Machine Learning applications in high energy physics

Viraj Thakkar
MSc Thesis Presentation
Supervised by Dr. Bedangadas Mohanty
NISER,Bhubaneswar

Motivation

- The aim of this project is to use **Machine Learning** techniques for the purpose of classification of signal and background events of resonance particle K^{*0} and **improve the significance** of the signal as compared to traditional approach.

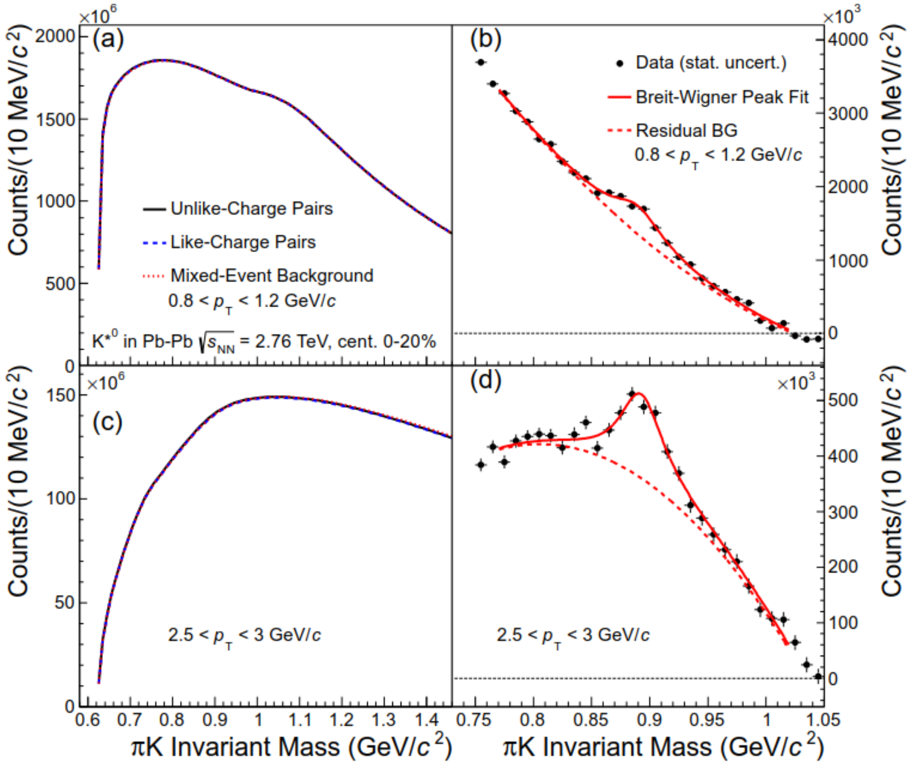


Fig: Large combinatorial background (left), signal is rarely visible

Reference: B.Abelev et al, ALICE Collaboration: Phys. Rev. C 91 (2015) 024609

Machine Learning

- Machine Learning(ML) is a field of Artificial Intelligence(AI) that uses statistical techniques to provide computer systems the ability to ``learn" from data, without being explicitly programmed.

Classification Problem

ML algorithm

Identifying data point

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

Belongs to

which class?

y=0 (background) OR y=1 (signal)

Classifier: $h_\theta(x)$
θ denotes ``weights" of the classifier.
$h_\theta(x)$ =0 or 1

Features: n number of discriminating variables defines the data point **x**

Supervised Machine Learning: Data is labelled.

The Machine Learning process

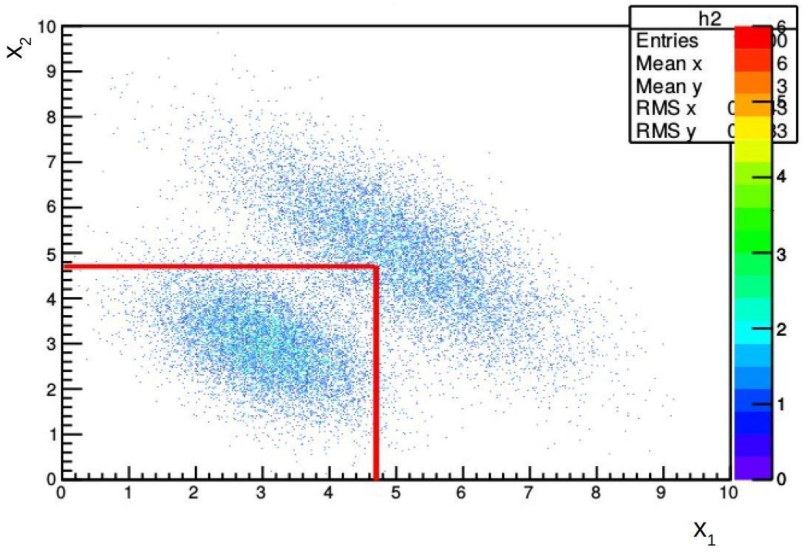
Training → Testing → Application

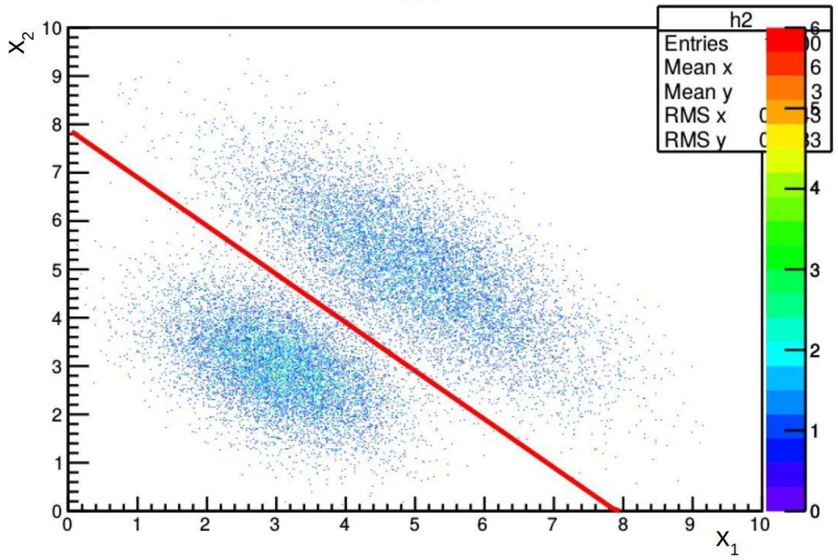
Training	**Testing**	**Application**
→The classifier uses data points **x** and learns about the features	→The classifier makes predictions on untrained data points **x**	→The classifier makes predictions on new unknown data points **x**
→Classifier learns the data in the form of weights $\boldsymbol{\theta}$	→Compare the classifier predicted output $\mathbf{h_{\theta}(x)}$ with the actual label y of the data **x**	→Predictions are made using the weights $\boldsymbol{\theta}$ which the classifier learned during training
→Uses the label y=0 or 1 for the data point **x**	→Does not use the labels y=0 or 1 for the data point **x** while testing.	→Labels y=0 or 1 are not known. Classifier predicts it.

Toolkit For Multivariate Analysis (TMVA)

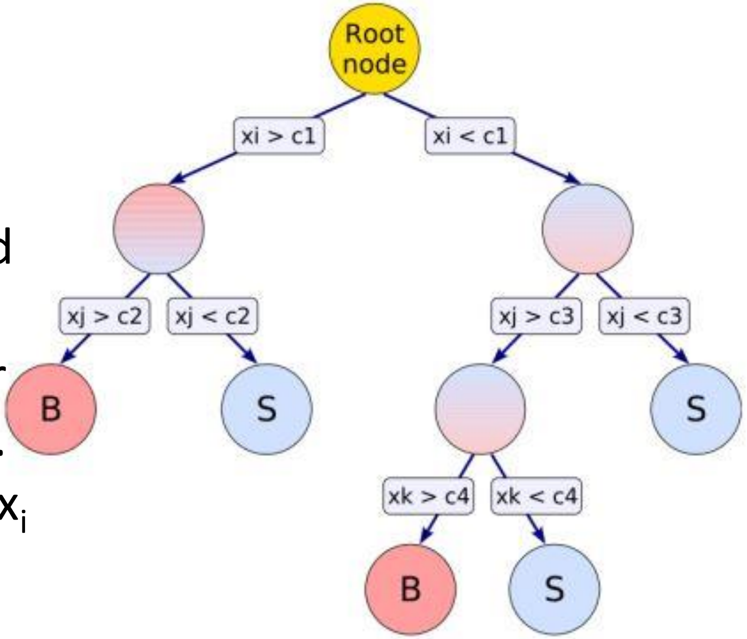
- TMVA provides a framework for multivariate analysis (ML classification algorithms) implemented in ROOT
- We have used Boosted Decision Tree(BDT) as our ML classification algorithm.

Why we need Multivariate Analysis? : Better separation





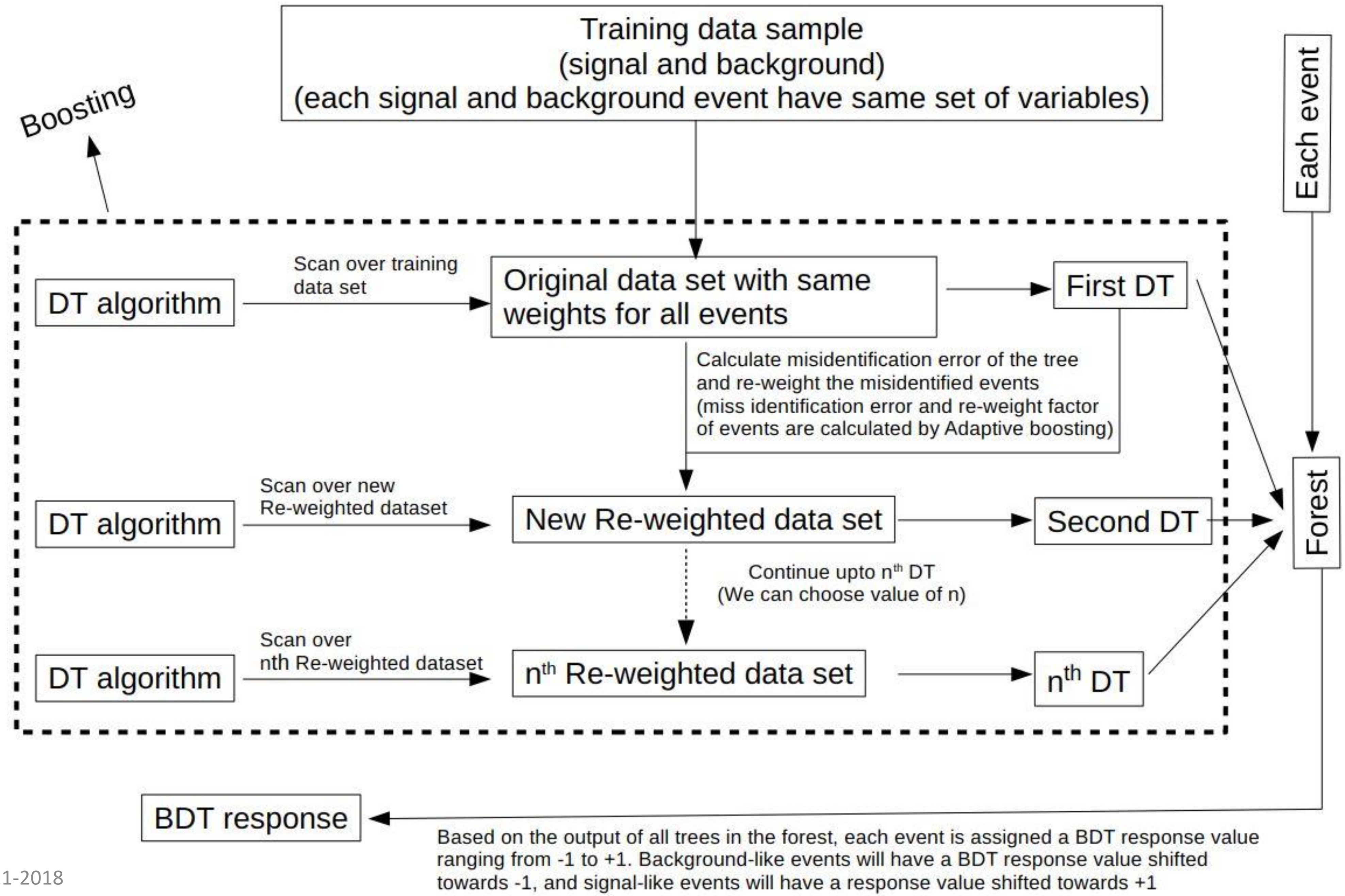
Decision Trees



- A decision tree(DT) is a tree-like structure that uses a branching method to illustrate possible outcome of a decision.
- An event is either classified as signal or background by either passing or not passing a condition (cut) on a specific node until a decision is made.
- A sequence of binary splits using the discriminating variables(features) x_i is applied to the data.
- The split is such that we get the best separation between signal and background.

