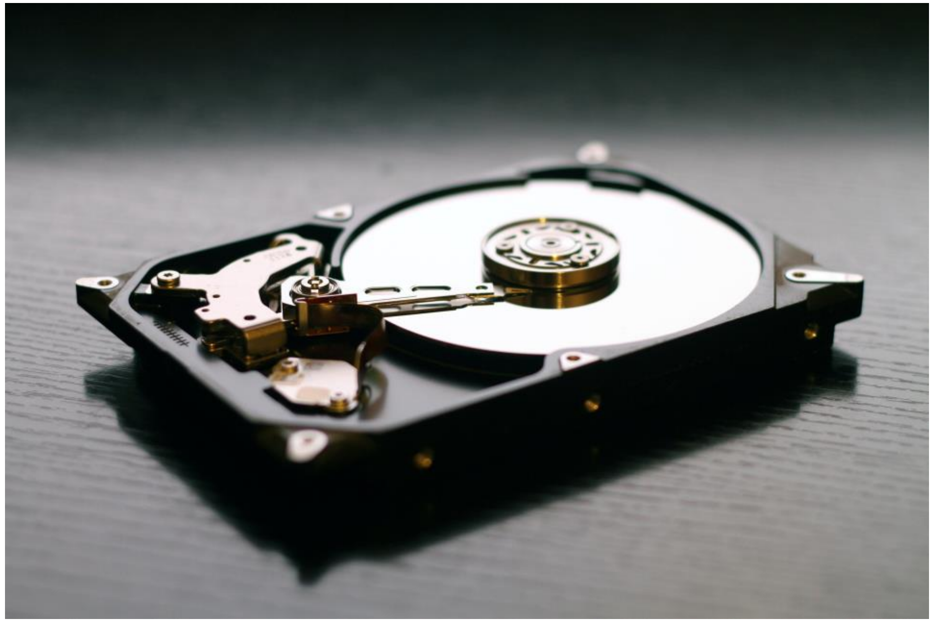
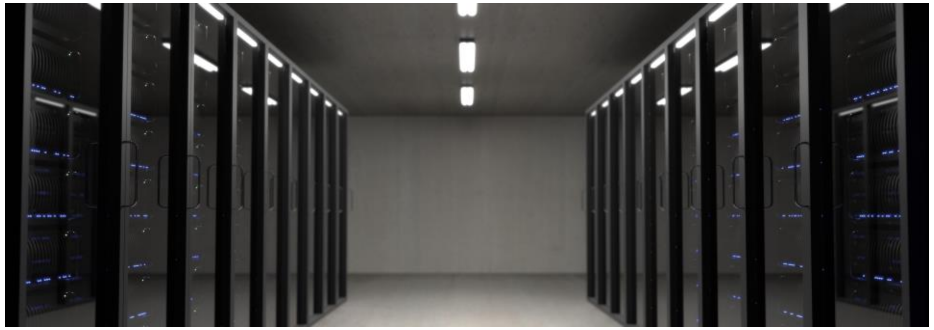
Intro

- Robust failure prediction can prevent downtime in many contexts.
- Modern hard-drives periodically measure and report usage and health statistics (called SMART metrics).
- This project looks at whether hard-drive failure can be predicted ahead of time based on these metrics.





Backblaze Hard Drive Data

- Backblaze has been publishing failure data for hard drives in their datacenters since 2014.
- For each drive, they record values of SMART metrics once per day along with whether the drive failed that day.
- There are ~20 different drive models and a total of 187 metrics, only a subset of which are available for a given drive model.
- We focus on 2021 data for one drive model (ST4000DM000).
    - After removing features with no or constant data, there are **22 SMART metrics** for this drive model.
    - There are 18,611 drives and a total of **324 failures**

