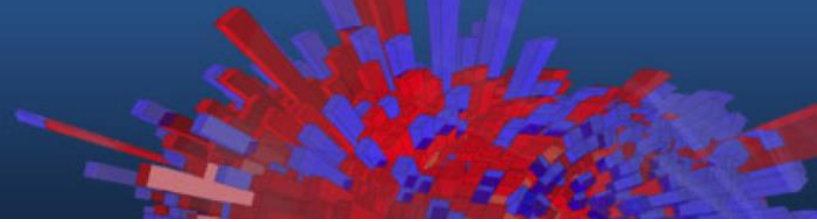


Integration of *T*MVA Output into Jupyter notebooks

Albulena Saliji - CERN Summer Student

Supervisors:
Sergei Gleyzer
Enric Tejedor Saavedra

15.08.2016

*T*MVA output can be very formidable!

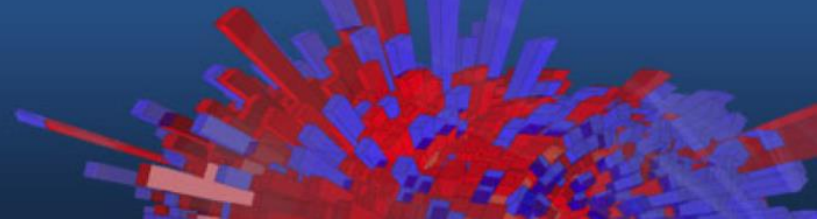
```
--- Factory             : Booking method: BDT DataSet Name: dataset1
--- DataSetFactory      : Dataset[dataset1] : Splitmode is: "RANDOM" the mixmode is: "SAMEASSPLITMODE"
--- DataSetFactory      : Dataset[dataset1] : Create training and testing trees -- looping over class "Signal" ...
--- DataSetFactory      : Dataset[dataset1] : Weight expression for class 'Signal': ""
--- DataSetFactory      : Dataset[dataset1] : Create training and testing trees -- looping over class "Background" ...
--- DataSetFactory      : Dataset[dataset1] : Weight expression for class 'Background': ""
--- DataSetFactory      : Dataset[dataset1] : Number of events in input trees (after possible flattening of arrays):
--- DataSetFactory      : Dataset[dataset1] :     Signal          -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory      : Dataset[dataset1] :     Background      -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory      : Dataset[dataset1] :     Signal     tree -- total number of entries: 6000
--- DataSetFactory      : Dataset[dataset1] :     Background tree -- total number of entries: 6000
--- DataSetFactory      : Dataset[dataset1] : Preselection: (will NOT affect number of requested training and testing ev
--- DataSetFactory      : Dataset[dataset1] :     No preselection cuts applied on event classes
--- DataSetFactory      : Dataset[dataset1] : Weight renormalisation mode: "NumEvents": renormalises all event classes
--- DataSetFactory      : Dataset[dataset1] :  such that the effective (weighted) number of events in each class equals
--- DataSetFactory      : Dataset[dataset1] :  number of events (entries) that you demanded in PrepareTrainingAndTestTre
--- DataSetFactory      : Dataset[dataset1] :  ... i.e. such that Sum[i=1..N_j]{w_i} = N_j, j=0,1,2...
--- DataSetFactory      : Dataset[dataset1] :  ... (note that N_j is the sum of TRAINING events (nTrain_j...with j=Signa
--- DataSetFactory      : Dataset[dataset1] :  ..... Testing events are not renormalised nor included in the renormalisa
--- DataSetFactory      : Dataset[dataset1] : --> Rescale Signal     event weights by factor: 1
--- DataSetFactory      : Dataset[dataset1] : --> Rescale Background event weights by factor: 1
--- DataSetFactory      : Dataset[dataset1] : Number of training and testing events after rescaling:
--- DataSetFactory      : Dataset[dataset1] : ----------------------------------------------------------------------------
--- DataSetFactory      : Dataset[dataset1] : Signal     -- training events            : 1000 (sum of weights: 1000) - r
--- DataSetFactory      : Dataset[dataset1] : Signal     -- testing events             : 5000 (sum of weights: 5000) - r
--- DataSetFactory      : Dataset[dataset1] : Signal     -- training and testing events: 6000 (sum of weights: 6000)
--- DataSetFactory      : Dataset[dataset1] : Background -- training events            : 1000 (sum of weights: 1000) - r
--- DataSetFactory      : Dataset[dataset1] : Background -- testing events             : 5000 (sum of weights: 5000) - r
--- DataSetFactory      : Dataset[dataset1] : Background -- training and testing events: 6000 (sum of weights: 6000)
--- DataSetFactory      : Dataset[dataset1] : Create internal training tree
--- DataSetFactory      : Dataset[dataset1] : Create internal testing tree
```

