Queen Mary
University of London

ADRIAN BEVAN

ATLAS UK MEETING 2020: MACHINE LEARNING SESSION

DECISION TREES: TUTORIAL

(SEE SEPARATE SLIDES ON DECISION TREES FOR DETAILS OF THE ALGORITHM)

This tutorial has been written for ROOT version 6.18.04

WORKPLAN

- 1. Train a boosted decision tree (BDT) out of the box (no hyperparameter (HP) optimisation). [very quick]
- 2. Optimise HPs:
  - Adaboost beta, max depth, node size, number of trees. [takes a while, but you should always optimise the hyperparameters once your code has been setup]
- 3. Repeat de-correlating the input variables.
- 4. Compare results.
- 5. The H→$\tau\tau$ Kaggle Data Challenge Problem. [in your own time]

1. TRAIN A BDT OUT OF THE BOX

- Download the TMVAClassifier.C example provided on the agenda web page (and copy to lxplus). This is a modified version of the one that comes with ROOT and is specifically for this tutorial.
- We will start using the default example in TMVA.
- There are two types of example:
  - signal and background.
- There are 4 variables in the input feature space
- We are going to train a BDT to reduce the dimensionality to a single output variable to separate signal from background.

1. TRAIN A BDT OUT OF THE BOX

- The dataloader will prepare test and training trees.
- Each tree will have 1000 examples to train (randomly selected).
- The training is weighted by the number of events.
- Note that this may not be what you want if you have examples that have generator level event weights as unequal the training samples (or sum of weights) will place greater importance on one example type in the loss function being used for training.

1. TRAIN A BDT OUT OF THE BOX

- There are two types of example: signal and background.
- The dataloader will prepare test and training trees.
- Each tree will have 1000 examples to train (randomly selected).
  - The remainder (5000 examples) will be used for testing.
- The training is weighted by the number of events.
  - Note that this may not be what you want if you have examples that have generator level event weights as unequal the training samples (or sum of weights) will place greater importance on one example type in the loss function being used for training.

1. TRAIN A BDT OUT OF THE BOX

Execute

- On lxplus setup ROOT:

```
lsetup "root 6.18.04-x86_64-centos7-gcc8-opt"
```

- In the directory where you have coped the example macro to, execute the following command:

```
root -l -b -q TMVAClassification.C
```

- If you want to retain the training log file then pipe the output into a log:

```
root -l -b -q TMVAClassification.C > trainOOTB.log
```

- TMVA will do the rest… and output a directory of weights (`dataset`) and a directory containing an output files (`files`). The output file will be interested in is made in the directory you run ROOT in, `TMVA.root`.

1. TRAIN A BDT OUT OF THE BOX

Execute

- On lxplus setup ROOT:
  - TMVA has the option to do data pre-processing before examples are presented to the algorithm. We have not used it for this example! (we will come back to this later).
- The output (logfile) contains a lot of information, including a ranking of variables

```
Variable       Mean        RMS    [        Min        Max ]
-----------------------------------------------------------
  myvar1:    0.20301     1.7142   [    -9.8605     7.9024 ]
  myvar2:  -0.048747     1.1049   [    -4.0854     4.0291 ]
    var3:    0.15975     1.0530   [    -5.3563     4.6430 ]
    var4:    0.42792     1.2213   [    -6.9675     5.0307 ]
-----------------------------------------------------------
```

Better

Worse

1. TRAIN A BDT OUT OF THE BOX

Execute

- At the end of the output you get a breakdown of the data set and the model performance:

```
Variable        Mean        RMS    [        Min        Max ]
-----------------------------------------------------------
  myvar1:     0.20301     1.7142   [    -9.8605     7.9024 ]
  myvar2:   -0.048747     1.1049   [    -4.0854     4.0291 ]
    var3:     0.15975     1.0530   [    -5.3563     4.6430 ]
    var4:     0.42792     1.2213   [    -6.9675     5.0307 ]
-----------------------------------------------------------
```

ROC integral is a commonly used figure of merit to compare MVA performance.

Better to use the end result (e.g. expected limit) if you can.

```
Evaluation results ranked by best signal efficiency and purity (area)
------------------------------------------------------------------------------

DataSet       MVA
Name:         Method:          ROC-integ
dataset       BDT            : 0.877
------------------------------------------------------------------------------


Testing efficiency compared to training efficiency (overtraining check)
------------------------------------------------------------------------------

DataSet              MVA              Signal efficiency: from test sample (from training sample)
Name:                Method:          @B=0.01             @B=0.10          @B=0.30
------------------------------------------------------------------------------

dataset              BDT            : 0.256 (0.393)       0.633 (0.743)    0.863 (0.915)
------------------------------------------------------------------------------
```

Differences in test/training sample performance at different working points indicates the variance of the trained model.