https://www.backblaze.com/b2/hard-drive-test-data.html

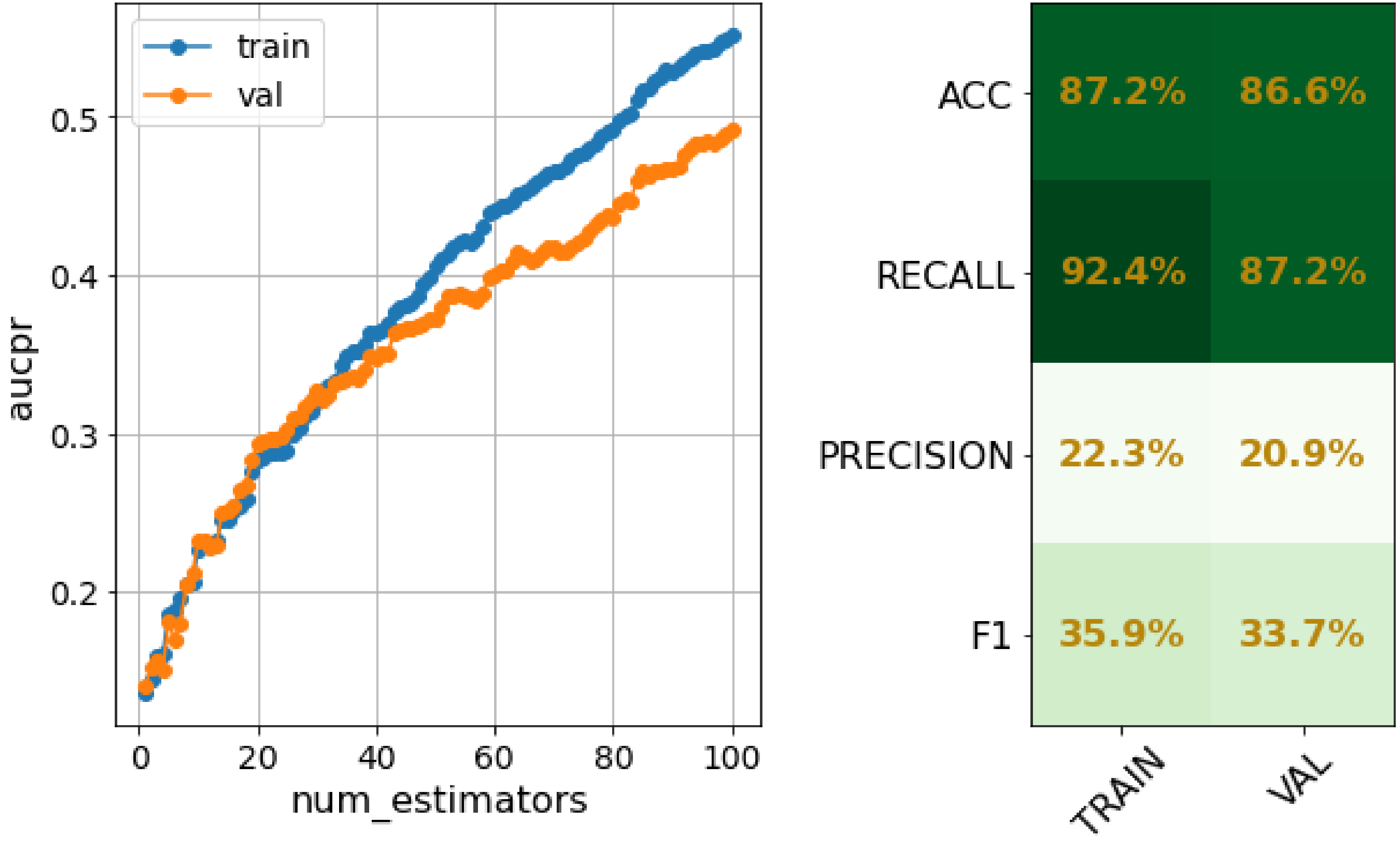
Modeling Approach

- **Problem Definition:** Predict **whether** a given drive will fail in the next **7 days**.
- **Feature Selection:** Use all 22 available features.
- **Model type:** Train a gradient-boosted decision tree model (**xgboost**).
- **Data sampling:** Use the latest 90 days for each drive. Use all failed drives + as many (324) randomly selected healthy drives. => 59,320 rows of data, **3.9%** of which belongs to the positive class.
- **Data splits:** **90/10** train/val split.
- **Accounting for Class Imbalance:** **Weight the failure rows** by the inverse of their relative frequency:

