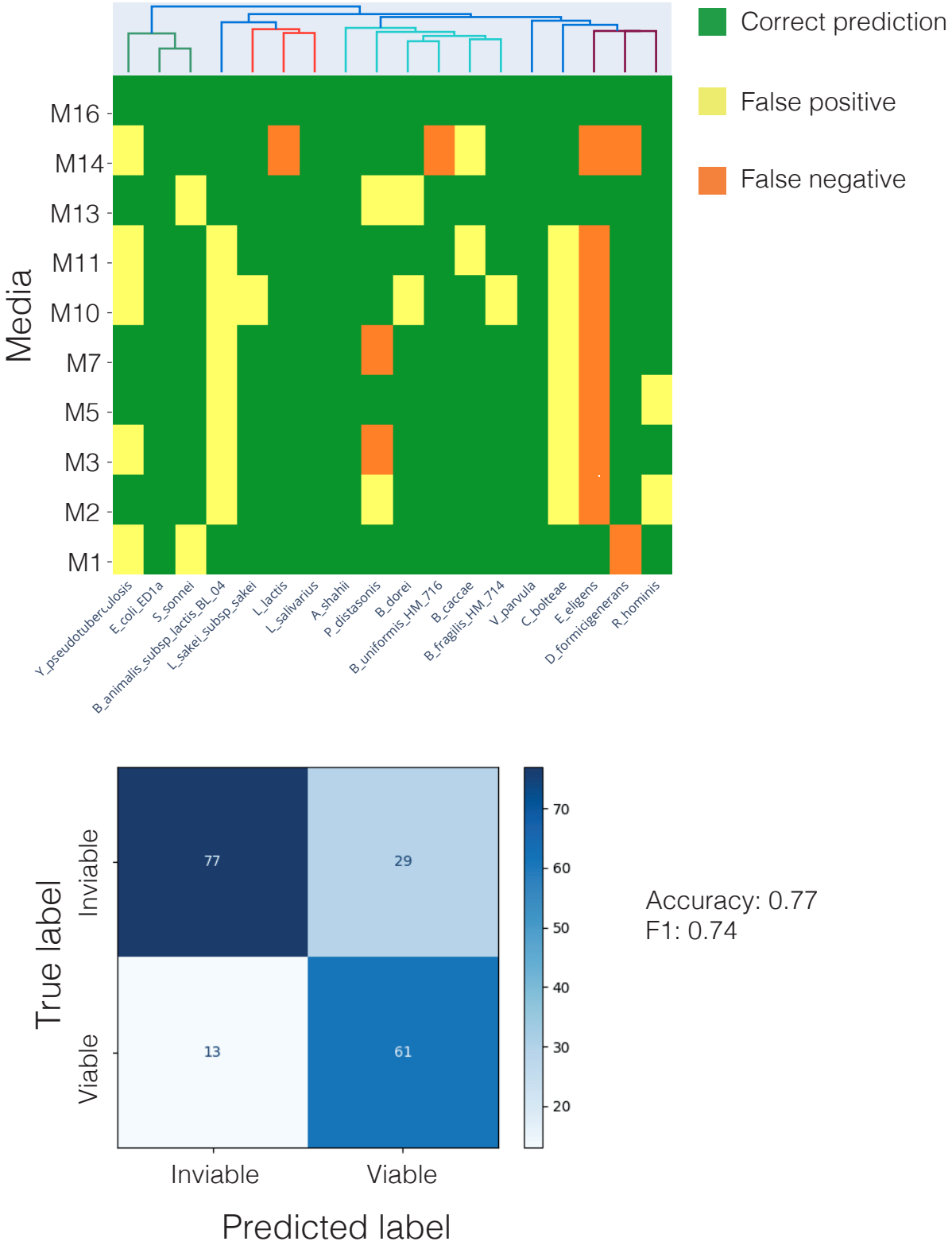

Figure 4: Performance of optimal random forest model on test set.

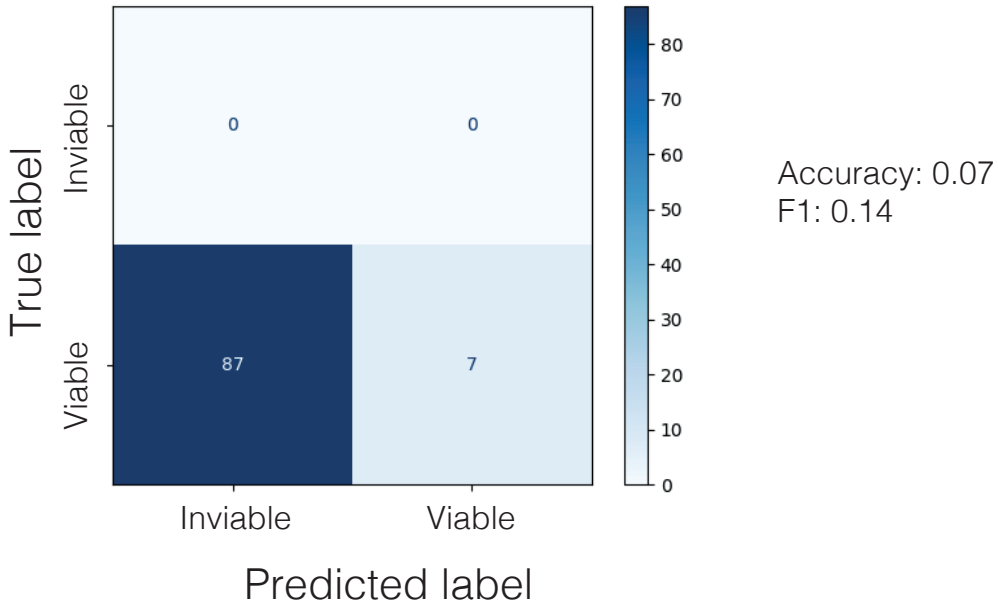

Figure 5: Performance of KOMODO models following machine learning informed gapfilling positive culture data.

3 Methods

Model selection hyperparameters

Models were optimized using a grid search across hyperparameters shown in Table 3. All parameters refer to those used in the scikit-learn package. The red highlighted parameters are those which produced the optimal performance for each model.

Random Forest	
n estimators	50, 250, 500, 750
max depth	null, 10
max features	sqrt, log2
Boosted trees	
n estimators	50, 250, 500, 750
learning rate	0.1, 0.01, 0.05
subsample	1.0, 0.5, 0.75
max depth	null, 10
max features	sqrt, log2
tol	1e-4
SVM	
C	0.1, 0.01, 1.0, 10.0
kernel	rbf, linear, poly, sigmoid
gamma	scale, auto
degree	3
tol	1e-3, 1e-4, 1e-2
shrinking	True
Decision tree	
criterion	gini, entropy, log loss
splitter	best, random
max depth	null
max features	auto, sqrt, log2

Table 1: Table showing hyperparameter grid search for used for model selection. Highlighted in red are the optimal combination of parameters identified for each model