Boosting

- Boosted Decision Trees (BDT) is a prediction model which uses an ensemble of ``weak classifiers" (decision trees).
- It is a model designed to make fewer and fewer mistakes as more trees are added to it.
- BDT is an algorithm which combines forests of DTs, each weighted according to their importance.


Training data sample
(signal and background)
(each signal and background event have same set of variables)
Boosting
Each event
DT algorithm
Scan over training data set
Original data set with same weights for all events
First DT
Calculate misidentification error of the tree and re-weight the misidentified events (miss identification error and re-weight factor of events are calculated by Adaptive boosting)
DT algorithm
Scan over new Re-weighted dataset
New Re-weighted data set
Second DT
Forest
Continue upto nth DT
(We can choose value of n)
DT algorithm
Scan over nth Re-weighted dataset
nth Re-weighted data set
nth DT
BDT response
Based on the output of all trees in the forest, each event is assigned a BDT response value ranging from -1 to +1. Background-like events will have a BDT response value shifted towards -1, and signal-like events will have a response value shifted towards +1

Data Set

We have used pp collisions data at $\sqrt{s}$= 7 TeV from ALICE detector at LHC

Reconstructed Monte Carlo

System: p+p

Period: LHC10f6a

Event generator: PYTHIA

Events: ~10M

Data

System: p+p

Period: LHC10d

Events: ~10M

Event Selection: $|V_z| < 10$ cm.

Track Selection: ITS-TPC 2010 cuts

Reference: B. Abelev et al ALICE Collaboration: Eur. Phys. J. C72 (2012) 2183

Preparation of data for training, testing and application

$$K^{*0} \rightarrow K^{+} + \pi^{-}$$

- K^{*0} := Mother Particle
- K^{+} := Daughter Particle 1
- π^{-} := Daughter Particle 2

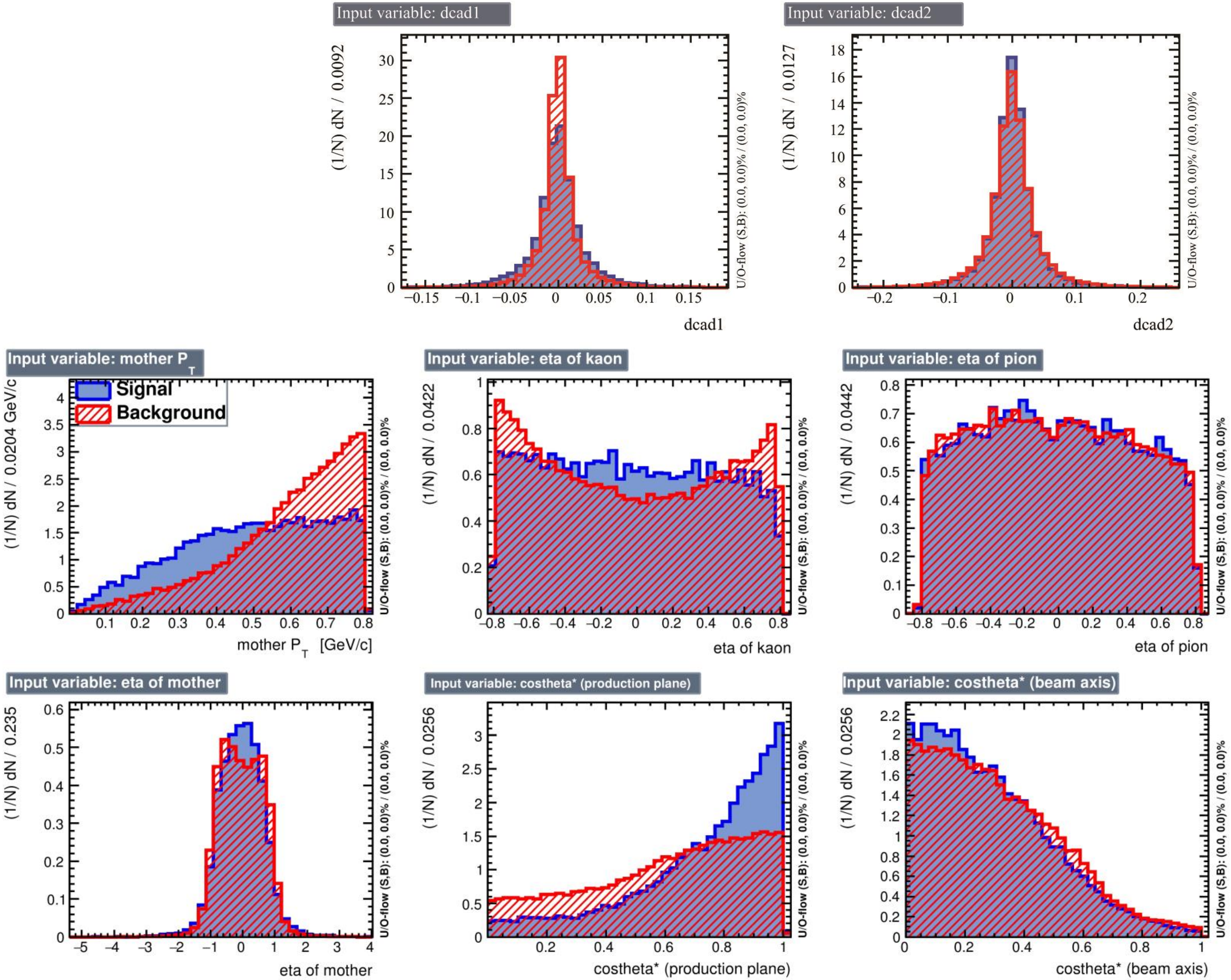
→**Signal** candidates are formed by K^{+} and π^{-} pairs which are decayed from K^{*0}. Generated level MC PID is used to select true K^{+} and π^{-}

→**Background** candidates are formed by same sign K and π like pairs which are selected by using energy loss information in TPC detector.

→**Unlike** pairs are all combinations of K^{+} and π^{-} pairs which constitutes signal plus background events.

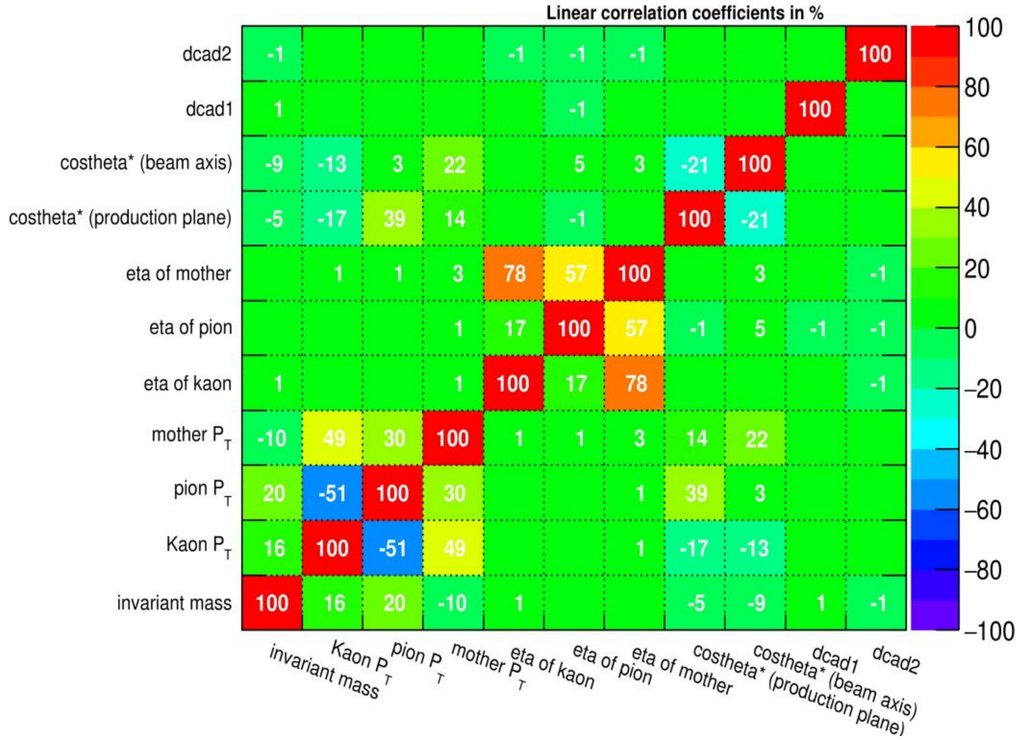
Input features: MC (PYTHIA)

$0<p_T(\text{mother})<0.8$ GeV/c

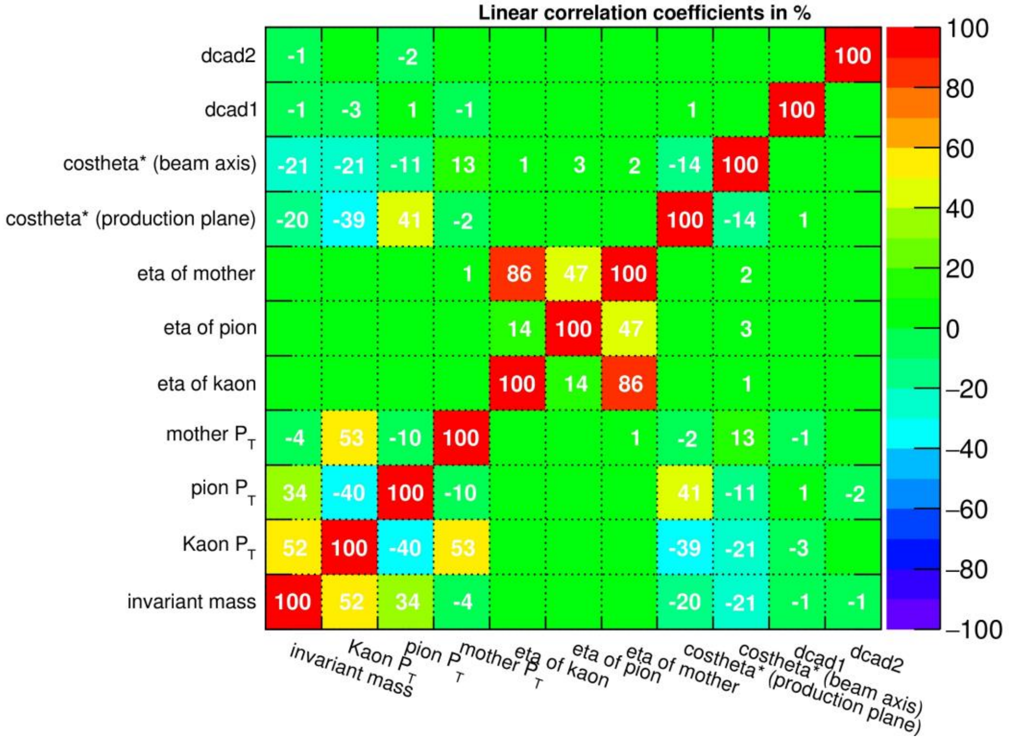


Correlation coefficient percentage

Correlation Matrix (signal)