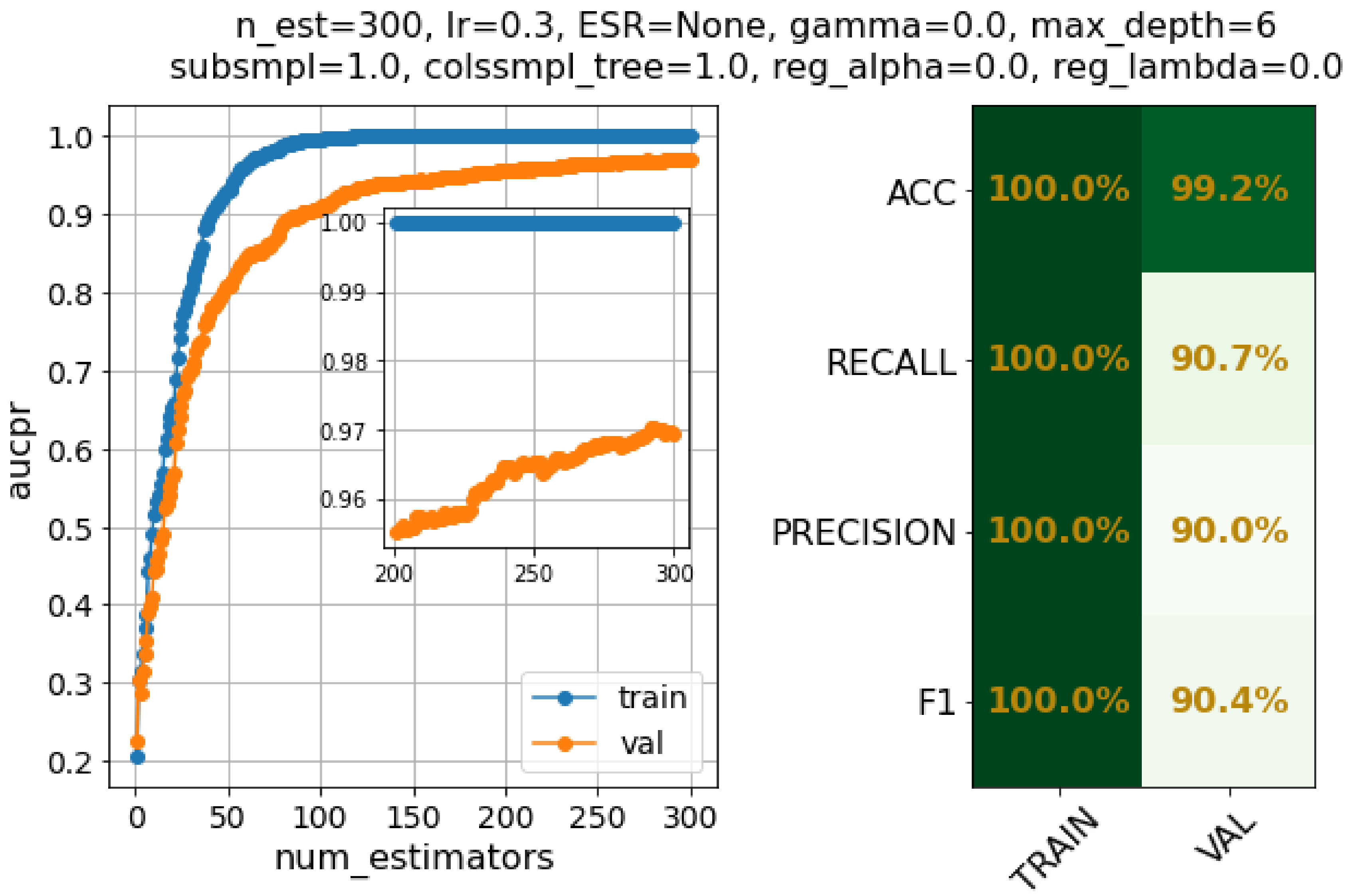
$$failed\ rows\ weight\ scale = \frac{number\ of\ healthy\ rows}{number\ of\ failed\ rows}$$

First Run with Default Parameters

