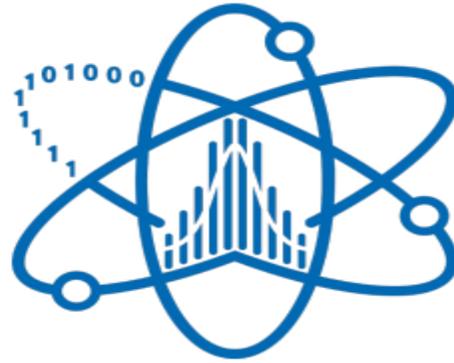




National Research
**Tomsk
State
University**



**Лаборатория
анализа данных
физики высоких энергий**

Томского
государственного
университета

**Measurement of differential cross-sections of a single top quark
produced in association with a W boson with ATLAS at
 $\sqrt{s} = 13$ TeV**

Progress Report

Neda Firoz

Goal: separate tW (top+anti-top) from $t\bar{t}$ in the 1j1b dilepton region

•Inputs (9 variables used):

bdt_centrality_1l_recalc_NOSYS,
bdt_delta_pT_1l_MET_recalc_NOSYS,
S,
bdt_delta_pT_1lb_MET_recalc_NOSYS,
YS,
bdt_eta_1lMetB_recalc_NOSYS,
bdt_m_11b_recalc_NOSYS,
bdt_m_12b_recalc_NOSYS,
bdt_pT_1lMetB_recalc_NOSYS,
bdt_pT_1lb_recalc_NOSYS,
bdt_sum_ET_recalc_NOSYS.

•**Samples / tree:** all files' tree name is analysis

Signal: tW (top) + $t\bar{W}$ (anti-top)

Background: $t\bar{t}$ (non-all-had)

•**Event weights:** auto-resolved to
 $\text{weight_mc_NOSYS} * \text{weight_pileup_NOSYS}$
 $* \text{globalTriggerEffSF_NOSYS}$.

•**Bad values:** any variable ≤ -990 or non-finite is masked per event.

Article's Report on BDT

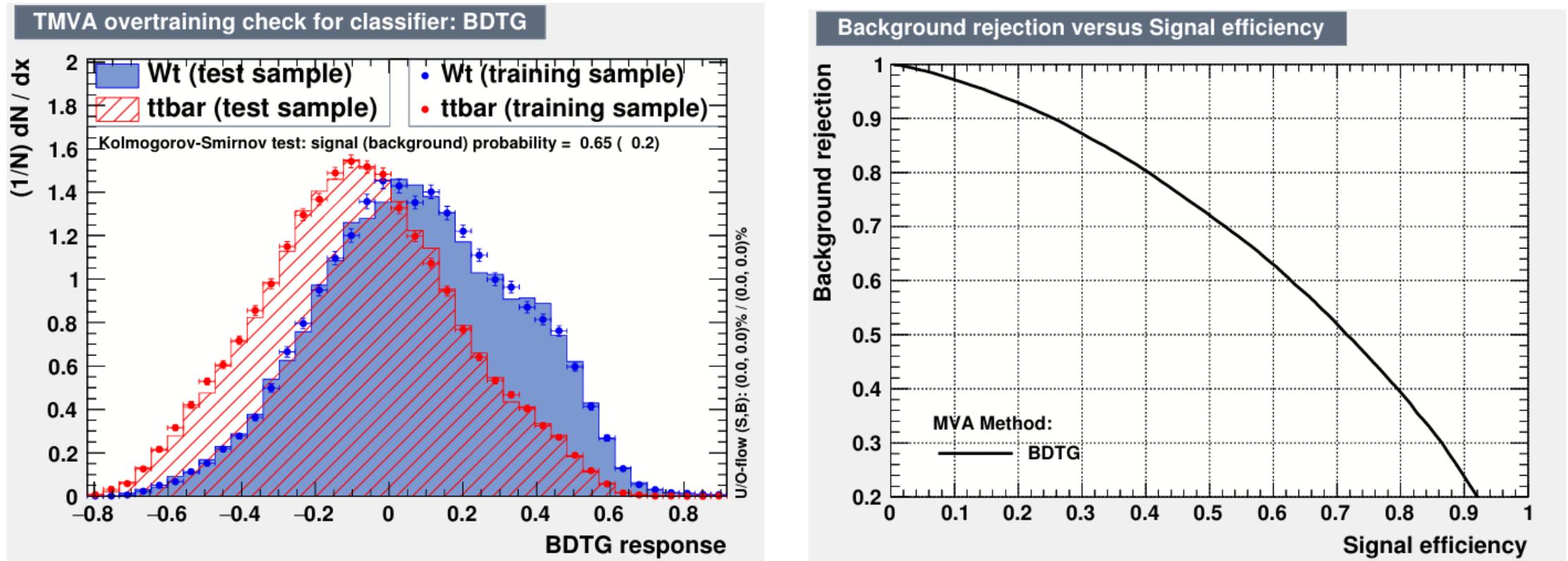


Figure 9: Comparison of test/training sample distributions and background rejection factor versus signal efficiency.

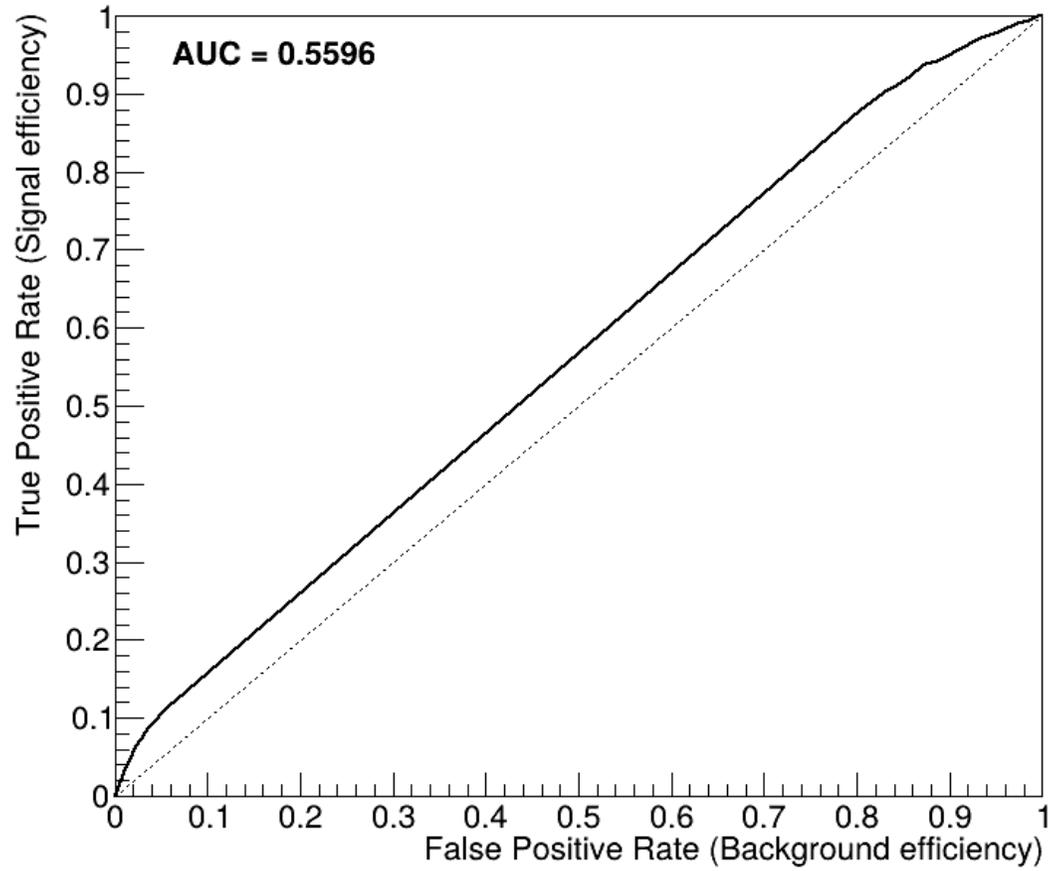
Results of Performance of ML Algorithms using K-fold on TMVA

27th January 2026

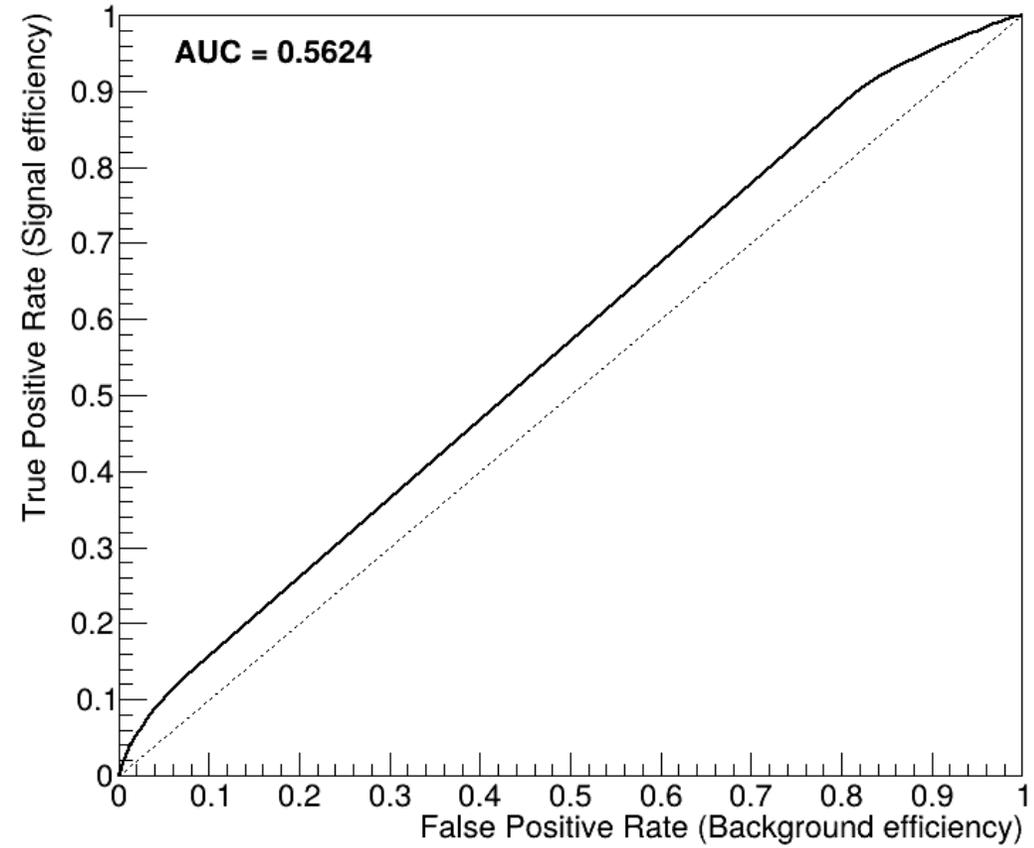
Algorithms	ROC_AUC_MEAN	ROC_AUC_STD	n_folds_used
BDTB	0.560854	0.003277	5
BDTG	0.55978	0.003049	5
Fisher	0.550102	0.00274	5
KNN	0.549632	0.003916	5
LD	0.547795	0.001821	5
Likelihood	0.523526	0.004158	5
MLP	0.561797	0.002013	5

I was doing an ordinary train/test split before cross-validation (CV). So, CV was effectively training on (roughly) half the data, then splitting that again into folds. That alone could have pushed AUC down noticeably and make results unstable. So, I faced three times unstable results on tmva. I am still trying to prepare correct code with kfold on TMVA.

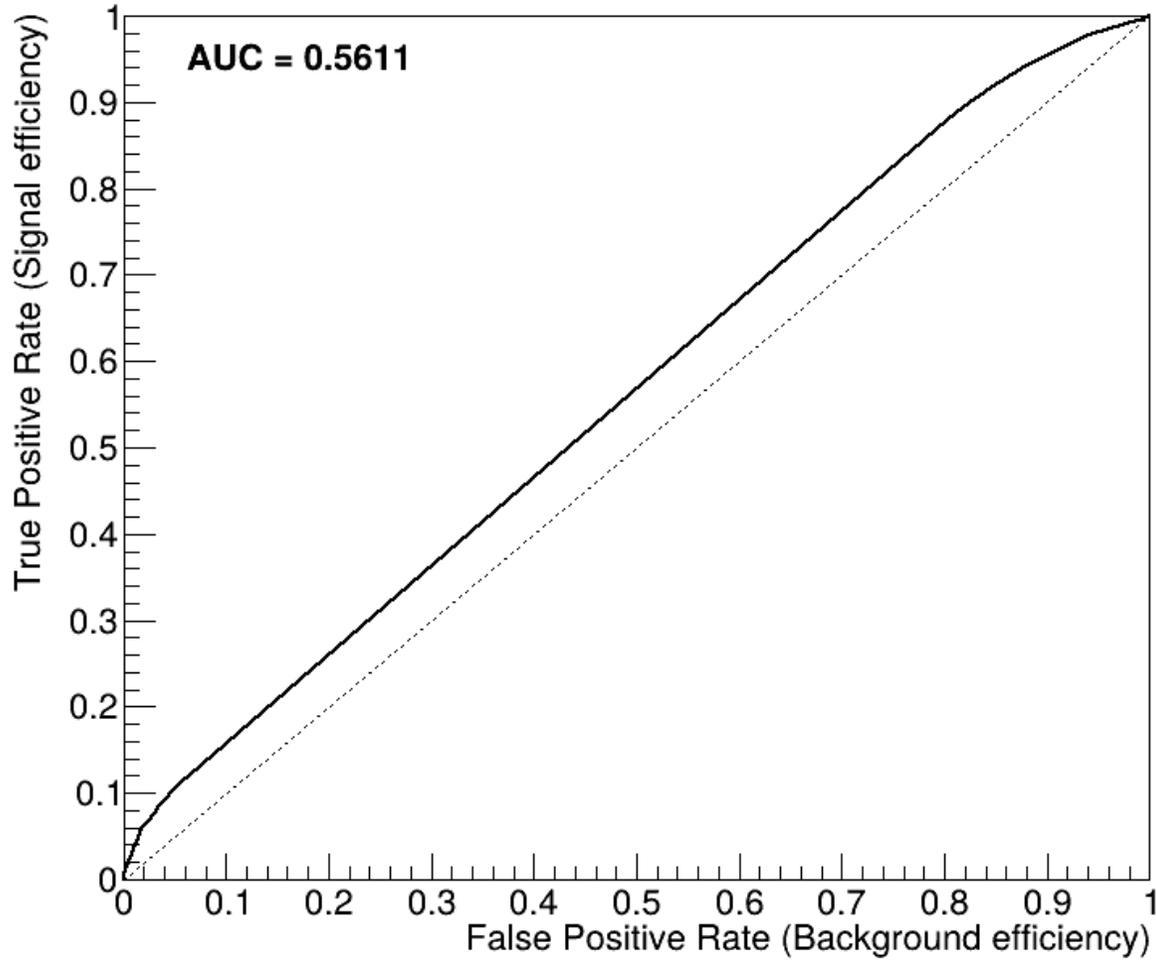
BDTG fold 3 ROC



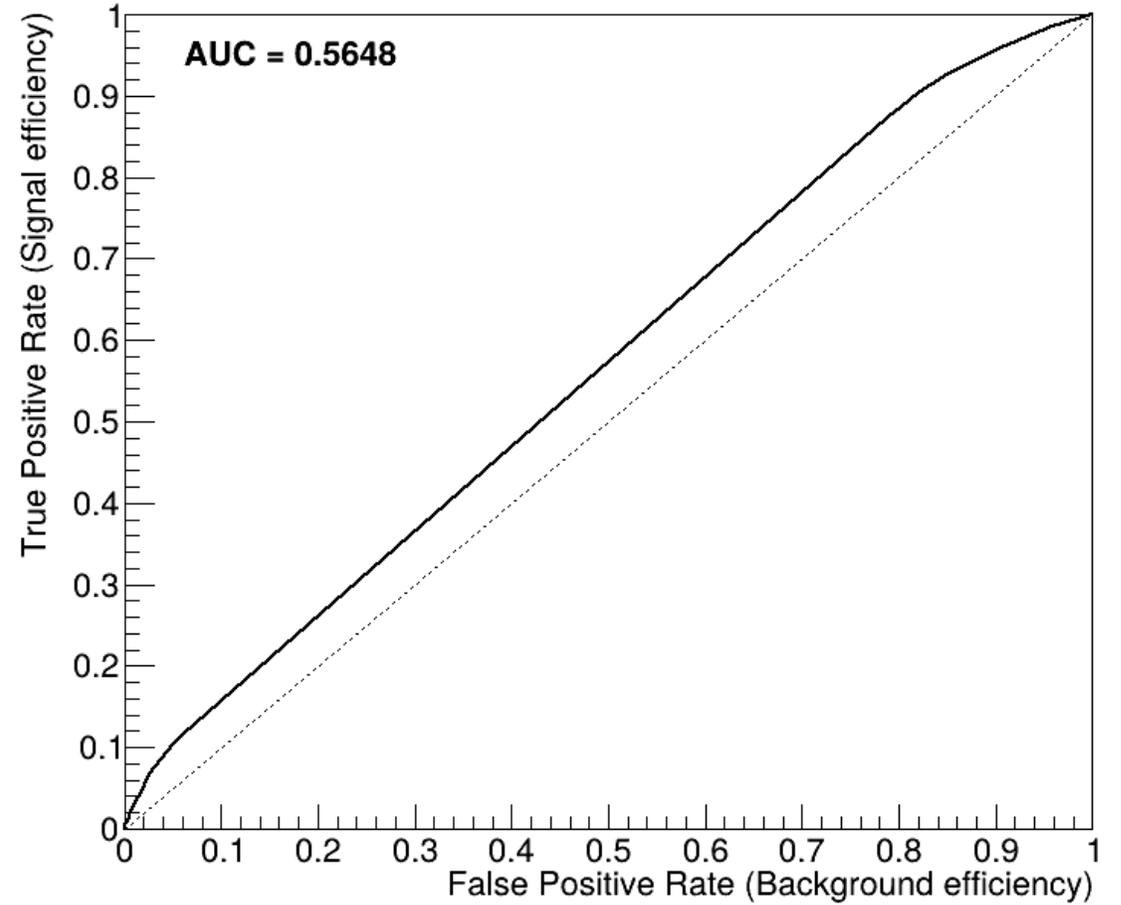
BDTG fold 5 ROC



MLP fold 3 ROC



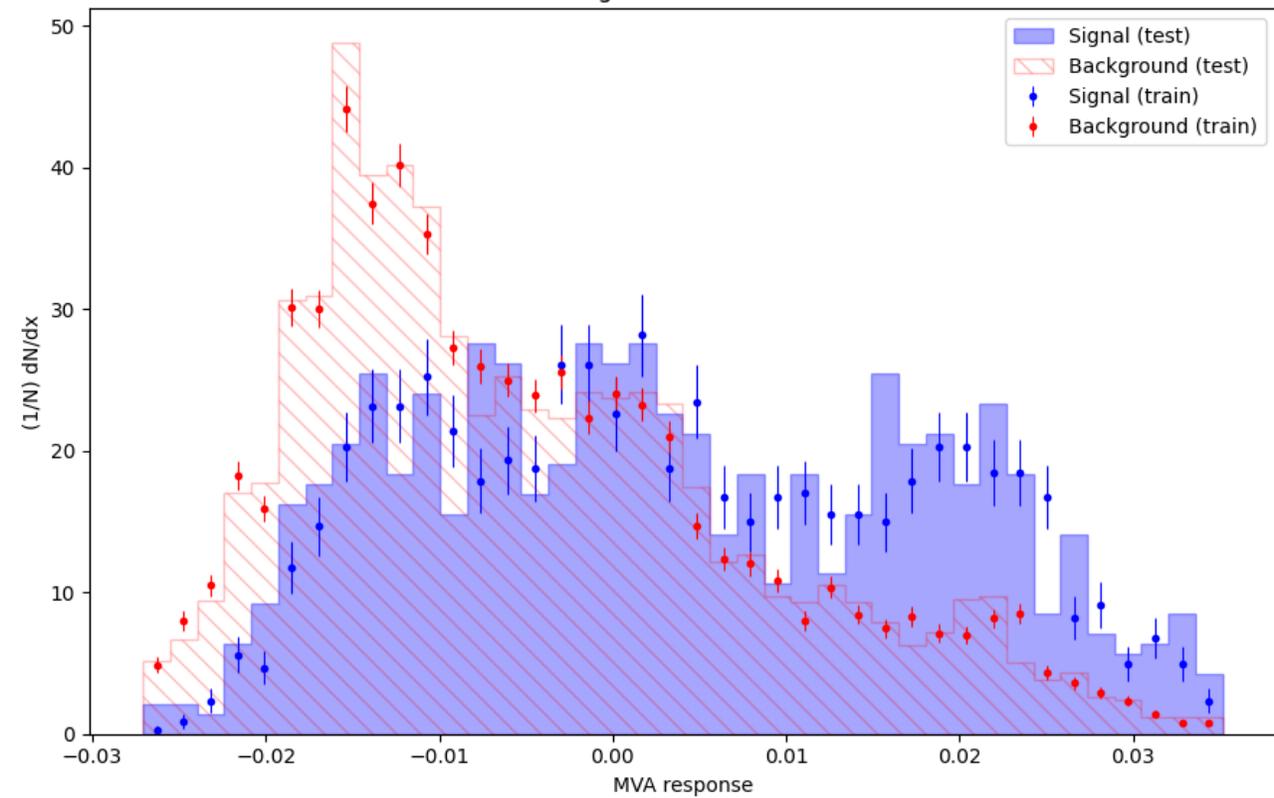
MLP fold 5 ROC



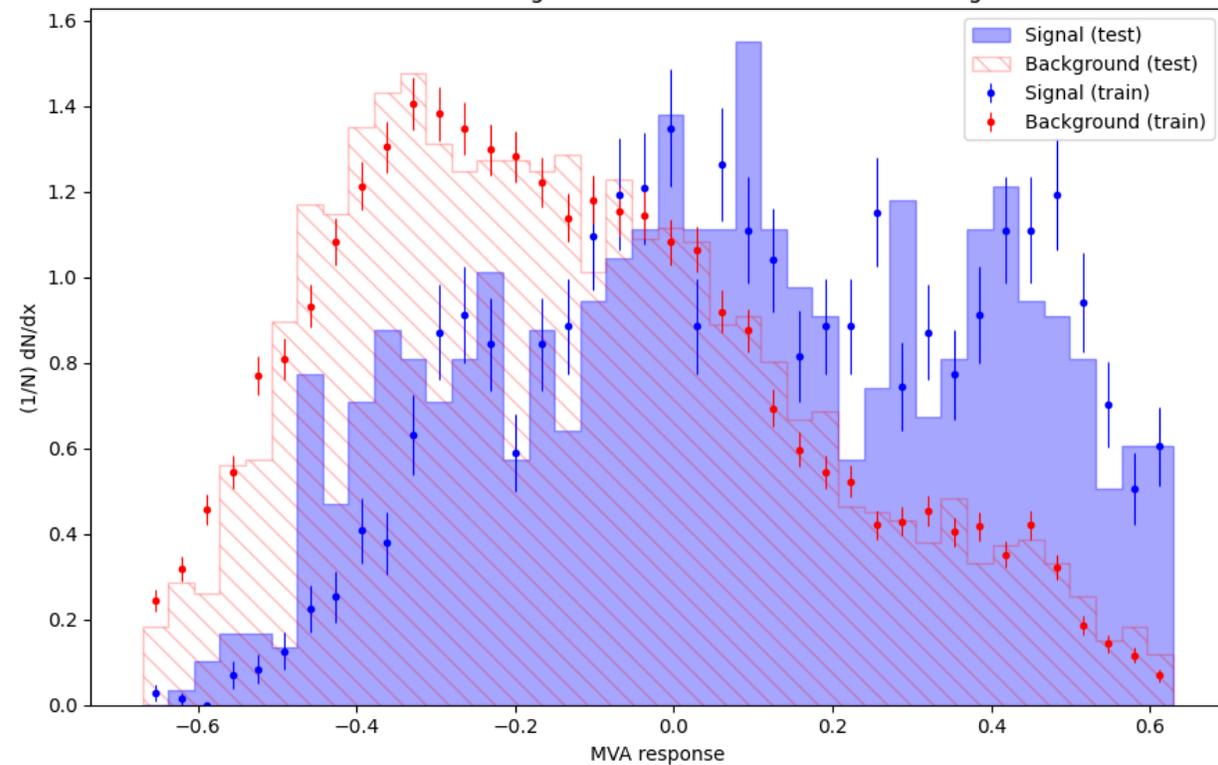
**Results of Performance of ML
Algorithms using K-fold on
Python Trial 3
27th January 2026**

Algorithms	outer_k	inner_k	n_iters	mean_fold_auc	std_fold_auc	overall_oof_auc
AdaBoost	5	3	25	0.668024	0.008952	0.662409
Gaussian-NB	5	3	25	0.6407	0.006545	0.639867
Gradient Boosting	5	3	25	0.675134	0.01051	0.67482
KNN	5	3	25	0.643668	0.007173	0.643566
Logistic Regression	5	3	25	0.639506	0.011228	0.639507
MLP	5	3	25	0.665761	0.010691	0.664427
Random Forest	5	3	25	0.672187	0.010309	0.671608
SVC	5	3	10	0.676572	0.01301	0.676277

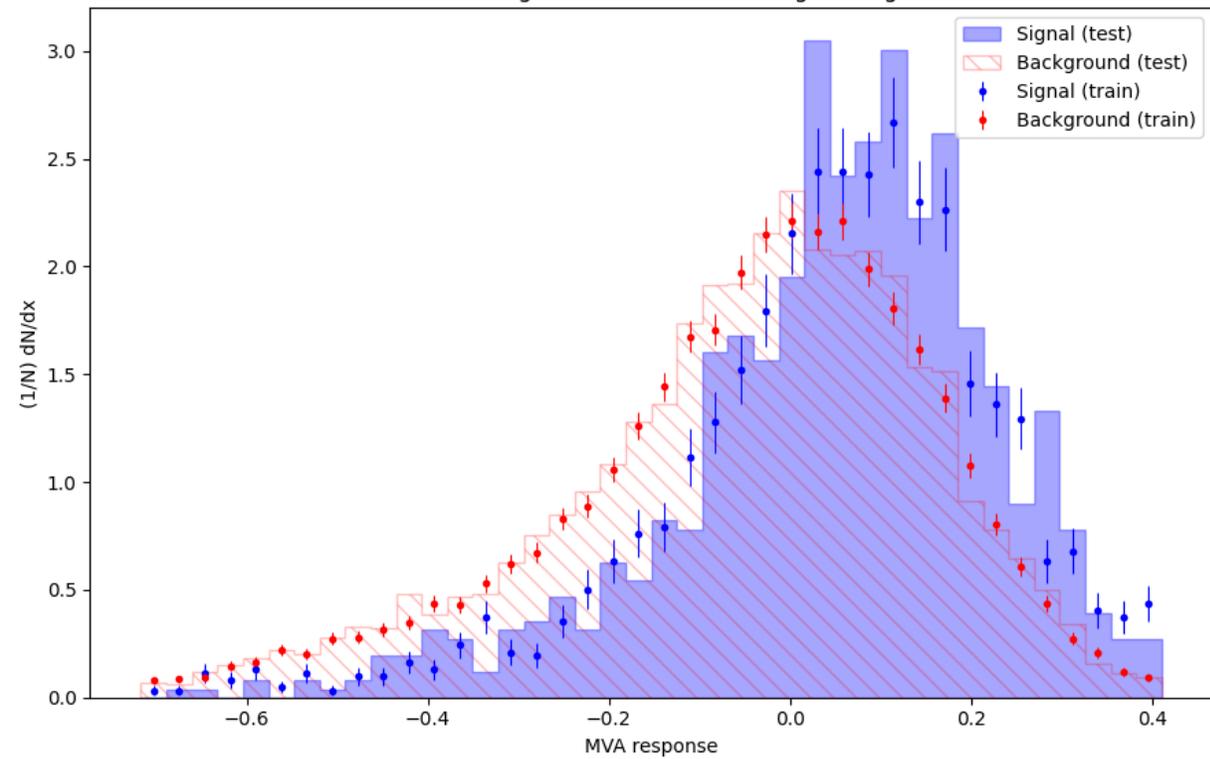
TMVA overtraining check for classifier: AdaBoost



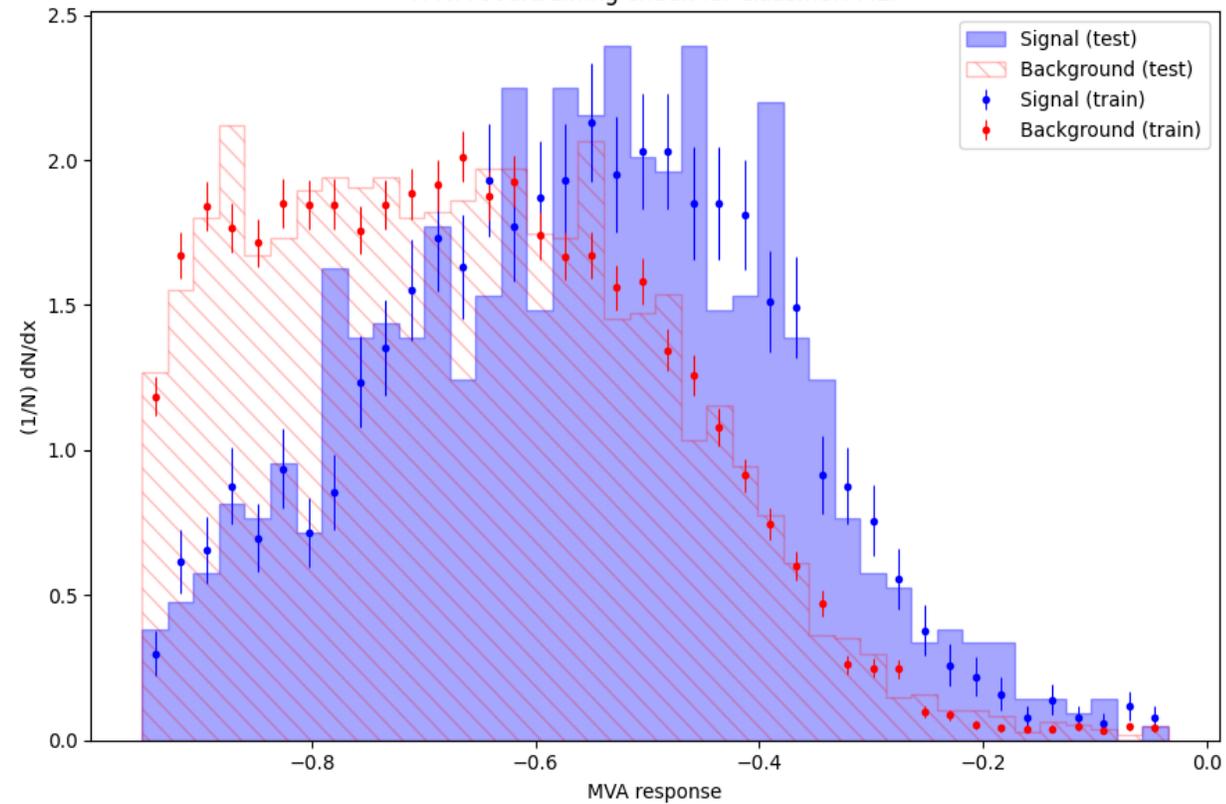
TMVA overtraining check for classifier: GradientBoosting



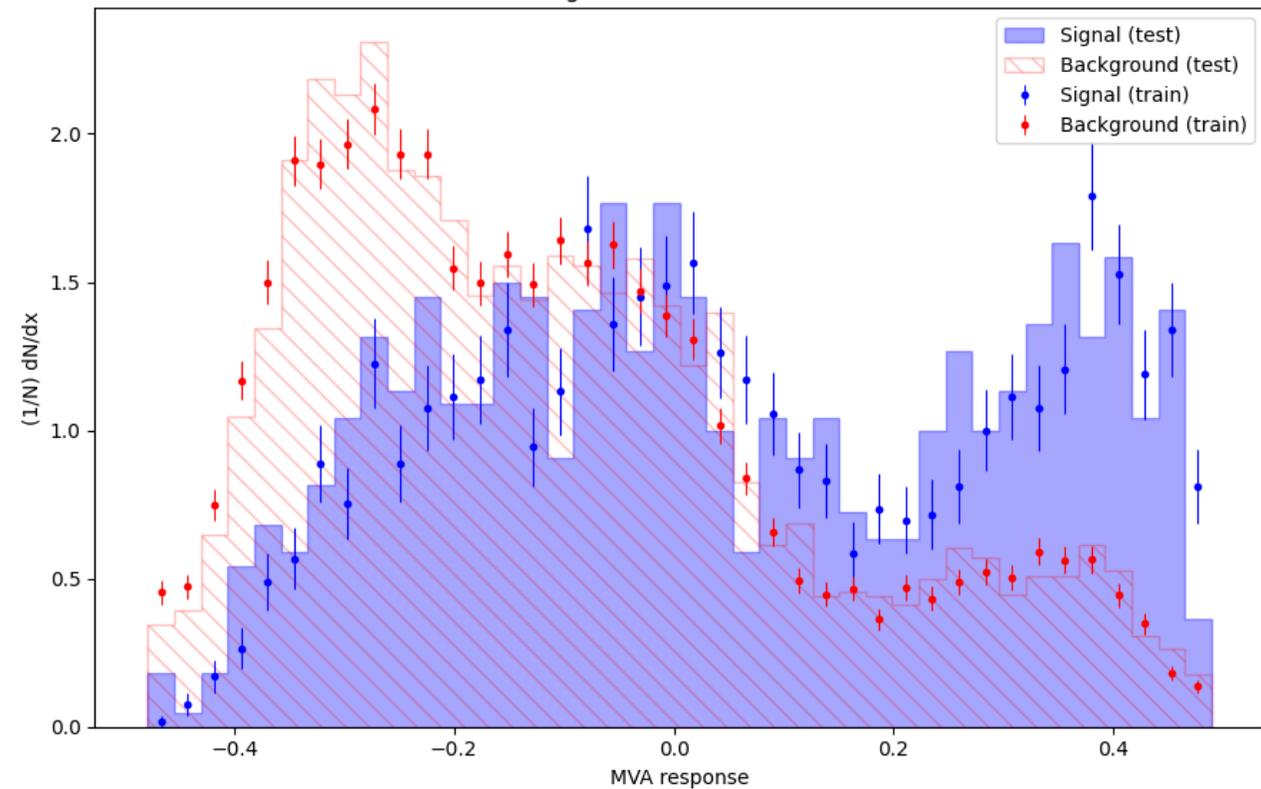
TMVA overtraining check for classifier: LogisticRegression