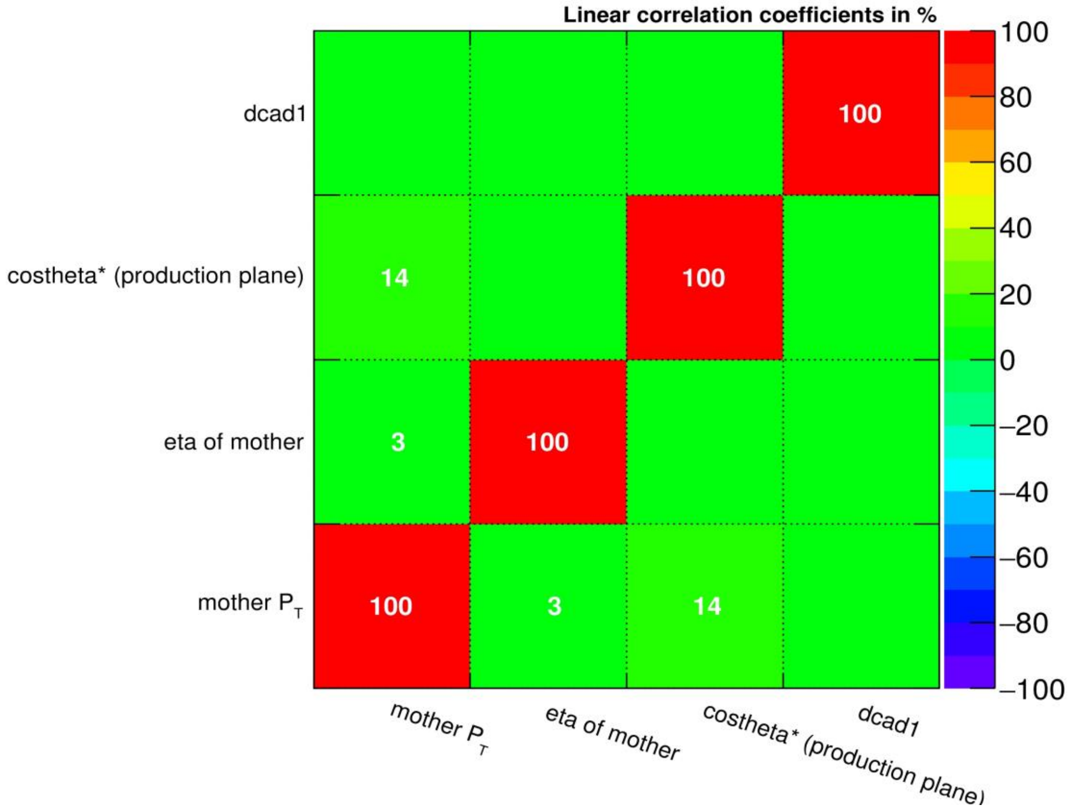
Correlation Matrix (background)



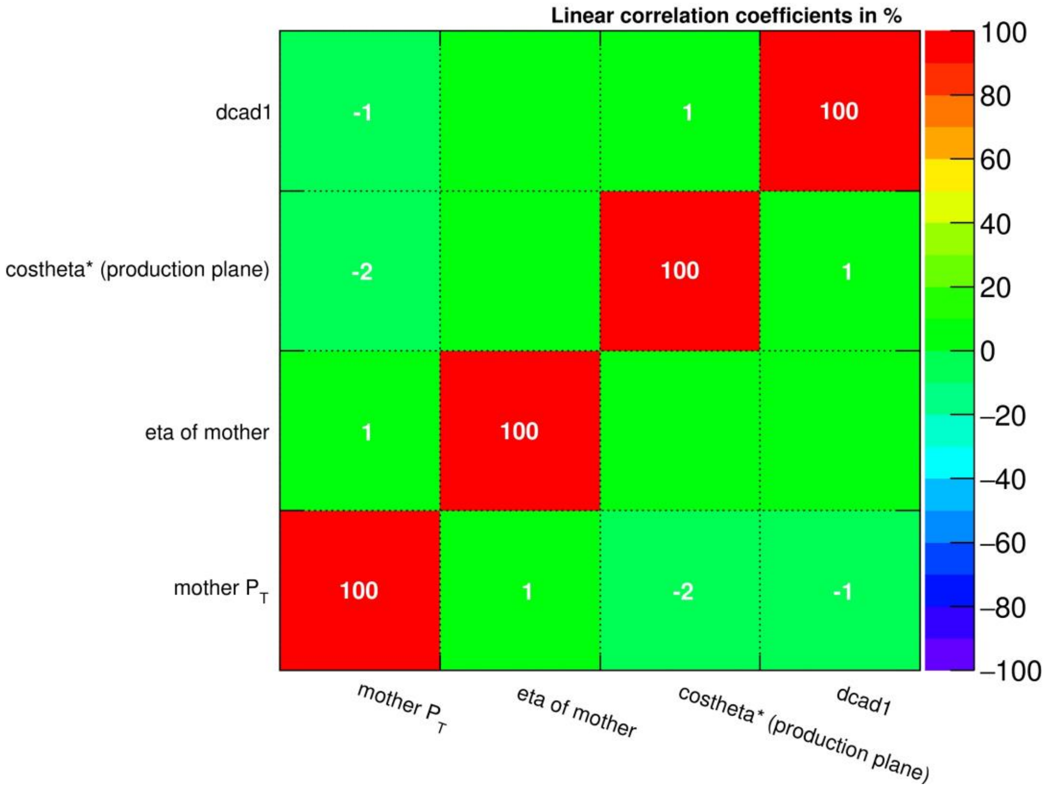
Input features used for training phase

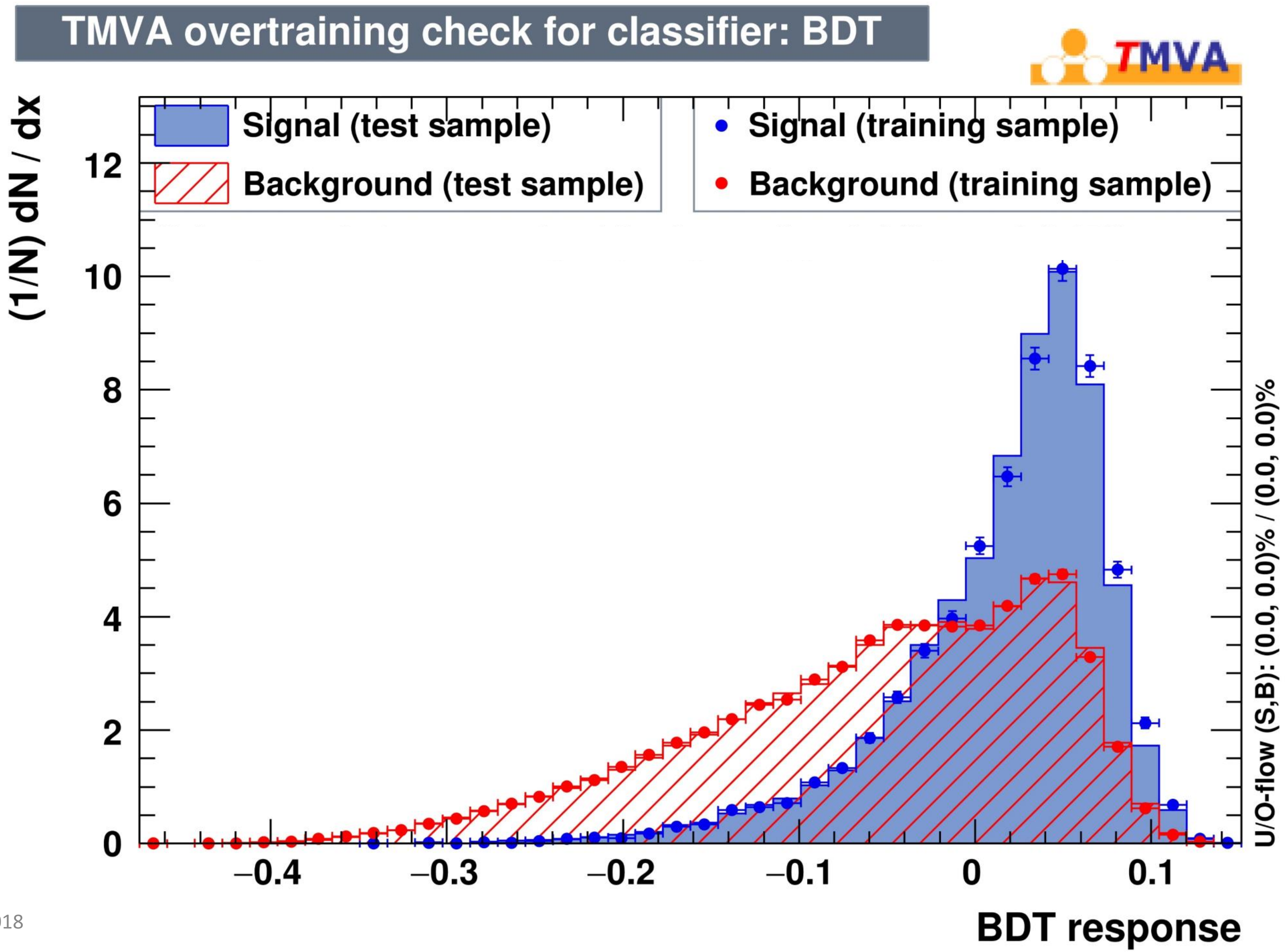
Selected features have less correlation with invariant mass

Correlation Matrix (signal)