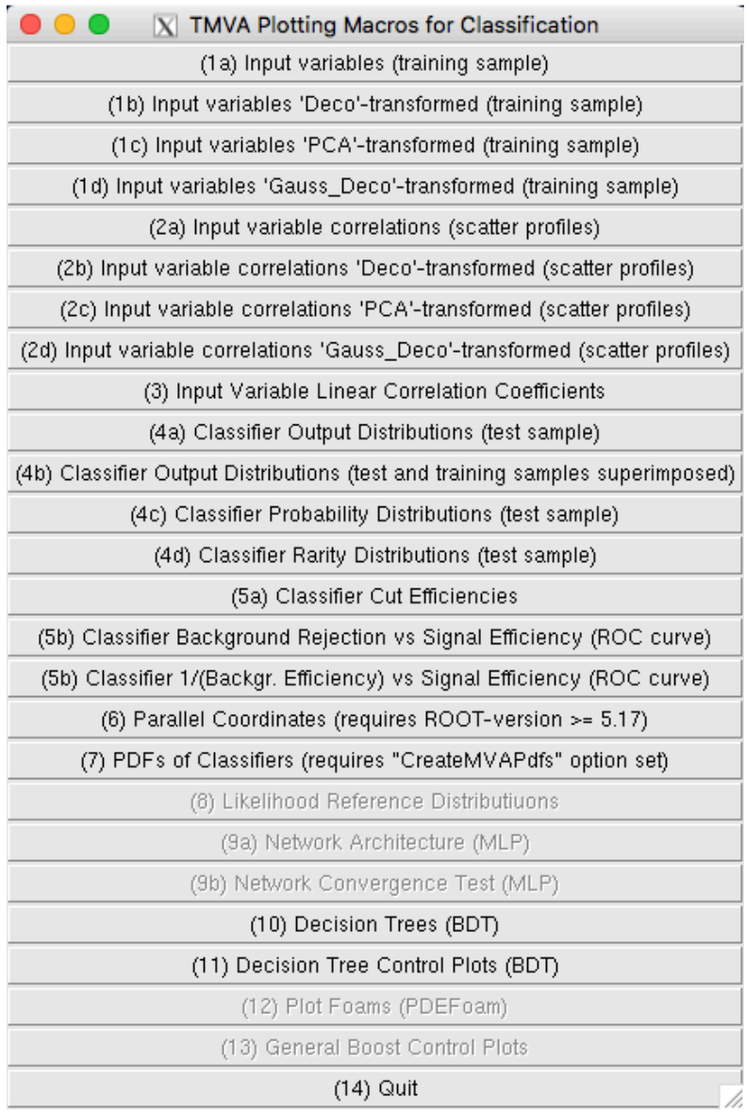
Models with large variance are overtrained.

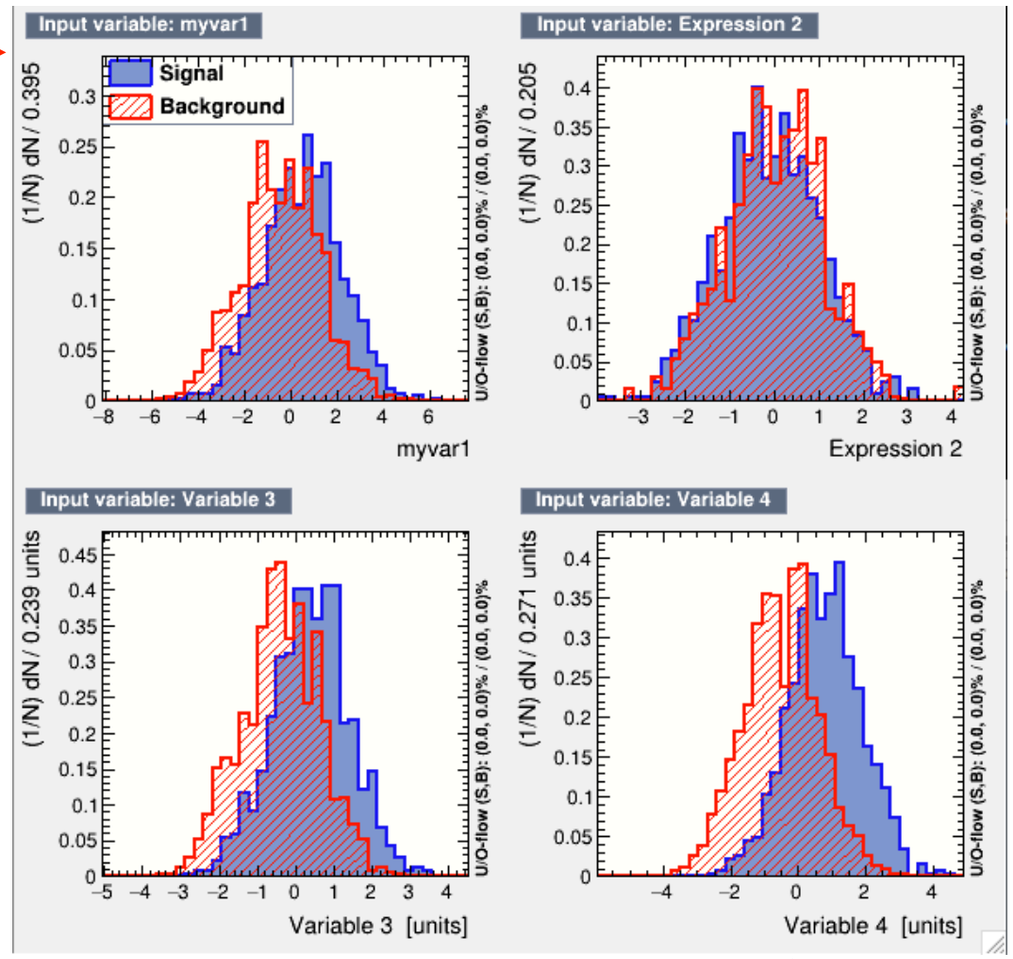
1. TRAIN A BDT OUT OF THE BOX

Review Results

- The output data file contains a tree of the data, and a number of histograms related to the test/train performance of the model. Run the following to launch the TMVA inspection GUI

```
TMVA::TMVAGui("TMVA.root")
```