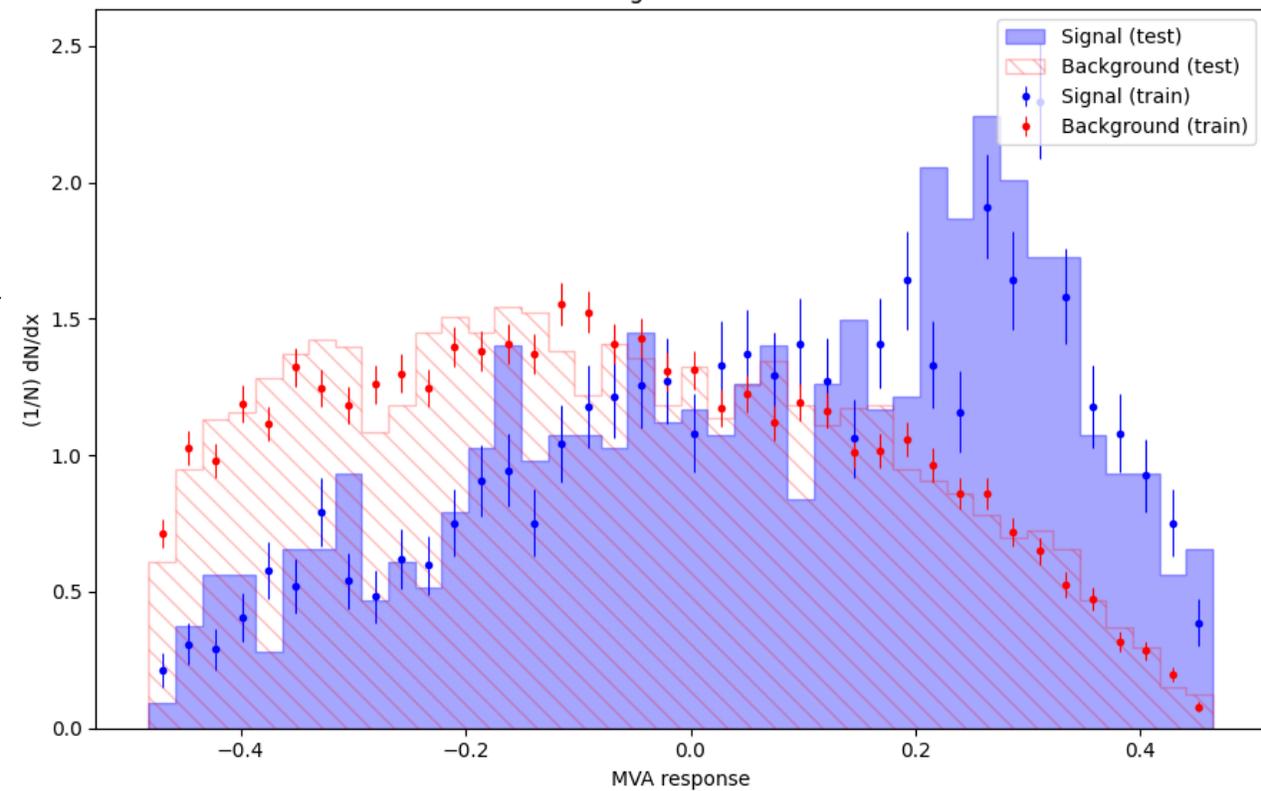
TMVA overtraining check for classifier: MLP



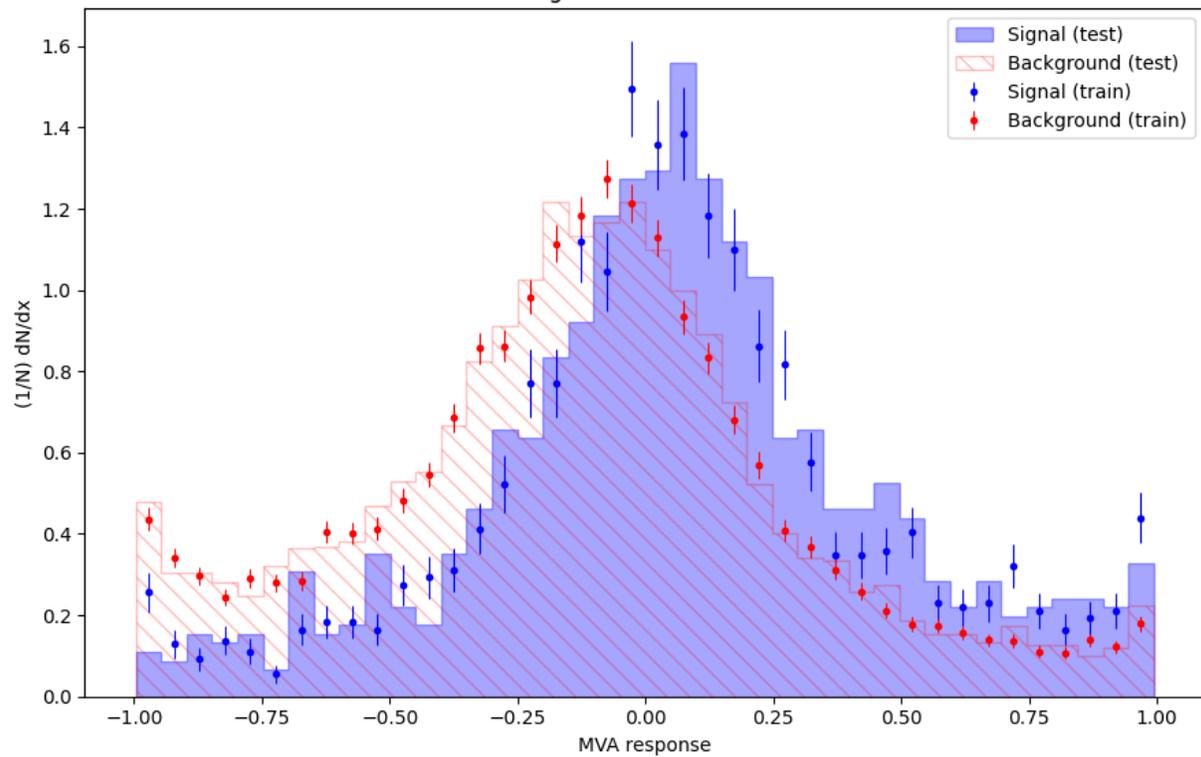
TMVA overtraining check for classifier: RandomForest



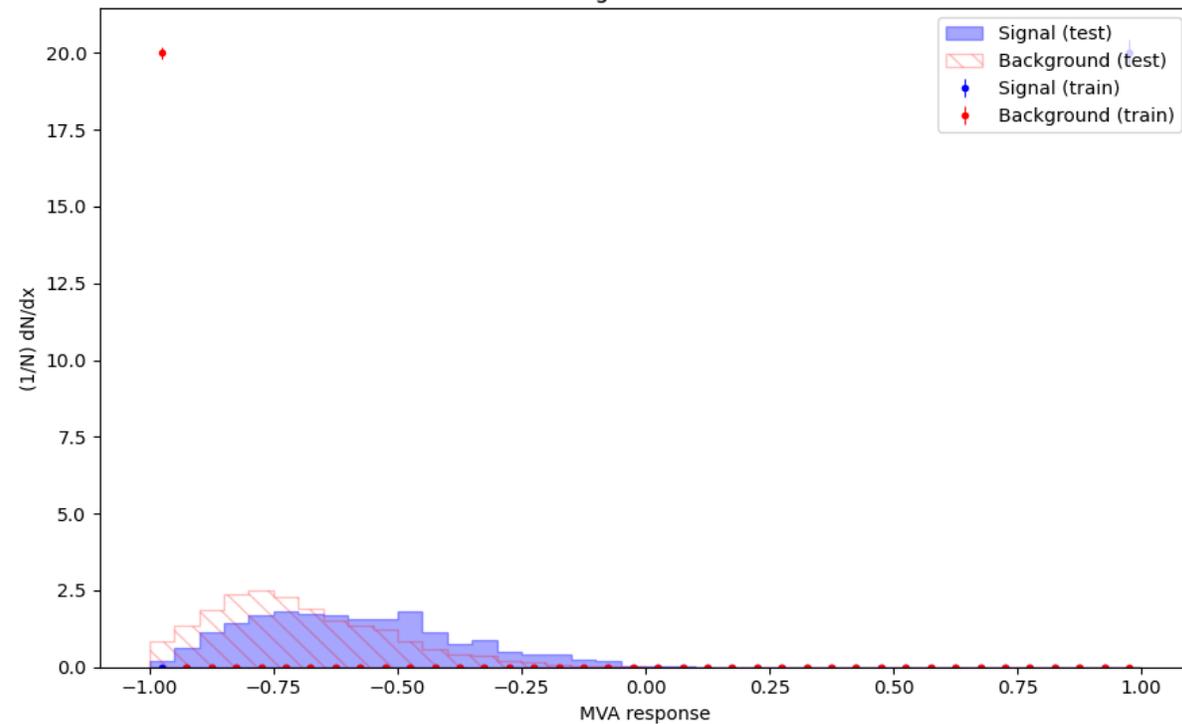
TMVA overtraining check for classifier: SVC



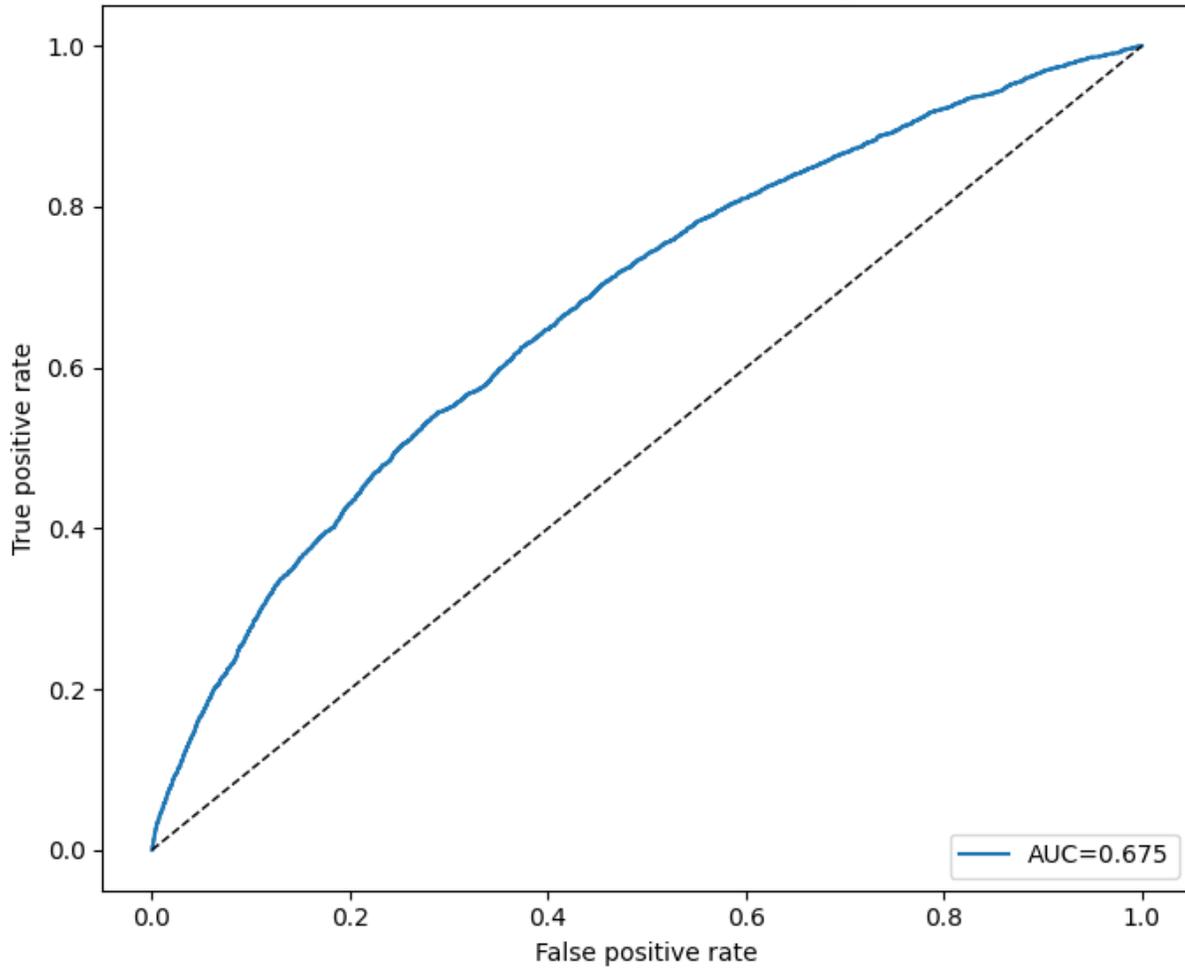
TMVA overtraining check for classifier: GaussianNB



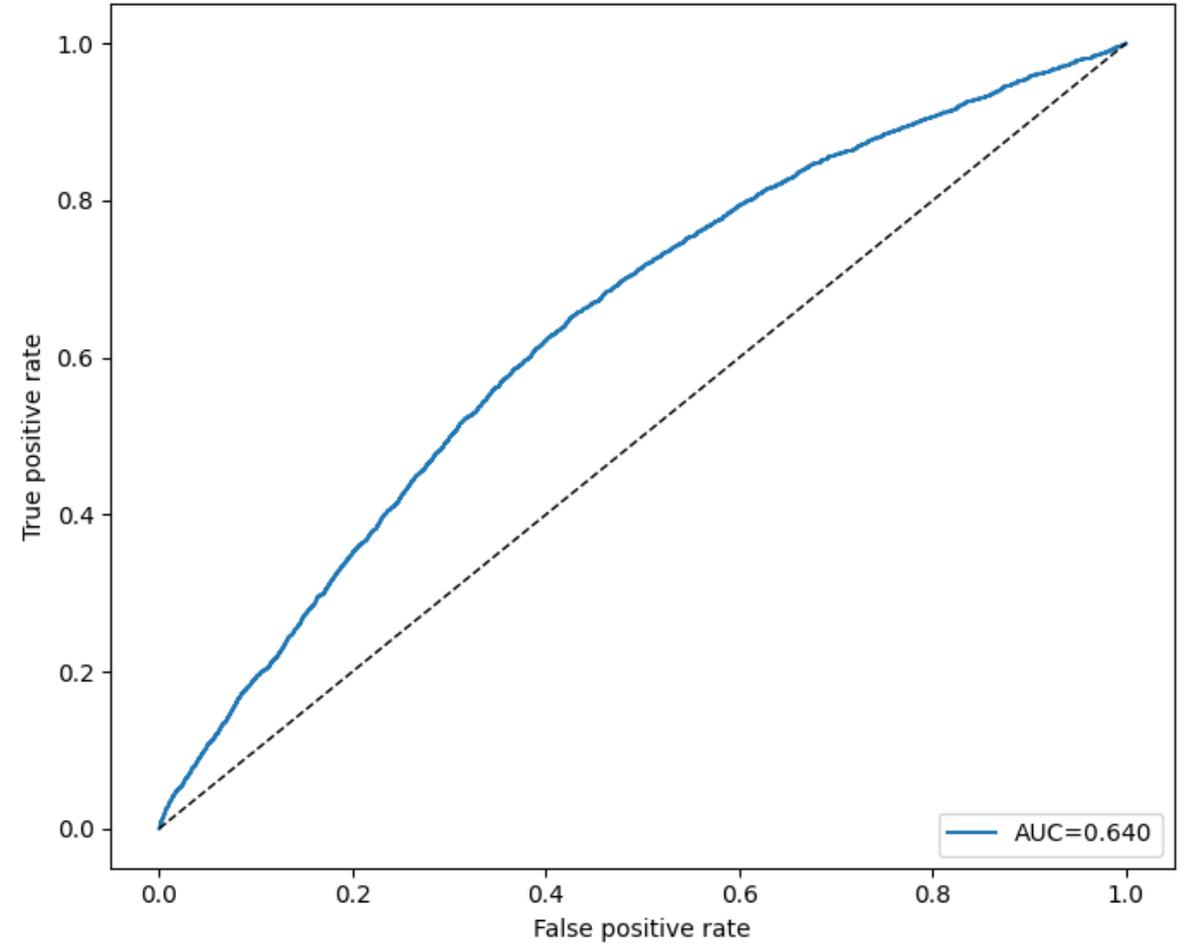
TMVA overtraining check for classifier: KNN