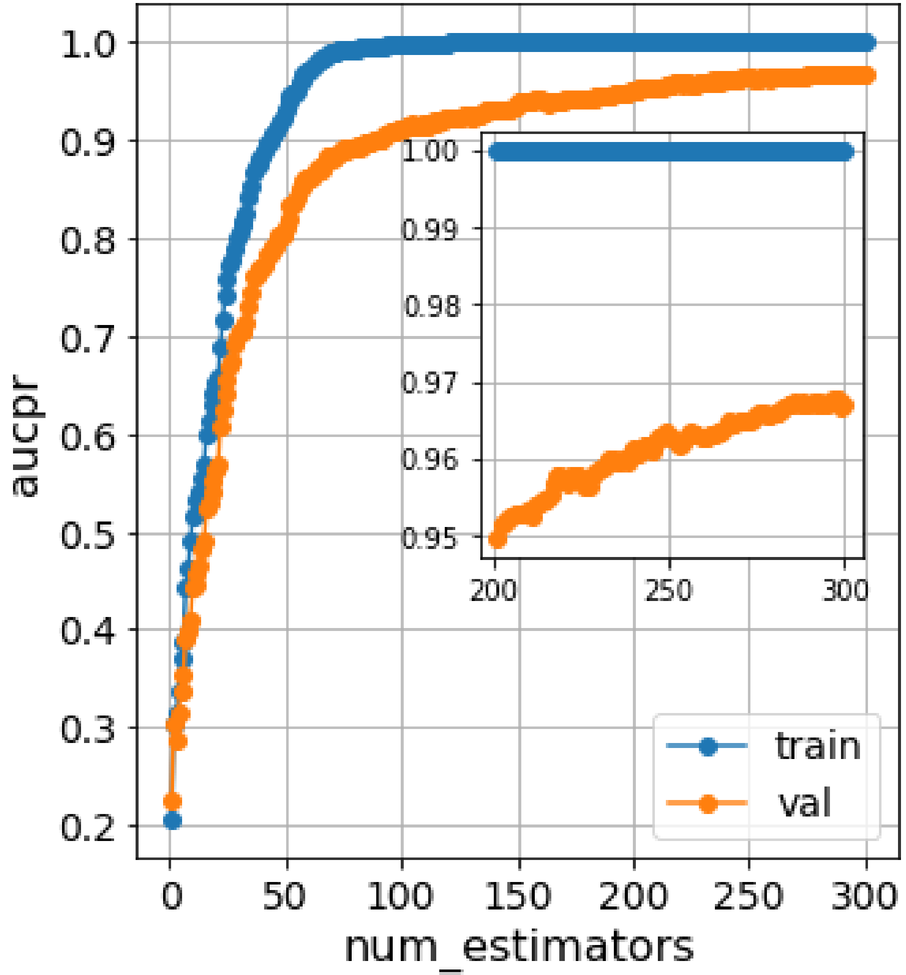
n_est=100, lr=0.3, ESR=None, gamma=0.0, max_depth=3
subsmpl=1.0, colssmpl_tree=1.0, reg_alpha=0.0, reg_lambda=0.0