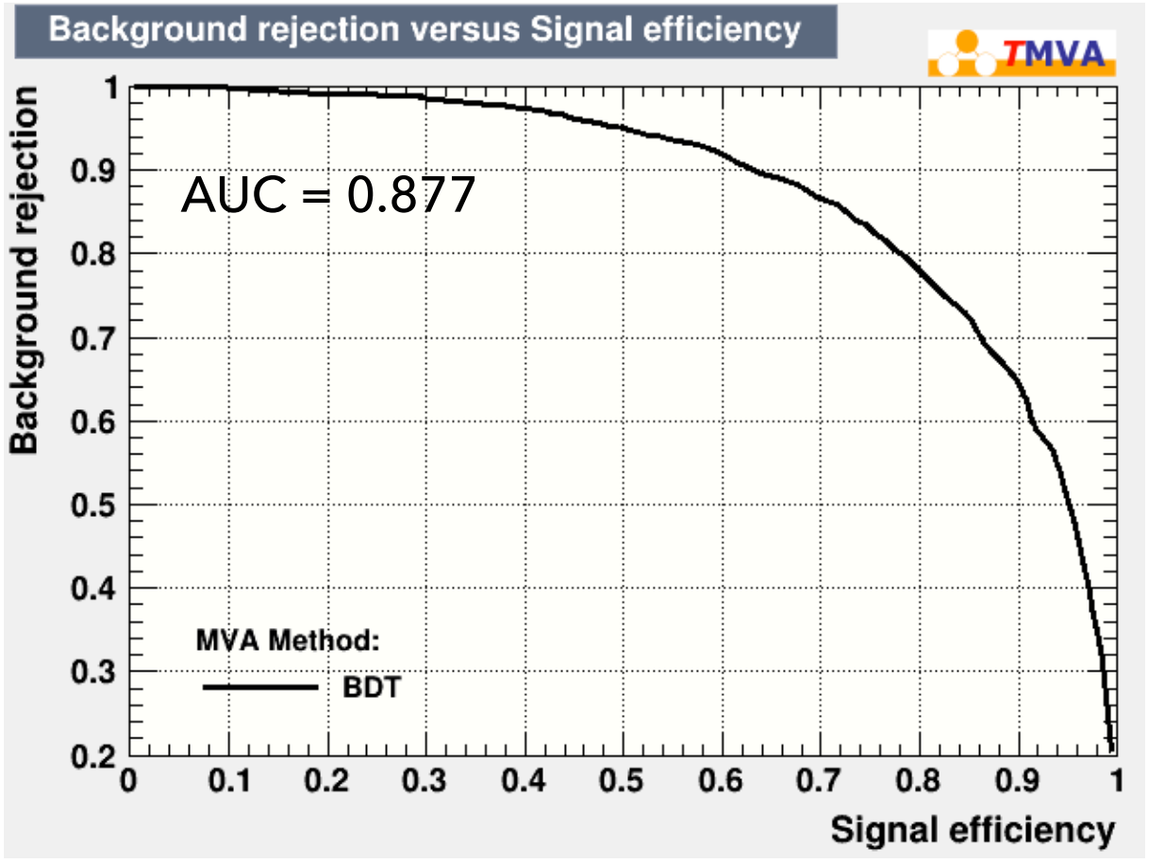


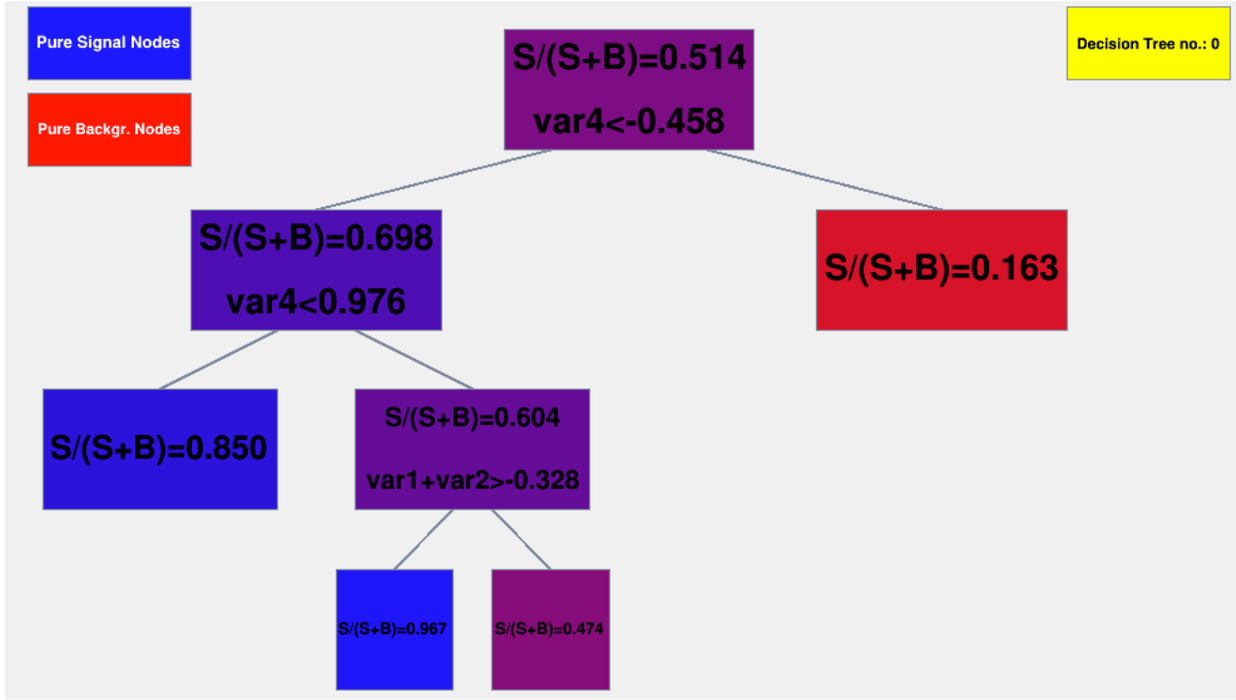


1. TRAIN A BDT OUT OF THE BOX

Review Results

- The ROC integral or ROC curve is often used as a figure of merit proxy for selecting a given model. Here is the one from our first training, alongside a diagram of the decision tree model.