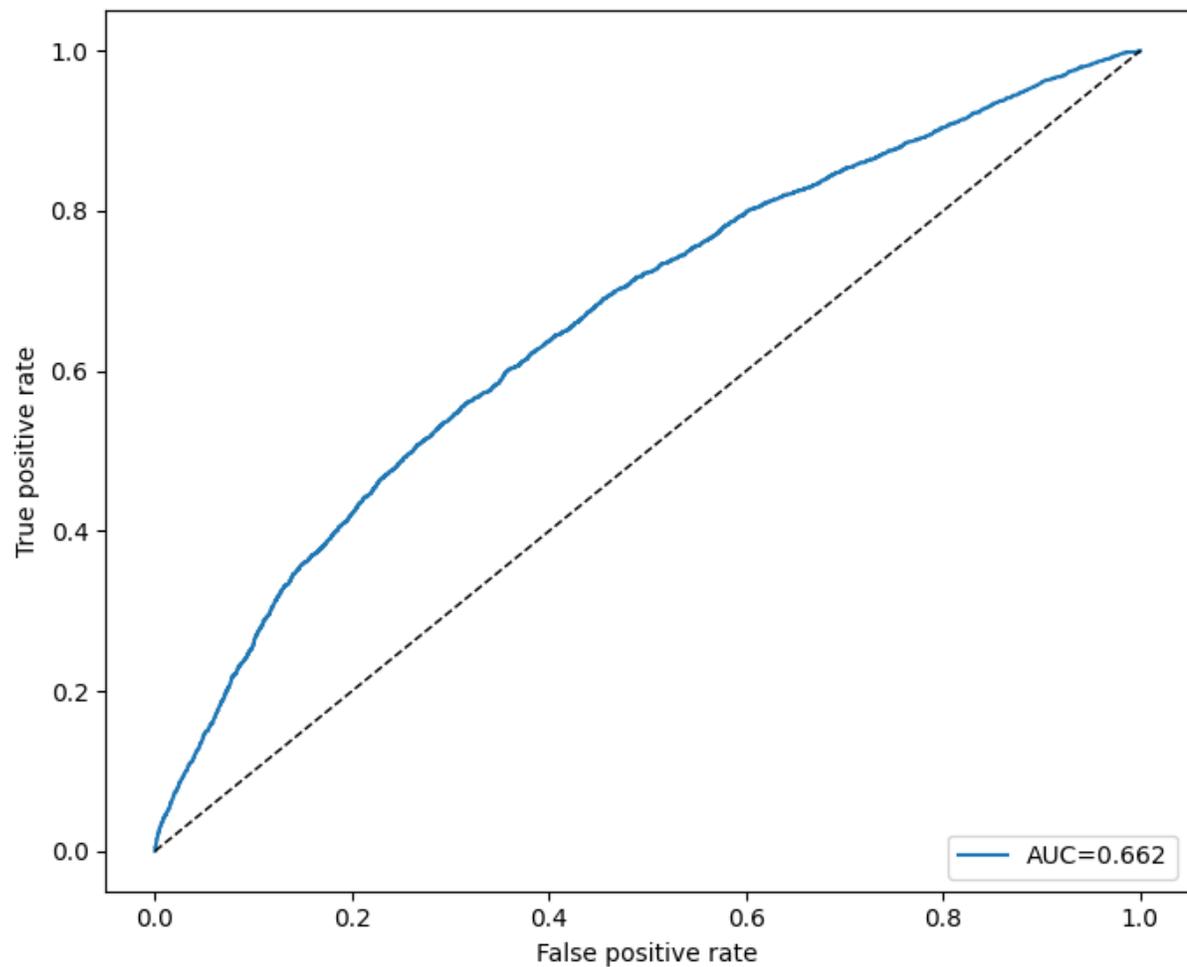
OOF ROC - GradientBoosting



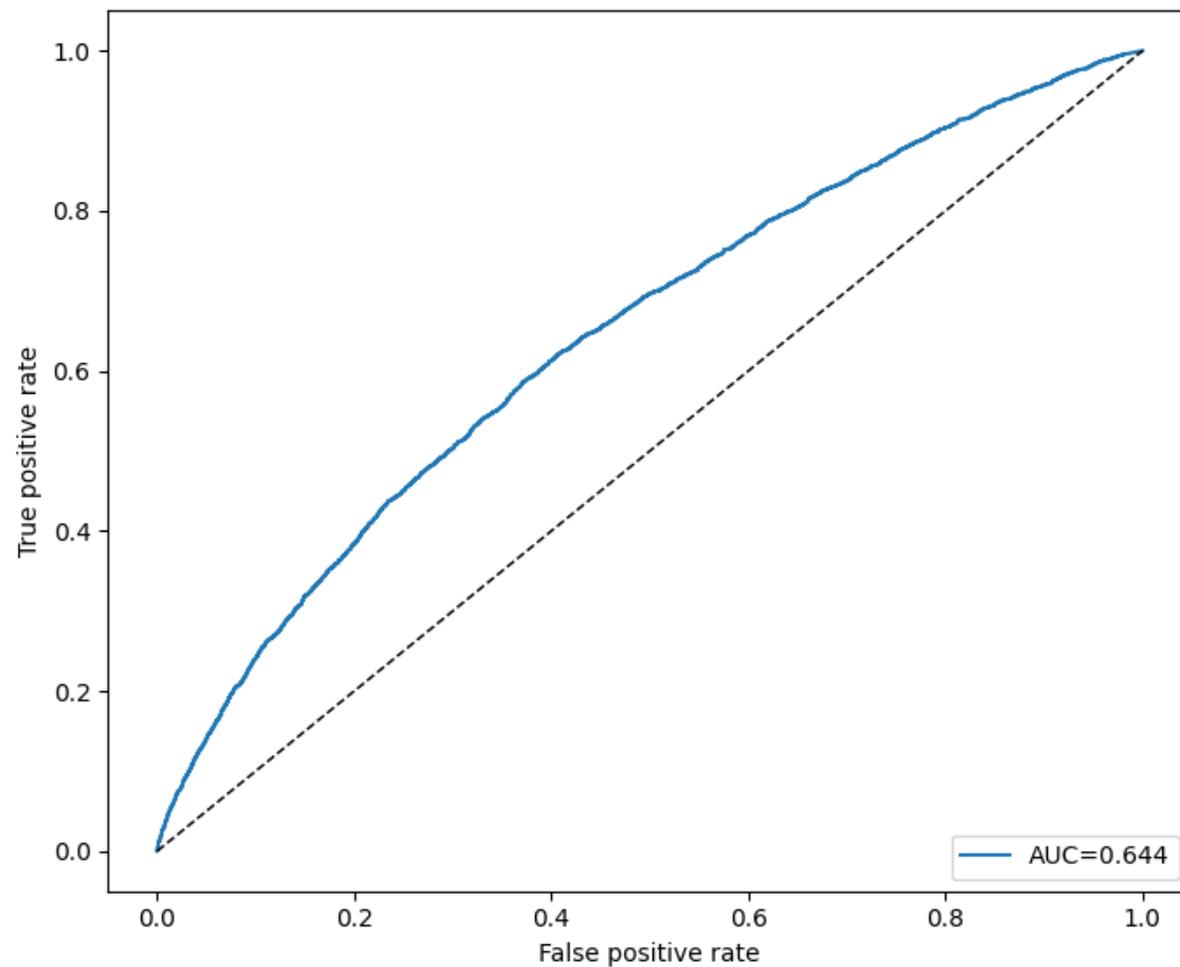
OOF ROC - GaussianNB



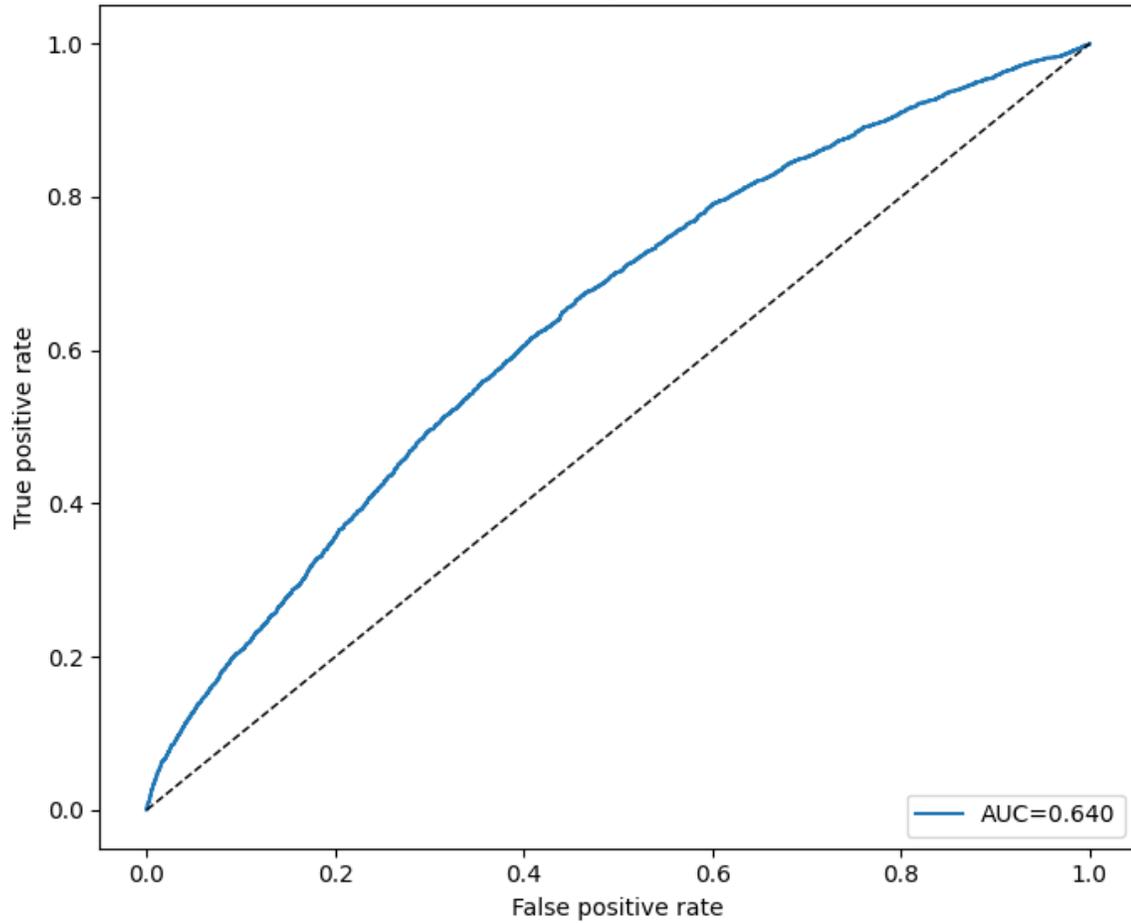
OOF ROC - AdaBoost



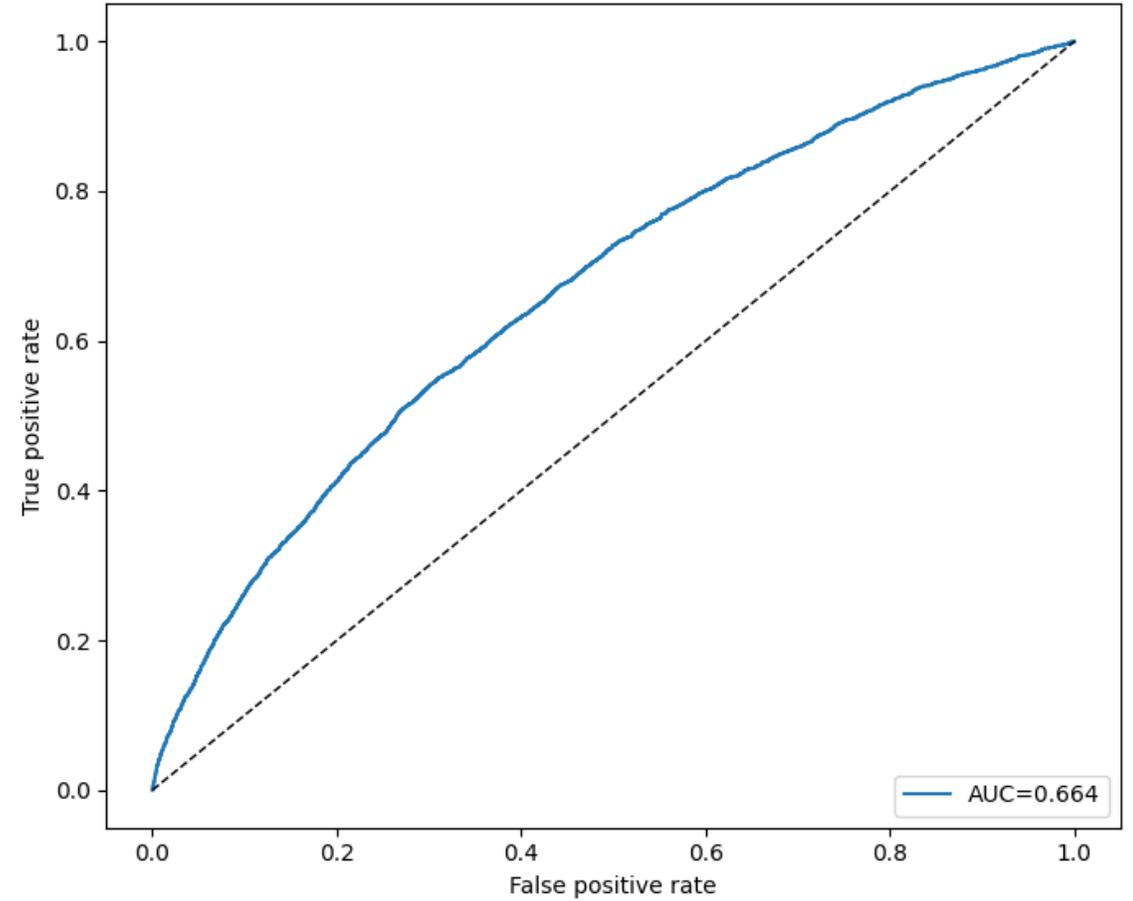
OOF ROC - KNN



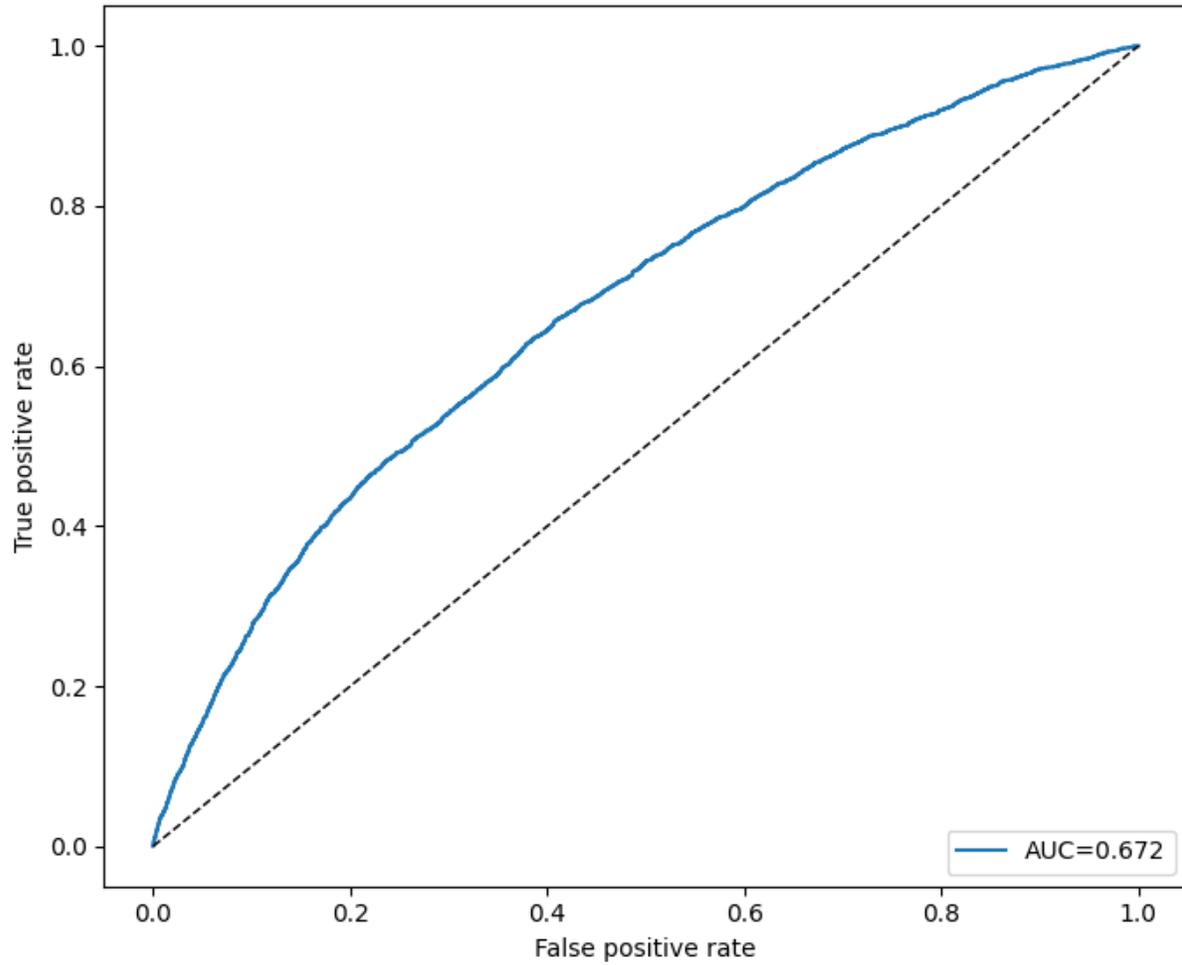
OOF ROC - LogisticRegression



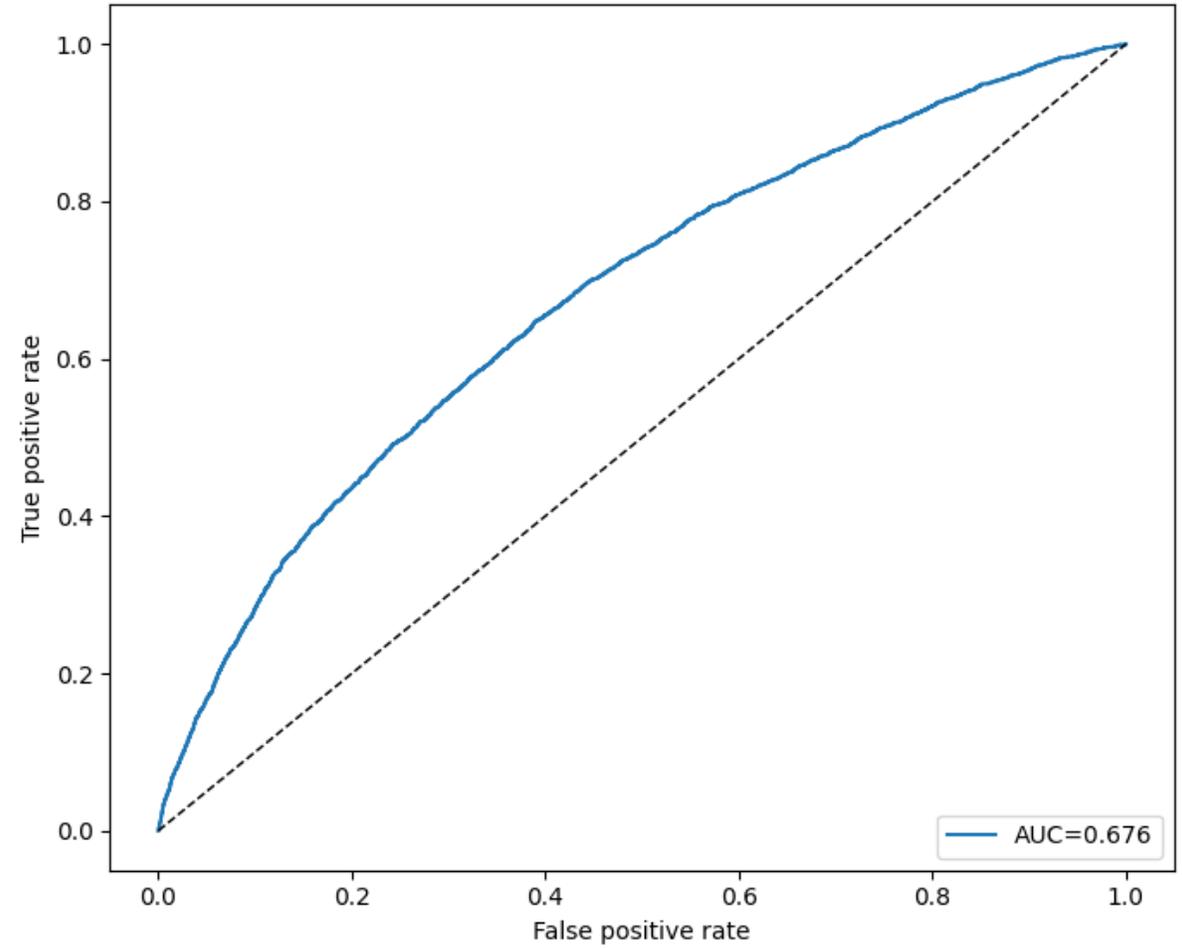
OOF ROC - MLP



OOF ROC - RandomForest



OOF ROC - SVC



Results of Performance of ML Algorithms using K-fold on R without caret package. 03 Feb 2026

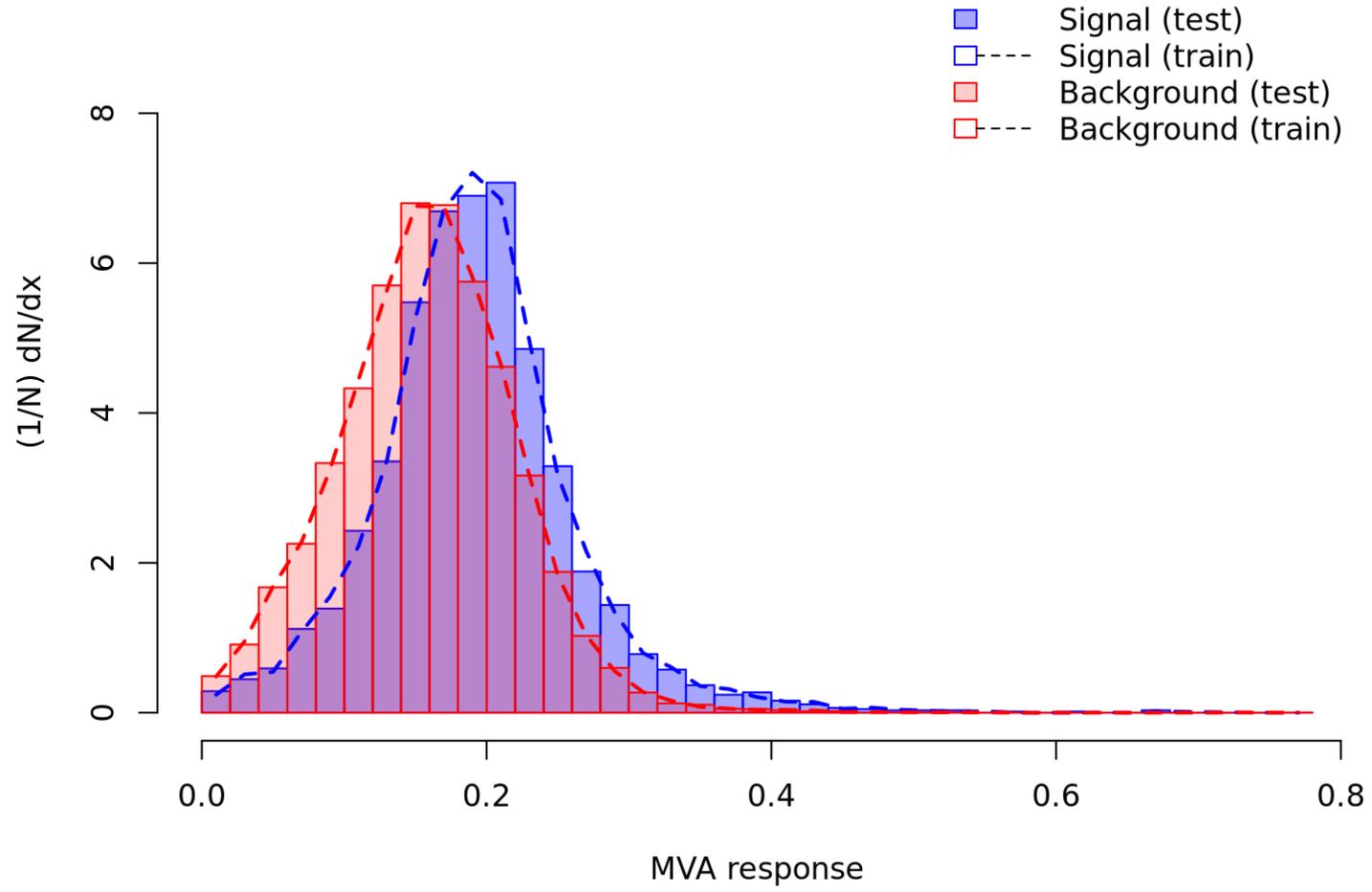
Methods: GLM, RF, GBM, XGBOOST, SVM,
KNN, NB, MLP

R-Platform (k-fold) Results on New Data

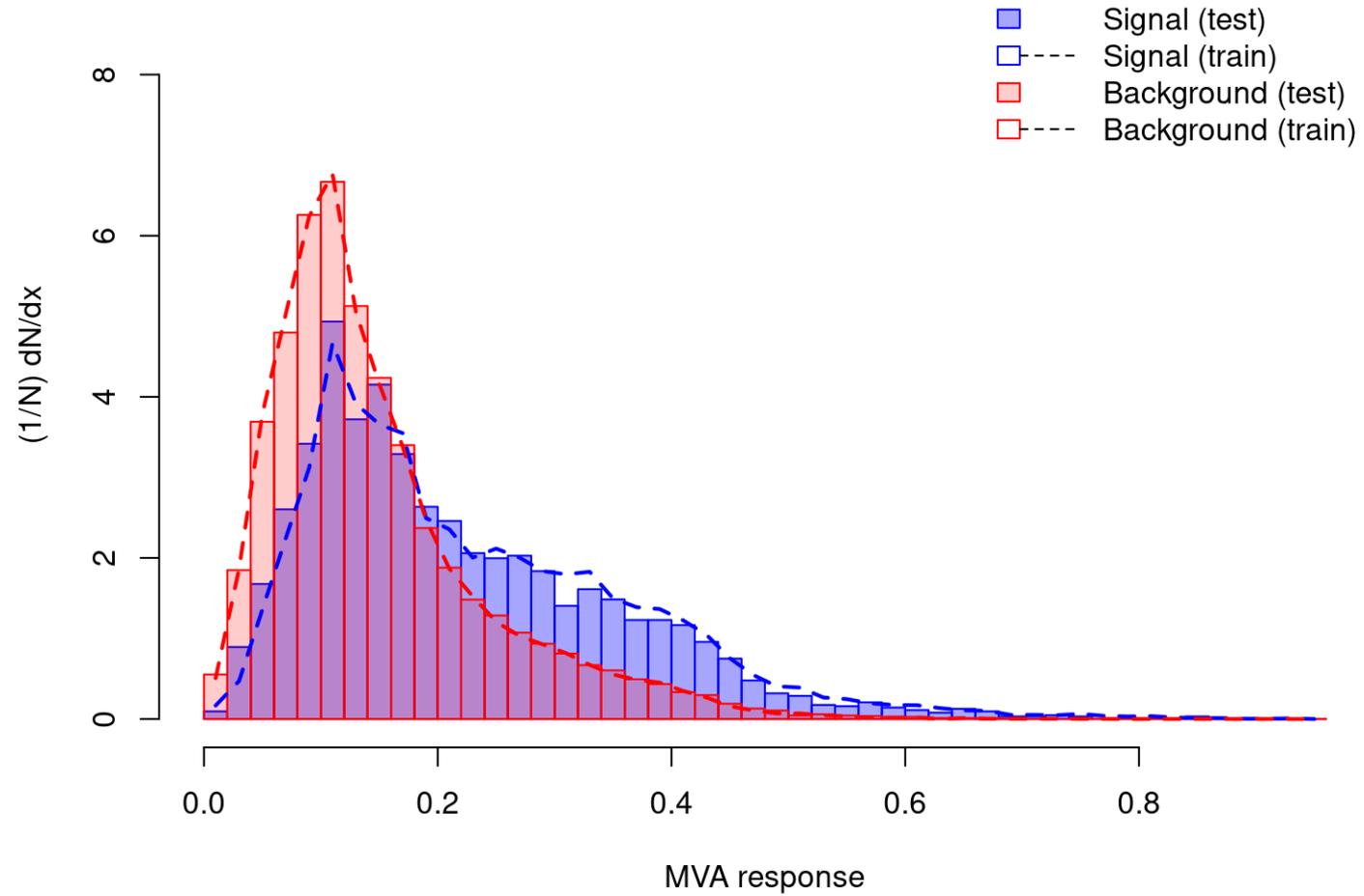
<i>Algorithm (R)</i>	<i>auc_oof_weighted</i>	<i>acc_oof_weighted</i>
GBM	1	1
MLP	0.660152	0.836995
XGBOOST	0.659179	0.835844
RF	0.656103	0.835792
GLM	0.643484	0.836315
NB	0.640344	0.813926
SVM	0.590384	0.836263

R overtraining check for classifier: glm

KS p-values (unweighted): S=0.974 B=0.999

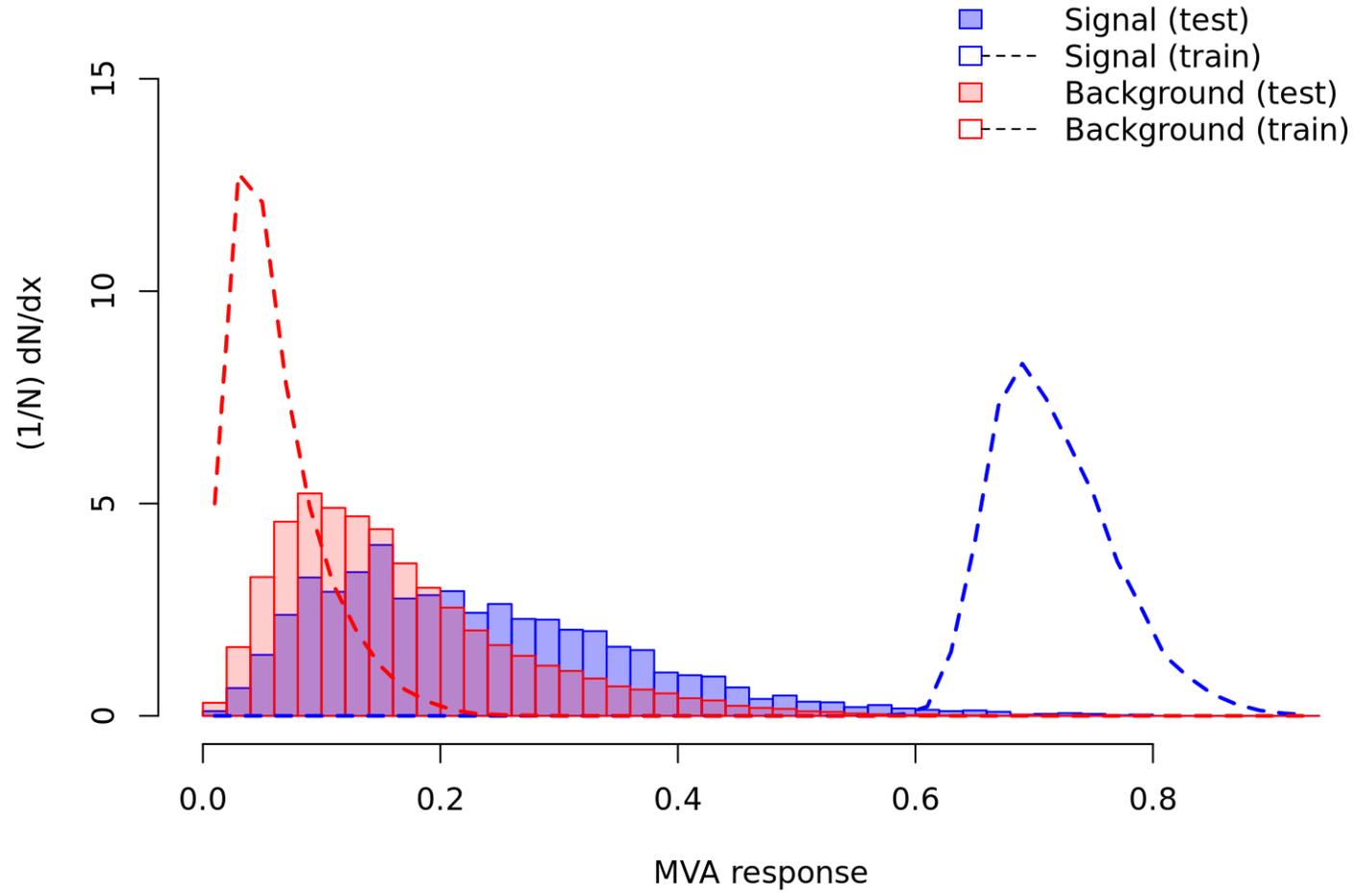


R overtraining check for classifier: mlp

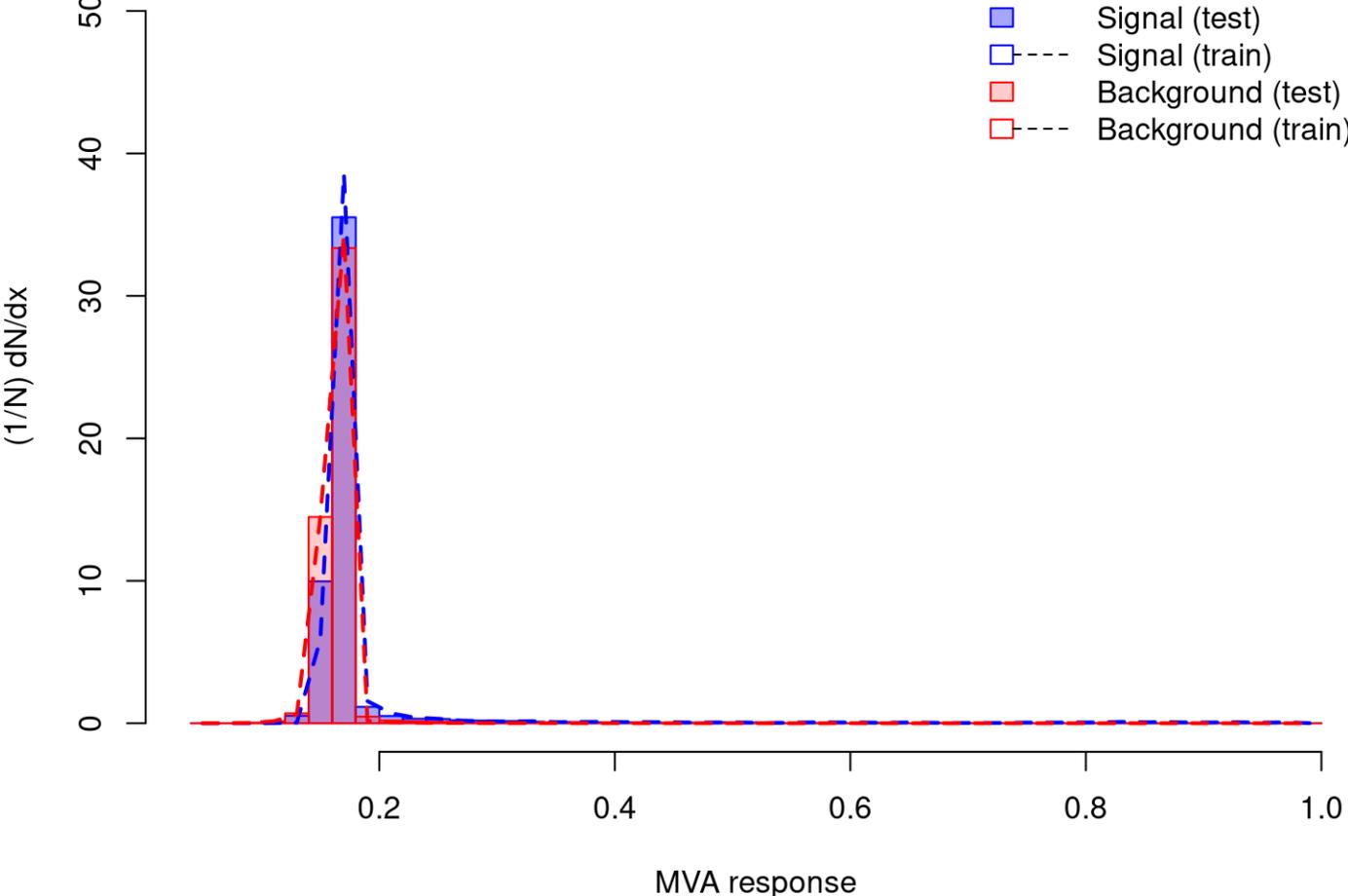


R overtraining check for classifier: rf

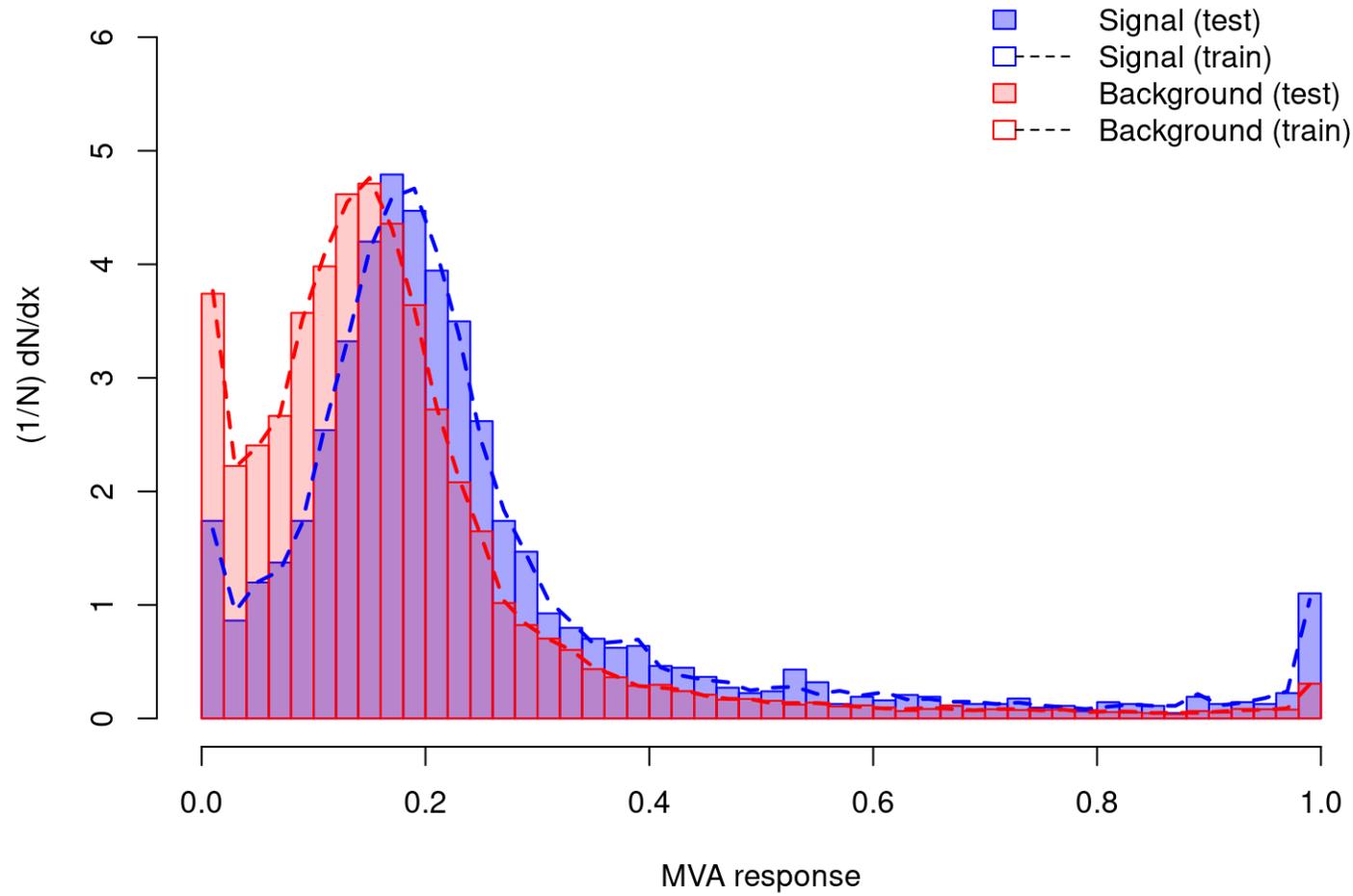
KS p-values (unweighted): S=0 B=0



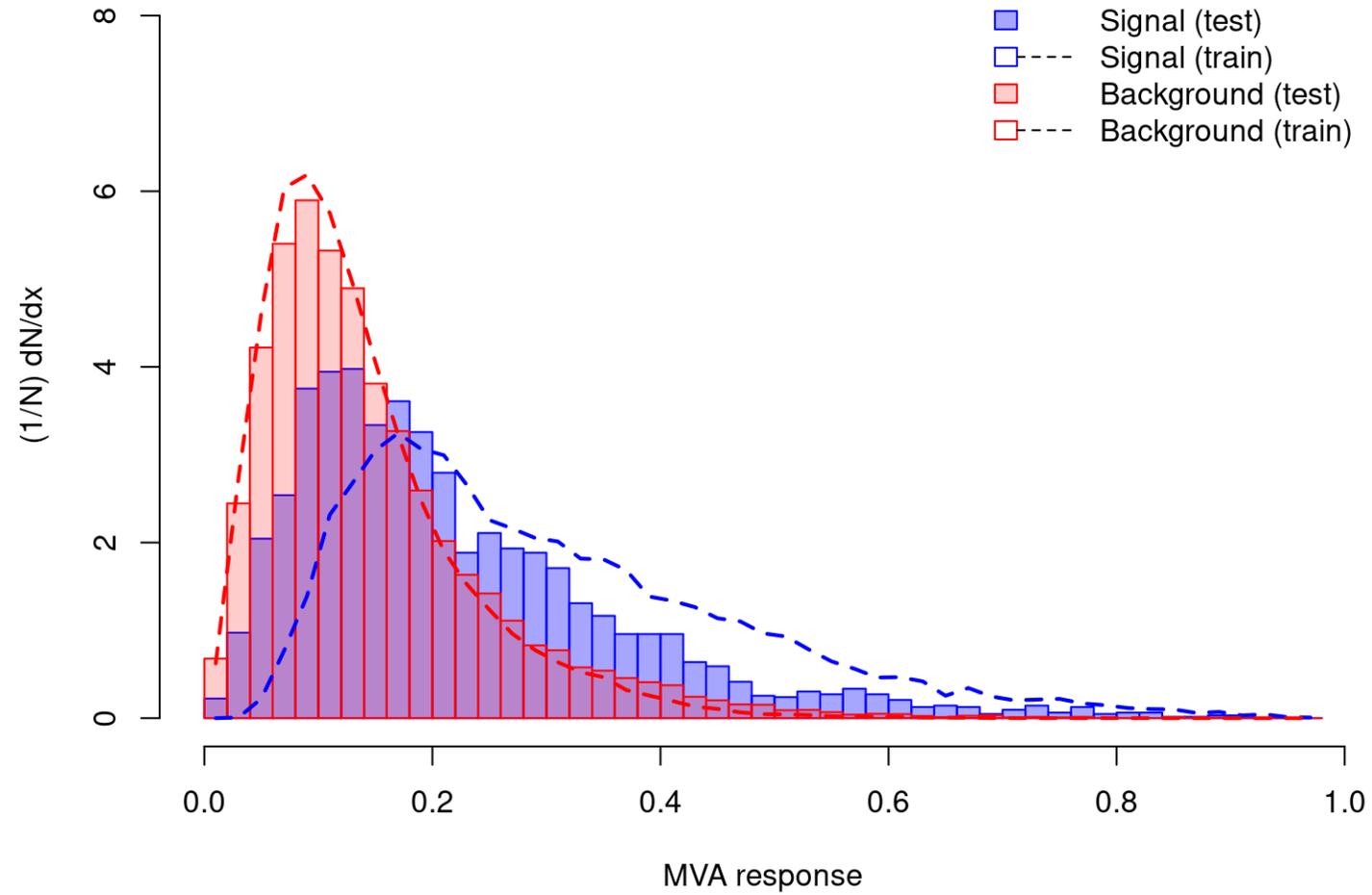
R overtraining check for classifier: svm



R overtraining check for classifier: nb

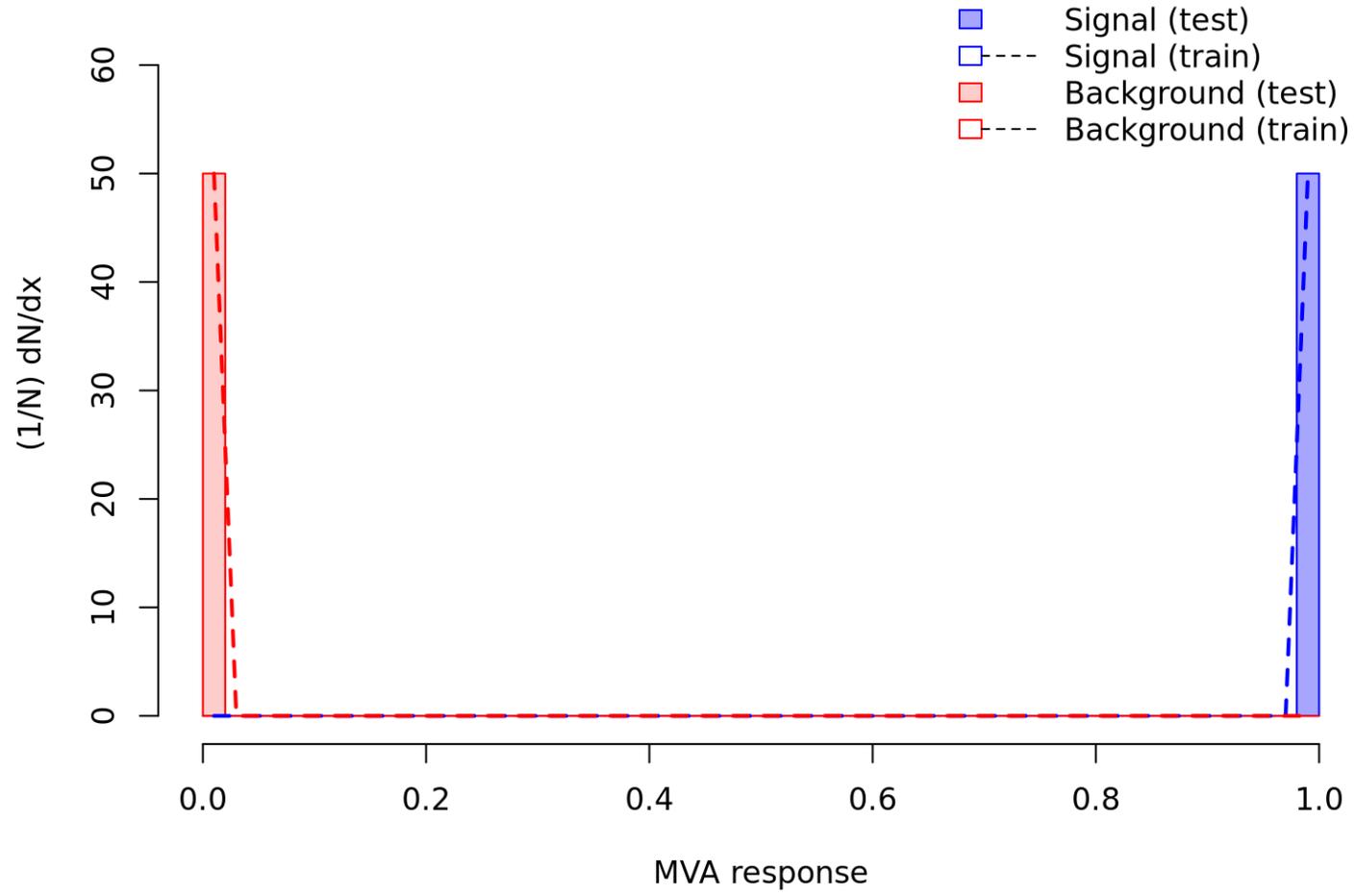


R overtraining check for classifier: xgboost

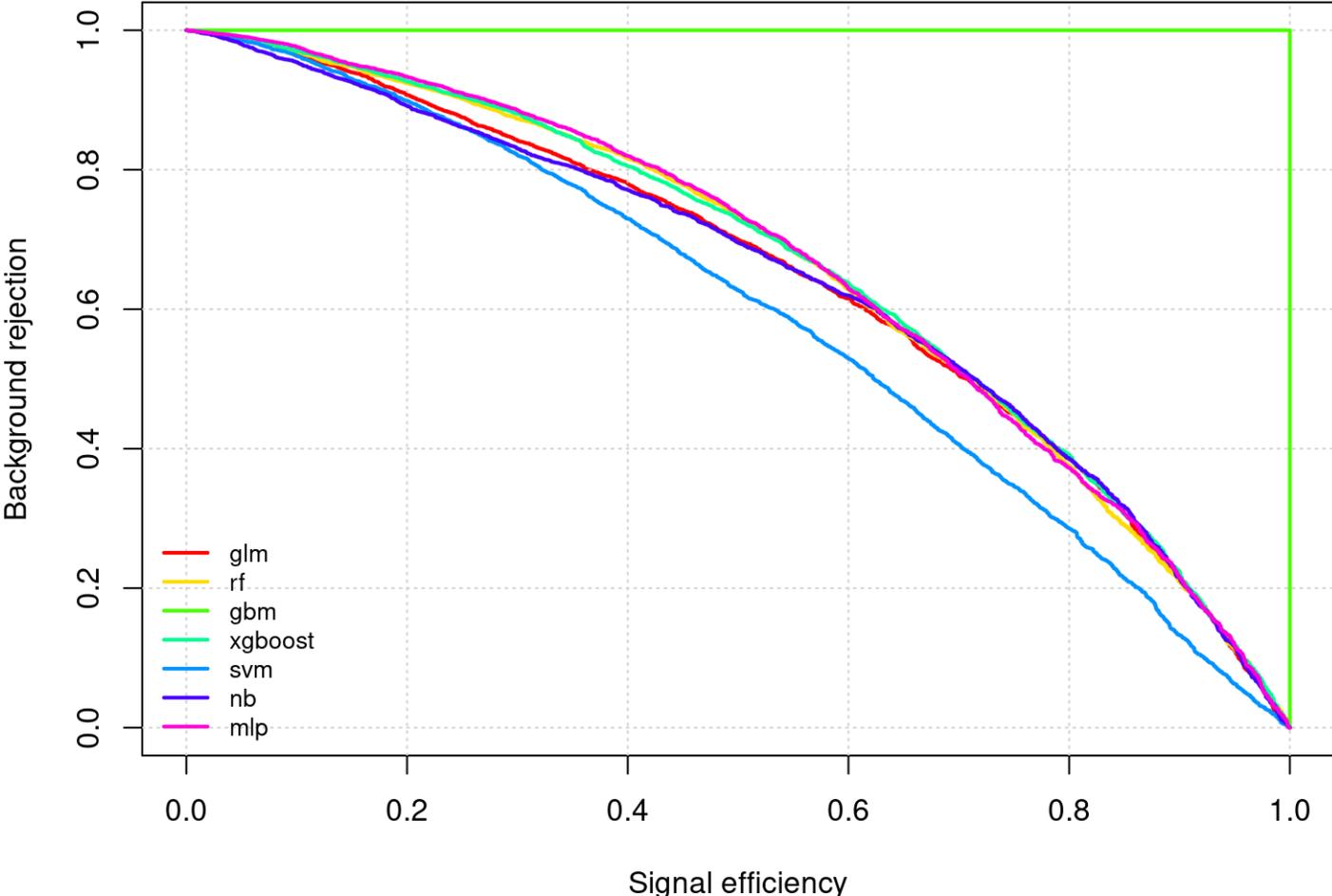


R overtraining check for classifier: gbm

KS p-values (unweighted): S=1 B=0.828



ROC curves (TMVA style): all methods



Comparison of Results from Python and R (k-fold)