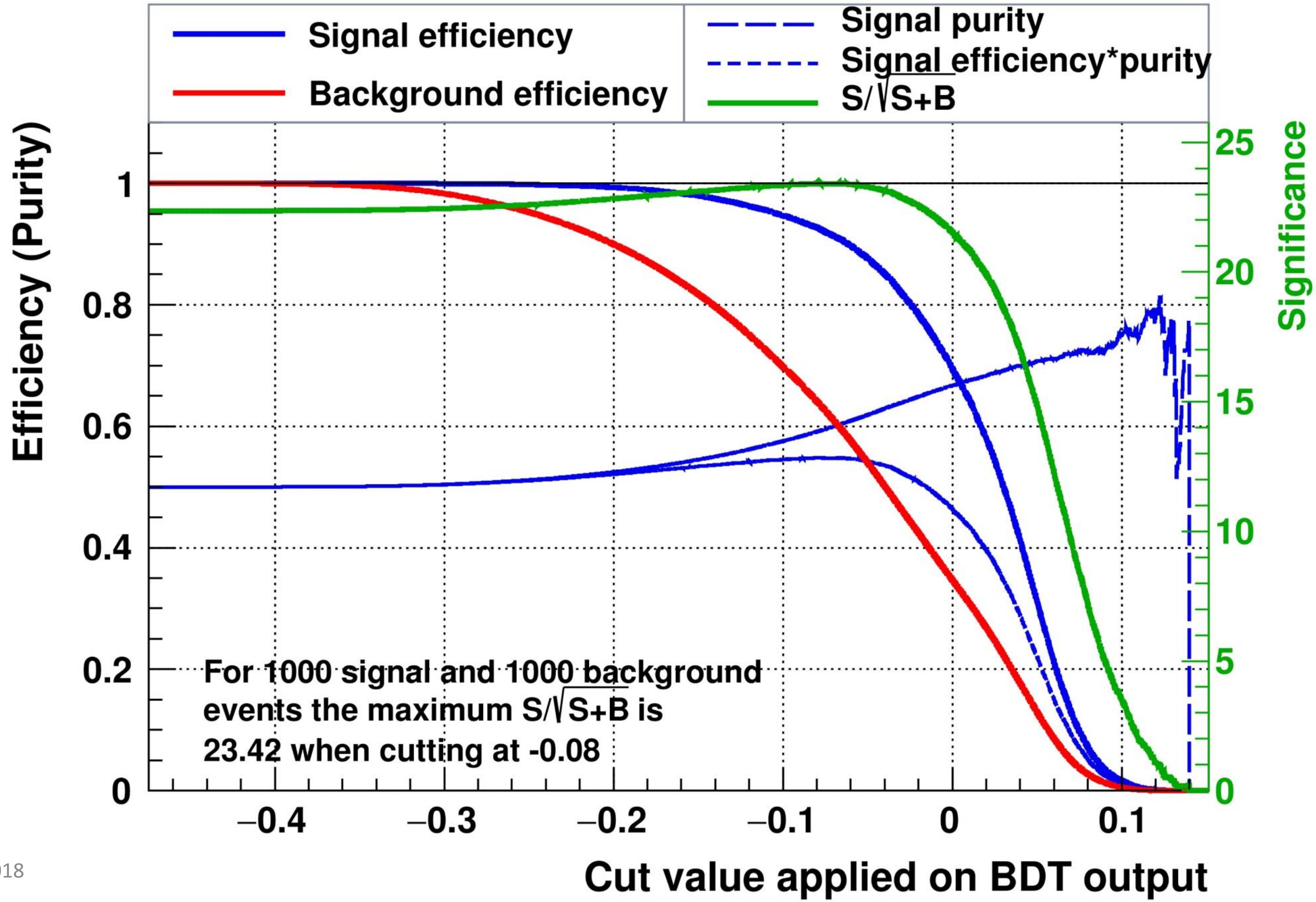


Correlation Matrix (background)

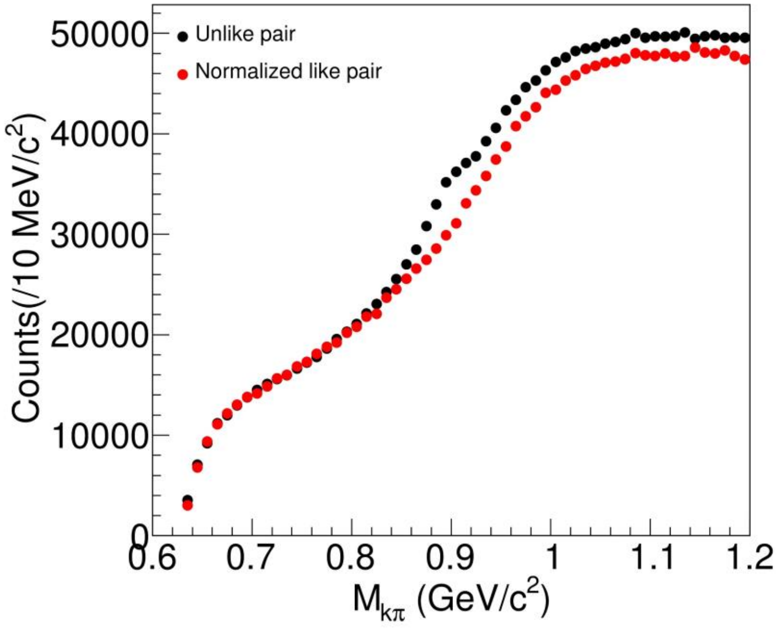


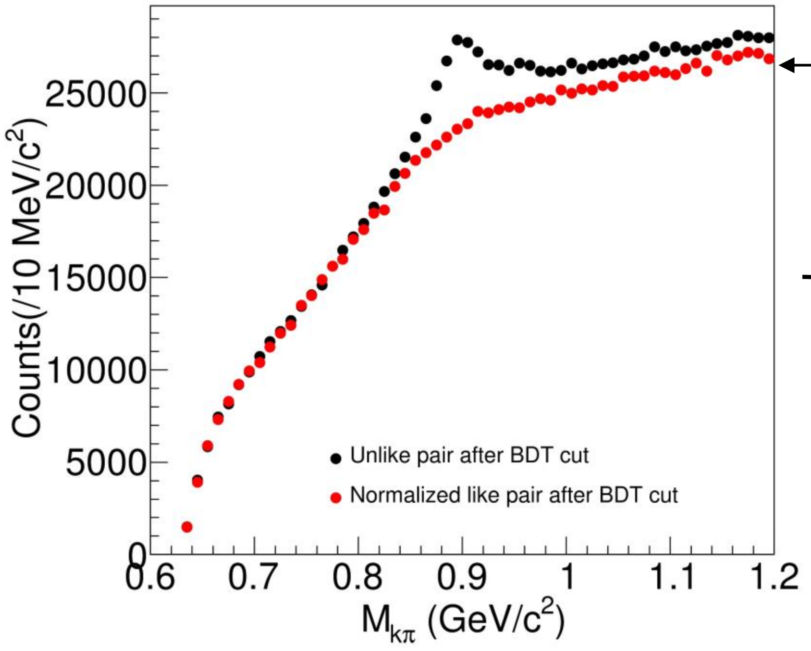

TMVA overtraining check for classifier: BDT
TMVA
Signal (test sample)
Background (test sample)
Signal (training sample)
Background (training sample)
(1/N) dN / dx
0
2
4
6
8
10
12
−0.4
−0.3
−0.2
−0.1
0
0.1
BDT response
U/O-flow (S,B): (0.0, 0.0)% / (0.0, 0.0)%

Cut efficiencies and optimal cut value



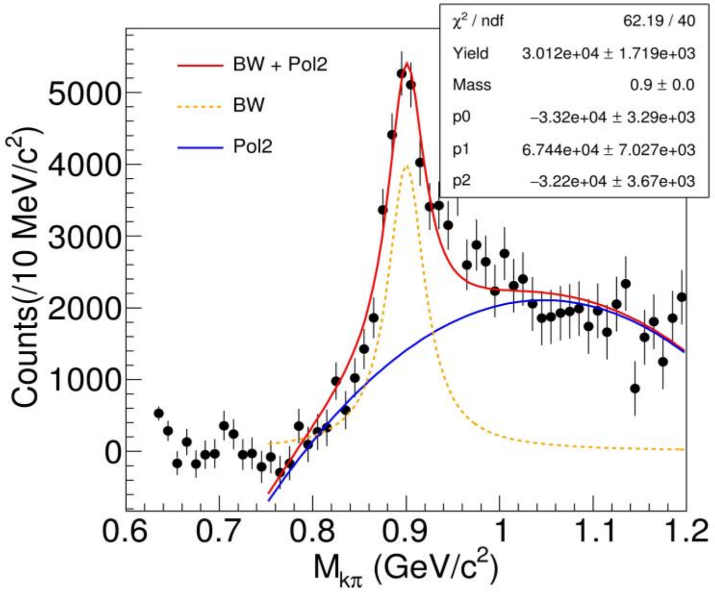
Invariant mass distribution: MC (PYTHIA)

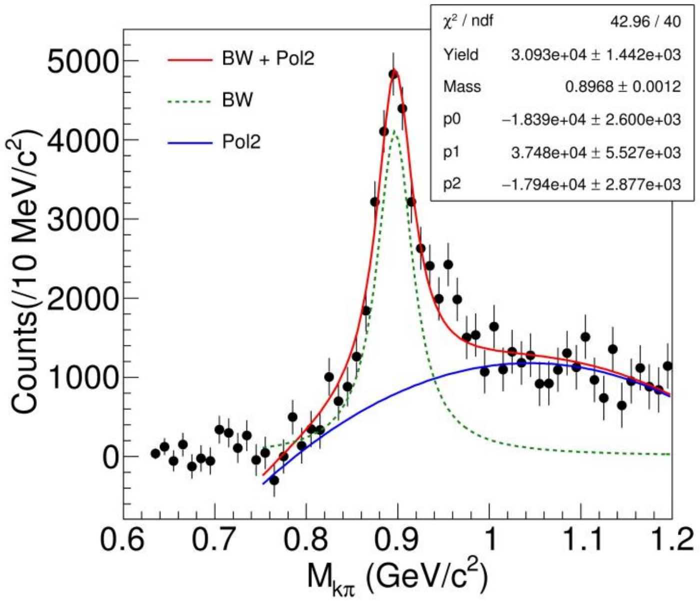




Signal is more visible!

Traditional approach





ML approach with BDT

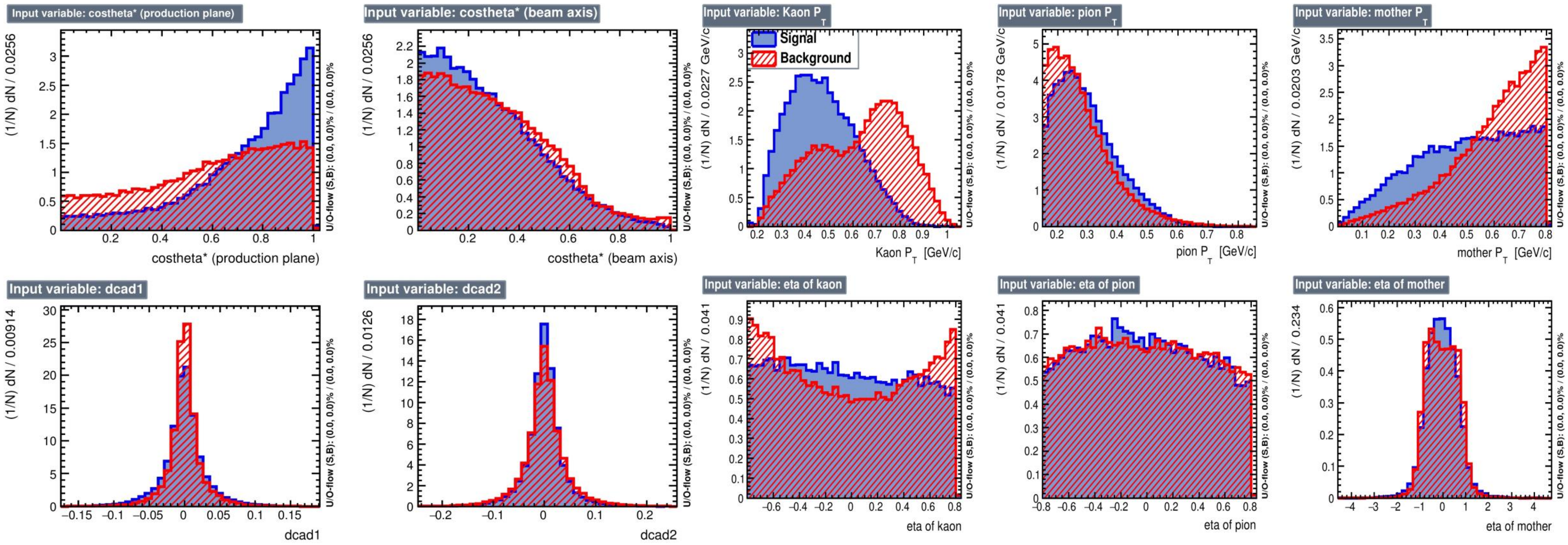
Comparison of signal obtained by Monte-Carlo