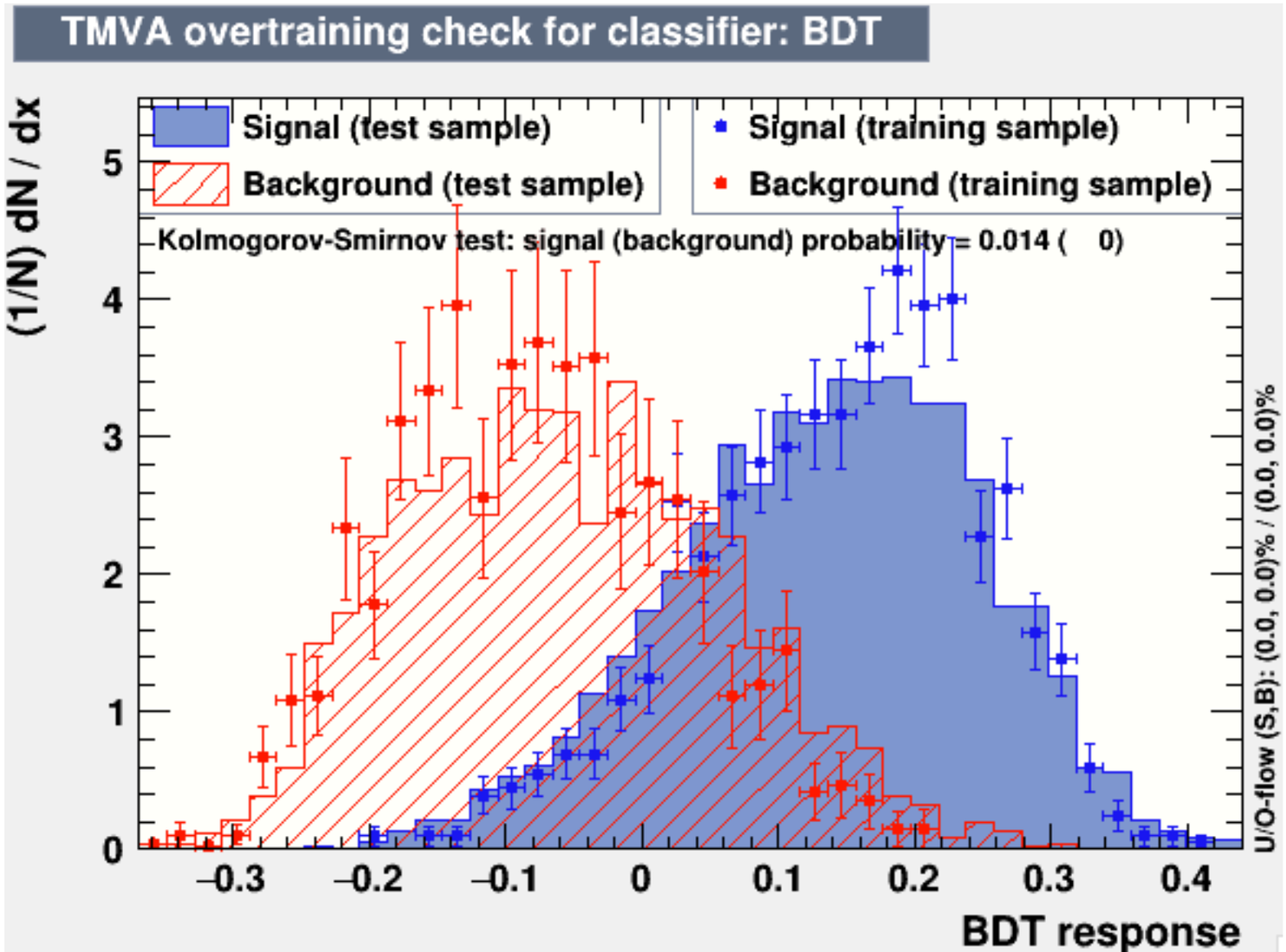




1. TRAIN A BDT OUT OF THE BOX

Review Results

- Overtraining - how do you check this?
- No agreed standardised approach in the field. However TMVA uses a comparison of outputs for test-train samples. The similarity of the plots is taken as an indication of agreement using [WARNING] a binned KS test.



WARNING

The binned KS test used here is known to be biased, and so a small KS probability does not necessarily mean and overtrained model.

Zero is definitely overtrained, and common sense is generally used to gauge agreement (chi^2 by eye).

- Signal is overtrained.
- Background is probably OK.

1. TRAIN A BDT OUT OF THE BOX

Tidy-up

- Before proceeding you will want to rename the output ROOT file and the weights directory (if you want to process data using this model later), so that these are not overwritten by the subsequent examples:

```
mv TMVA.root TMVA-unoptimised.root
mv dataset/weights dataset/weights-unoptimised
```

2. OPTIMISE THE HYPERPARAMETERS

Execute

- Edit the macro TMVAClassification.C to uncomment line 229:

```
factory->OptimizeAllMethods();
```

- Re-run the macro and compare output results.

```
root -l -b -q TMVAClassification.C > trainOpt.log
```