My task structure:

Integration of TMVA Output into Jupyter notebooks

Improve TMVA output in terminal

Transform TMVA output into HTML



Improve TMVA output from terminal

Before:

```
--- Factory              : Booking method: BDT DataSet Name: dataset1
--- DataSetFactory       : Dataset[dataset1] : Splitmode is: "RANDOM" the mixmode is: "SAMEASSPLITMODE"
--- DataSetFactory       : Dataset[dataset1] : Create training and testing trees -- looping over class "Signal" ...
--- DataSetFactory       : Dataset[dataset1] : Weight expression for class 'Signal': ""
--- DataSetFactory       : Dataset[dataset1] : Create training and testing trees -- looping over class "Background" ...
--- DataSetFactory       : Dataset[dataset1] : Weight expression for class 'Background': ""
--- DataSetFactory       : Dataset[dataset1] : Number of events in input trees (after possible flattening of arrays):
--- DataSetFactory       : Dataset[dataset1] :     Signal          -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory       : Dataset[dataset1] :     Background      -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory       : Dataset[dataset1] :     Signal     tree -- total number of entries: 6000
--- DataSetFactory       : Dataset[dataset1] :     Background tree -- total number of entries: 6000
--- DataSetFactory       : Dataset[dataset1] : Preselection: (will NOT affect number of requested training and testing ev
--- DataSetFactory       : Dataset[dataset1] :     No preselection cuts applied on event classes
--- DataSetFactory       : Dataset[dataset1] : Weight renormalisation mode: "NumEvents": renormalises all event classes
--- DataSetFactory       : Dataset[dataset1] :  such that the effective (weighted) number of events in each class equals
--- DataSetFactory       : Dataset[dataset1] :  number of events (entries) that you demanded in PrepareTrainingAndTestTre
--- DataSetFactory       : Dataset[dataset1] :  ... i.e. such that Sum[i=1..N_j]{w_i} = N_j, j=0,1,2...
--- DataSetFactory       : Dataset[dataset1] :  ... (note that N_j is the sum of TRAINING events (nTrain_j...with j=Signa
--- DataSetFactory       : Dataset[dataset1] :  ..... Testing events are not renormalised nor included in the renormalisa
--- DataSetFactory       : Dataset[dataset1] : --> Rescale Signal     event weights by factor: 1
--- DataSetFactory       : Dataset[dataset1] : --> Rescale Background event weights by factor: 1
--- DataSetFactory       : Dataset[dataset1] : Number of training and testing events after rescaling:
--- DataSetFactory       : Dataset[dataset1] : ---------------------------------------------------------------------------
--- DataSetFactory       : Dataset[dataset1] : Signal     -- training events            : 1000 (sum of weights: 1000) - r
--- DataSetFactory       : Dataset[dataset1] : Signal     -- testing events             : 5000 (sum of weights: 5000) - r
--- DataSetFactory       : Dataset[dataset1] : Signal     -- training and testing events: 6000 (sum of weights: 6000)
--- DataSetFactory       : Dataset[dataset1] : Background -- training events            : 1000 (sum of weights: 1000) - r
--- DataSetFactory       : Dataset[dataset1] : Background -- testing events             : 5000 (sum of weights: 5000) - r
--- DataSetFactory       : Dataset[dataset1] : Background -- training and testing events: 6000 (sum of weights: 6000)
```

After:

```
Factory              : Booking method: BDT
                     :
DataSetFactory       : [dataset1] : Number of events in input trees
                     : Number of training and testing events
                     : ---------------------------------------------------------------------------
                     : Signal     -- training events            : 1000
                     : Signal     -- testing events             : 5000
                     : Signal     -- training and testing events: 6000
                     : Background -- training events            : 1000
                     : Background -- testing events             : 5000
                     : Background -- training and testing events: 6000
```