Model	R AUC (oof_weighted)	Python mean_fold_auc	Python overall_oof_auc	Δ AUC (R – Py overall)
Gradient Boosting (gbm)	1.000000	0.675134	0.674820	+0.325180
MLP (mlp)	0.661500	0.665761	0.664427	-0.002927
Random Forest (rf)	0.656103	0.672187	0.671608	-0.015505
Logistic Regression (glm)	0.643484	0.639506	0.639507	+0.003977
Naive Bayes (nb)	0.640344	0.640700	0.639867	+0.000477
SVM / SVC (svm)	0.590934	0.676572	0.676277	-0.085343

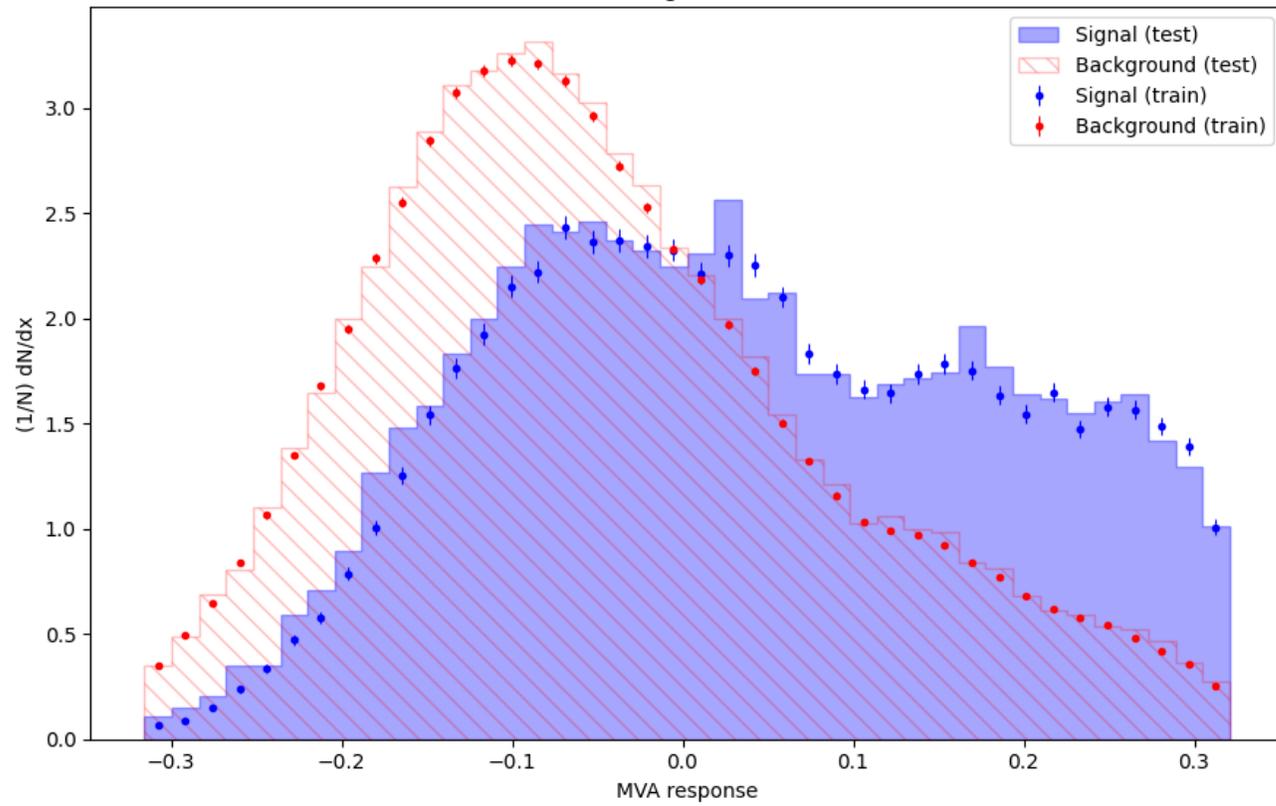
Results of Performance of ML Algorithms using K-fold on TMVA

10th February 2026

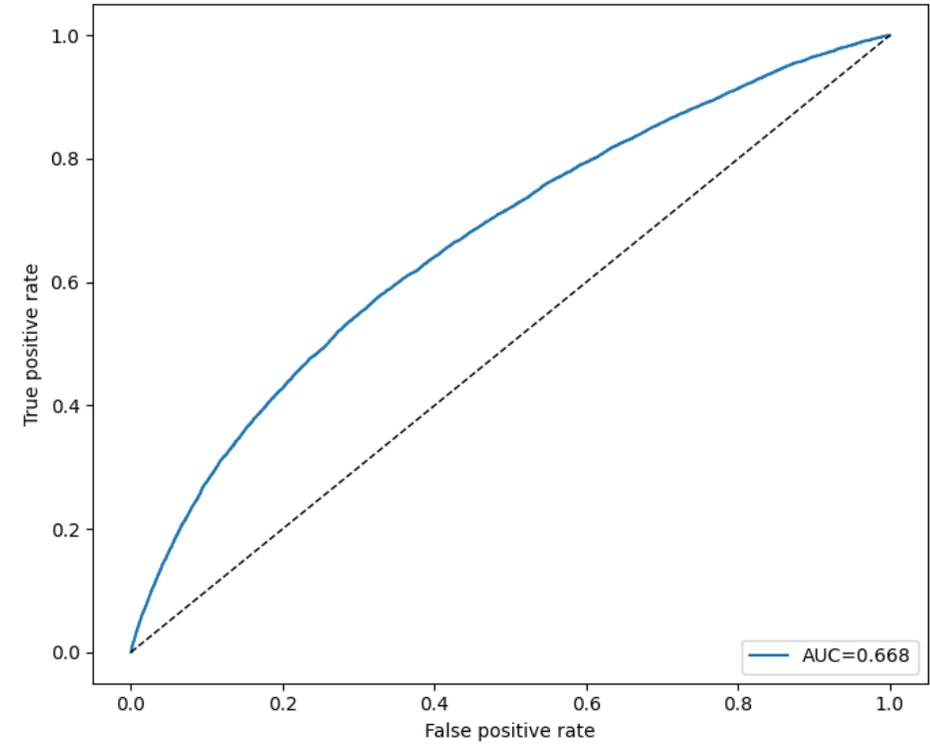
TMVA (k-fold) Results on New Data

Algorithm (TMVA)	kfold	n_trials	best_mean_auc
BDT	5	20	0.668182
BDTG	5	20	0.670094
Fisher	5	20	0.637319
kNN	5	20	0.651456
Likelihood	5	20	0.650749
MLP	5	20	0.668192

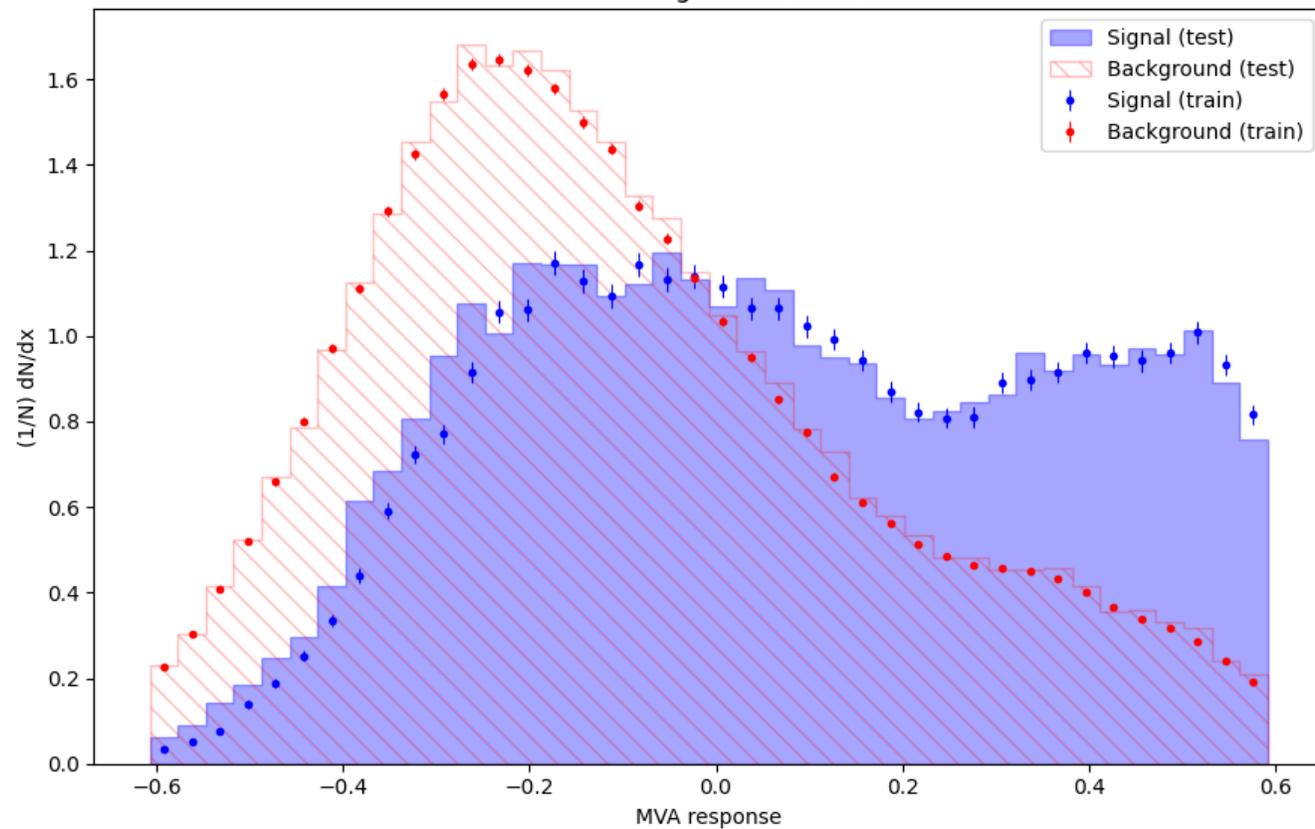
TMVA overtraining check (k-fold): BDT



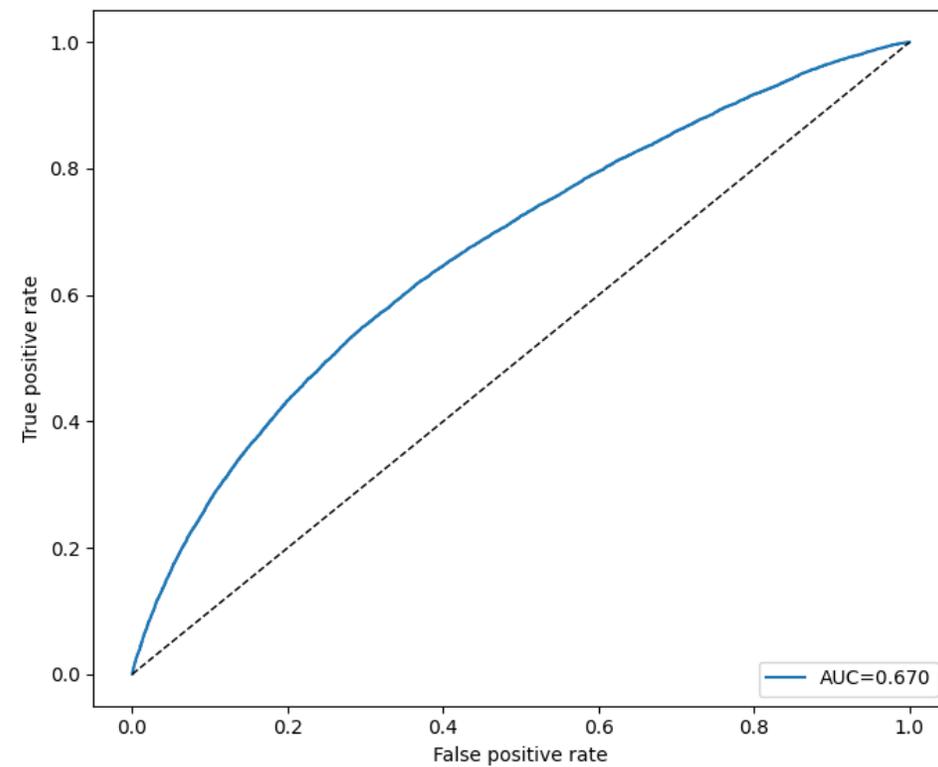
TMVA k-fold ROC - BDT



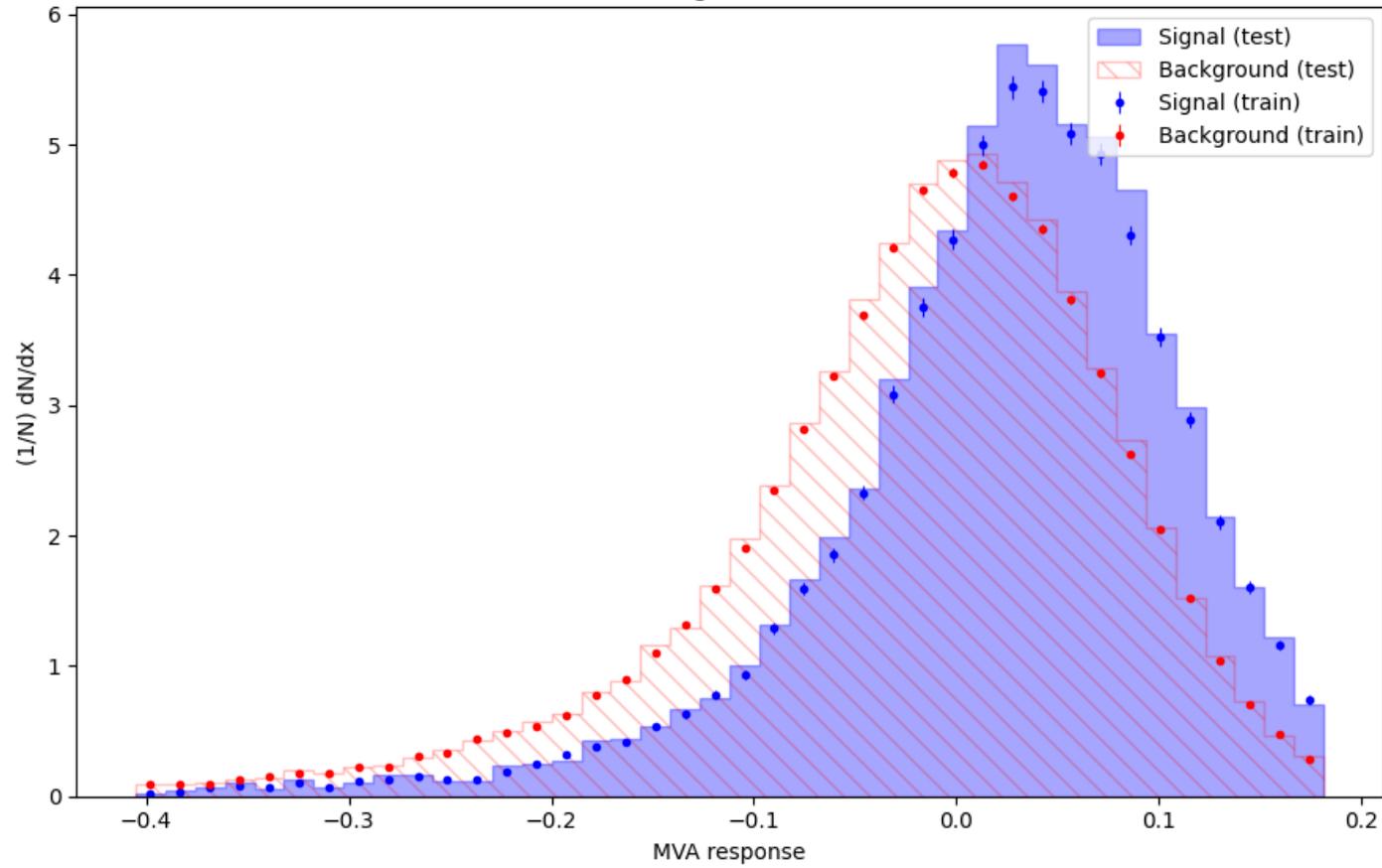
TMVA overtraining check (k-fold): BDTG



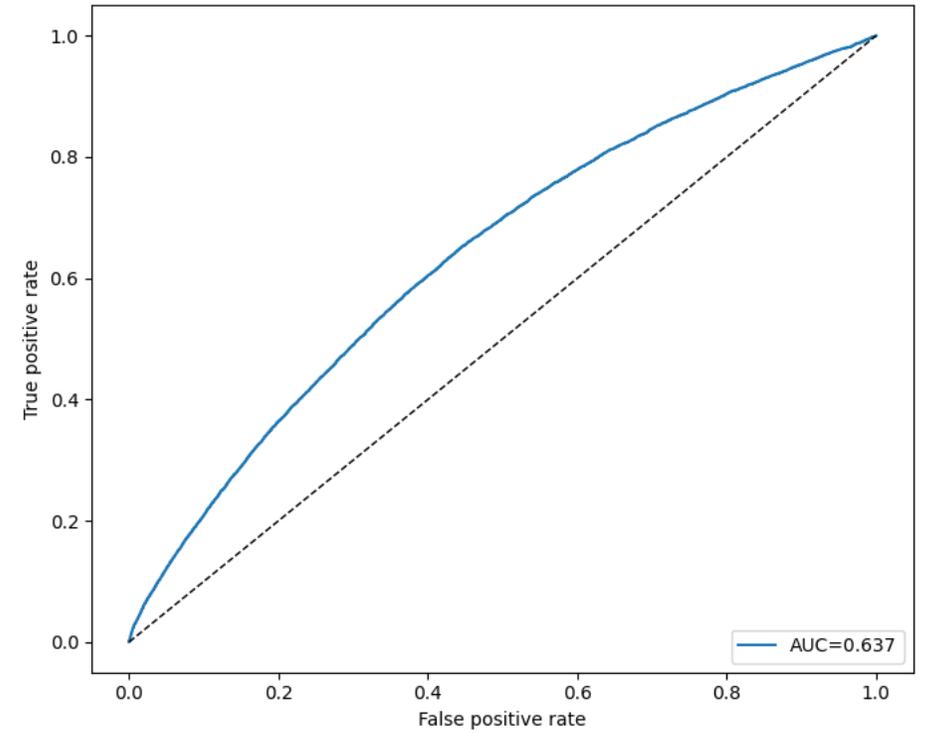
TMVA k-fold ROC - BDTG



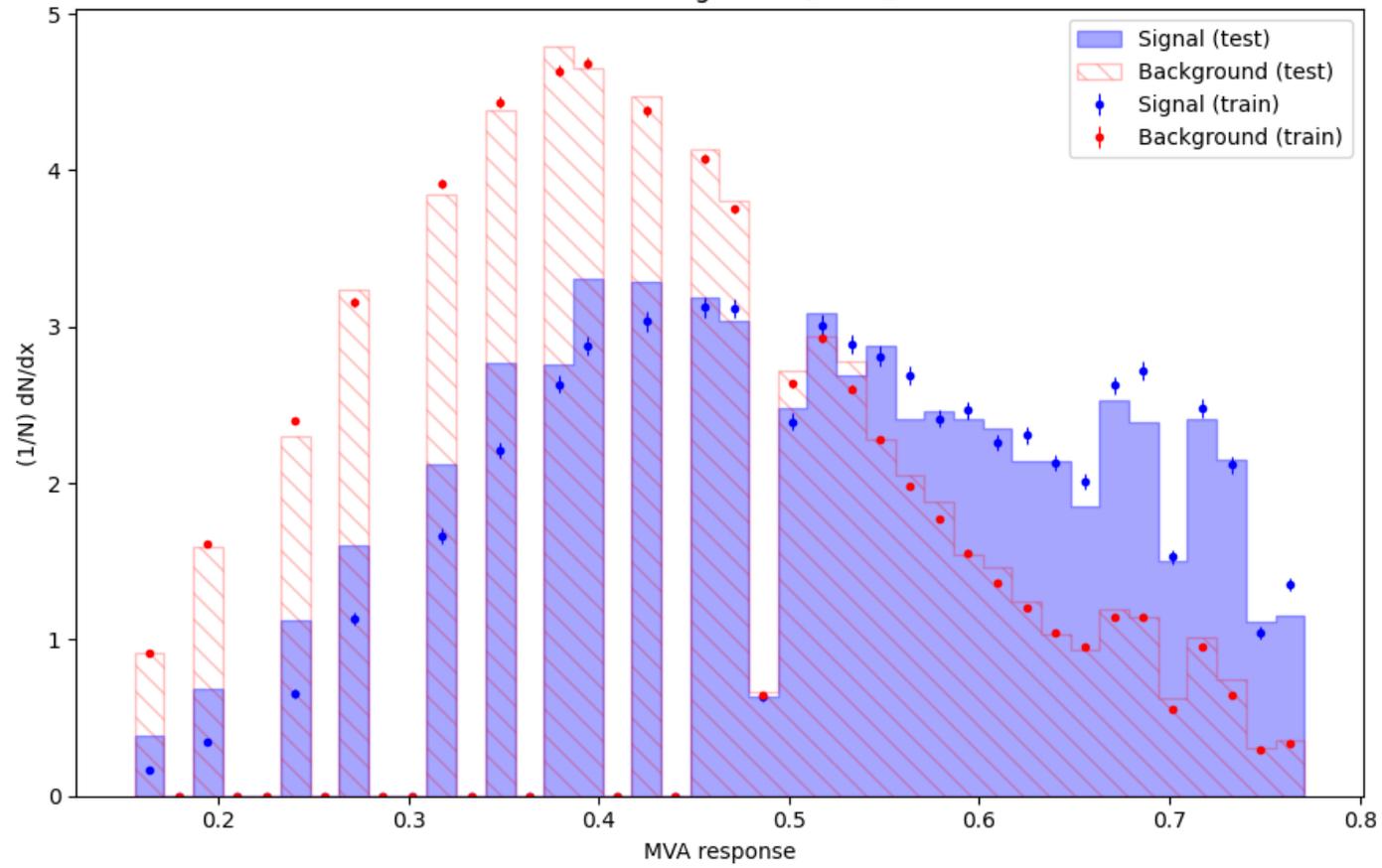
TMVA overtraining check (k-fold): Fisher