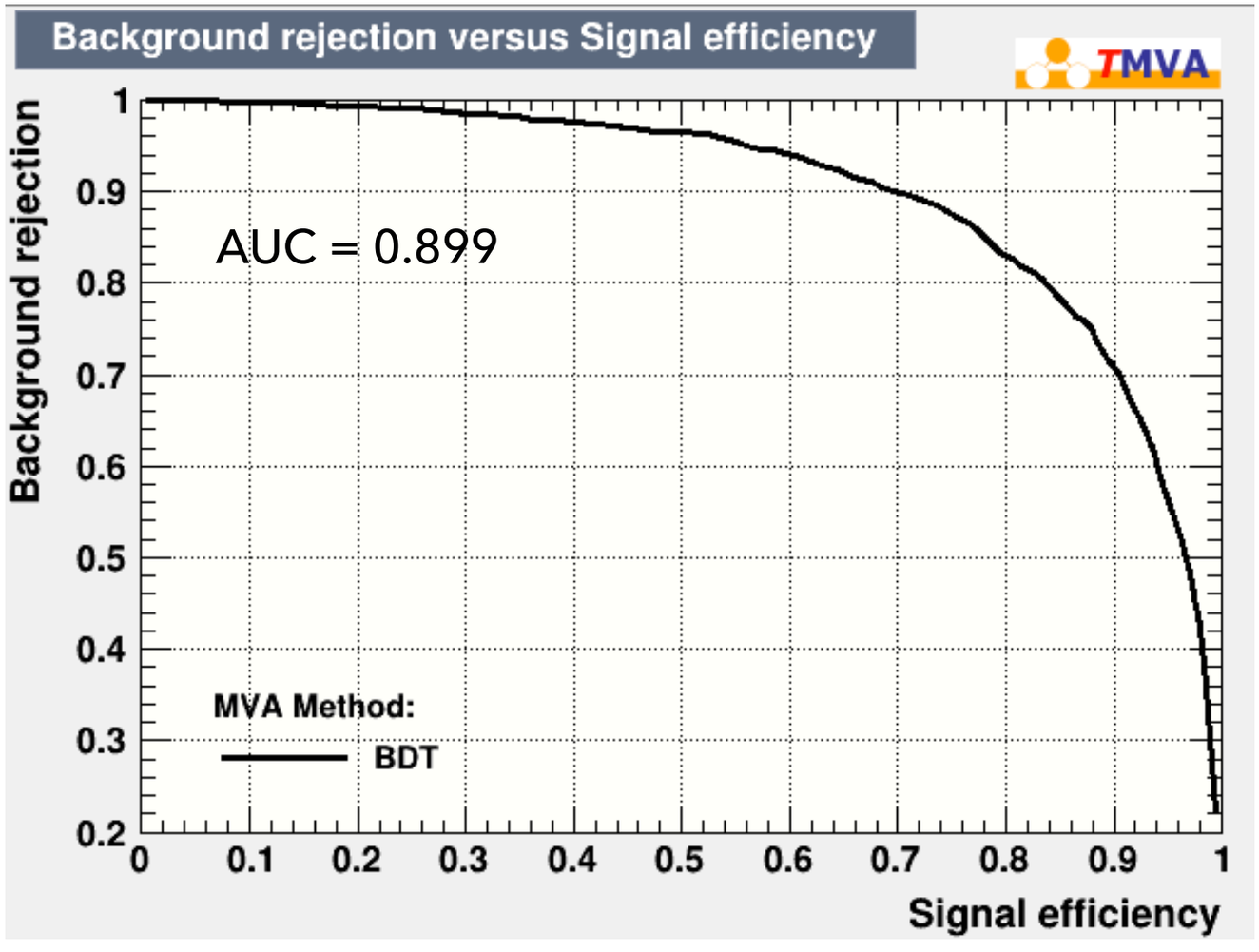
This should take about 12 mins on lxplus.

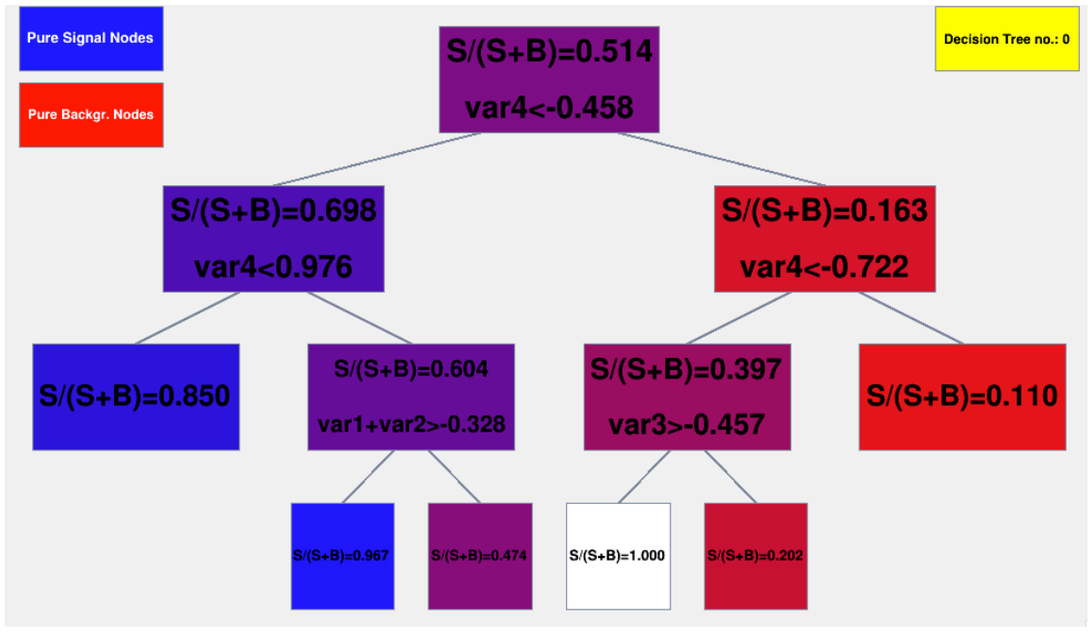
2. OPTIMISE THE HYPERPARAMETERS

Review Results

- Again you can inspect the output of the training using the TMVA GUI

```
TMVA::TMVAGui("TMVA.root")
```





2. OPTIMISE THE HYPER-PARAMETERS

Tidy-up

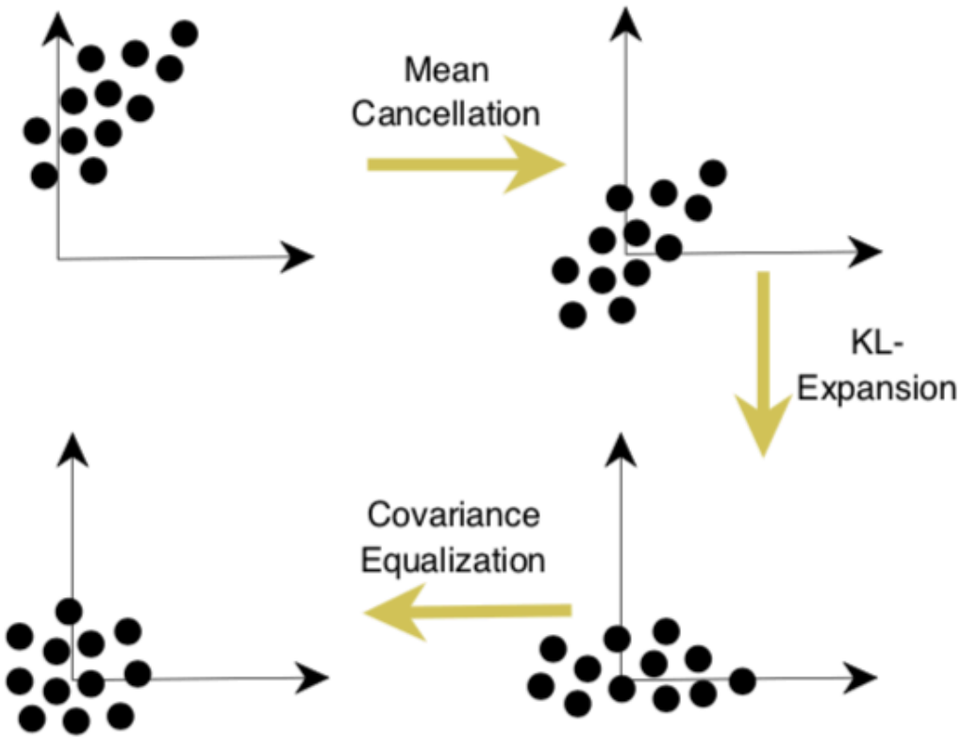
- Before proceeding you will want to rename the output ROOT file and the weights directory (if you want to process data using this model later), so that these are not overwritten by the subsequent examples:

```
mv TMVA.root TMVA-optimised.root
mv dataset/weights dataset/weights-optimised
```