Method	χ^2/ndf	S	$S+B$	S/$\sqrt{S+B}$	Input Yield
Machine Learning	42.96/40	$\frac{30931.6}{0.92} = 33621.3$	906953	32.48	34253
Traditional Approach	62.19/40	30115	1355500	25.87	34253

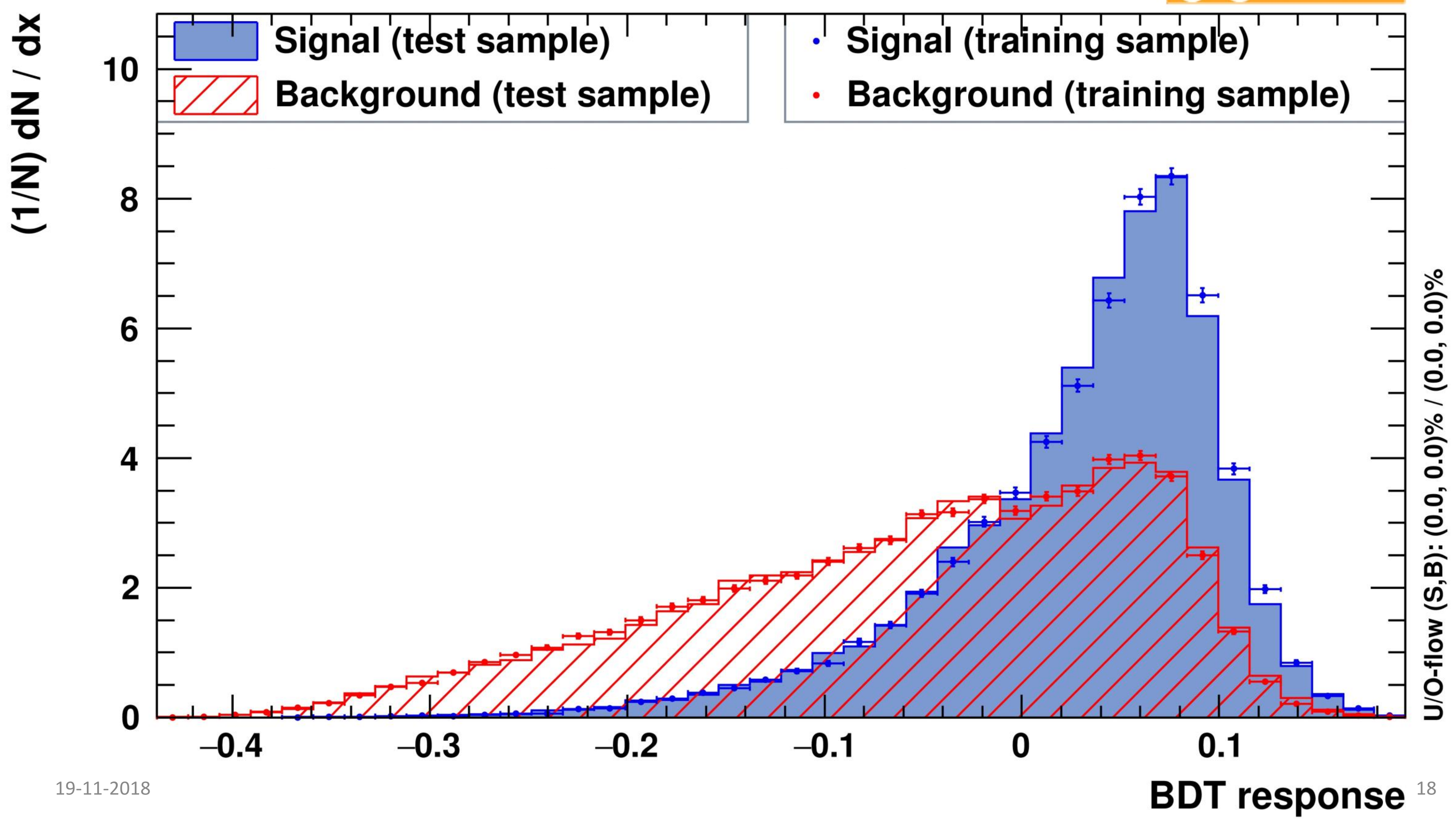
Extraction yield, significance and χ^2/ndf of the fit suggests a better extraction of signal by using Machine Learning via BDT for MC data.