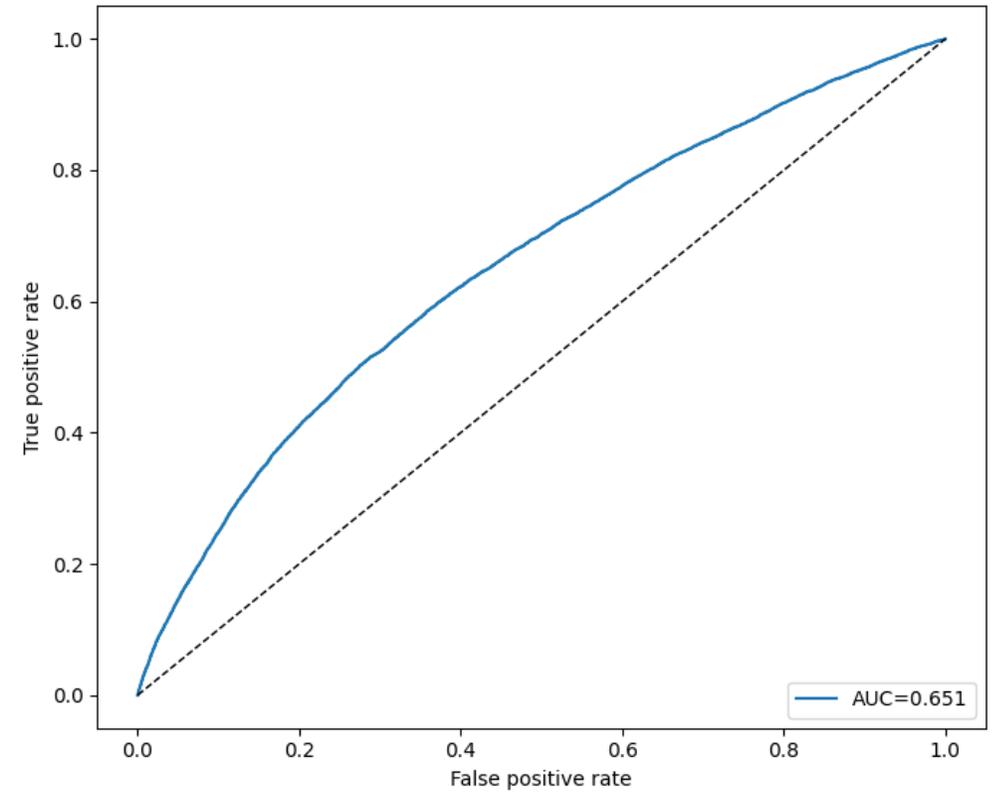
TMVA k-fold ROC - Fisher



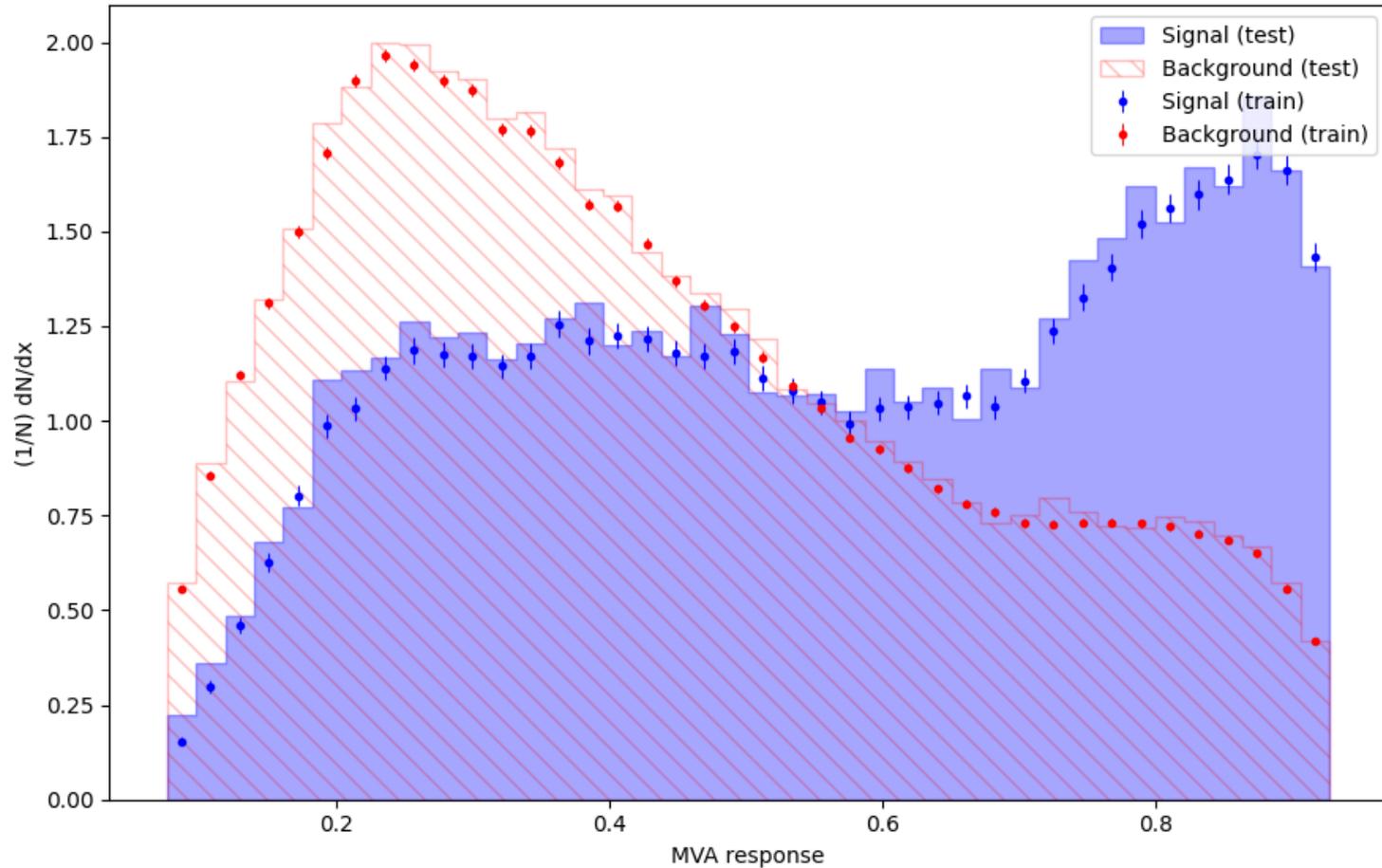
TMVA overtraining check (k-fold): kNN



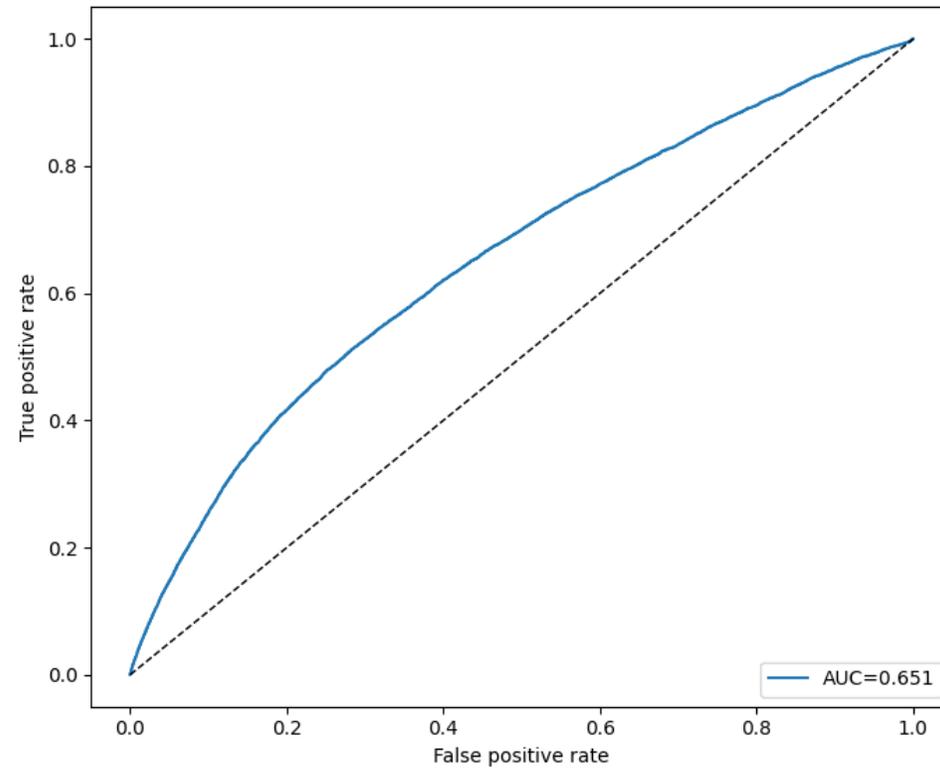
TMVA k-fold ROC - kNN



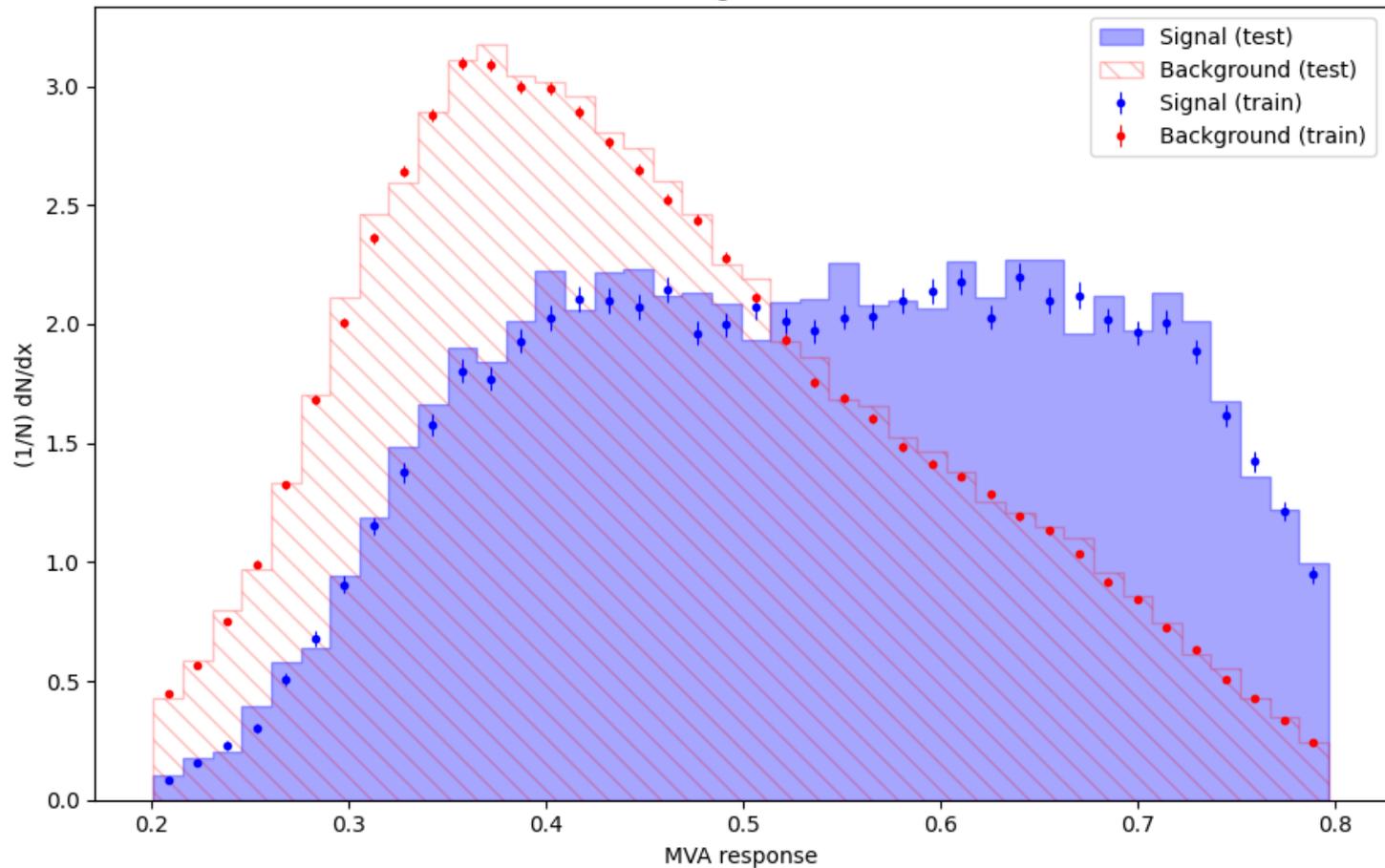
TMVA overtraining check (k-fold): Likelihood



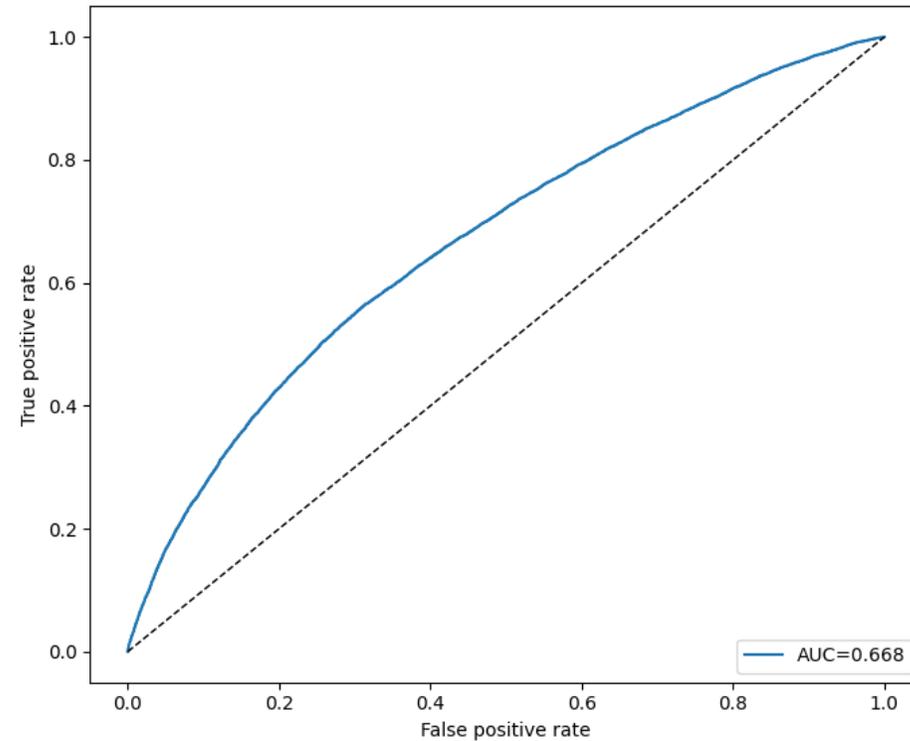
TMVA k-fold ROC - Likelihood



TMVA overtraining check (k-fold): MLP



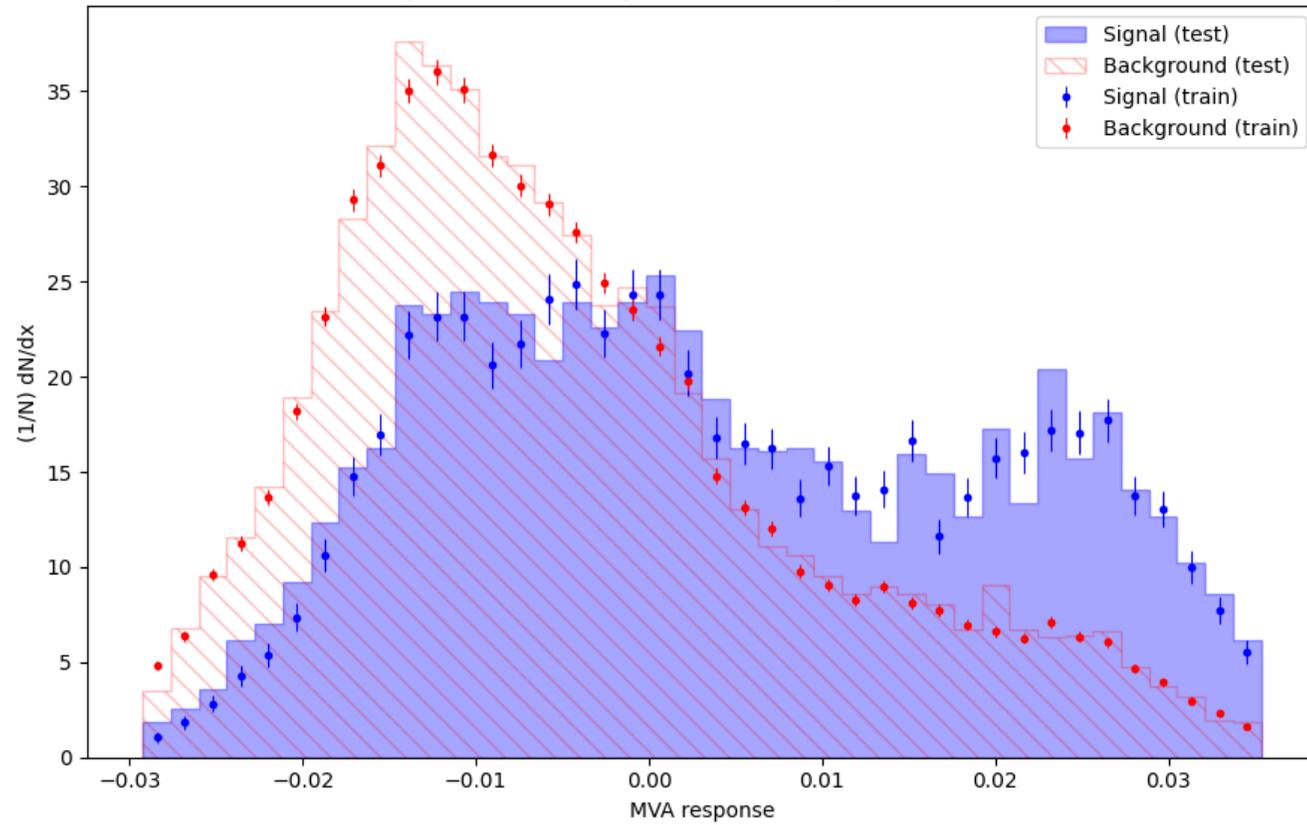
TMVA k-fold ROC - MLP



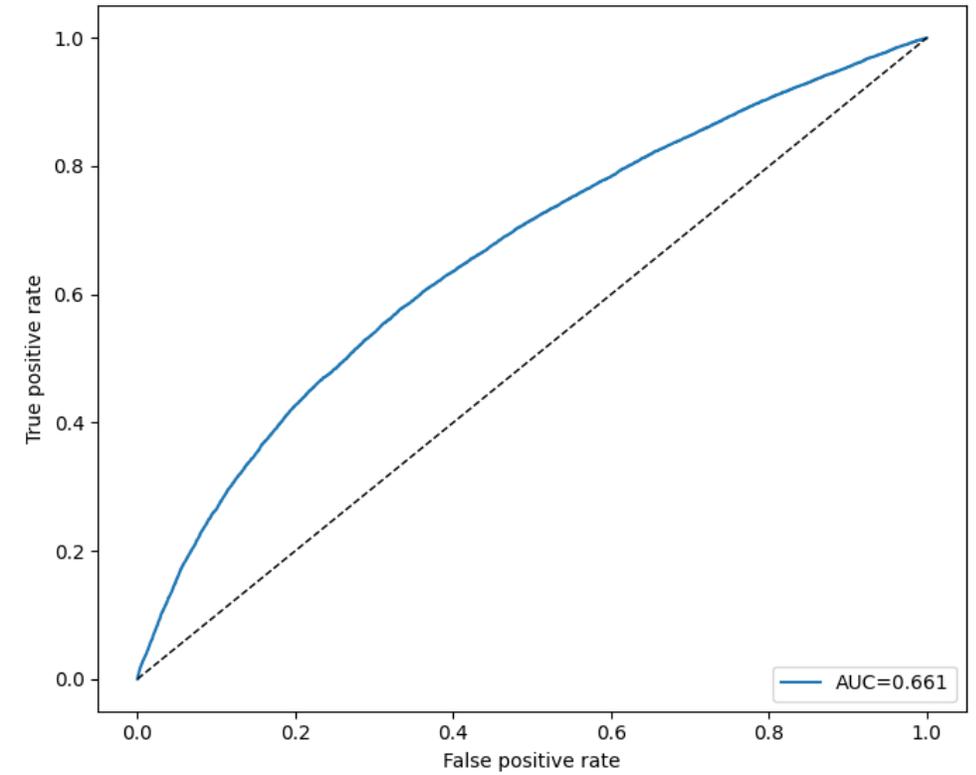
**Results of Performance of ML
Algorithms using K-fold on
Python 4th Trial More
algorithms added
10th February 2026**

Algorithm (Python)	outer_k	inner_k	n_iter	mean_fold_auc	std_fold_auc	overall_oof_auc	mean_fold_wacc	std_fold_wacc	overall_oof_wacc
AdaBoost	5	3	25	0.663612	0.006768	0.660667	0.62078	0.004724	0.62078
Bagging	5	3	25	0.647918	0.007531	0.647809	0.502429	0.000625	0.502429
ExtraTrees	5	3	25	0.667711	0.006485	0.667169	0.622879	0.005654	0.622879
GaussianNB	5	3	25	0.636746	0.008755	0.636637	0.59563	0.004879	0.59563
GradientBoosting	5	3	25	0.670242	0.007283	0.670033	0.625561	0.006146	0.62556
HistGradientBoosting	5	3	25	0.668588	0.006794	0.668074	0.624038	0.005583	0.624038
KNN	5	3	25	0.649829	0.006528	0.649802	0.500247	0.000234	0.500247
LDA	5	3	25	0.633161	0.009455	0.632984	0.500227	4.95E-05	0.500227
LinearSVC	5	3	25	0.638989	0.00895	0.638861	0.602301	0.008101	0.6023
LogisticRegression	5	3	25	0.641267	0.008797	0.641124	0.604398	0.007429	0.604398
MLP	5	3	25	0.666970	0.007911	0.666548	0.50295	0.00077	0.50295
QDA	5	3	25	0.645348	0.007435	0.645249	0.51138	0.002164	0.511379
RandomForest	5	3	25	0.667747	0.006265	0.667129	0.620657	0.006323	0.620657
SGDClassifier	5	3	25	0.652858	0.006399	0.652702	0.570652	0.004155	0.570652
SVC	5	3	12	0.668126	0.006504	0.667912	0.62335	0.00485	0.62335

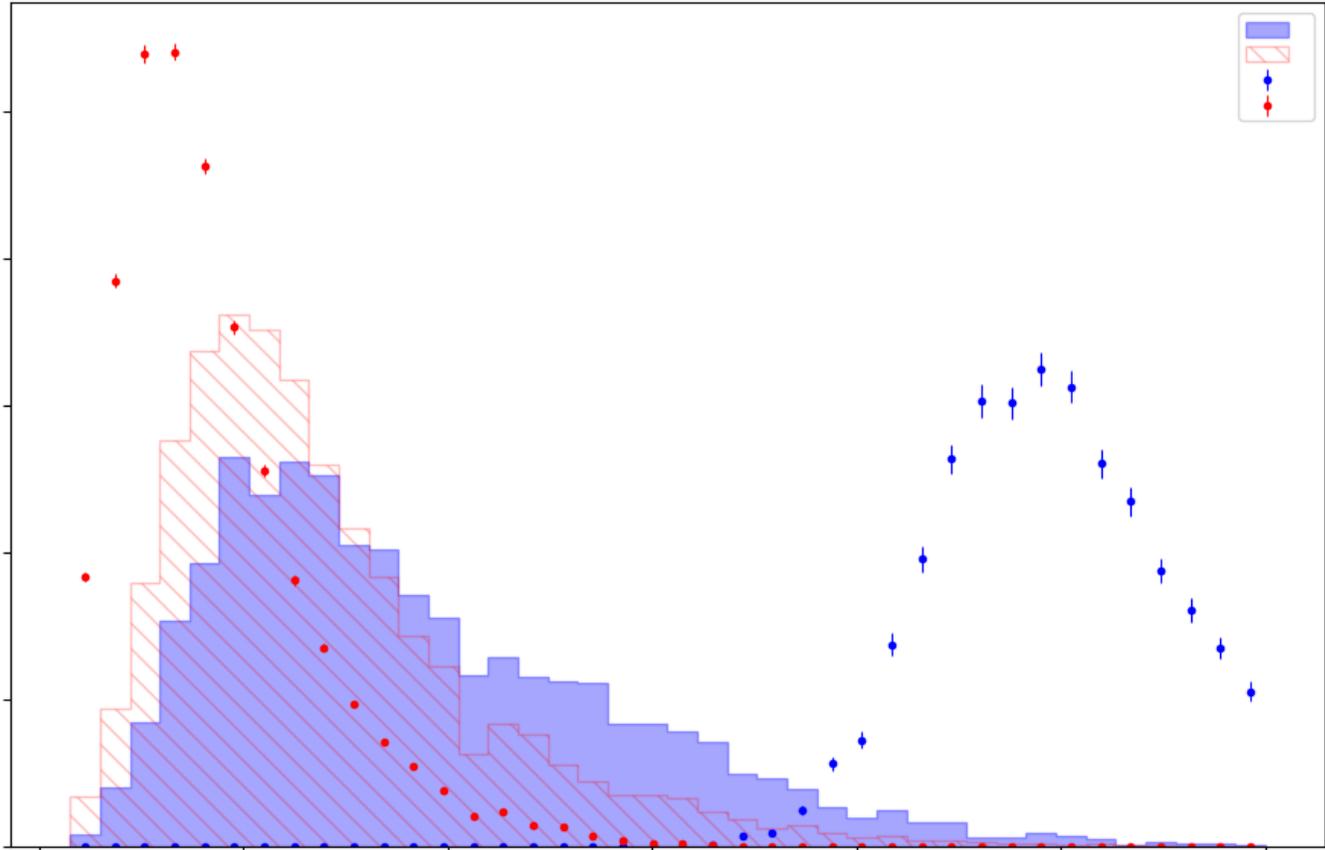
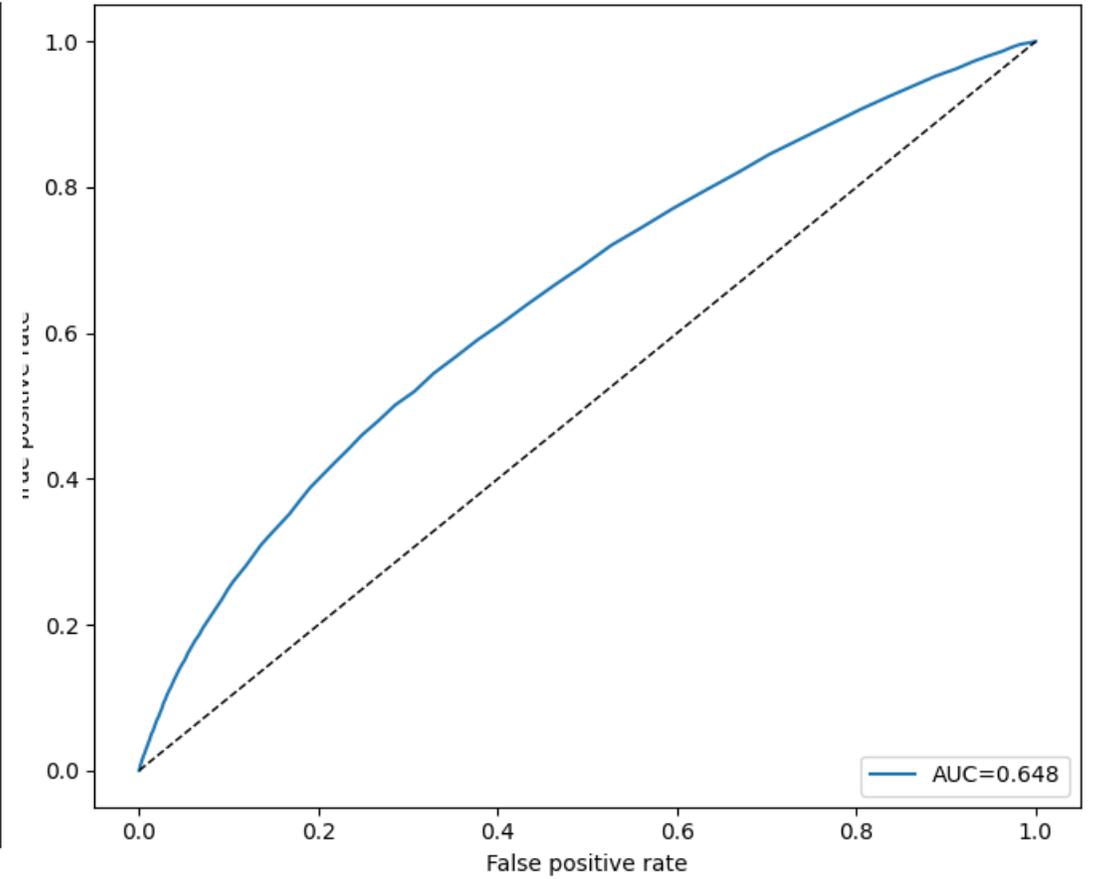
Python overtraining check (TMVA-style): AdaBoost



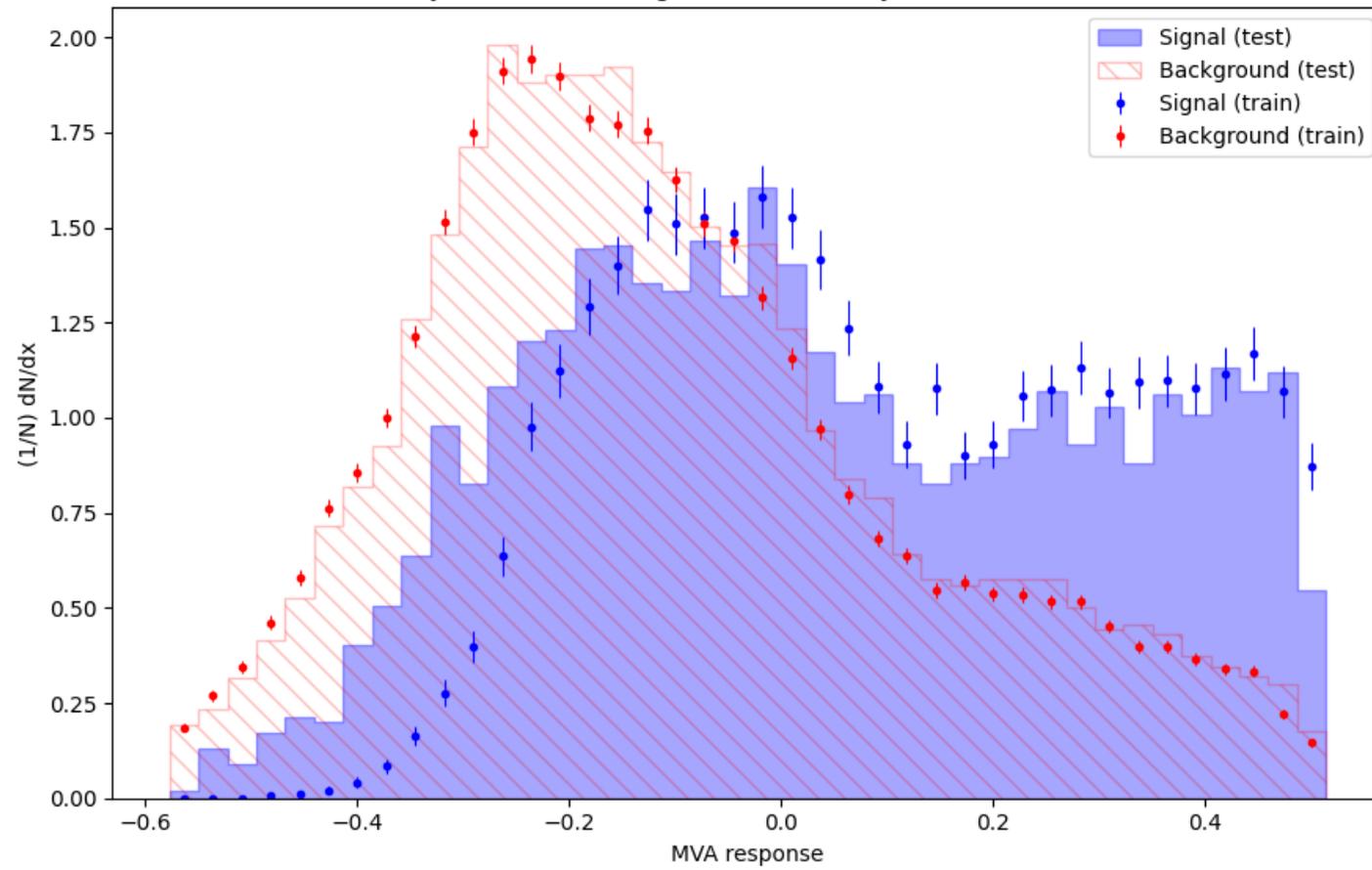
OOF ROC - AdaBoost



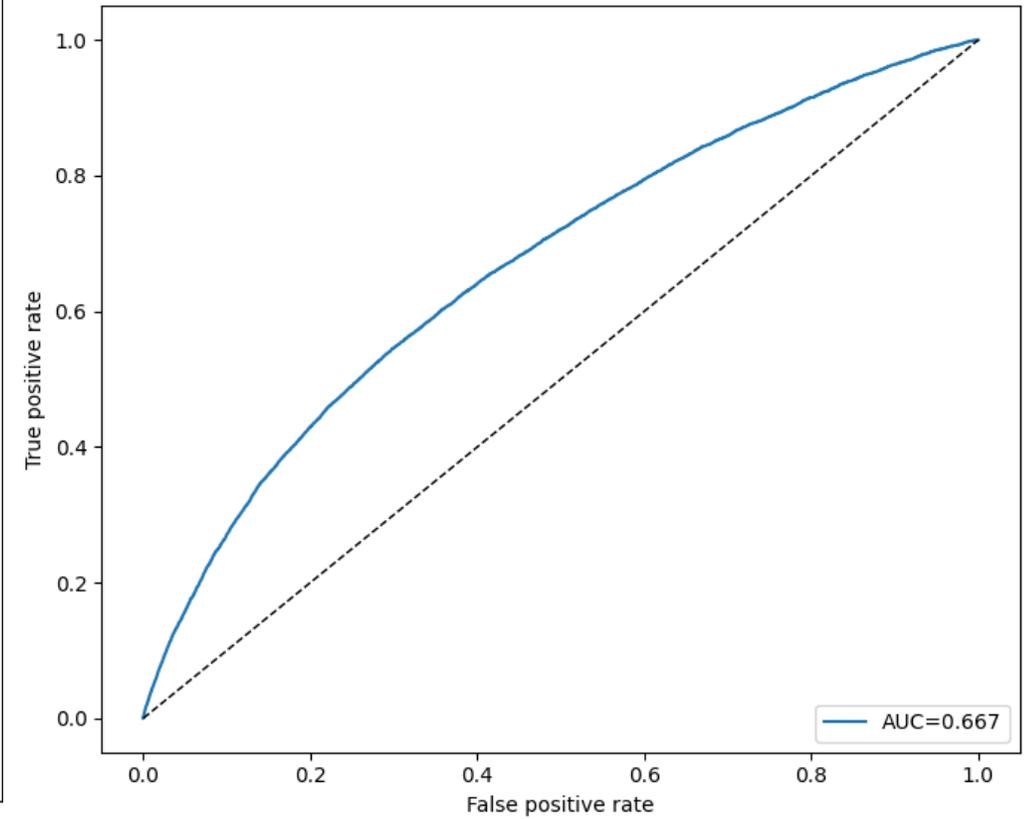
OOF ROC - Bagging



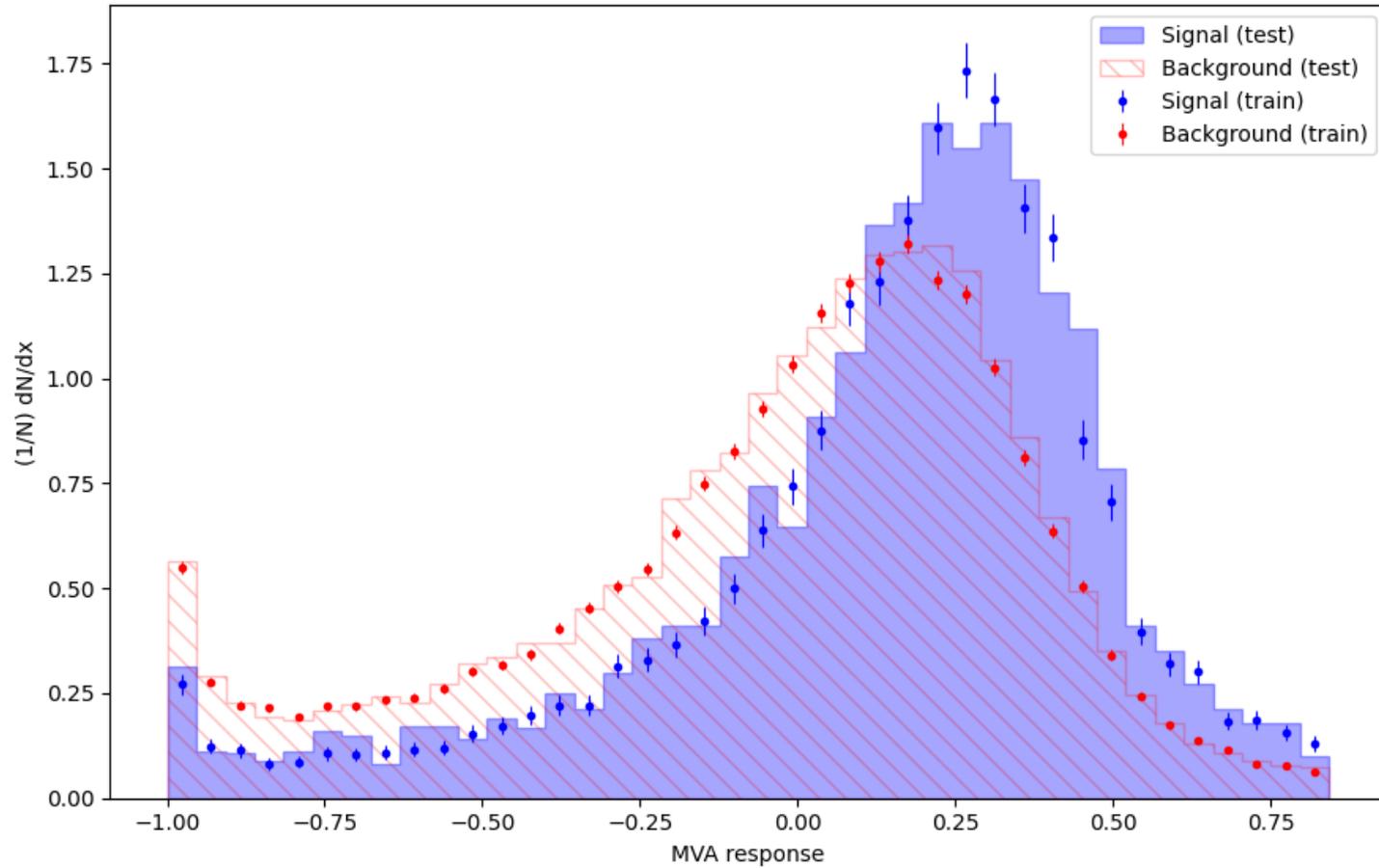
Python overtraining check (TMVA-style): ExtraTrees