3. DECORRELATING INPUT VARIABLES

Background

- We are using rectangular cuts to separate signal from background; some of the variables are correlated and this can make it harder to separate the components of the data.
- We can do some data wrangling to pre-process our input feature space before it gets passed to the model training.



These steps are generally good for gradient descent algorithm performance optimisation (e.g. Neural Networks). Here we care about the rotation for de-correlating linearly correlated features.

3. DECORRELATING INPUT VARIABLES

Execute

- Modify the booking method function call to include a VarTransform option. Add `:VarTransform=D` to the end of line 215.

- Transform options are:
  - **N**orm: Normalise the input features to [-1, 1]
  - **D**eco: Linearly de-correlate input features
  - **P**CA: Use PCA (similar to Deco)
  - **U**niform: Map features into a uniform distribution
  - **G**aus: Map features into a Gaussian distribution

- Re-run the macro with(out) optimisation:

```
root -l -b -q TMVAClassification.C > trainDeco.log
```

3. DECORRELATING INPUT VARIABLES

Review Results

- Again you can inspect the output of the training using the TMVA GUI

```
TMVA::TMVAGui("TMVA.root")
```

- You should see similar outputs as for the first two trainings when doing this.

3. DECORRELATING INPUT VARIABLES

Tidy-up

- Before proceeding you will want to rename the output ROOT file and the weights directory (if you want to process data using this model later), so that these are not overwritten by the subsequent examples:

```
mv TMVA.root TMVA-deco.root
mv dataset/weights dataset/weights-deco
```

- Use the file stem `decoOpt` for the optimised and decorrelated feature space training.

3. DECORRELATING INPUT VARIABLES & OPTIMISING

Execute

- As before uncomment the `OptimizeAllMethods` function call and re-run the macro.

```
mv TMVA.root TMVA-decoOpt.root
mv dataset/weights dataset/weights-decoOpt
```

- You can review the results as before.

4. COMPARE RESULTS

Review Results

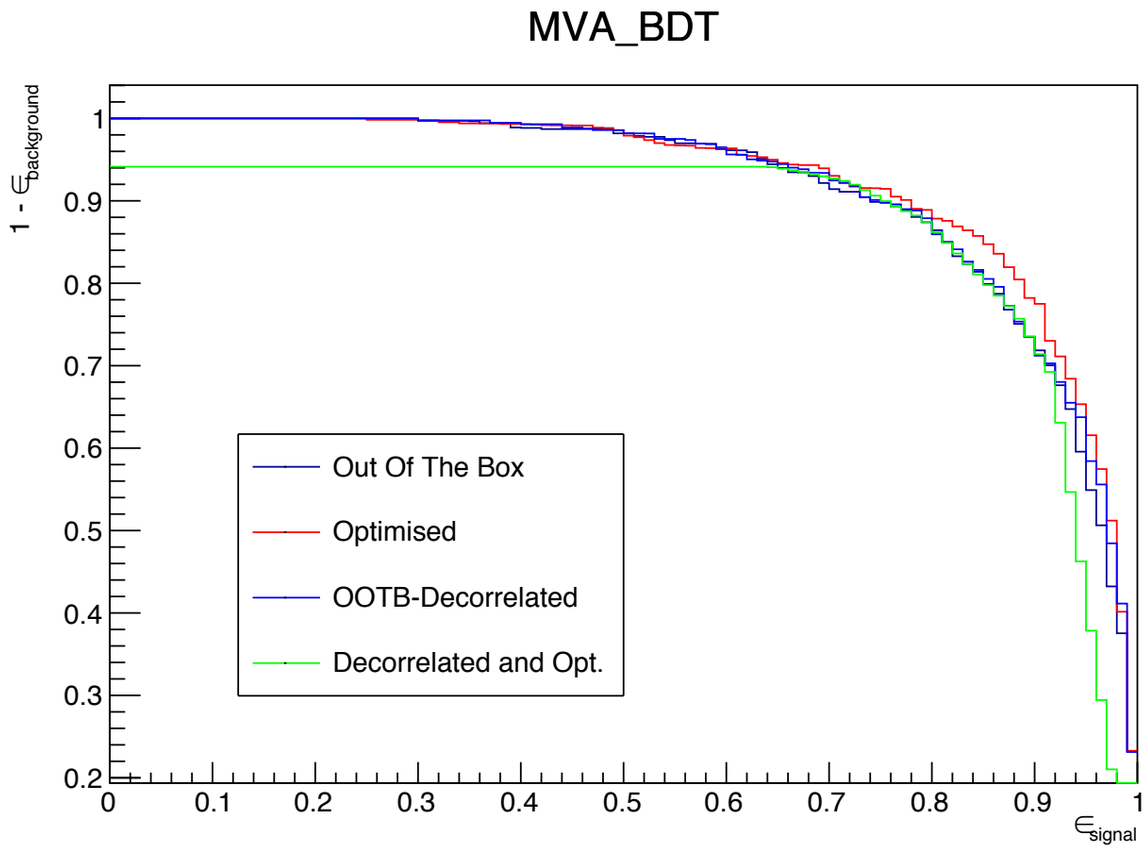
- We have looked at 4 different trainings of a BDT model using the AdaBoost.M1 method in TMVA:
  - Out of the box (OOTB);
  - Optimising the HPs;
  - decorrelating input feature space variables (and optimising);
- Here we compare the trainings