Input Features: ALICE data

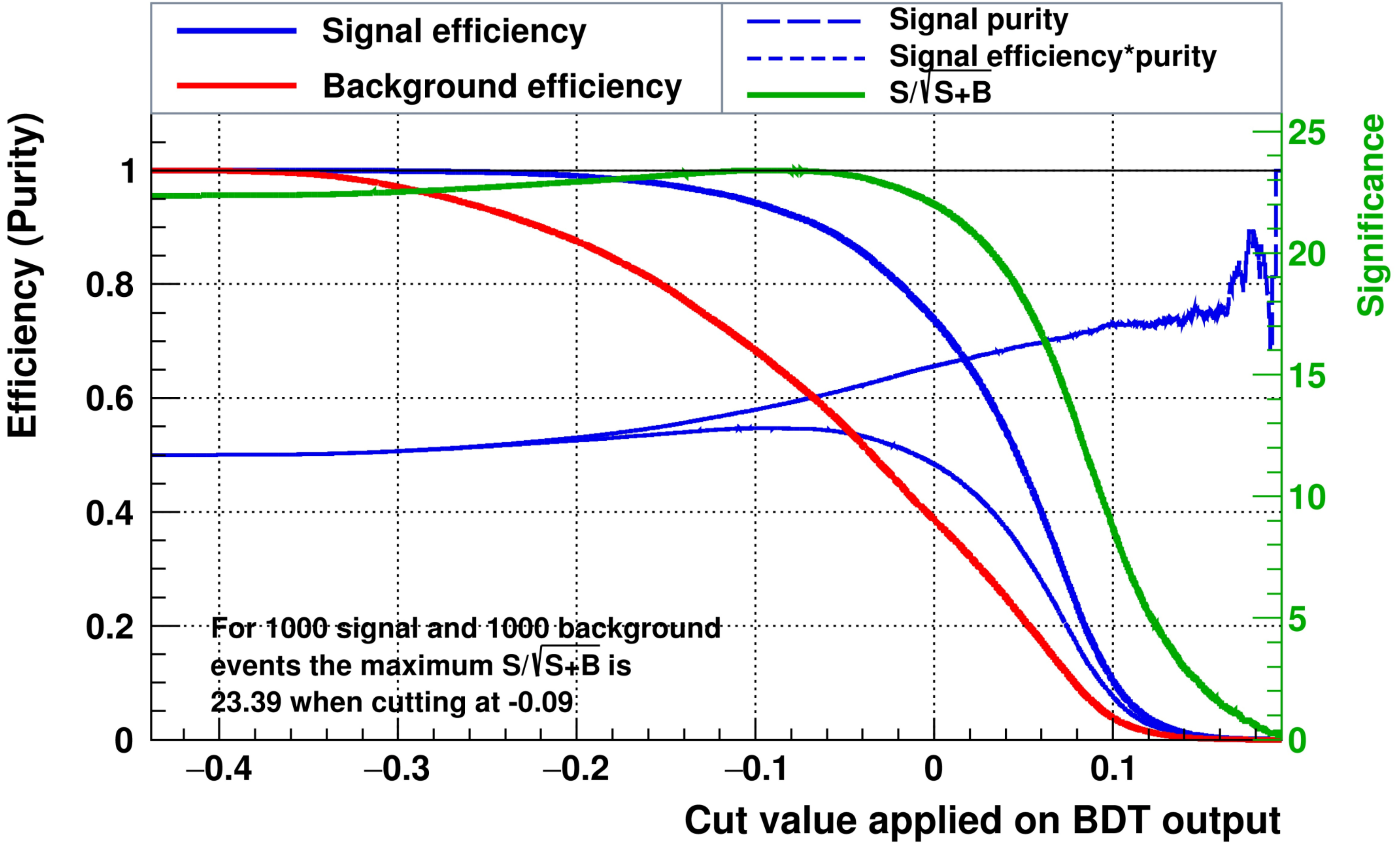
$0<p_T(\text{mother})<0.8$ GeV/c



TMVA overtraining check for classifier: BDT

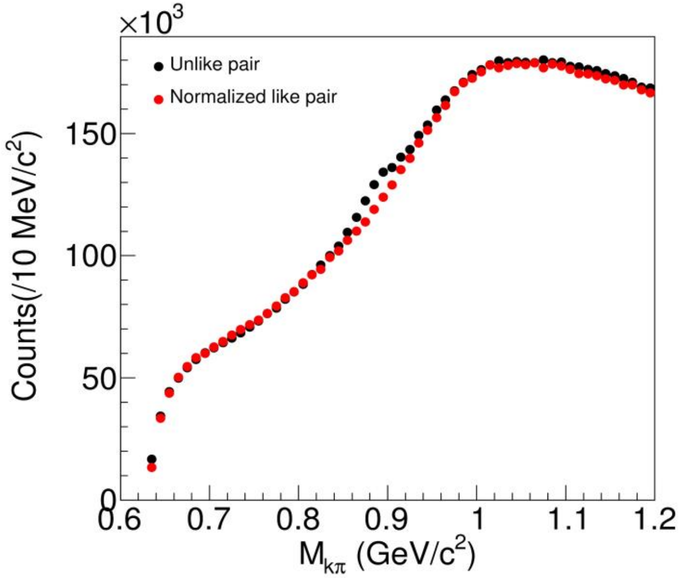


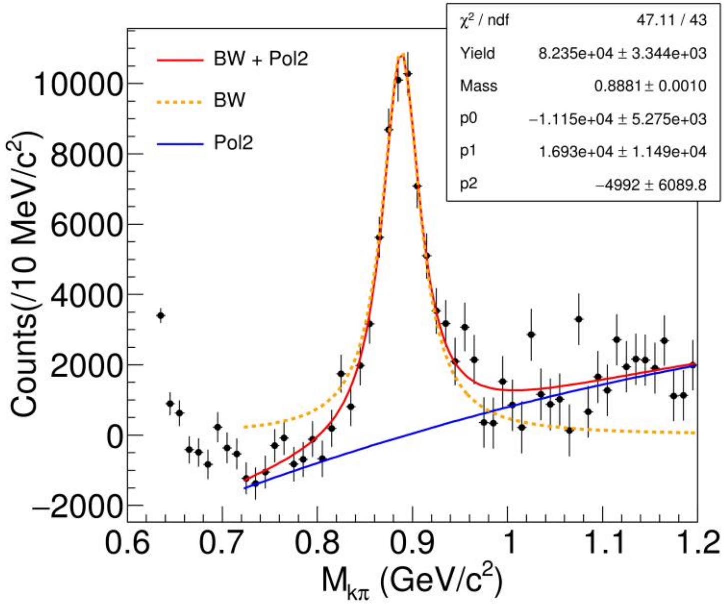
Cut efficiencies and optimal cut value

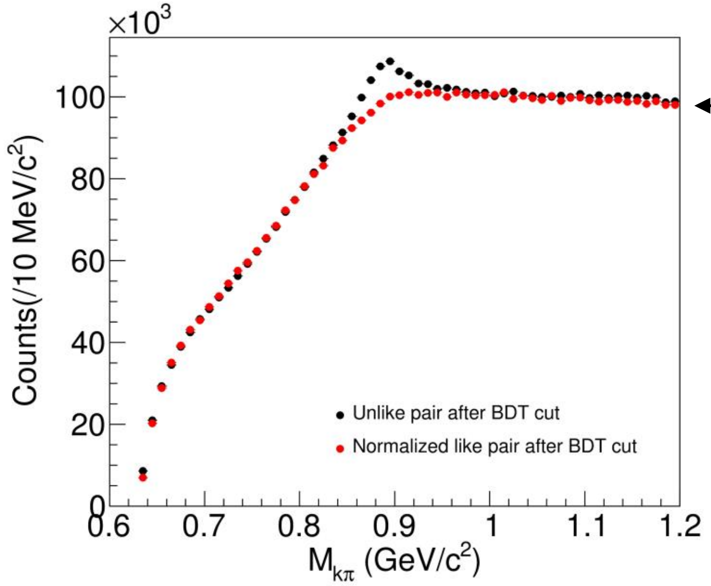


Invariant mass distribution: ALICE data

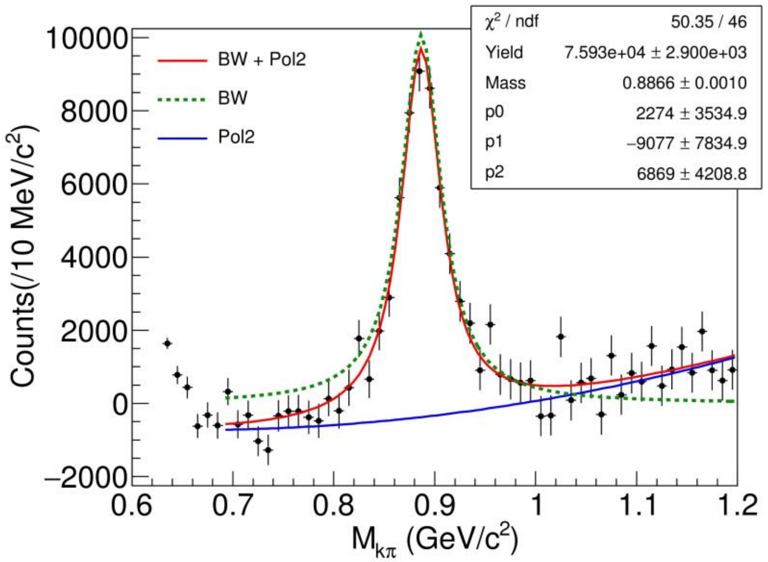
Traditional approach







Signal is more visible!



ML approach with BDT

Comparison of signal obtained from ALICE data

Method	χ^2/ndf	S	$S+B$	S/$\sqrt{S+B}$
Machine Learning	50.35/46	$\frac{75934.8}{0.93} = 81650.3$	3626890	38.54
Traditional Approach	47.11/43	82346.8	5227420	34.82

Extraction yield, significance and χ^2/ndf of the fit suggests a better extraction of signal by using Machine Learning via BDT for real data too.

Summary

In this analysis, we have extracted K^{*0} signal by two methods
1)**Traditional invariant mass technique** 2)**ML classification by BDT**

In both cases of Monte-Carlo and real data, we obtained a better significance by machine learning method as compared to the traditional method.

Outlook

- In future, we will extend this analysis for resonances which undergo 3-body decay.
- We will implement this analysis on Pb-Pb nuclear collisions data
- Machine Learning can also be applied to extract signal from rare resonances such as K*(1410), K*(1680) and Ξ(1820).

Backup

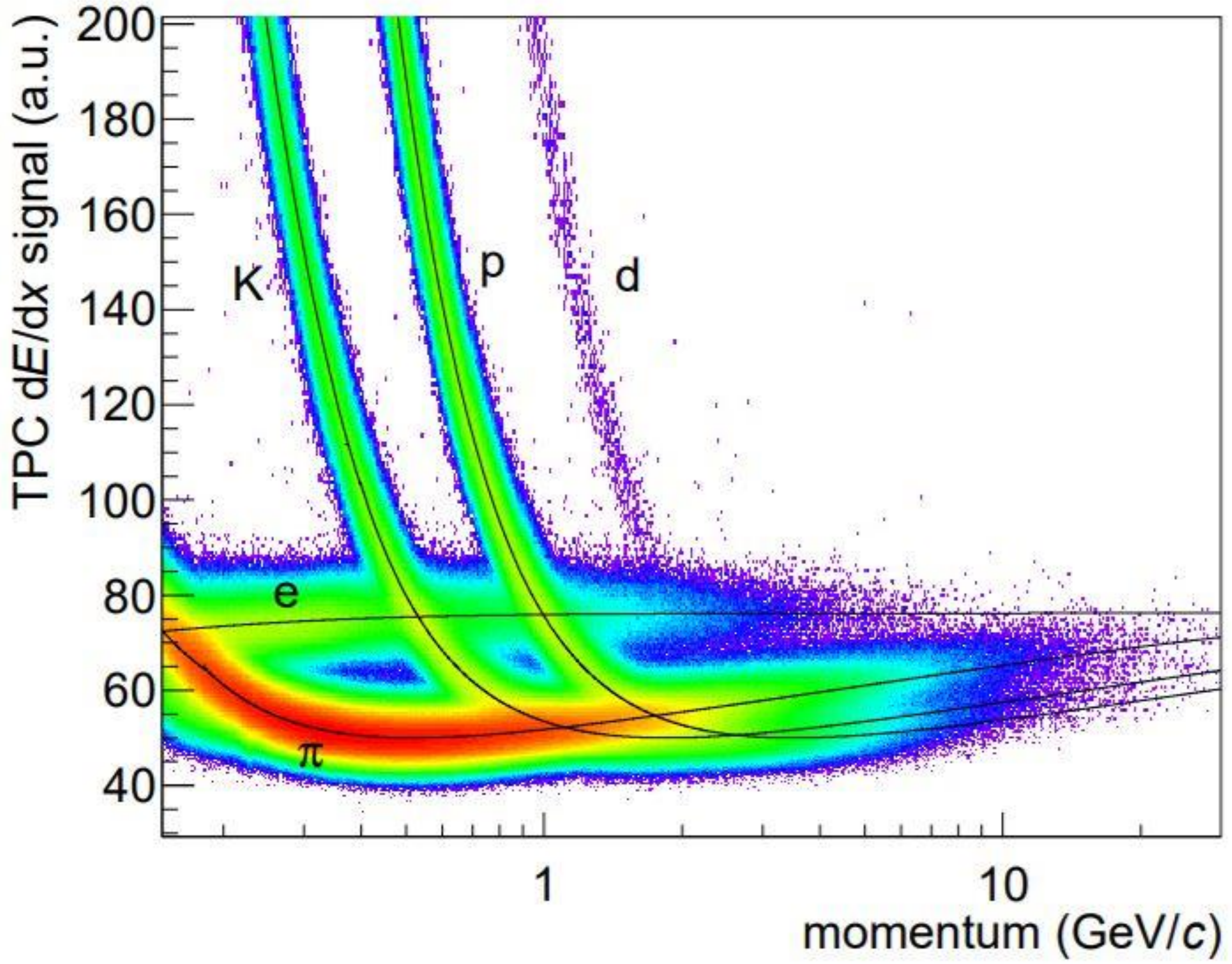
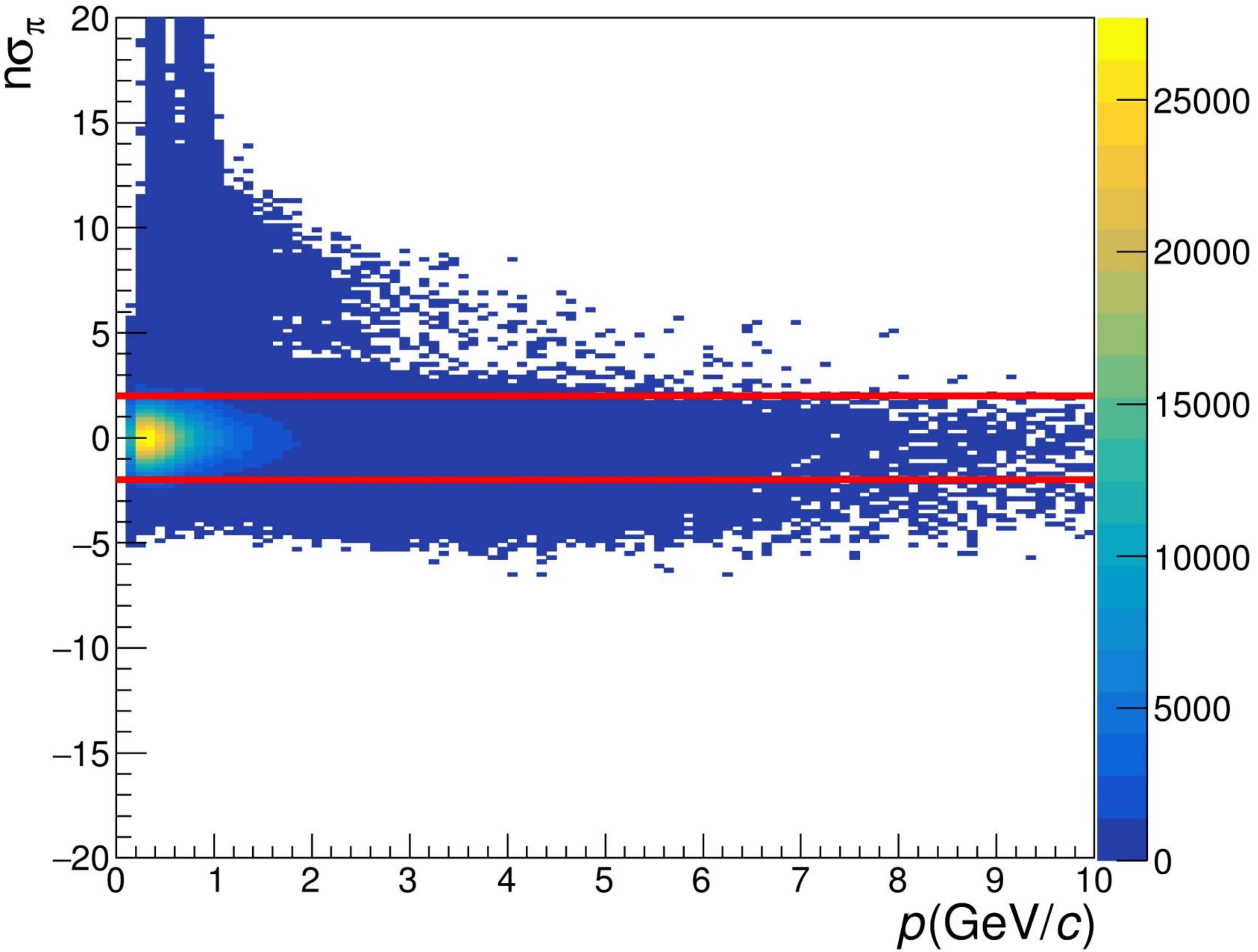