Before:

```
--- Factory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
--- DataSetFactory
```

```
: Booking method: BDT DataSet Name: dataset1
: Dataset[dataset1] : Splitmode is: "RANDOM" the mixmode is: "SAMEASSPLITMODE"
: Dataset[dataset1] : Create training and testing trees -- looping over class "Signal" ...
: Dataset[dataset1] : Weight expression for class 'Signal': ""
: Dataset[dataset1] : Create training and testing trees -- looping over class "Background" ...
: Dataset[dataset1] : Weight expression for class 'Background': ""
: Dataset[dataset1] : Number of events in input trees (after possible flattening of arrays):
: Dataset[dataset1] :    Signal          -- number of events       : 6000   / sum of weights: 6
: Dataset[dataset1] :    Background      -- number of events       : 6000   / sum of weights: 6
: Dataset[dataset1] :    Signal     tree -- total number of entries: 6000
: Dataset[dataset1] :    Background tree -- total number of entries: 6000
: Dataset[dataset1] : Preselection: (will NOT affect number of requested training and testing ev
: Dataset[dataset1] :    No preselection cuts applied on event classes
: Dataset[dataset1] : Weight renormalisation mode: "NumEvents": renormalises all event classes
: Dataset[dataset1] :  such that the effective (weighted) number of events in each class equals
: Dataset[dataset1] :  number of events (entries) that you demanded in PrepareTrainingAndTestTre
: Dataset[dataset1] :  ... i.e. such that Sum[i=1..N_j]{w_i} = N_j, j=0,1,2...
: Dataset[dataset1] :  ... (note that N_j is the sum of TRAINING events (nTrain_j...with j=Signa
: Dataset[dataset1] :  ..... Testing events are not renormalised nor included in the renormalisa
: Dataset[dataset1] :  -> Rescale Signal     event weights by factor: 1
: Dataset[dataset1] :  -> Rescale Background event weights by factor: 1
: Dataset[dataset1] : Number of training and testing events after rescaling:
: Dataset[dataset1] : ---------------------------------------------------------------------------
: Dataset[dataset1] : Signal     -- training events            : 1000 (sum of weights: 1000) - r
: Dataset[dataset1] : Signal     -- testing events             : 5000 (sum of weights: 5000) - r
: Dataset[dataset1] : Signal     -- training and testing events: 6000 (sum of weights: 6000)
: Dataset[dataset1] : Background -- training events            : 1000 (sum of weights: 1000) - r
: Dataset[dataset1] : Background -- testing events             : 5000 (sum of weights: 5000) - r
: Dataset[dataset1] : Background -- training and testing events: 6000 (sum of weights: 6000)
```

After:

```
Factory

DataSetFactory
```

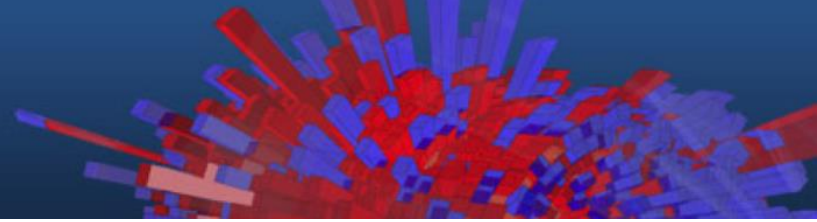
```
: Booking method: BDT
:
: [dataset1] : Number of events in input trees
: Number of training and testing events
: ---------------------------------------------------------------------------
: Signal     -- training events            : 1000
: Signal     -- testing events             : 5000
: Signal     -- training and testing events: 6000
: Background -- training events            : 1000
: Background -- testing events             : 5000
: Background -- training and testing events: 6000
```