	1. OOTB	2. Optimise	3. Decorrelate	4. Decor+Opt
AUC	0.877	0.899	0.881	0.883
Max Tree depth	3	3	3	2
N trees	850	257.5	850	10
Ada boost β	0.5	0.2	0.5	0.2
Min Node Size	2.5%	1%	2.5%	29%
Opt. Time	–	~12min	–	~6min

4. COMPARE RESULTS

Review Results

- While the AUC may enough useful information to determine the “best” model, comparison of the ROC curves is also useful.



Run the macro Compare.C to make this plot.

The best training is using optimisation.

The result using decorrelation and optimisation does not look as good as the other trainings.

- Remember that ultimately one wants to run the full analysis chain and select the best model (stat+syst) based on the relevant figure of merit (e.g. best limit or best measurement of some parameter). However this is not often done as it can be logistically challenging.

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM

- This is an additional problem for you to work through in your own time.
- The macro Htautau.C is setup to train a BDT to separate signal from background as a first part of developing a model.
- If you look at the Kaggle Challenge website you will see that this is only the first part of the challenge. The Approximate Median Significance, Z, is the FOM to be optimised.
- The Higgs Kaggle Web page can be found at:
  - https://www.kaggle.com/c/higgs-boson
- Documentation can also be found on the preprint archive:
  - https://higgsml.lal.in2p3.fr

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM

- The decay probability for a Higgs particle to pairs of fermions is

Decay channel	Probability (%)
H→bb	57.7
H→WW	21.5
H→ττ	6.3
H→ZZ	2.6
H→γγ	0.2

- In addition to signal, there are significant background channels that mean that the the best measured channels are for γγ and ZZ final states.

- The $H \rightarrow \tau^+\tau^-$ channel is an important decay to measure, and this requires separation of signal from background.

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM

- Develop a model that can be used to distinguish between signal and background for the $H \rightarrow \tau^{+}\tau^{-}$ sample.
- There are several simplifications for this task relative to a normal HEP analysis:
  - Events with negative weights have been removed (comes from Monte Carlo generators of some simulators).
  - Only the dominant background sources are included.
  - Some correction factors have been neglected.

Additional Work

5. THE H→ττ KAGGLE DATA CHALLENGE PROBLEM

- The approximate median significance is a metric used to compare results. This is given by

$$\mathrm{AMS} = \sqrt{2\left((s+b+b_{\mathrm{reg}})\ln\left(1+\frac{s}{b+b_{\mathrm{reg}}}\right)-s\right)}$$

- The term b_{reg} is used to stop the search reverting to small regions of the feature space where statistical fluctuations can become significant. This is set to 10 for the challenge.
- s and b:
  - are defined in the challenge notes as the sum over the example weights for the signal and background events used in the search region.
  - They are unbiased estimators of the number of signal and background events, respectively.

(see sec. 2 of the challenge notes)

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: FEATURES

- Some formulae (physics) are included in Appendix A of "Learning to discover: the Higgs boson machine learning challenge" for the interested student (context).
- Features are listed in Appendix B.
- The following features are **NOT** to be used in the classifier:
  - EventId: A unique integer identifier of the example[1].
  - Weight: Event weight.
  - Label[2]: The event label (string) y_i {s, b} (s for signal, b for background).
  - KaggleSet: Specific to the opendata.cern.ch dataset: string specifying to which Kaggle set the event belongs: "t":training, "b":public leaderboard, "v":private leaderboard, "u":unused.
  - KaggleWeight: Specific to the opendata.cern.ch dataset: weight normalized within each Kaggle data set according to:

(see Appendix B of the challenge notes) $$w'_j = w_j \frac{\sum_i w_i \mathbb{1}\{y_i = y_j\}}{\sum_{i \in S'} w_i \mathbb{1}\{y_i = y_j\}}$$

[1] In HEP training examples are normally referred to as events, as is the case in the documentation associated with this challenge.
[2]Not available in the test sample.

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: FEATURES

- Features listed from here on are usable in the classifier.
- Features with names prefixed with PRI and DER are:
  - PRI: Primary features “raw“ quantities measured on objects like jets.
  - DER: Derived features - combinations of the primary features derived from lower level (raw) information.

(see Appendix B of the challenge notes)

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: FEATURES

DER_mass_MMC The estimated mass m_H of the Higgs boson candidate, obtained through a probabilistic phase space integration (may be undefined if the topology of the event is too far from the expected topology)

DER_mass_transverse_met_lep The transverse mass (22) between the missing transverse energy and the lepton.

DER_mass_vis The invariant mass (21) of the hadronic tau and the lepton.

DER_pt_h The modulus (20) of the vector sum of the transverse momentum of the hadronic tau, the lepton, and the missing transverse energy vector.

DER_deltaeta_jet_jet The absolute value of the pseudorapidity separation (23) between the two jets (undefined if PRI_jet_num ≤ 1).

DER_mass_jet_jet The invariant mass (21) of the two jets (undefined if PRI_jet_num ≤ 1).

DER_prodeta_jet_jet The product of the pseudorapidities of the two jets (undefined if PRI_jet_num ≤ 1).

DER_deltar_tau_lep The R separation (24) between the hadronic tau and the lepton.

DER_pt_tot The modulus (20) of the vector sum of the missing transverse momenta and the transverse momenta of the hadronic tau, the lepton, the leading jet (if PRI_jet_num ≥ 1) and the subleading jet (if PRI_jet_num $= 2$) (but not of any additional jets).