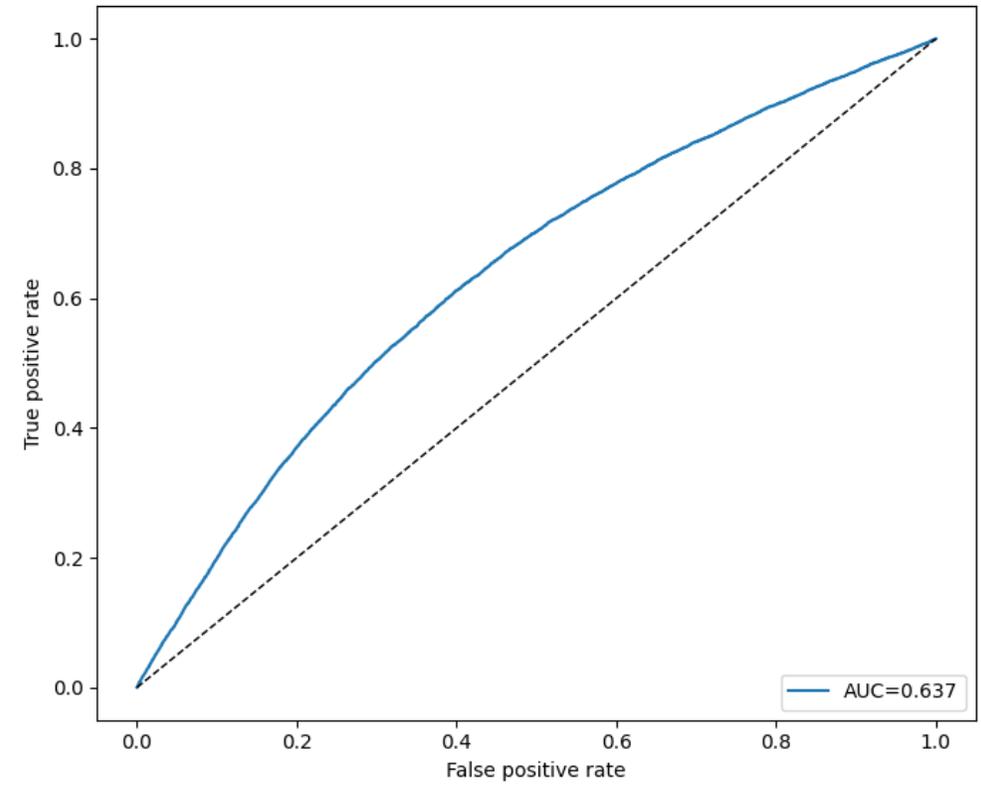
OOF ROC - ExtraTrees



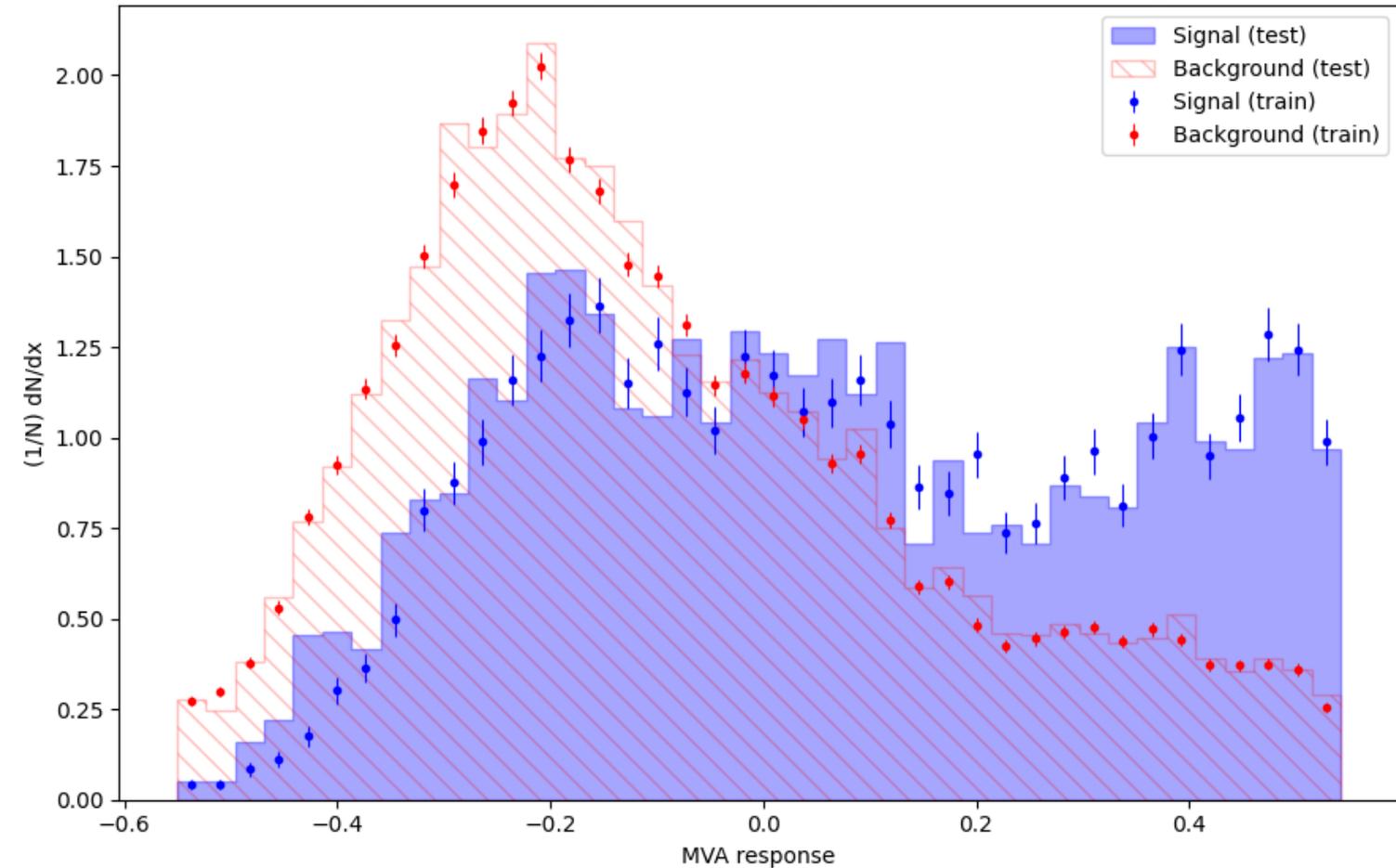
Python overtraining check (TMVA-style): GaussianNB



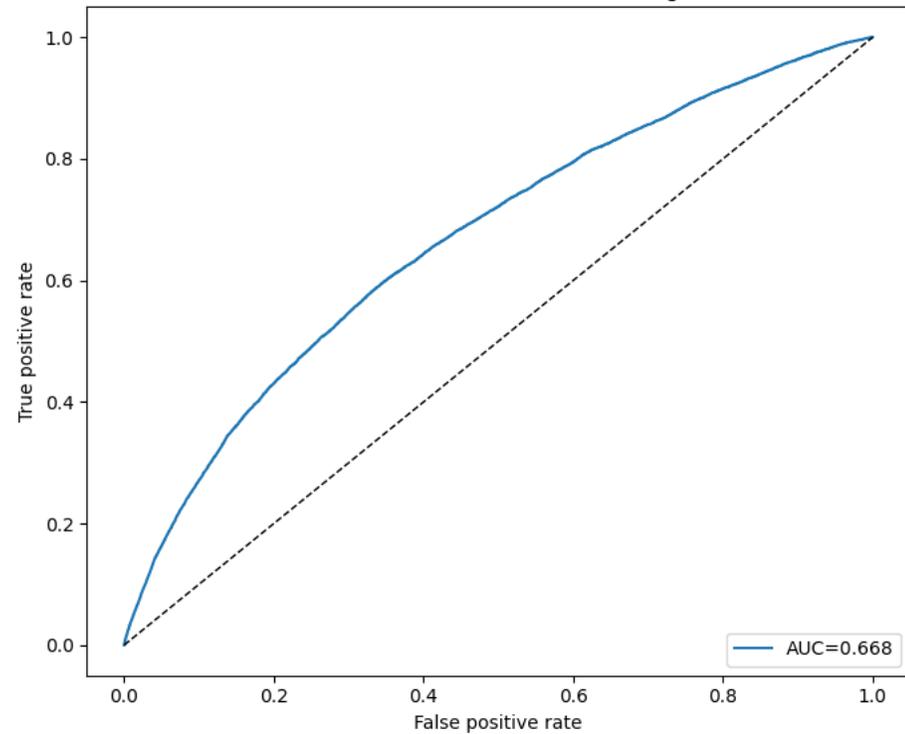
OOF ROC - GaussianNB



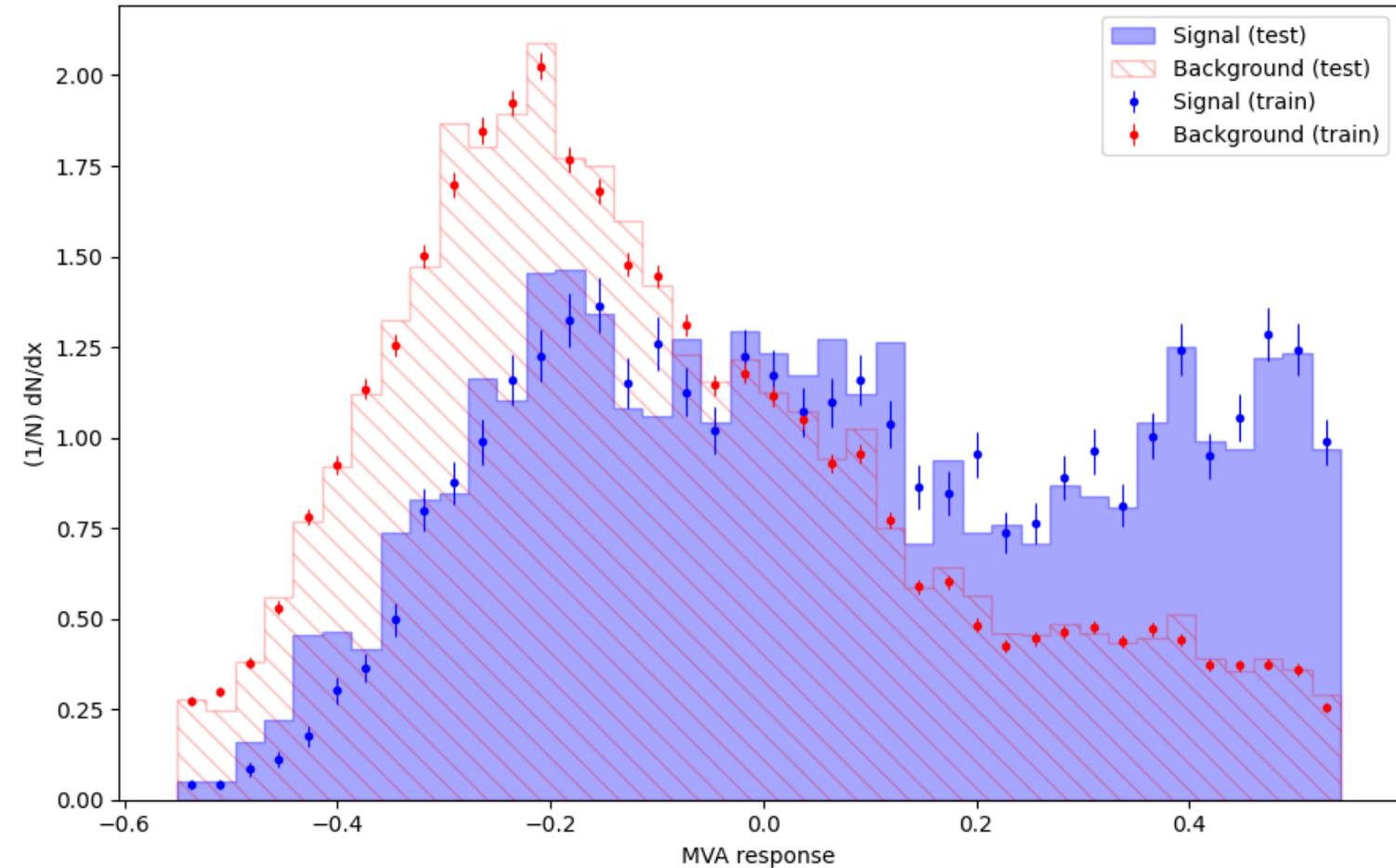
Python overtraining check (TMVA-style): HistGradientBoosting



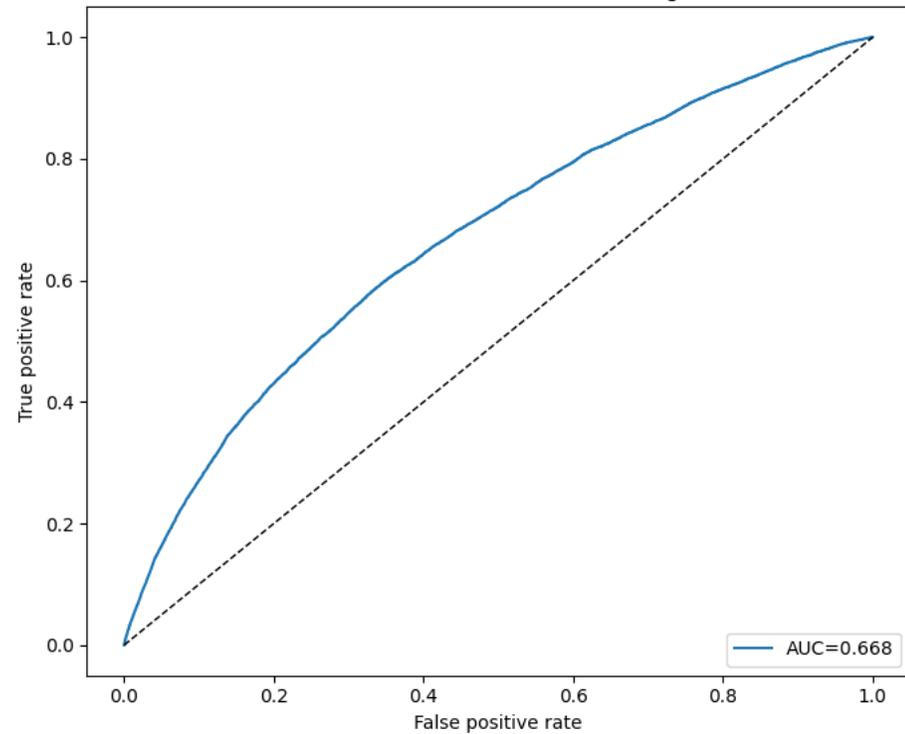
OOF ROC - HistGradientBoosting



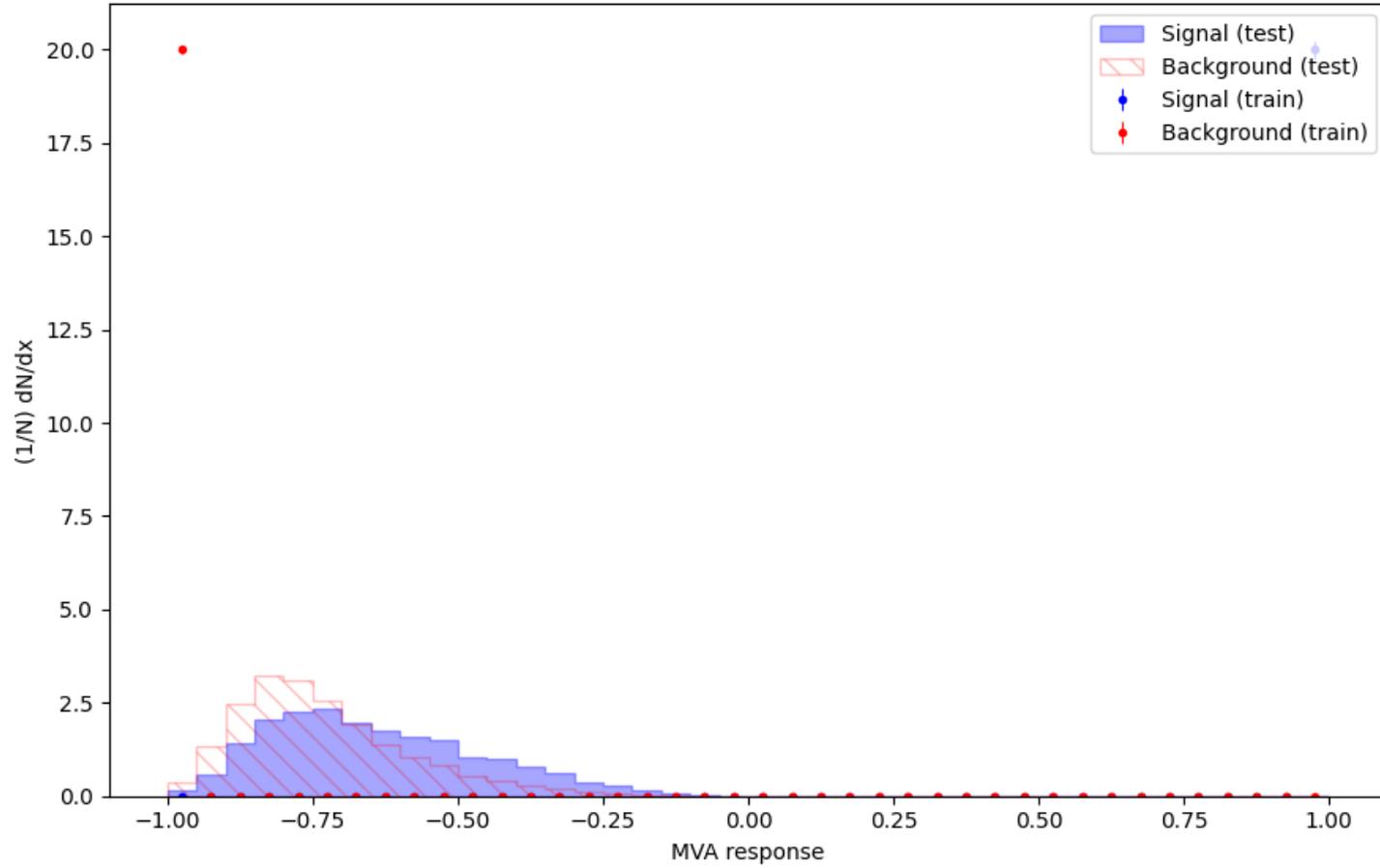
Python overtraining check (TMVA-style): HistGradientBoosting



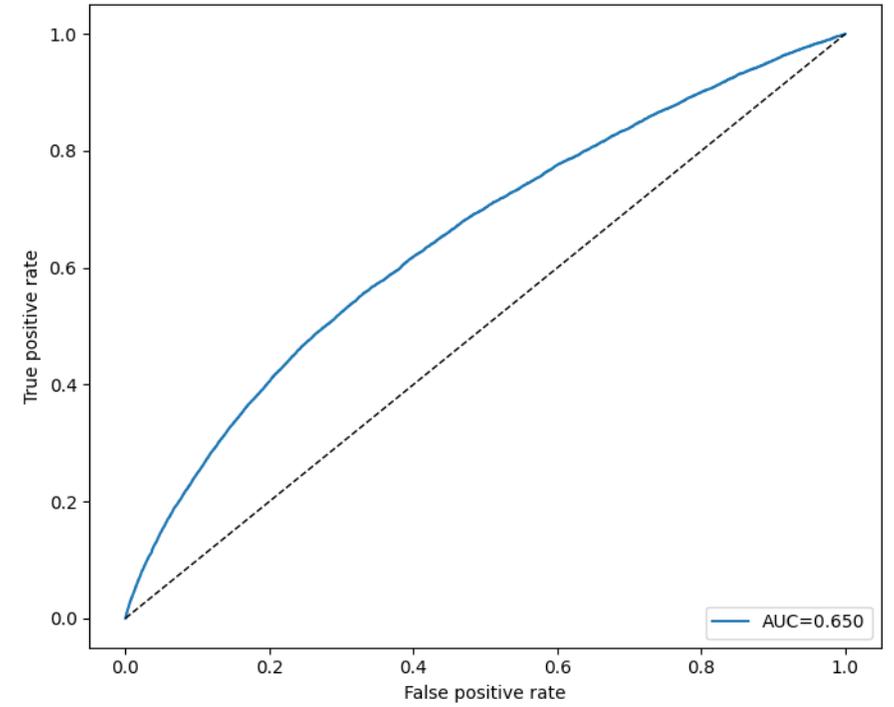
OOF ROC - HistGradientBoosting



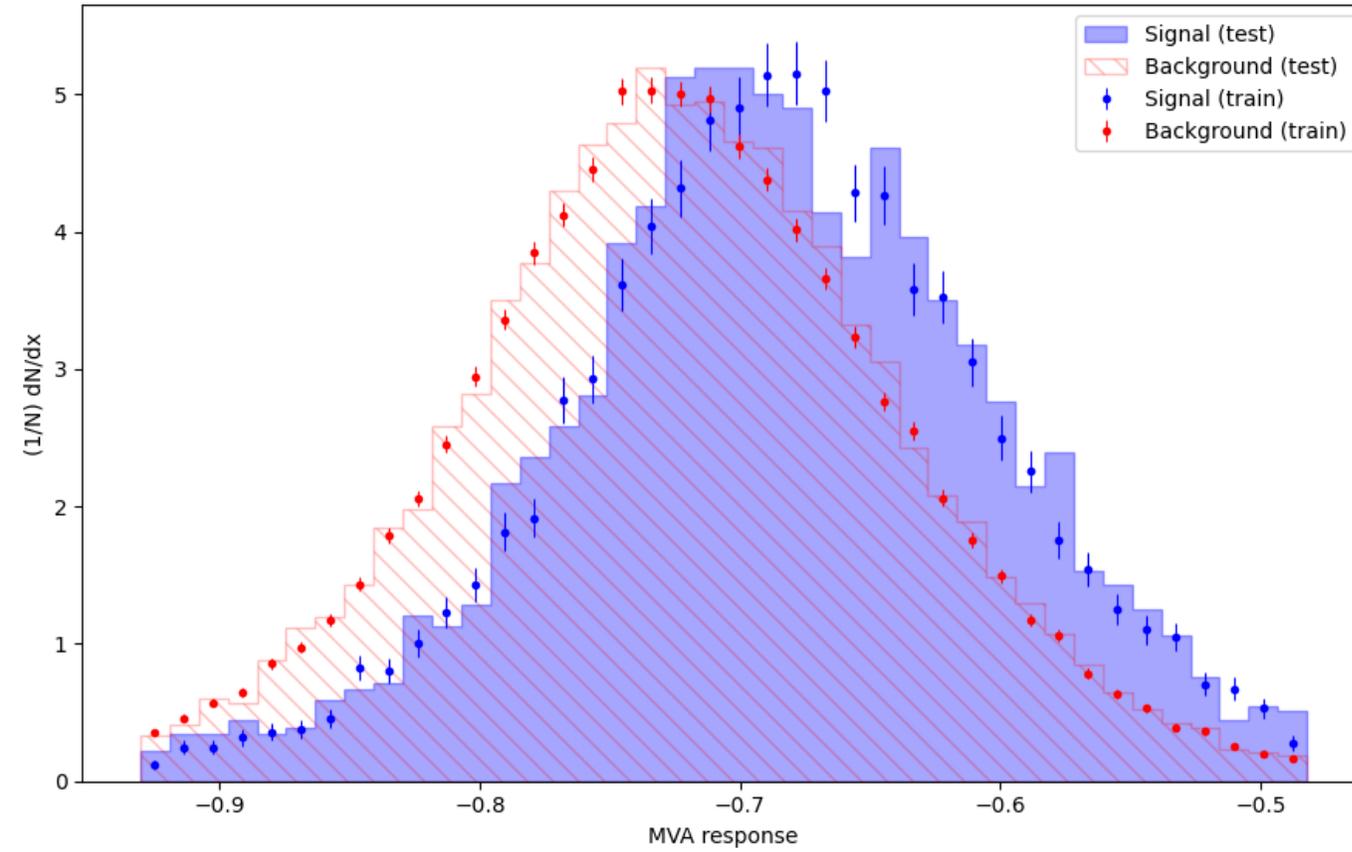
Python overtraining check (TMVA-style): KNN



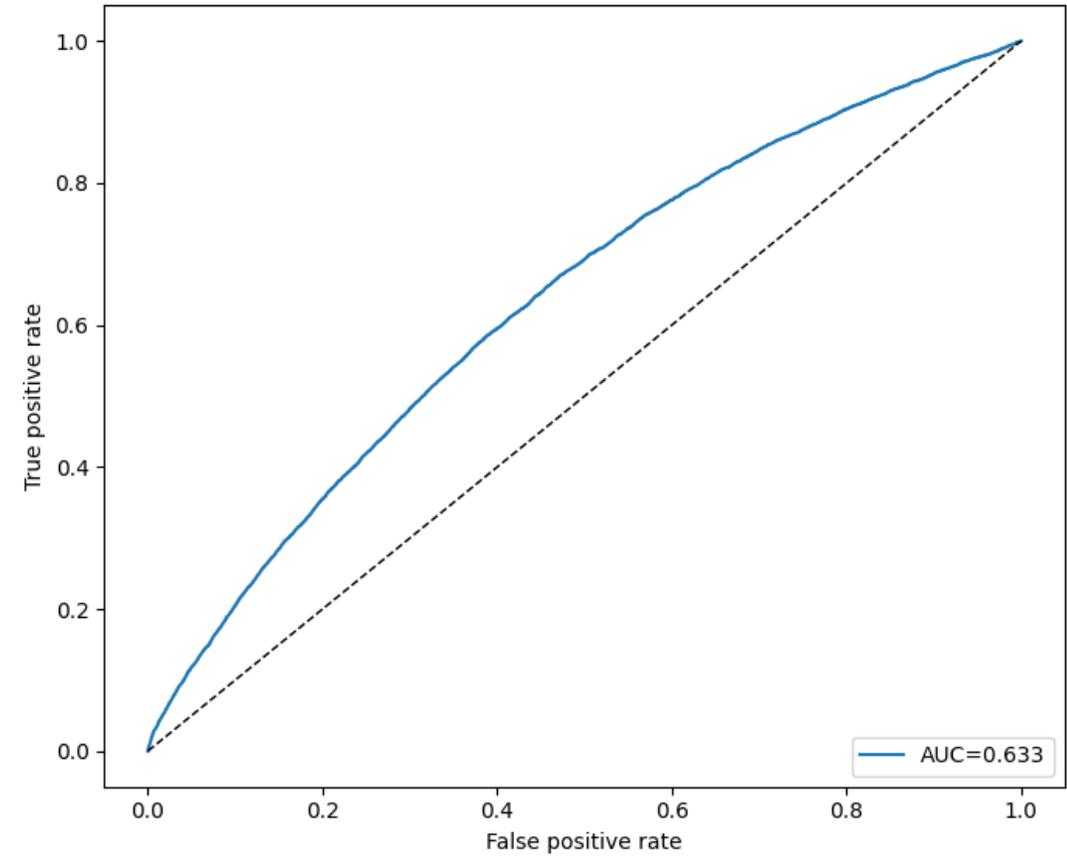
OOF ROC - KNN



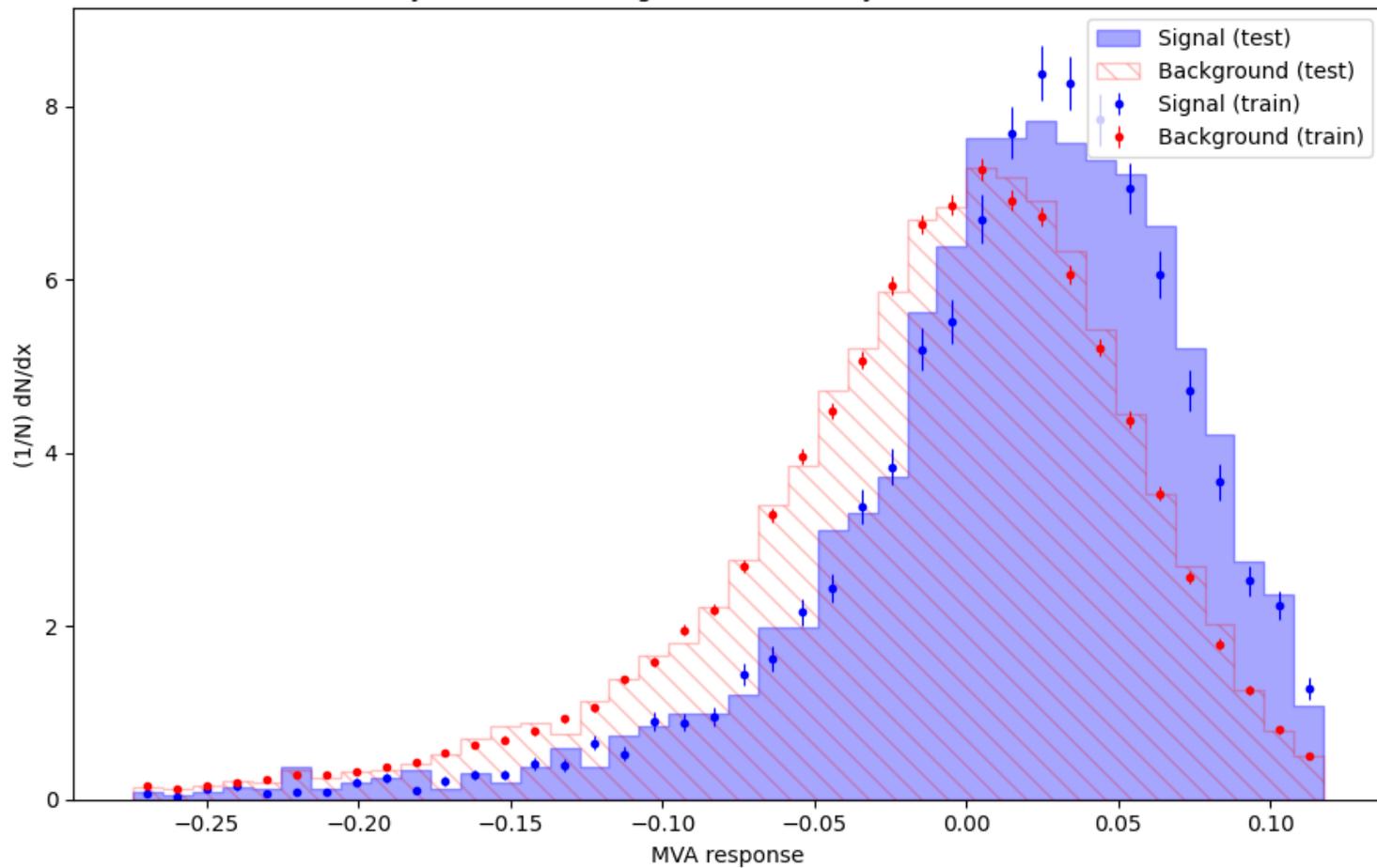
Python overtraining check (TMVA-style): LDA



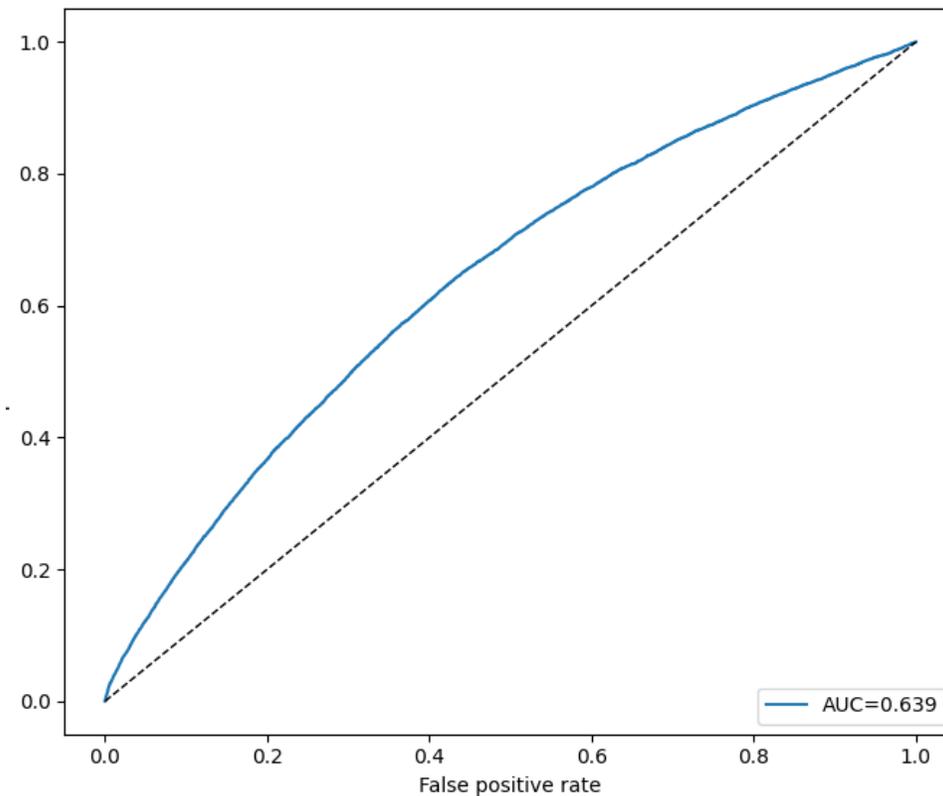
OOF ROC - LDA



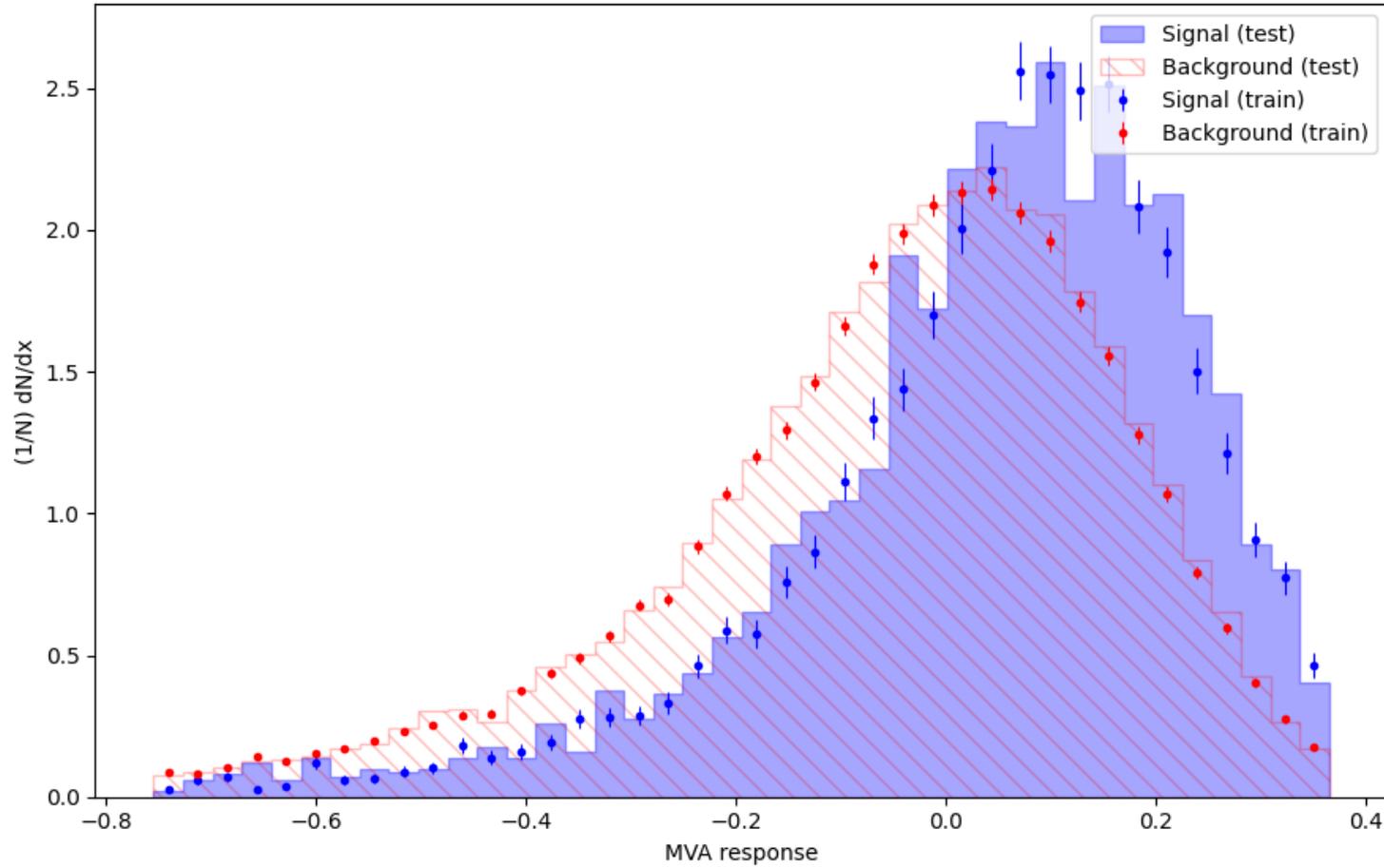
Python overtraining check (TMVA-style): LinearSVC