Before:

```
--- Factory              : Booking method: BDT DataSet Name: dataset1
--- DataSetFactory       : Dataset[dataset1] : Splitmode is: "RANDOM" the mixmode is: "SAMEASSPLITMODE"
--- DataSetFactory       : Dataset[dataset1] : Create training and testing trees -- looping over class "Signal" ...
--- DataSetFactory       : Dataset[dataset1] : Weight expression for class 'Signal': ""
--- DataSetFactory       : Dataset[dataset1] : Create training and testing trees -- looping over class "Background" ...
--- DataSetFactory       : Dataset[dataset1] : Weight expression for class 'Background': ""
--- DataSetFactory       : Dataset[dataset1] : Number of events in input trees (after possible flattening of arrays):
--- DataSetFactory       : Dataset[dataset1] :     Signal          -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory       : Dataset[dataset1] :     Background      -- number of events       : 6000   / sum of weights: 6
--- DataSetFactory       : Dataset[dataset1] :     Signal     tree -- total number of entries: 6000
--- DataSetFactory       : Dataset[dataset1] :     Background tree -- total number of entries: 6000
--- DataSetFactory       : Dataset[dataset1] : Preselection: (will NOT affect number of requested training and testing ev
--- DataSetFactory       : Dataset[dataset1] :     No preselection cuts applied on event classes
--- DataSetFactory       : Dataset[dataset1] : Weight renormalisation mode: "NumEvents": renormalises all event classes
--- DataSetFactory       : Dataset[dataset1] :  such that the effective (weighted) number of events in each class equals
--- DataSetFactory       : Dataset[dataset1] :  number of events (entries) that you demanded in PrepareTrainingAndTestTre
--- DataSetFactory       : Dataset[dataset1] :  ... i.e. such that Sum[i=1..N_j]{w_i} = N_j, j=0,1,2...
--- DataSetFactory       : Dataset[dataset1] :  ... (note that N_j is the sum of TRAINING events (nTrain_j...with j=Signa
--- DataSetFactory       : Dataset[dataset1] :  ..... Testing events are not renormalised nor included in the renormalisa
--- DataSetFactory       : Dataset[dataset1] : --> Rescale Signal     event weights by factor: 1
--- DataSetFactory       : Dataset[dataset1] : --> Rescale Background event weights by factor: 1
--- DataSetFactory       : Dataset[dataset1] : Number of training and testing events after rescaling:
--- DataSetFactory       : Dataset[dataset1] : ---------------------------------------------------------------------------
--- DataSetFactory       : Dataset[dataset1] : Signal     -- training events           : 1000 (sum of weights: 1000) - r
--- DataSetFactory       : Dataset[dataset1] : Signal     -- testing events            : 5000 (sum of weights: 5000) - r
--- DataSetFactory       : Dataset[dataset1] : Signal     -- training and testing events: 6000 (sum of weights: 6000)
--- DataSetFactory       : Dataset[dataset1] : Background -- training events           : 1000 (sum of weights: 1000) - r
--- DataSetFactory       : Dataset[dataset1] : Background -- testing events            : 5000 (sum of weights: 5000) - r
--- DataSetFactory       : Dataset[dataset1] : Background -- training and testing events: 6000 (sum of weights: 6000)
--- DataSetFactory       : Dataset[dataset1] : Create internal training tree
--- DataSetFactory       : Dataset[dataset1] : Create internal testing tree
--- DataSetInfo          : Dataset[dataset1] : Correlation matrix (Signal):
--- DataSetInfo          : --------------------------------
--- DataSetInfo          :            var1    var2    var3
--- DataSetInfo          :    var1:  +1.000  +0.386  +0.597
--- DataSetInfo          :    var2:  +0.386  +1.000  +0.696
--- DataSetInfo          :    var3:  +0.597  +0.696  +1.000
--- DataSetInfo          : --------------------------------
--- DataSetInfo          : Dataset[dataset1] : Correlation matrix (Background):
--- DataSetInfo          : --------------------------------
--- DataSetInfo          :            var1    var2    var3
--- DataSetInfo          :    var1:  +1.000  +0.856  +0.914
--- DataSetInfo          :    var2:  +0.856  +1.000  +0.927
--- DataSetInfo          :    var3:  +0.914  +0.927  +1.000
--- DataSetInfo          : --------------------------------
--- DataSetFactory       : Dataset[dataset1] :
--- Factory              : Booking method: MLP DataSet Name: dataset1
--- MLP                  : Dataset[dataset1] : Create Transformation "N" with events from all classes.
--- Norm                 : Transformation, Variable selection :
--- Norm                 : Input : variable 'var1' (index=0).   <---> Output : variable 'var1' (index=0).
--- Norm                 : Input : variable 'var2' (index=1).   <---> Output : variable 'var2' (index=1).
--- Norm                 : Input : variable 'var3' (index=2).   <---> Output : variable 'var3' (index=2).
--- MLP                  : Building Network
--- MLP                  : Initializing weights
```