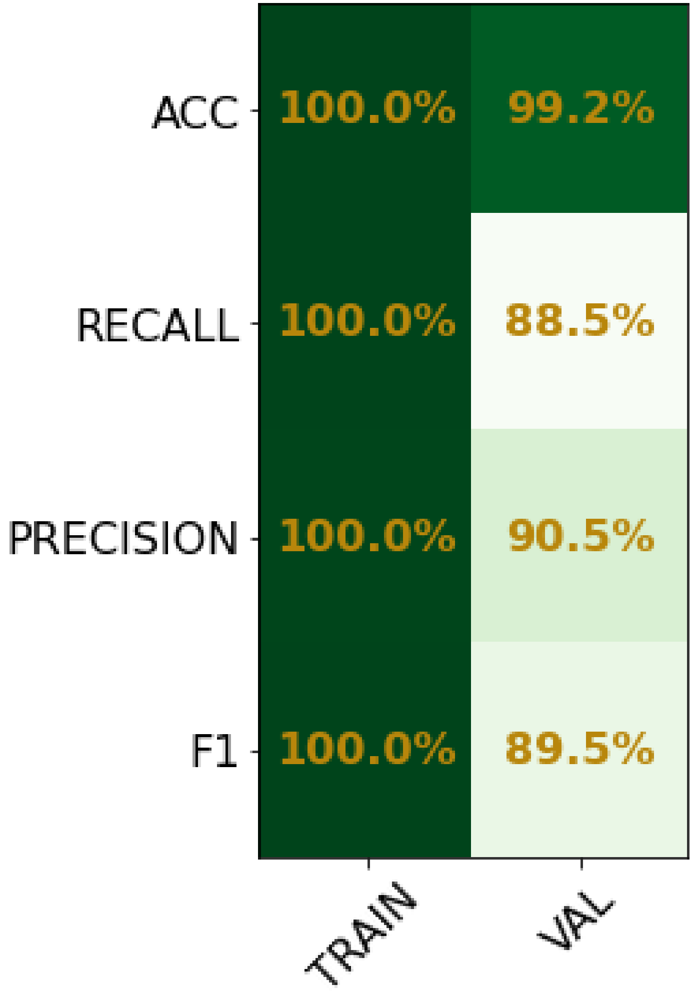
Increase max_depth and add more trees



Add gamma

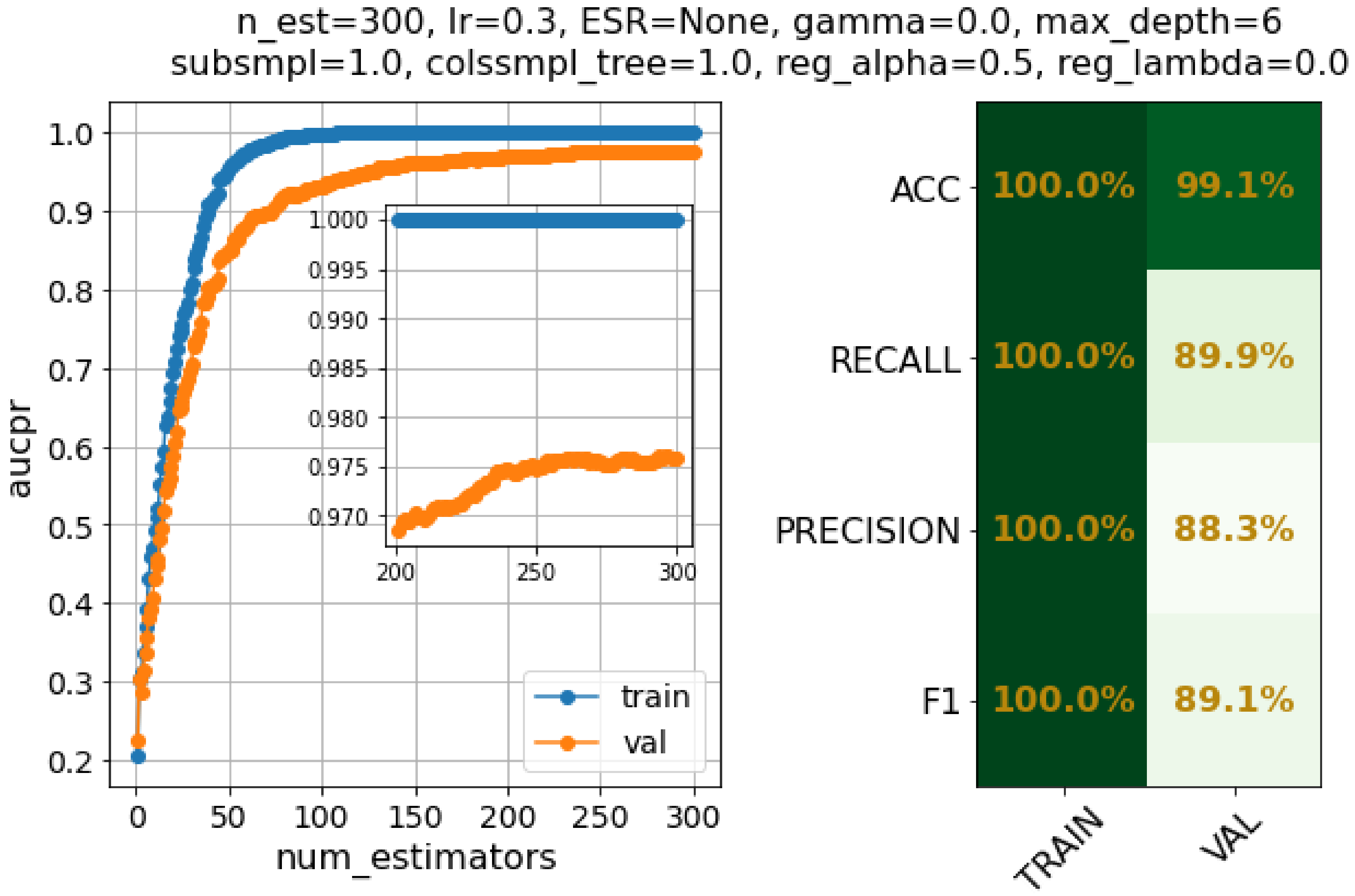


n_est=300, lr=0.3, ESR=None, gamma=0.05, max_depth=6
subsmpl=1.0, colssmpl_tree=1.0, reg_alpha=0.0, reg_lambda=0.0
aucpr
num_estimators
train
val
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
50
100
150
200
250
300
1.00
0.99
0.98
0.97
0.96
0.95
200
250
300