DER_sum_pt The sum of the moduli (20) of the transverse momenta of the hadronic tau, the lepton, the leading jet (if PRI_jet_num ≥ 1) and the subleading jet (if PRI_jet_num $= 2$) and the other jets (if PRI_jet_num $= 3$).

DER_pt_ratio_lep_tau The ratio of the transverse momenta of the lepton and the hadronic tau.

(see Appendix B of the challenge notes)

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: FEATURES

Additional Work

DER_met_phi_centrality The centrality of the azimuthal angle of the missing transverse energy vector w.r.t. the hadronic tau and the lepton

$$C = \frac{A + B}{\sqrt{A^2 + B^2}},$$

where $A = \sin(\phi_{\text{met}} - \phi_{\text{lep}}) * \text{sign}(\sin(\phi_{\text{had}} - \phi_{\text{lep}}))$, $B = \sin(\phi_{\text{had}} - \phi_{\text{met}}) * \text{sign}(\sin(\phi_{\text{had}} - \phi_{\text{lep}}))$, and ϕ_{met}, ϕ_{lep}, and ϕ_{had} are the azimuthal angles of the missing transverse energy vector, the lepton, and the hadronic tau, respectively. The centrality is $\sqrt{2}$ if the missing transverse energy vector $\vec{E}_{\text{T}}^{\text{miss}}$ is on the bisector of the transverse momenta of the lepton and the hadronic tau. It decreases to 1 if $\vec{E}_{\text{T}}^{\text{miss}}$ is collinear with one of these vectors and it decreases further to $-\sqrt{2}$ when $\vec{E}_{\text{T}}^{\text{miss}}$ is exactly opposite to the bisector.

DER_lep_eta_centrality The centrality of the pseudorapidity of the lepton w.r.t. the two jets (undefined if `PRI_jet_num` ≤ 1)

$$\exp\left[\frac{-4}{(\eta_1 - \eta_2)^2}\left(\eta_{\text{lep}} - \frac{\eta_1 + \eta_2}{2}\right)^2\right],$$

where η_{lep} is the pseudorapidity of the lepton and η_1 and η_2 are the pseudorapidities of the two jets. The centrality is 1 when the lepton is on the bisector of the two jets, decreases to $1/e$ when it is collinear to one of the jets, and decreases further to zero at infinity.

PRI_tau_pt The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the hadronic tau.

PRI_tau_eta The pseudorapidity η of the hadronic tau.

PRI_tau_phi The azimuth angle ϕ of the hadronic tau.

PRI_lep_pt The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the lepton (electron or muon).

(see Appendix B of the challenge notes)

Additional Work

5. THE H→ττ KAGGLE DATA CHALLENGE PROBLEM: FEATURES

PRI_lep_eta The pseudorapidity η of the lepton.

PRI_lep_phi The azimuth angle ϕ of the lepton.

PRI_met The missing transverse energy $\vec{E}_{\mathrm{T}}^{\mathrm{miss}}$.

PRI_met_phi The azimuth angle ϕ of the missing transverse energy.

PRI_met_sumet The total transverse energy in the detector.

PRI_jet_num The number of jets (integer with value of 0, 1, 2 or 3; possible larger values have been capped at 3).

PRI_jet_leading_pt The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the leading jet, that is the jet with largest transverse momentum (undefined if PRI_jet_num = 0).

PRI_jet_leading_eta The pseudorapidity η of the leading jet (undefined if PRI_jet_num = 0).

PRI_jet_leading_phi The azimuth angle ϕ of the leading jet (undefined if PRI_jet_num = 0).

PRI_jet_subleading_pt The transverse momentum $\sqrt{p_x^2 + p_y^2}$ of the leading jet, that is, the jet with second largest transverse momentum (undefined if PRI_jet_num ≤ 1).

PRI_jet_subleading_eta The pseudorapidity η of the subleading jet (undefined if PRI_jet_num ≤ 1).

PRI_jet_subleading_phi The azimuth angle ϕ of the subleading jet (undefined if PRI_jet_num ≤ 1).

PRI_jet_all_pt The scalar sum of the transverse momentum of all the jets of the events.

(see Appendix B of the challenge notes)

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: THE DATA

- The following data samples are available:
  https://pprc.qmul.ac.uk/~bevan/statistics/TF/data.zip
  - All of the data: atlas-higgs-challenge-2014-v2.csv
    (818239 examples: signal, background and all KaggleLabel types)
  - Training data: train_*.csv
    (85668 signal and 164334 background examples)
  - Test data: test_*.csv
    (34026 signal and 65976 background examples)
  - Unused data: unused_*.csv
    (6186 signal and 12054 background examples)
  - Private test data[1]: train_private_*.csv
    (153684 signal and 296318 background examples)
  - Training and test data are also split into signal and background for convenience (i.e. * = sig, bg).

[1]This sample was reserved for leaderboard validation of the test.

Additional Work

5. THE H→$\tau\tau$ KAGGLE DATA CHALLENGE PROBLEM: THE DATA

- The following data samples are available:
  https://pprc.qmul.ac.uk/~bevan/statistics/TF/data.zip
  - All of the data: atlas-higgs-challenge-2014-v2.csv
    (818239 examples: signal, background and all KaggleLabel types)
  - Training data: train_*.csv
  - Test da
  - Unuse
  - Private
- Training and test data are also split into signal and background for convenience (i.e. * = sig, bg).

Two small training files with 5k examples have been provided as train_sml_(sig/bg).csv to be used when debugging code and setting up model options.

Alternatively download from the Kaggle website and start to pre-process the files yourself.

Htautau.C assumes that the data files can be found in `./data/`, so make sure that you place the data in the appropriate location.

[1]This sample was reserved for leaderboard validation of the test.

REFERENCES & NOTES

- The TMVA v4 User Guide can be downloaded from:
  - https://root.cern.ch/download/doc/tmva/TMVAUsersGuide.pdf
  - In particular note that the TMVA Reader can be used to process data and add model outputs so that you can use this new variable. See `TMVAClassificationApplication.C` for an example of how to do this.