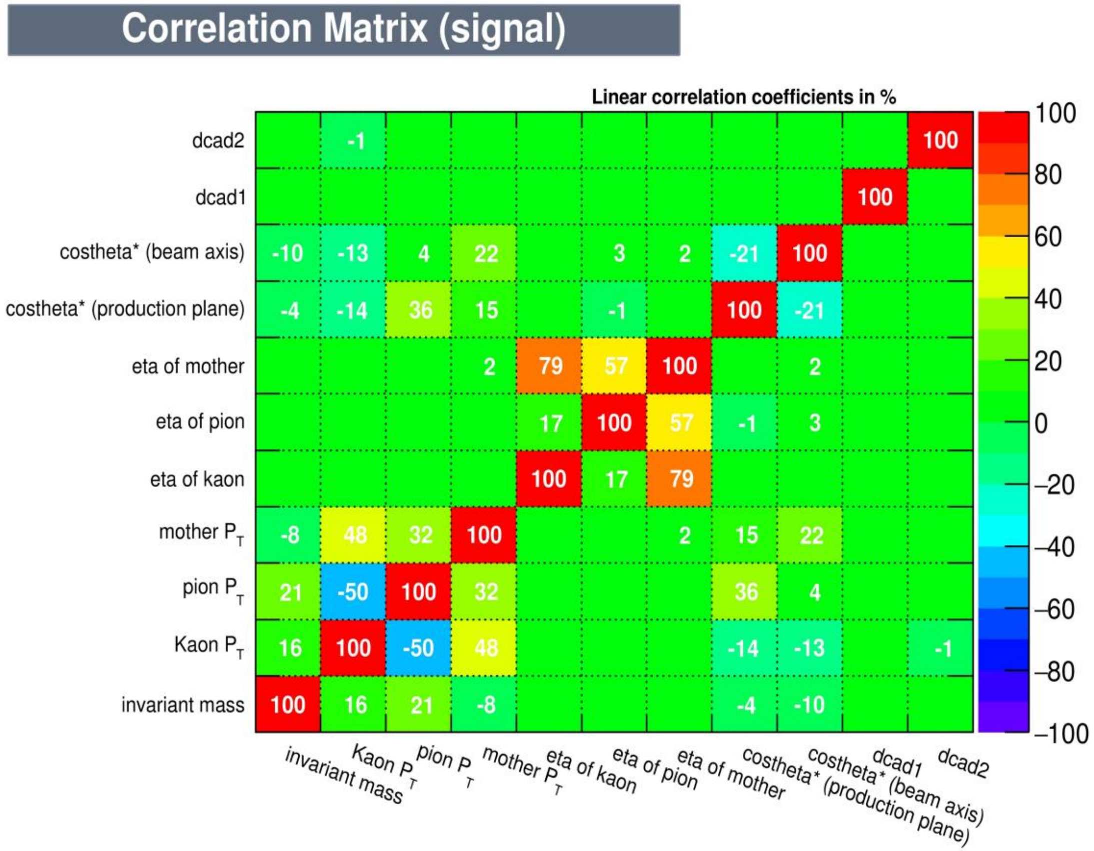
Fig. 1: (Colour online) Specific ionization energy loss dE/dx vs. momentum for tracks measured with the ALICE TPC. The solid lines are parametrizations of the Bethe-Bloch function [23].

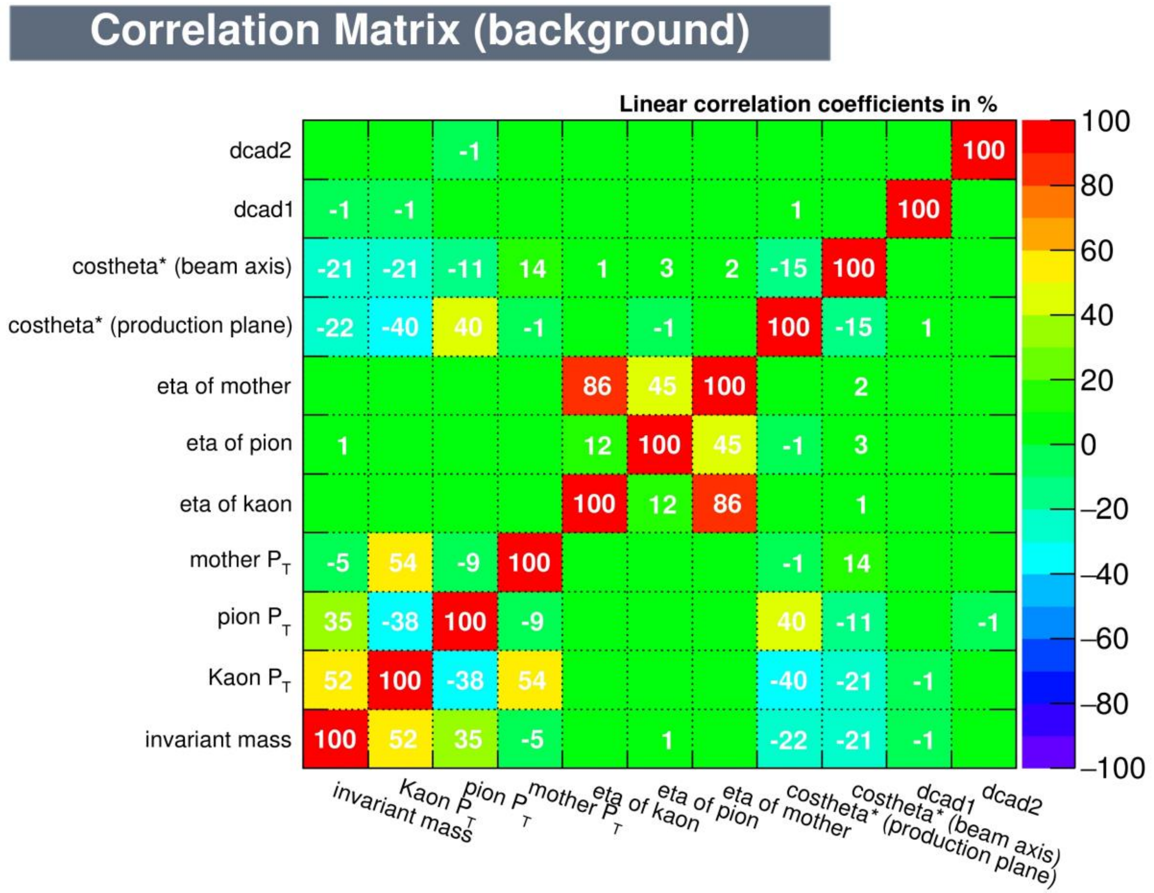

$n\sigma_\pi$
20
15
10
5
0
−5
−10
−15
−20
0
1
2
3
4
5
6
7
8
9
10
p(GeV/c)
25000
20000
15000
10000
5000
0

Standard Track Selection Cuts

1. $p_T > 0.15$ GeV/c
2. -0.8 $< \eta <$ 0.8
3. Reject kink daughters
4. Ratio of crossed rows over findable cluster > 0.8
5. Minimum(Maximum) number of rows crossed in TPC is 70(159).
6. TPC χ^2/clusters < 4.0
7. ITS χ^2/clusters < 36.0
8. $(DCA)_r\ (p_T) < 0.0182 + 0.0350 p_T^{-1.0}$ cm
9. $|DCA|_z < 2$ cm
10. $|y_{pair}| < 0.5$

Correlation coefficient percentage

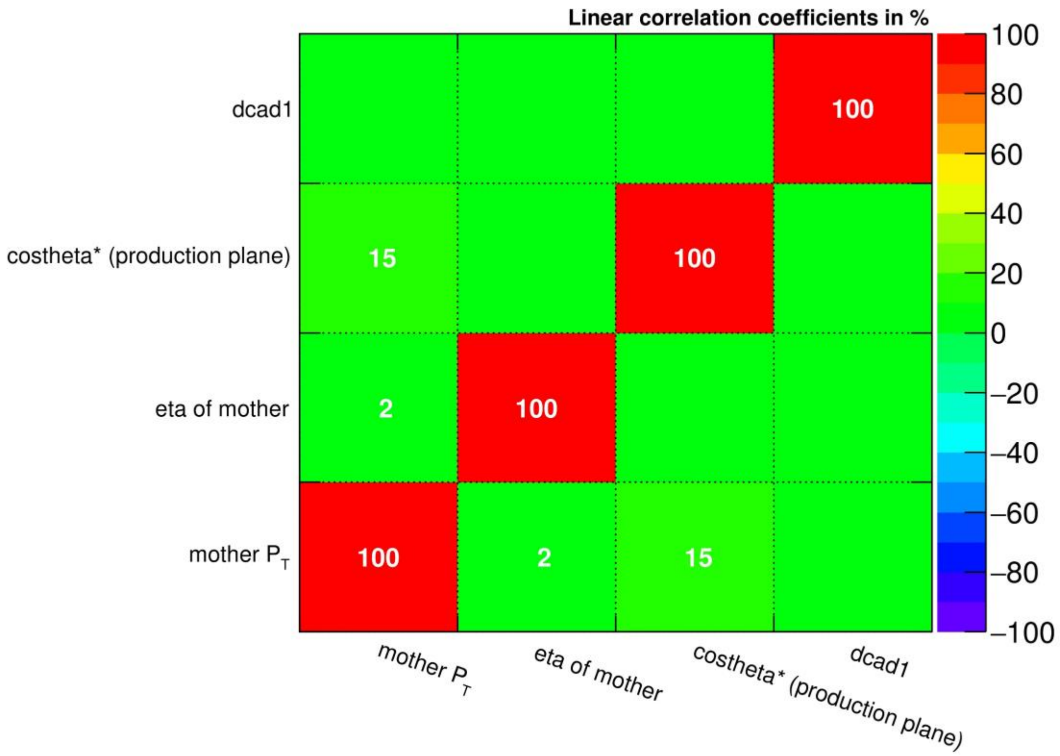




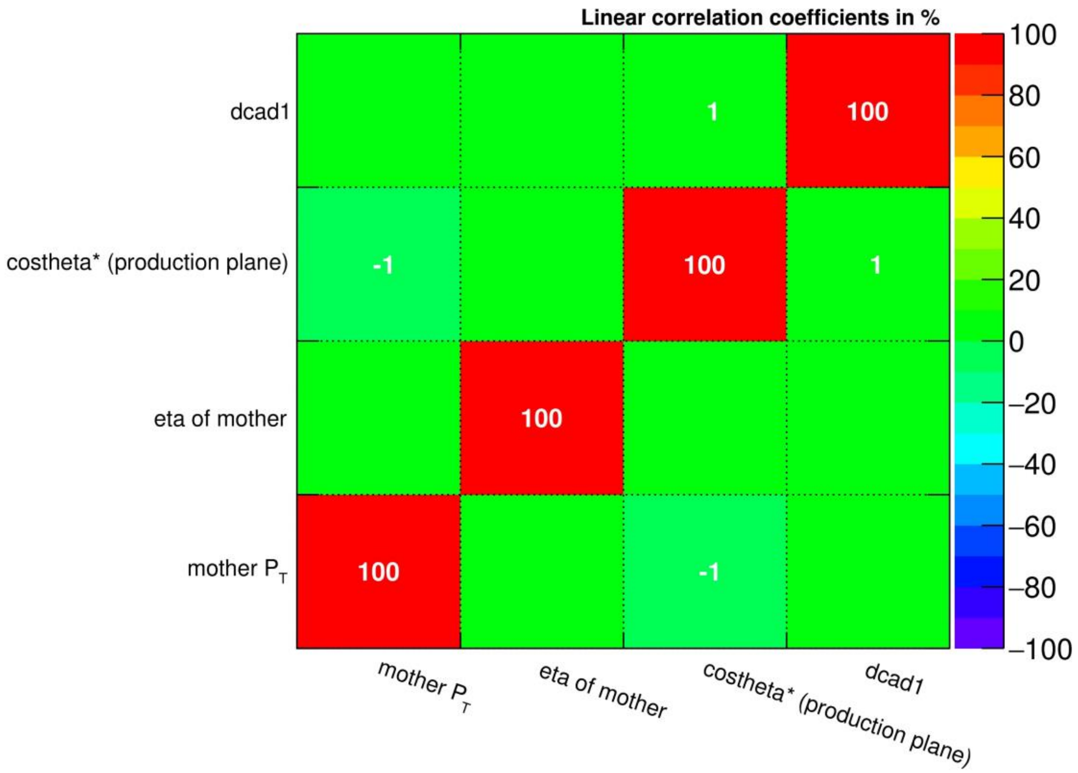
Input features used for training phase.