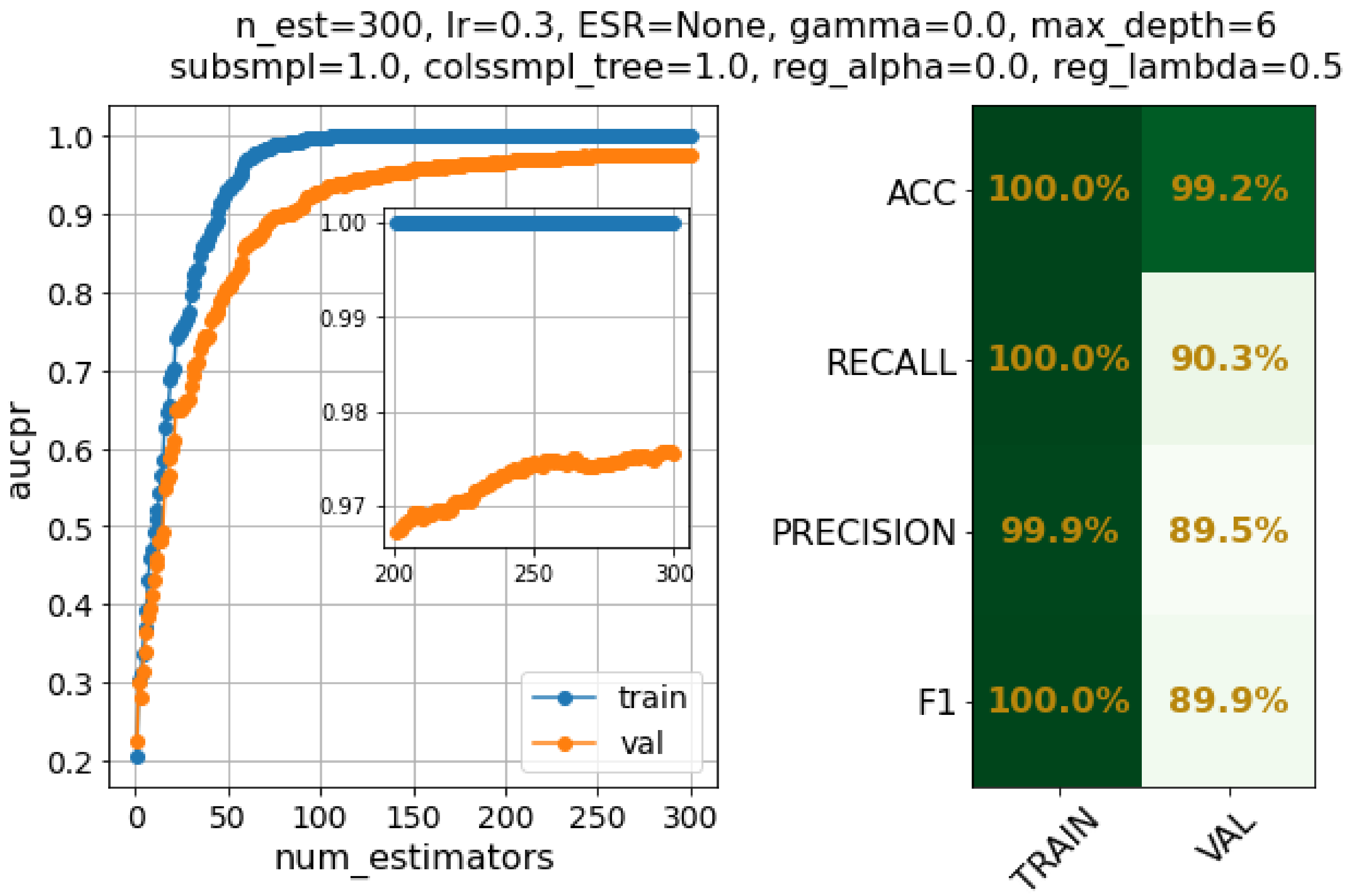


ACC
RECALL
PRECISION
F1
TRAIN
VAL
100.0%
99.2%
100.0%
88.5%
100.0%
90.5%
100.0%
89.5%

Add L1 regularization



Add L2 regularization



Hyperparameter Exploration: Grid Search

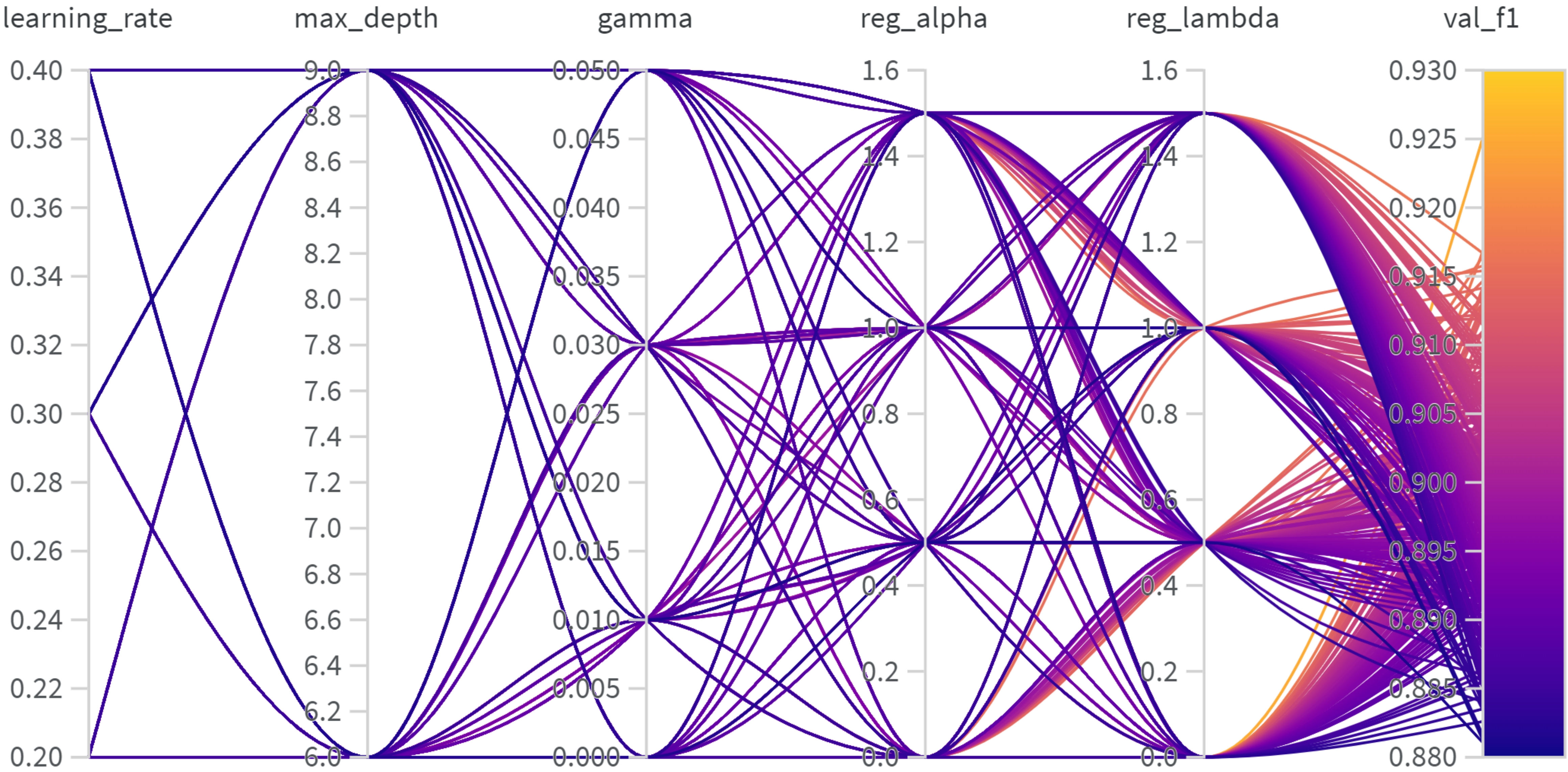
Hyperparameter Values Explored

eta (learning_rate)	0.2, 0.3, 0.4
num_boost_round (n_estimators)	$default\_num\_boost\_round \times default\_eta/eta \times default\_max\_depth/max\_depth$
max_depth	6, 9
gamma	0.0, 0.1, 0.03, 0.05
reg_alpha (L1)	0.0, 0.5, 1.0, 1.5
reg_lambda (L2)	0.0, 0.5, 1.0, 1.5
Total Combinations	384

Fixed Parameters

objective	binary::logistic
default_eta	0.3
default_num_boost_round	300
default_max_depth	6
subsample	1
colsample_by*	1
min_child_weight	1
Early_stopping	None