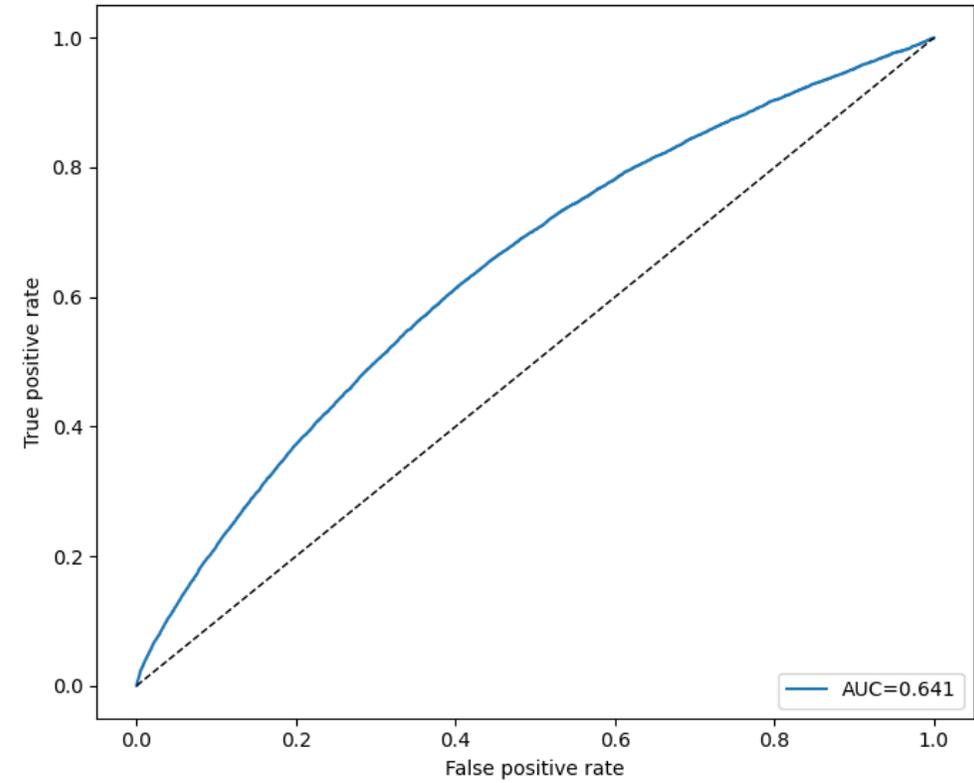
OOF ROC - LinearSVC



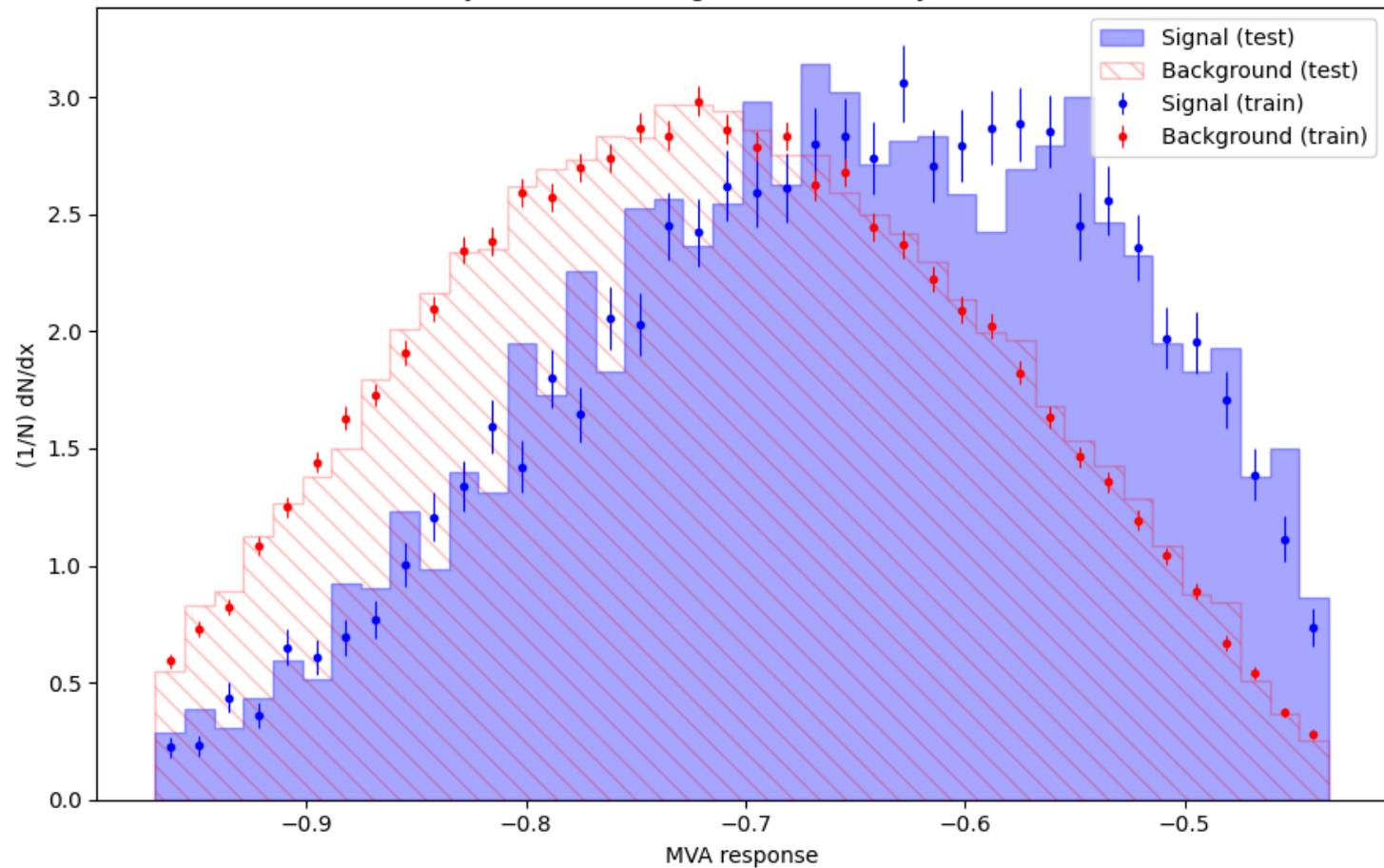
Python overtraining check (TMVA-style): LogisticRegression



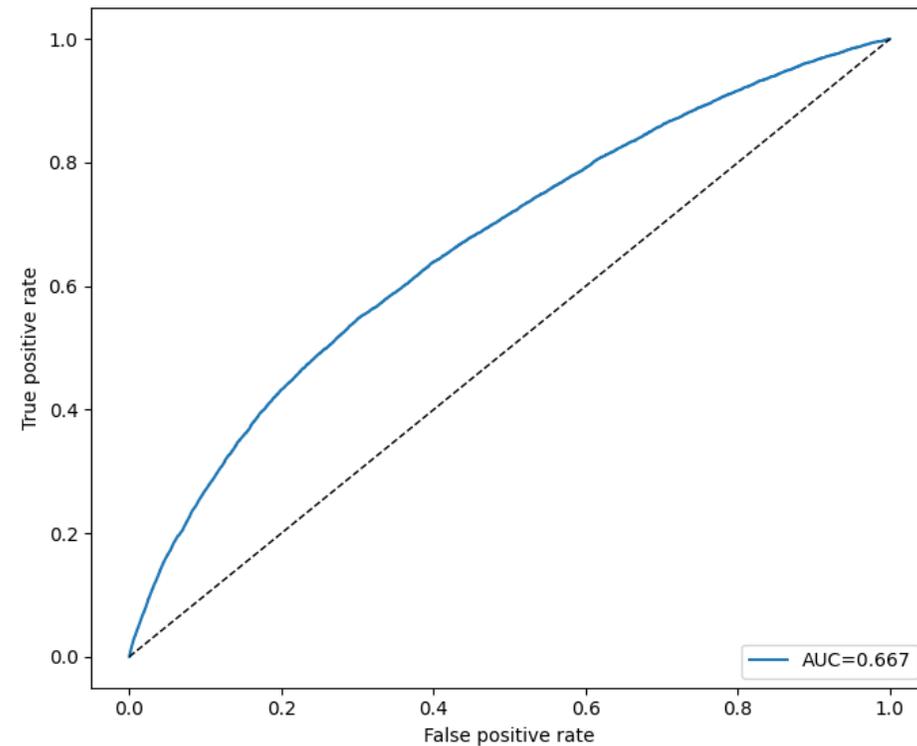
OOF ROC - LogisticRegression



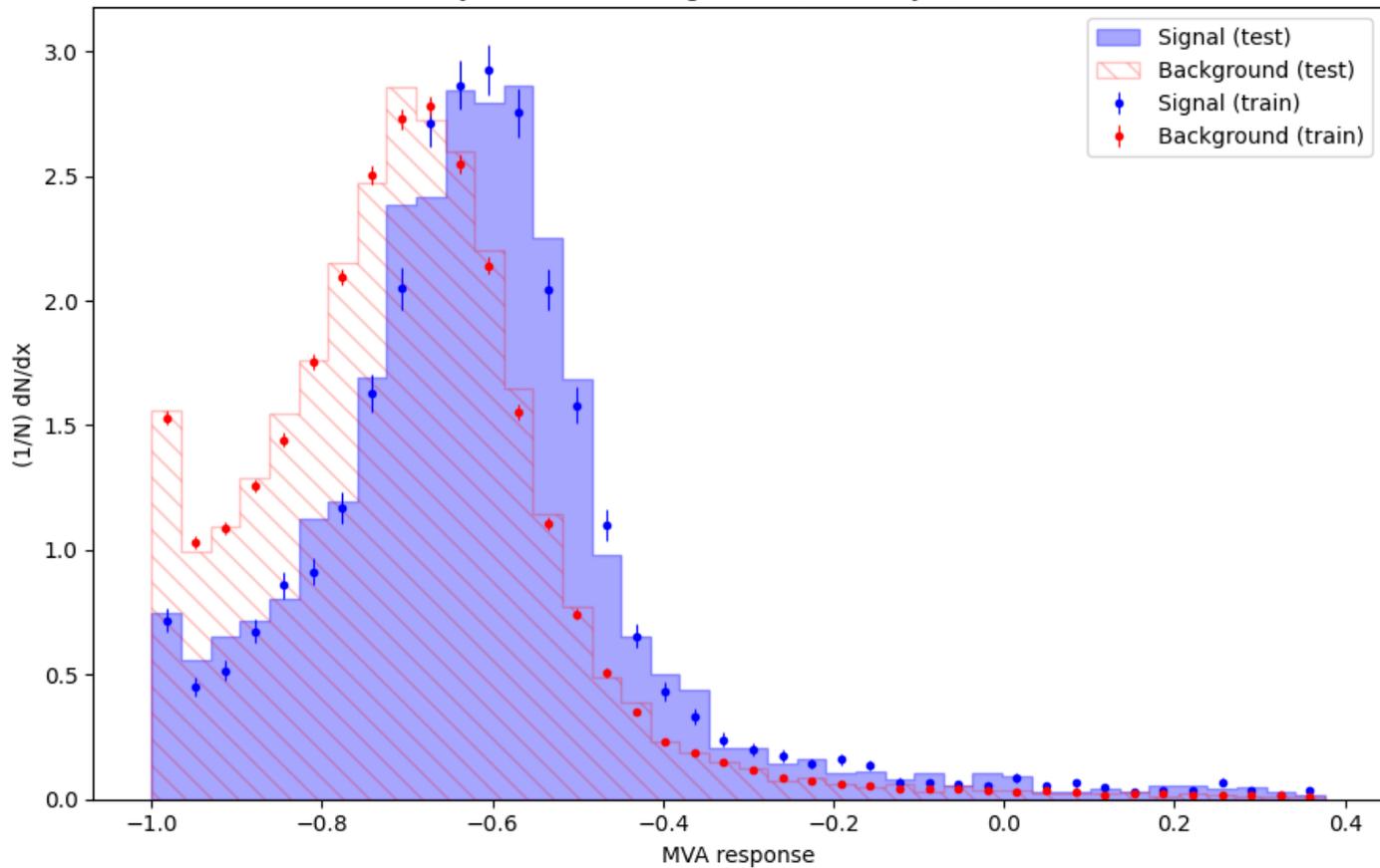
Python overtraining check (TMVA-style): MLP



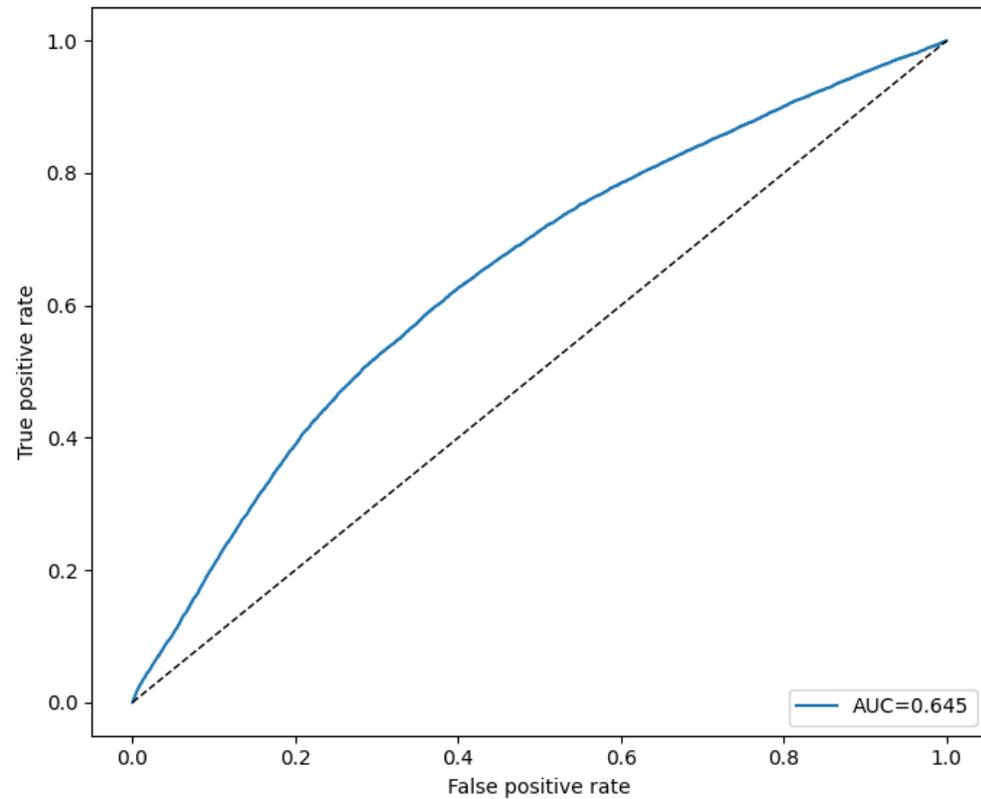
OOF ROC - MLP



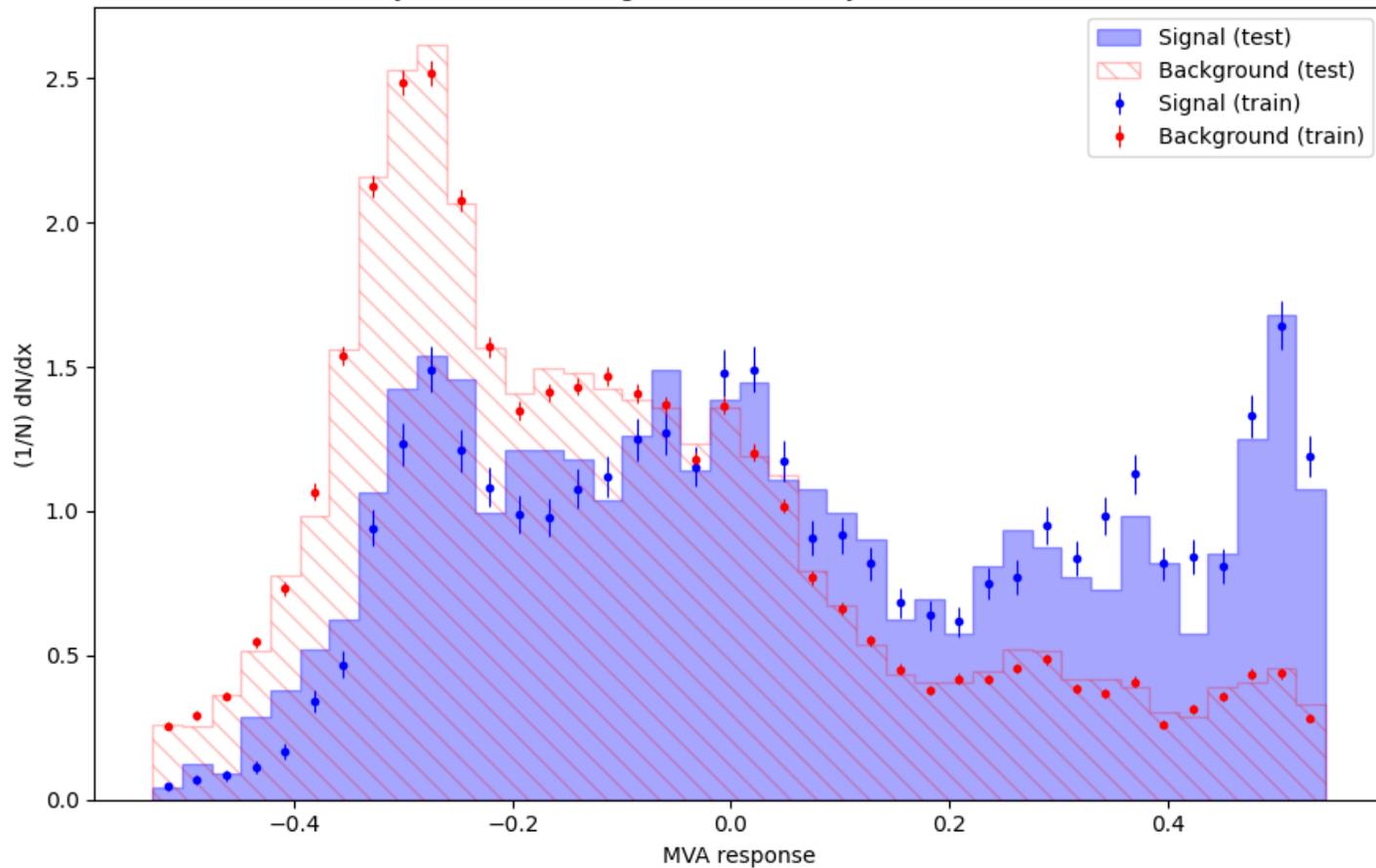
Python overtraining check (TMVA-style): QDA



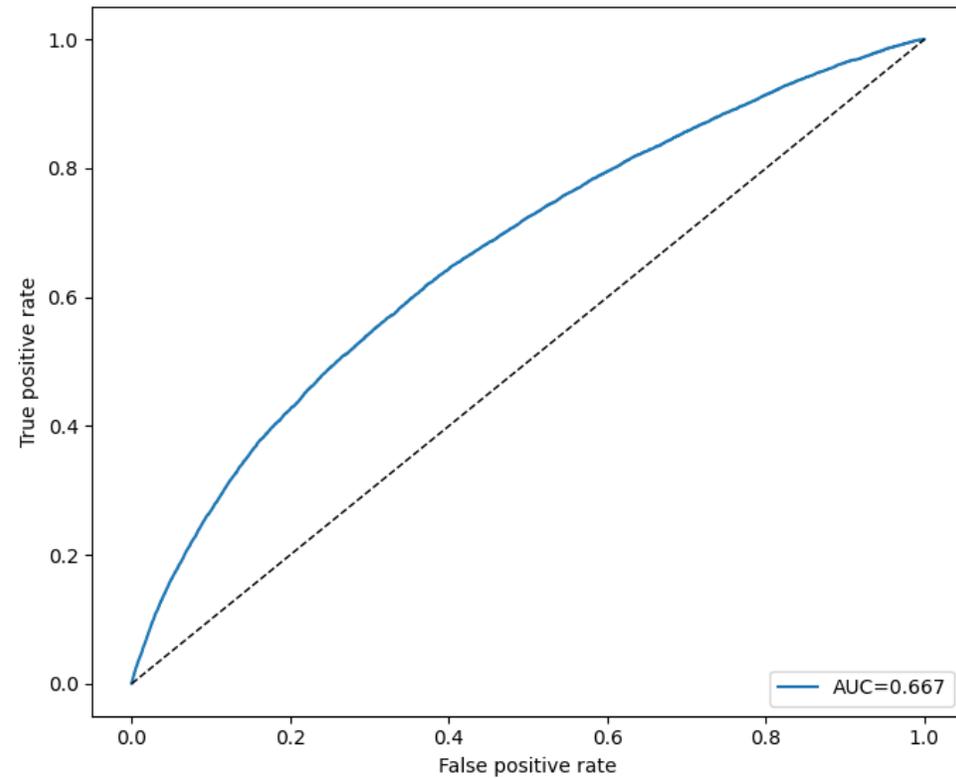
OOF ROC - QDA



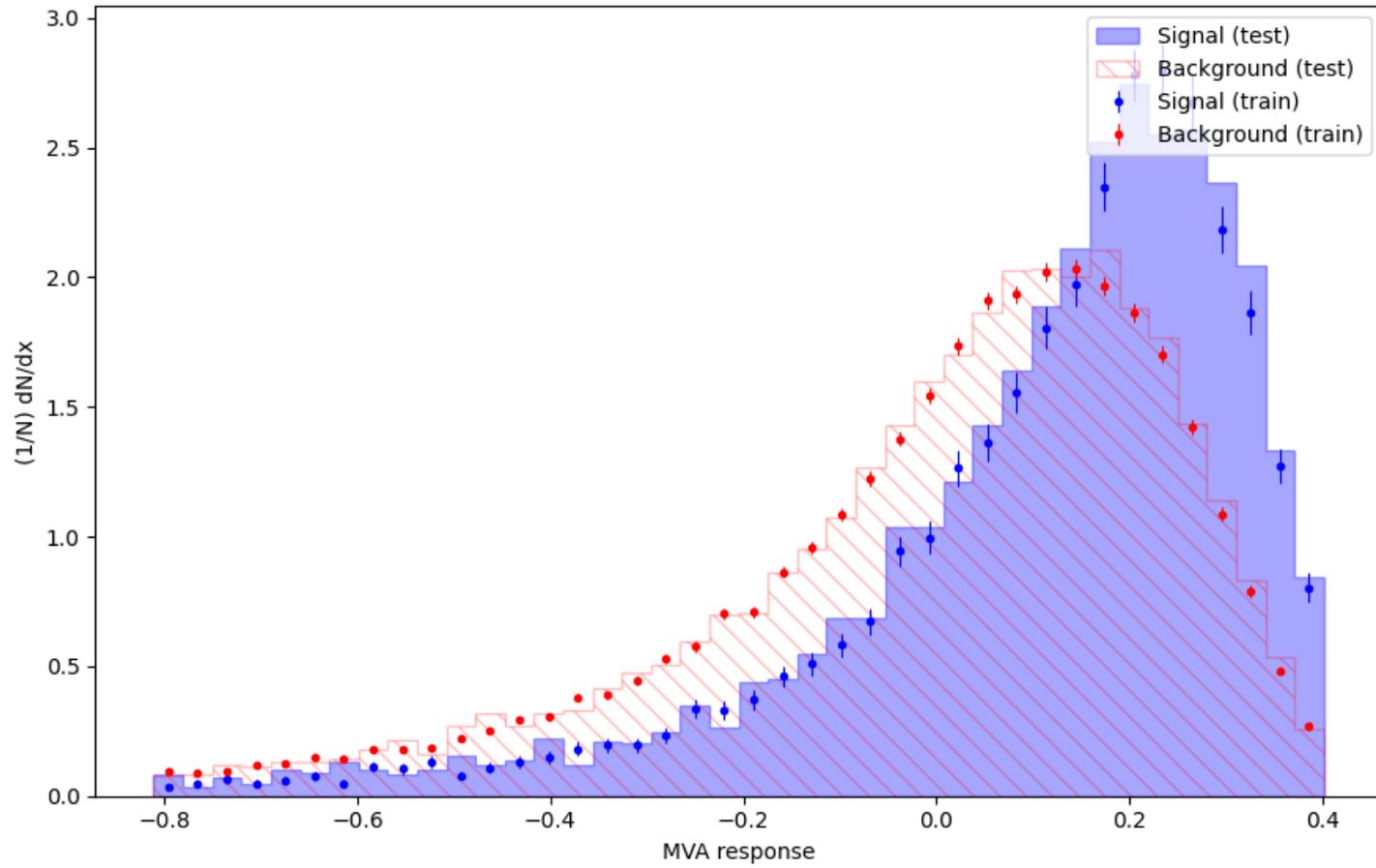
Python overtraining check (TMVA-style): RandomForest



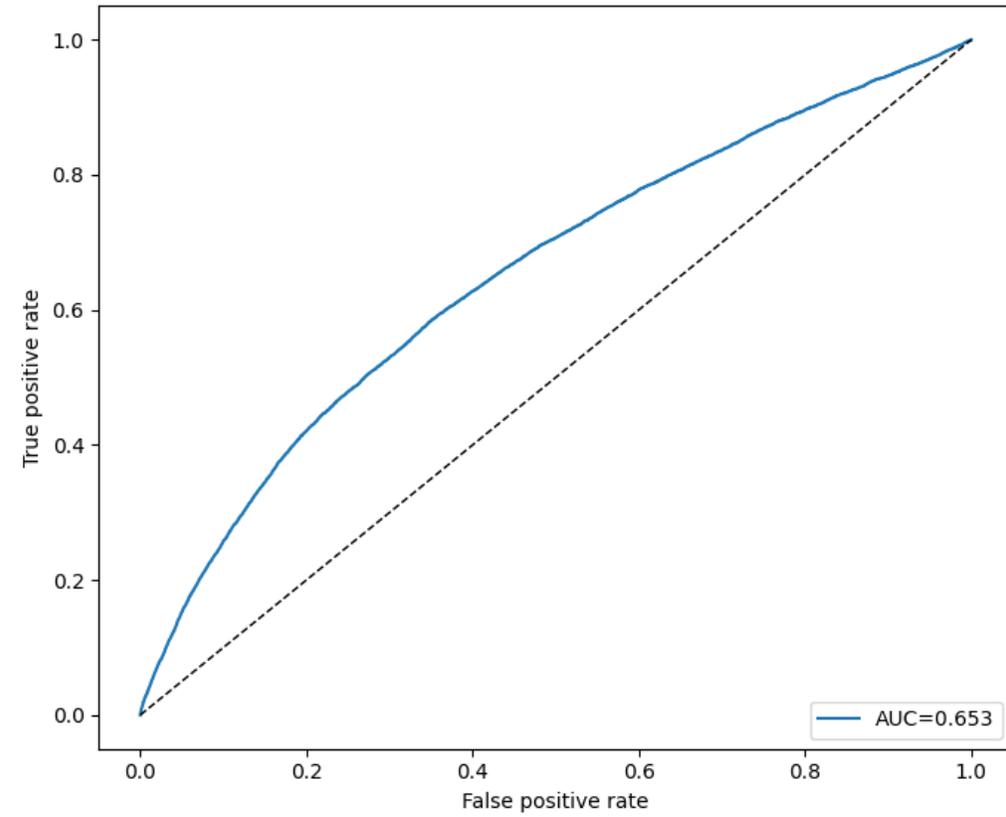
OOF ROC - RandomForest



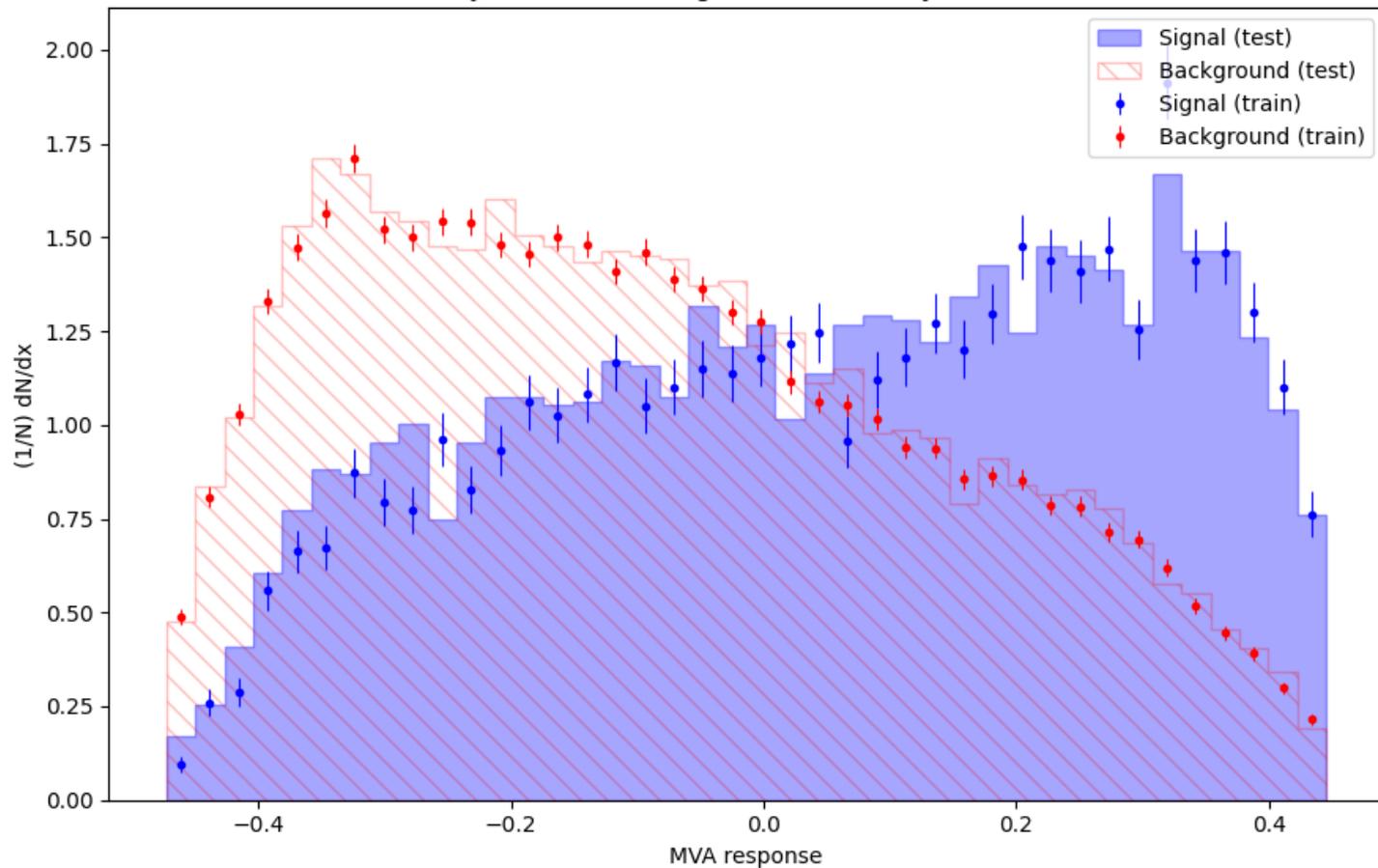
Python overtraining check (TMVA-style): SGDClassifier