Selected features have less correlation with invariant mass

Correlation Matrix (signal)



Correlation Matrix (background)



DT algorithm

- An event is either classified as signal or background by either passing or not passing a condition (cut) on a specific node until a decision is made.
- In order to determine these conditions(cuts) , the decision tree is grown starting from the root node.

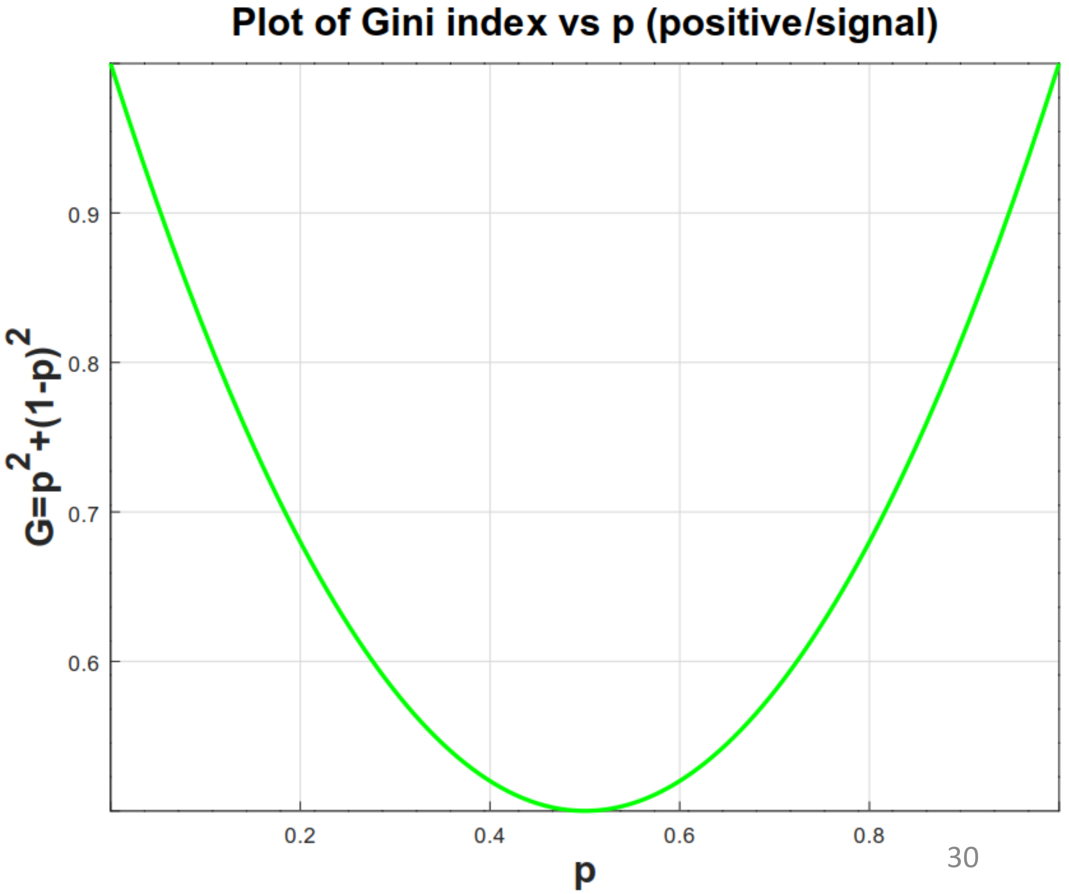
Where to stop ? Stop the splitting when we reach maximum tree depth or we have exhausted all our features!

Class of a leaf : If purity(S/S+B) > 0.5 then signal, otherwise background.

How do I select which feature I take first?

Gini Index : Define as $p^2 + q^2$ (at a given node)

- p: fraction of positive (signal) events
- q=1-p : fraction of negative (background) events



Weighted Gini Split:

$$\mathbf{I_G = f_{left} * G_{left} + f_{right} * G_{right}}$$

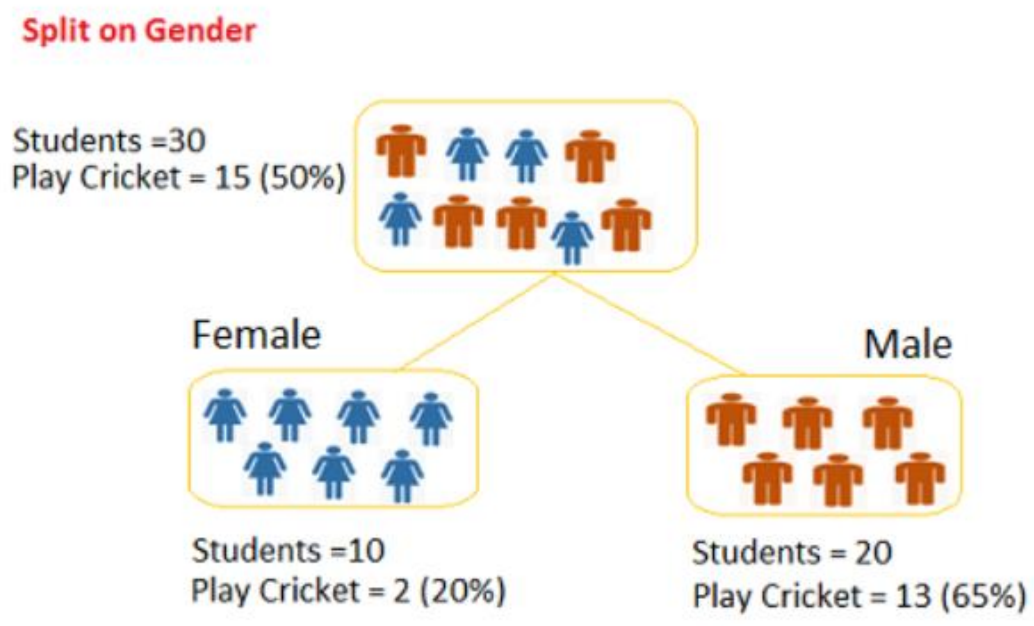
where,

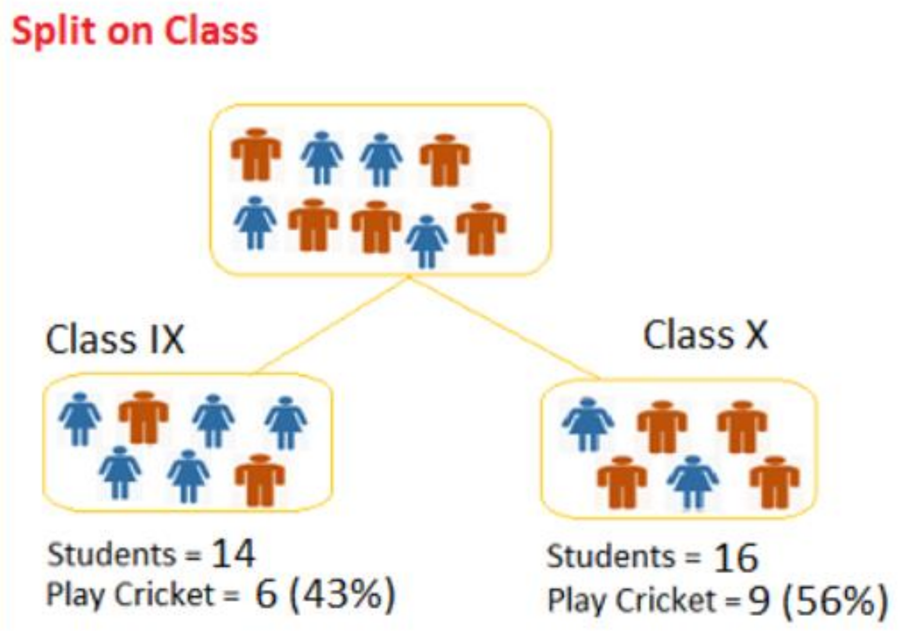
f_{left} = fraction of events which go in the left split

f_{right} = fraction of events which go in the right split

G_{left} = Gini index of left node

G_{right} = Gini index of right node





Split on Gender:

1. Gini for sub-node Female = $(0.2)*(0.2)+(0.8)*(0.8)=0.68$
2. Gini for sub-node Male = $(0.65)*(0.65)+(0.35)*(0.35)=0.55$
3. Weighted Gini for Split Gender = $(10/30)*0.68+(20/30)*0.55$ = **0.59**

Similar for Split on Class:

1. Gini for sub-node Class IX = $(0.43)*(0.43)+(0.57)*(0.57)=0.51$
2. Gini for sub-node Class X = $(0.56)*(0.56)+(0.44)*(0.44)=0.51$
3. Weighted Gini for Split Class = $(14/30)*0.51+(16/30)*0.51$ = **0.51**

Select this one first!

AdaBoost: Formula for computing coefficient $\hat{w}_t$ of classifier $f_t(\mathbf{x})$

$$\hat{\mathbf{w}}_t = \frac{1}{2}\ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

Is $f_t(\mathbf{x})$ good?

Yes

No

weighted_error(f_t) on training data	$\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}$	$\hat{w}_t$
0.01	$\frac{1-0.01}{0.01} = 99$	$\frac{1}{2}\ln(99) = 2.3$
0.5	$\frac{1-0.5}{0.5} = 1$	0
0.99	$\frac{1-0.99}{0.99} = 0.01$	$\frac{1}{2}\ln(0.01) = -2.3$

A terrible classifier!
But negative of that:
-f is awesome! ☺

AdaBoost: Formula for updating weights α_i

$$\alpha_i \leftarrow \begin{cases} \alpha_i \, e^{-\hat{W}_t}, & \text{if } f_t(\mathbf{x}_i)=y_i \\ \alpha_i \, e^{\hat{W}_t}, & \text{if } f_t(\mathbf{x}_i)\neq y_i \end{cases}$$

Did f_t get $\mathbf{x}_i$ right?

Yes

No

$f_t(\mathbf{x}_i)=y_i$?	$\hat{W}_t$	Multiply α_i by	Implication
Correct	2.3	$e^{2.3} = 9.98$	Increases importance at x_i
Correct	0	$e^{0} = 1$	Keeps importance same
Mistake	2.3	$e^{-2.3} = 0.1$	Decreases importance at x_i
Mistake	0	$e^{-0} = 1$	Keeps importance same

AdaBoost: learning ensemble

- Start same weight for all points: $\alpha_i = 1/N$
- For t = 1,…,T
  - Learn $f_t(\mathbf{x})$ with data weights α_i
  - $\boxed{\text{Compute coefficient } \hat{w}_t}$
  - $\boxed{\text{Recompute weights } \alpha_i}$
  - $\boxed{\text{Normalize weights } \alpha_i}$
- Final model predicts by:

$$\hat{y} = sign\left(\sum_{t=1}^{T} \hat{\mathbf{w}}_t f_t(\mathbf{x})\right)$$

$$\hat{\mathbf{w}}_t = \frac{1}{2} \ln\left(\frac{1 - weighted\_error(f_t)}{weighted\_error(f_t)}\right)$$

$$\alpha_i \leftarrow \begin{cases} \alpha_i e^{-\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) = y_i \\ \alpha_i e^{\hat{w}_t}, & \text{if } f_t(\mathbf{x}_i) \neq y_i \end{cases}$$

$$\alpha_i \leftarrow \frac{\alpha_i}{\sum_{j=1}^{N} \alpha_j}$$

Threshold split selection algorithm

Income

- Step 1: Sort the values of a feature $h_j(\mathbf{x})$:

  Let $\{v_1, v_2, v_3, \dots v_N\}$ denote sorted values

- Step 2:
  - For i = 1 … N-1
    - Consider split $t_i = (v_i + v_{i+1}) / 2$
    - Compute classification error for treshold split $h_j(\mathbf{x}) >= t_i$
  - Chose the t^* with the lowest classification error