Hyperparameter Exploration: Results

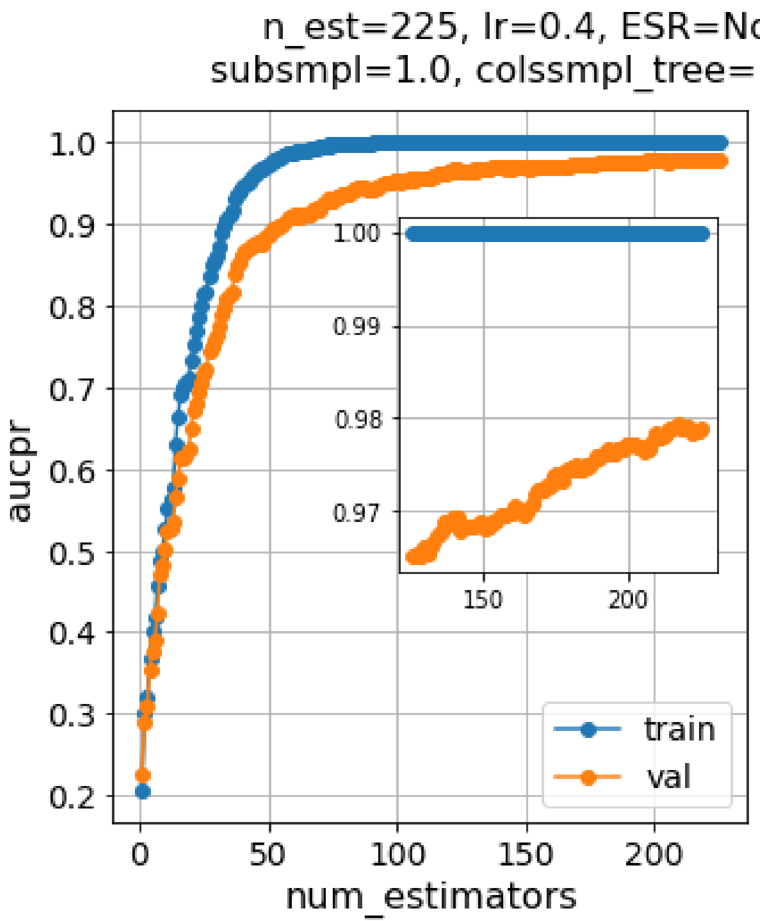


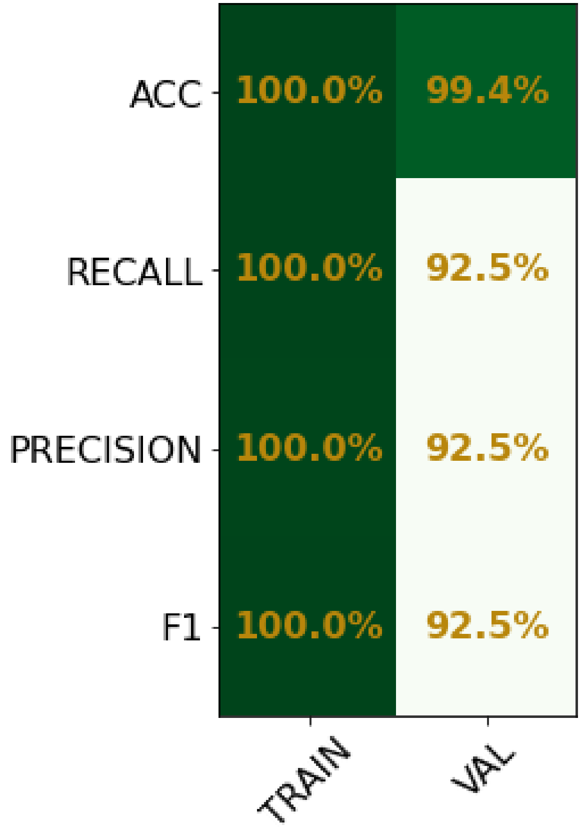
Hyperparameter Exploration: Best HP Values

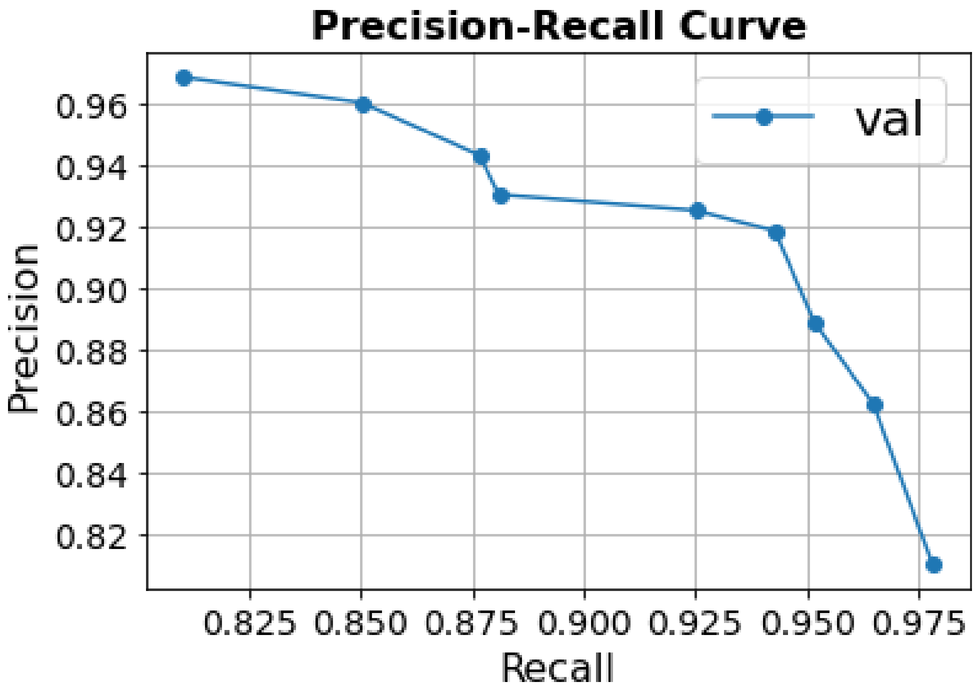
	Best F1-Score & Best Precision	Best Recall
eta (learning_rate)	0.4	0.2
num_boost_round (n_estimators)	225	450
max_depth	6	6
gamma	0	0.03
reg_alpha (L1)	0	1
reg_lambda (L2)	0	1.5
VAL. RECALL	92.5%	94.7%
VAL. PRECISION	92.5%	88.1%
VAL. F1-Score	0.925	0.913

Selected This One

Final Model: Training + Precision-Recall Curve







Final Model: Feature Importance