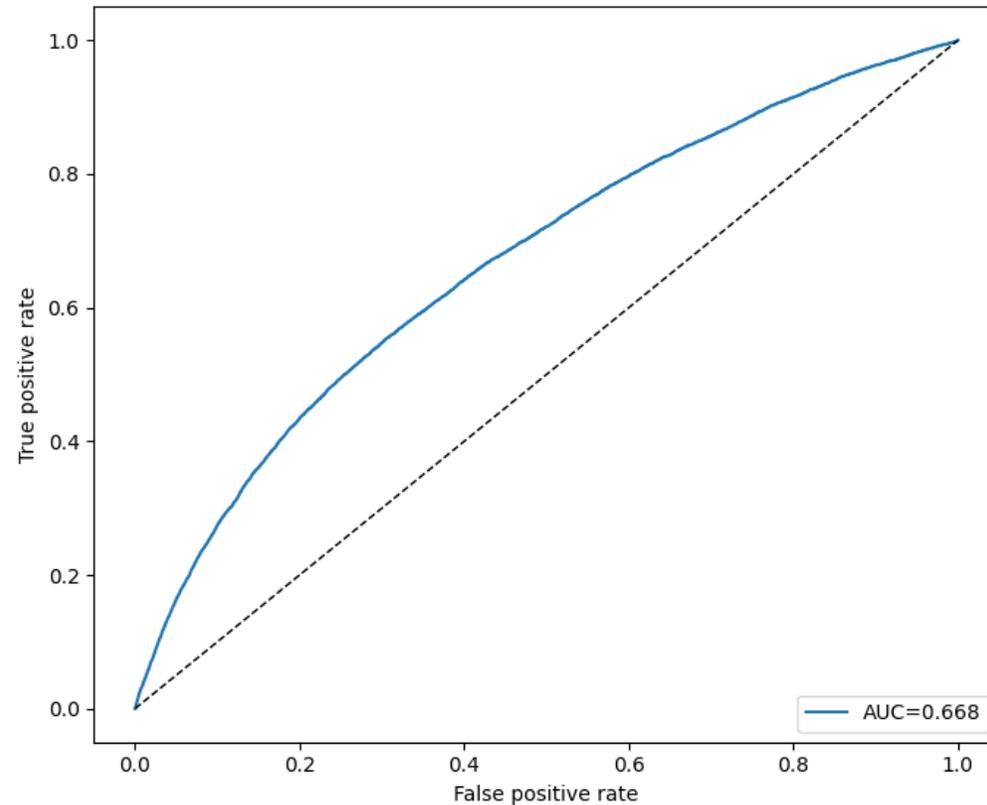
OOF ROC - SGDClassifier



Python overtraining check (TMVA-style): SVC



OOF ROC - SVC

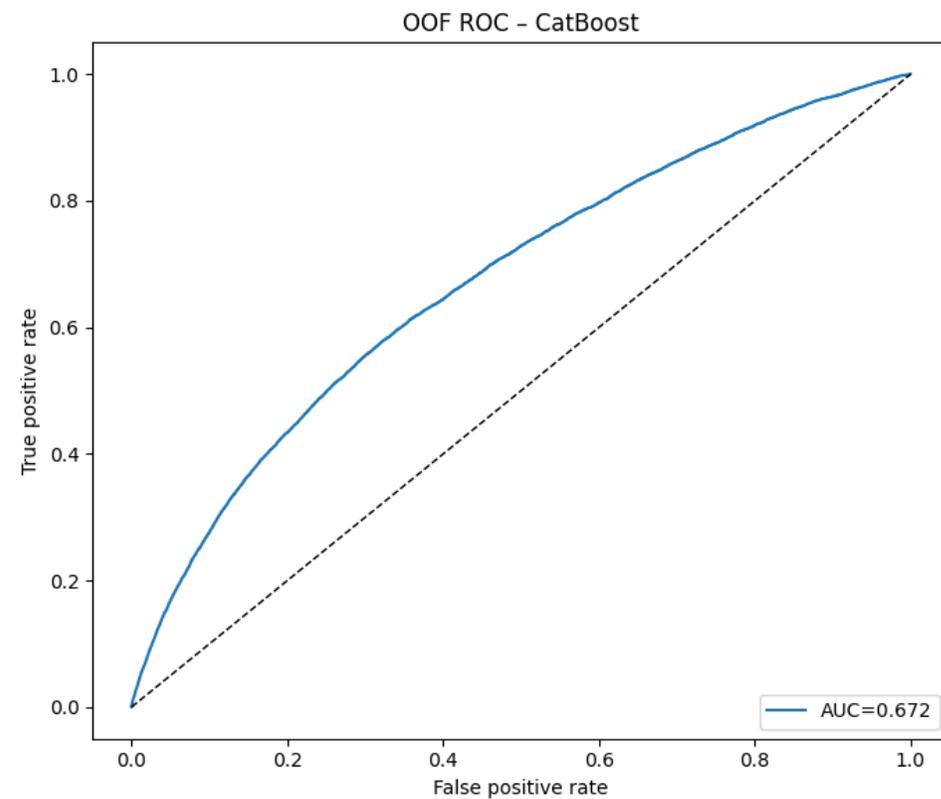
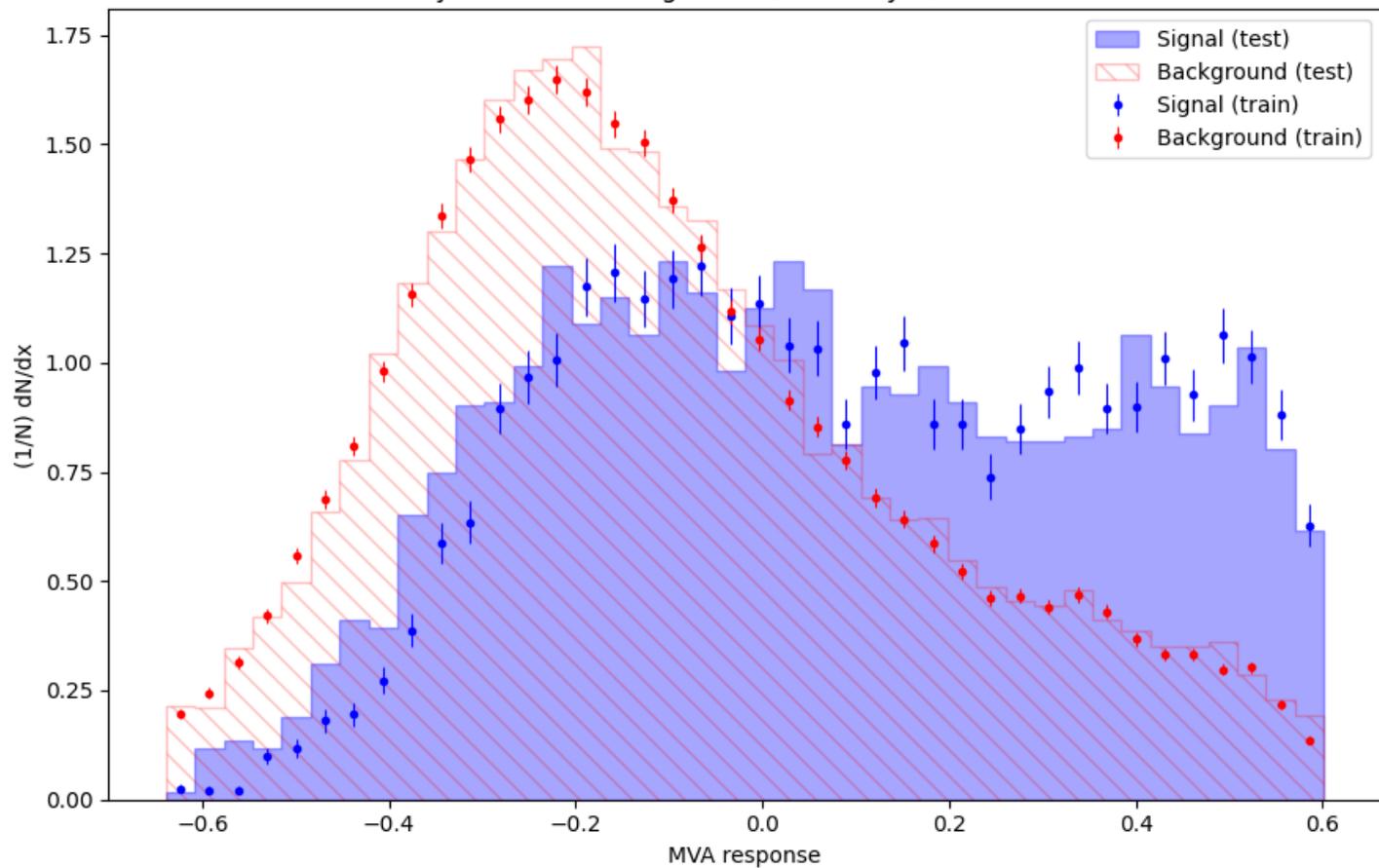


Comparison Table of all platforms with kfold=5

Model family (common name)	TMVA algorithm (if available)	TMVA best_mean_auc	R AUC (oof_weighted)	Python mean_fold_auc	Python overall_oof_auc	Δ AUC (R – Py overall)	Δ AUC (TMVA – Py overall)
Gradient Boosting (GBM)	BDTG	0.670094	1.000000	0.675134	0.674820	+0.325180	–0.004726
MLP	MLP	0.668192	0.661500	0.665761	0.664427	–0.002927	+0.003765
Random Forest (RF)	—	—	0.656103	0.672187	0.671608	–0.015505	—
Logistic Regression (GLM)	—	—	0.643484	0.639506	0.639507	+0.003977	—
Naive Bayes (NB)	—	—	0.640344	0.640700	0.639867	+0.000477	—
SVM (SVM/SVC)	—	—	0.590934	0.676572	0.676277	–0.085343	—

CATBOOST Algorithm added with 9 variables only

Python overtraining check (TMVA-style): CatBoost



Algorithm (Python)	outer_k	inner_k	n_iter	mean_fold_auc	std_fold_auc	overall_oof_auc	mean_fold_wacc	std_fold_wacc	overall_oof_wacc
AdaBoost	5	3	25	0.66	0.01	0.66	0.62	0	0.62
Bagging	5	3	25	0.65	0.01	0.65	0.5	0	0.5
ExtraTrees	5	3	25	0.67	0.01	0.67	0.62	0.01	0.62
GaussianNB	5	3	25	0.64	0.01	0.64	0.6	0	0.6
GradientBoosting	5	3	25	0.67	0.01	0.67	0.63	0.01	0.63
HistGradientBoosting	5	3	25	0.67	0.01	0.67	0.62	0.01	0.62
KNN	5	3	25	0.65	0.01	0.65	0.5	0	0.5
LDA	5	3	25	0.63	0.01	0.63	0.5	####	0.5
LinearSVC	5	3	25	0.64	0.01	0.64	0.6	0.01	0.6
LogisticRegression	5	3	25	0.64	0.01	0.64	0.6	0.01	0.6
MLP	5	3	25	0.67	0.01	0.67	0.5	0	0.5
QDA	5	3	25	0.65	0.01	0.65	0.51	0	0.51
RandomForest	5	3	25	0.67	0.01	0.67	0.62	0.01	0.62
SGDClassifier	5	3	25	0.65	0.01	0.65	0.57	0	0.57
SVC	5	3	12	0.67	0.01	0.67	0.62	0	0.62
CatBoost	5	3	25	0.67	0.006241	0.67	0.62	0.0049	0.627

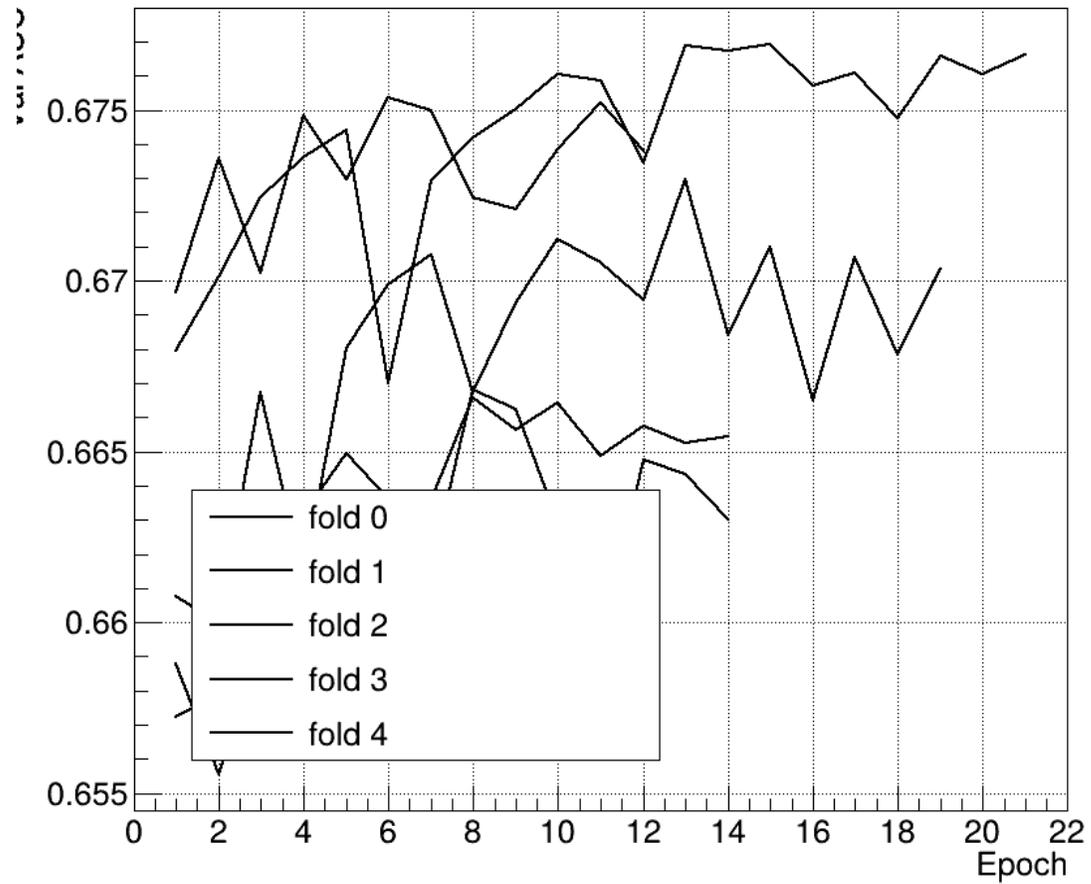
Metric	Description
outer_k	Number of outer cross-validation folds used to estimate generalization performance (the “test folds” in nested CV).
inner_k	Number of inner cross-validation folds used for hyperparameter tuning/model selection (the “validation folds” inside each outer split).
n_iter	How many tuning trials/iterations were run per inner-CV search (e.g., random search iterations or repeated fitting cycles).
mean_fold_auc	Mean ROC–AUC across the outer folds (average discrimination performance over all outer test folds).
std_fold_auc	Standard deviation of ROC–AUC across outer folds (how variable/unstable AUC is across splits).
overall_oof_auc	ROC–AUC computed from out-of-fold (OOF) predictions pooled across all outer folds (one prediction per case from a model that didn’t train on it).
mean_fold_wacc	Mean weighted accuracy across outer folds (accuracy adjusted by class weights or class frequency weighting, depending on your scoring definition).
std_fold_wacc	Standard deviation of weighted accuracy across outer folds (split-to-split variability in weighted accuracy).
overall_oof_wacc	Weighted accuracy computed on pooled OOF predictions across all outer folds (single “full-sample” estimate from OOF predictions).

Transformer's Model results

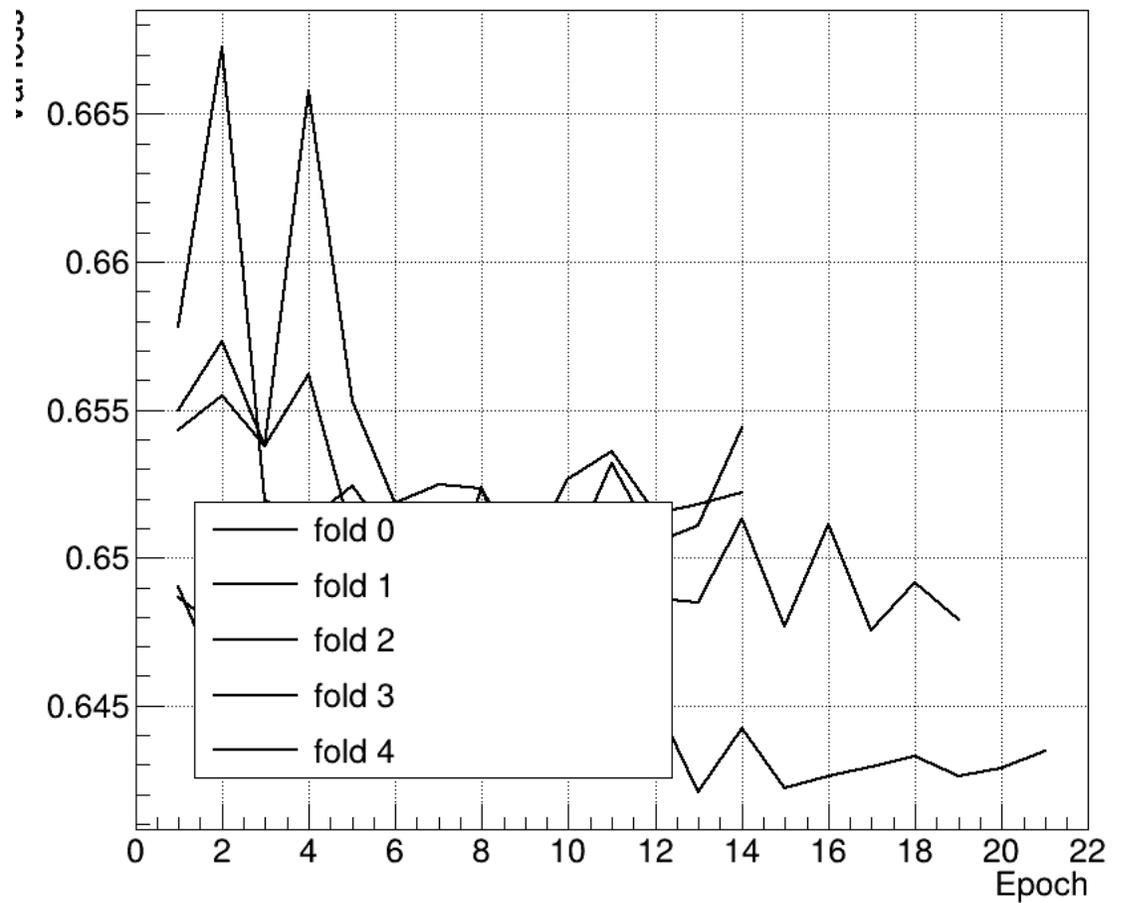
fold	n_val_events	auc	wacc	best_epoch	wall_sec	cpu_user_sec	cpu_sys_sec	SoverB_raw	SoverB_bal
0	17514	0.672968	0.621107	13	3650.041	13514	95.54988	0.166396	1
1	17514	0.6666	0.617789	8	4586.651	17450.66	131.4563	0.166396	1
2	17513	0.675379	0.634363	6	4713.912	16461.25	130.6868	0.166396	1
3	17513	0.666812	0.617936	8	2571.248	9011.478	37.38613	0.166396	1
4	17512	0.676911	0.629515	15	2532.314	9455.203	190.0953	0.166396	1
overall	87566	0.669226	0.624142						

Metric	Description
fold	k-fold split index (identifier for the specific cross-validation partition).
n_val	Number of validation events in that fold (size of the validation set).
auc	ROC AUC (Area Under the Curve) computed on the validation set for this fold.
wacc	Weighted accuracy (class-weighted accuracy) computed on the validation set for this fold.
best_epoch	The epoch (training iteration) at which the validation score was optimal for this fold.
wall_sec	Elapsed wall-clock time in seconds for this fold's training/validation run.
cpu_user_sec	CPU user-mode time in seconds consumed during this fold (time spent executing your code).
cpu_sys_sec	CPU system/kernel time in seconds for this fold (time spent on OS-level operations).
SoverB_raw	Signal-to-background ratio (S/B) computed on the raw event yields in this fold.
SoverB_bal	Signal-to-background ratio (S/B) computed after applying class balancing or reweighting techniques.

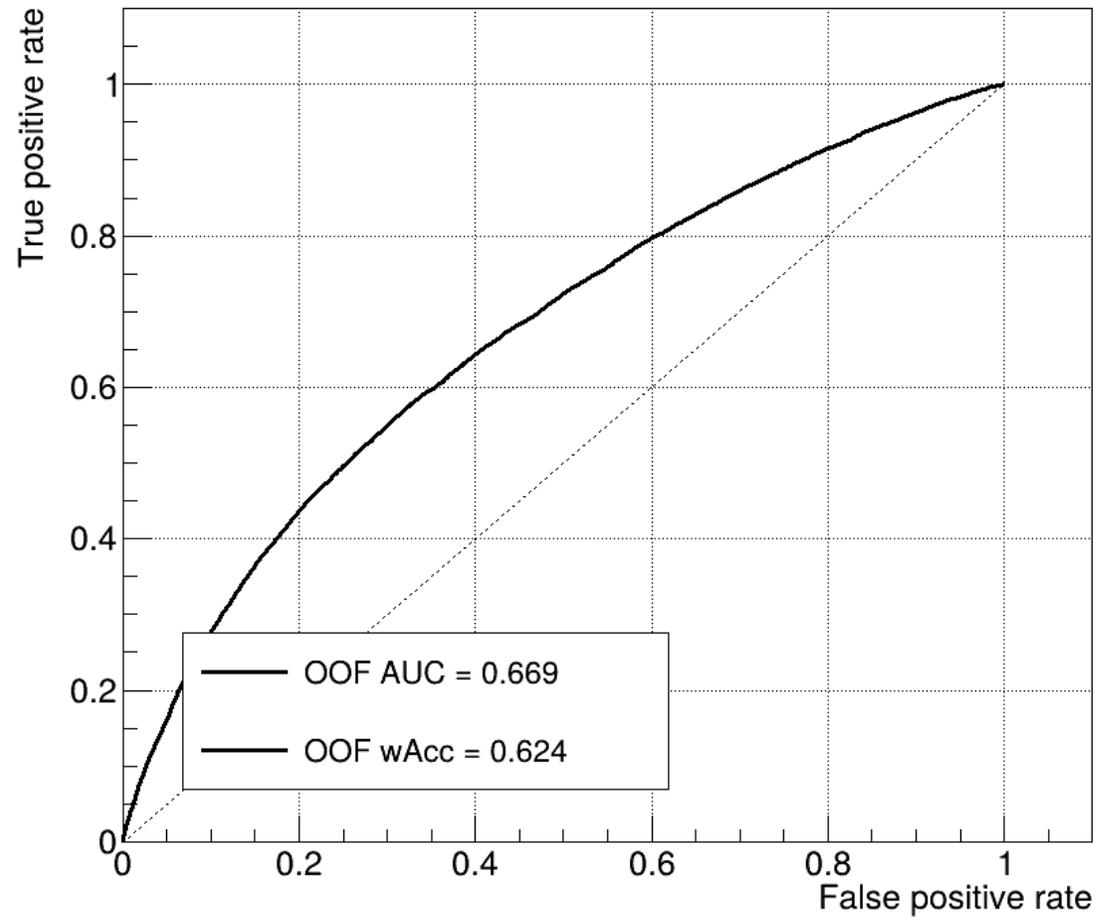
Val AUC by epoch



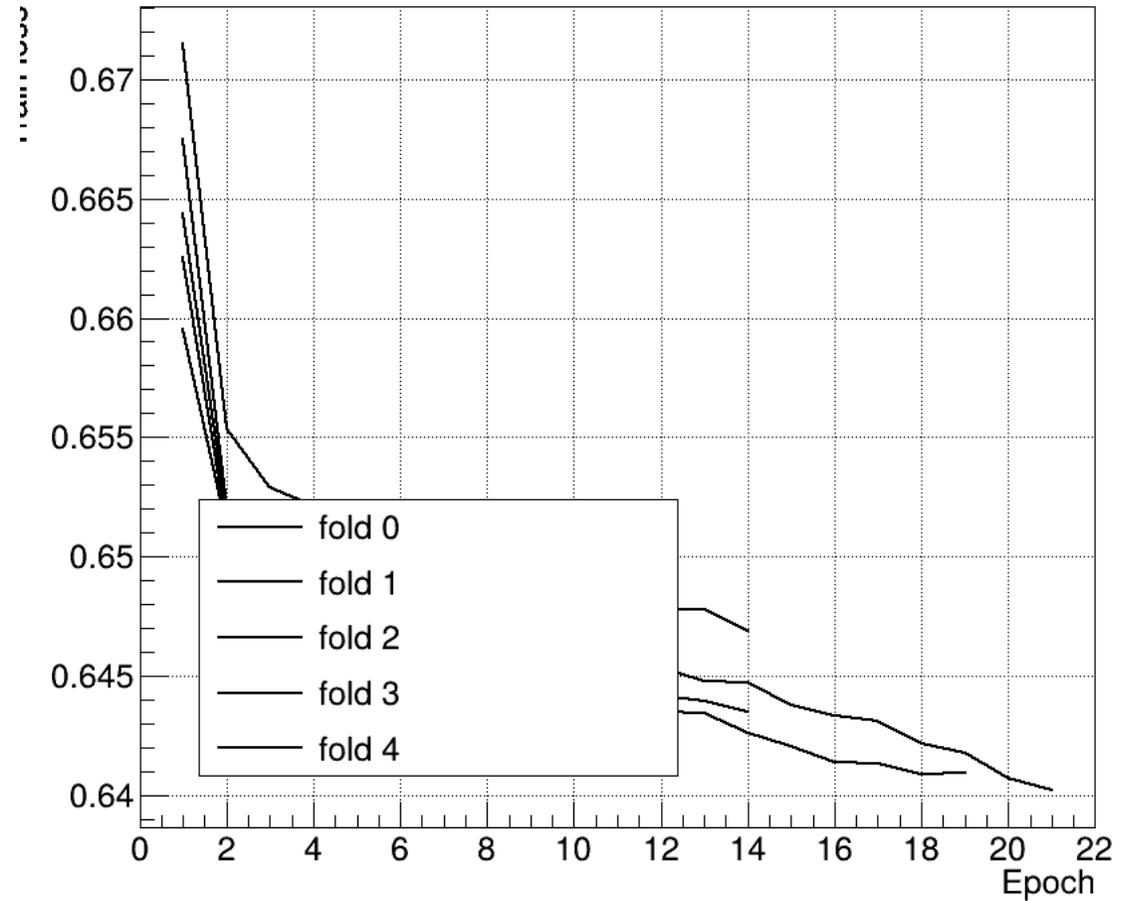
Val loss by epoch



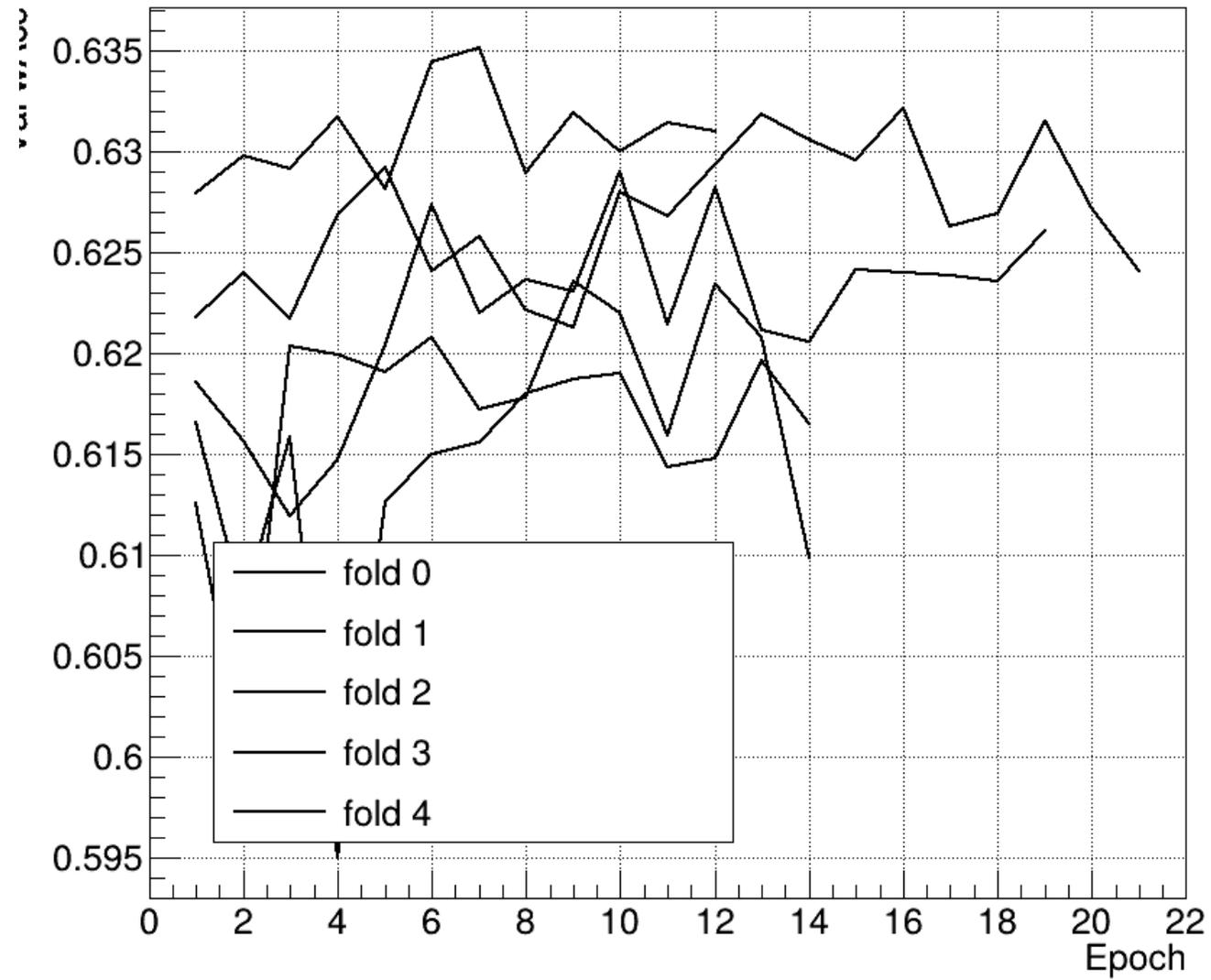
Transformer OOF ROC



Train loss by epoch



Val weighted accuracy by epoch



Best Performance across each platform

TMVA, Python and R

with k-fold=5

Platform	Best algorithm	Best score (AUC)	Score type (as reported)
TMVA	BDTG	0.670094	best_mean_auc (k-fold)
Python	GradientBoosting	0.670033	overall_oof_auc (out-of-fold)
R	MLP	0.660152	auc_oof_weighted (out-of-fold, weighted)
Condor	Transformers (input n-tuples)	0.669226	

```
-- Schedd: inter.phys.cluster : <192.168.192.12:9618?... @ 02/24/26 16:05:13
```

OWNER	BATCH_NAME	SUBMITTED	DONE	RUN	IDLE	TOTAL	JOB_IDS
nfiroz	ID: 2355	2/24 00:17	-	1	-	1	2355.0

```
Total for query: 1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```

```
Total for nfiroz: 1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```

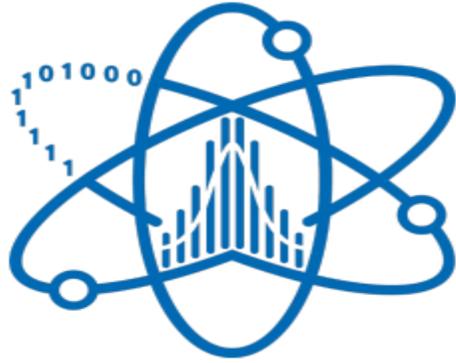
```
Total for all users: 101 jobs; 0 completed, 0 removed, 0 idle, 7 running, 94 held, 0 suspended
```

```
[nfiroz@inter ~]$
```

- 1. Could not work on Caret package this week
(Still running on condor, there were errors in terminal it took more than 12 hrs so I quit)**
- 2. And working on preparing script for RNN-LSTM and see how it goes this week with article writing**



National Research
**Tomsk
State
University**



**Лаборатория
анализа данных
физики высоких энергий**

Томского
государственного
университета

Thank you for your attention!!!