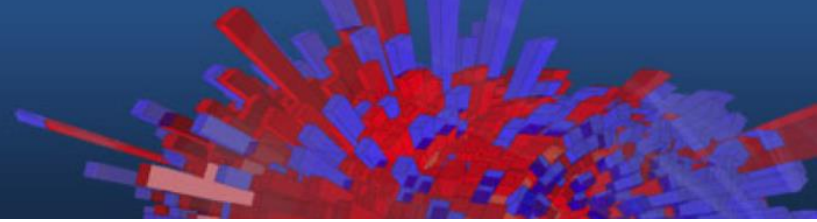
After:

```
Factory              : Booking method: BDT
                     :
DataSetFactory       : [dataset1] : Number of events in input trees
                     : Number of training and testing events
                     : ---------------------------------------------------------------------------
                     : Signal     -- training events            : 1000
                     : Signal     -- testing events             : 5000
                     : Signal     -- training and testing events: 6000
                     : Background -- training events            : 1000
                     : Background -- testing events             : 5000
                     : Background -- training and testing events: 6000
                     :
DataSetInfo          : Correlation matrix (Signal):
                     : --------------------------------
                     :            var1    var2    var3
                     :    var1:  +1.000  +0.386  +0.597
                     :    var2:  +0.386  +1.000  +0.696
                     :    var3:  +0.597  +0.696  +1.000
                     : --------------------------------
DataSetInfo          : Correlation matrix (Background):
                     : --------------------------------
                     :            var1    var2    var3
                     :    var1:  +1.000  +0.856  +0.914
                     :    var2:  +0.856  +1.000  +0.927
                     :    var3:  +0.914  +0.927  +1.000
                     : --------------------------------
DataSetFactory       : [dataset1] :
                     :
Factory              : Booking method: MLP
                     :
MLP                  : [dataset1] : Create Transformation "N" with events from all classes.
                     :
Norm                 : Transformation, Variable selection :
                     : Input : variable 'var1' <---> Output : variable 'var1'
                     : Input : variable 'var2' <---> Output : variable 'var2'
                     : Input : variable 'var3' <---> Output : variable 'var3'
                     :
MLP                  : Building Network.
                     : Initializing weights
```



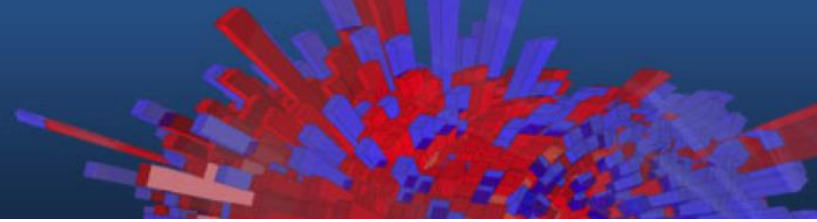
Transform TMVA output into HTML

Transform TMVA output into HTML



- Define a Python function
    - transform(output, error)
- Identify patterns from the TMVA Output and transform them into HTML
    - Regular expressions

Transform TMVA output into HTML



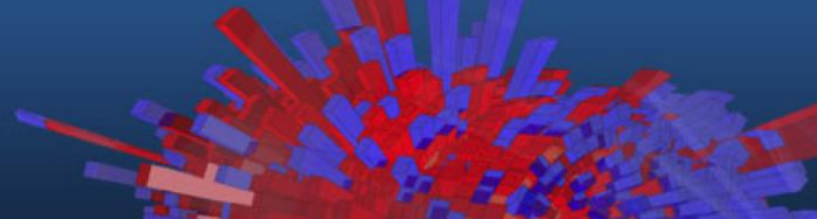
- Use Jupyter notebook as interface
- Return HTML output and error
    - Make the output more appealing to the user

Transformed elements into HTML:

- Font style and size
- Headers
- Specify datasets – add some color
- Create tables

Transformed elements

Next slide



Before:

```
DataSetInfo              : [dataset1] : Added class "Signal"
                         : Add Tree Sig of type Signal with 6000 events
```

Font style
Font size
Header
Dataset color

After:

DataSetInfo : : **[dataset1]**: Added class "Signal"
: Add Tree Sig of type Signal with 6000 events

Before:

```
DataSetInfo              : Correlation matrix (Signal):
                         : ------------------------------
                         :             var1    var2    var3
                         :    var1:  +1.000  +0.386  +0.597
                         :    var2:  +0.386  +1.000  +0.696
                         :    var3:  +0.597  +0.696  +1.000
                         : ------------------------------
```

After:

Header
Font size
Font style

DataSetInfo : : Correlation matrix (Signal):

: ----------

	var1	var2	var3
var1:	+1.000	+0.386	+0.597
var2:	+0.386	+1.000	+0.696
var3:	+0.597	+0.696	+1.000

: ----------

Table

Before:

```
DataSetFactory           : [dataset1] : Number of events in input trees
                         : Number of training and testing events
                         : -------------------------------------------------------------------------
                         : Signal     -- training events            : 1000
                         : Signal     -- testing events             : 5000
                         : Signal     -- training and testing events: 6000
                         : Background -- training events            : 1000
                         : Background -- testing events             : 5000
                         : Background -- training and testing events: 6000
```

After:

DataSetFactory : : **[dataset1]**: Number of events in input trees
: Number of training and testing events
: ---

Signal	training events	1000
Signal	testing events	5000
Signal	training and testing events	6000
Background	training events	1000
Background	testing events	5000
Background	training and testing events	6000

Before:

```
BDT
Factory
MLP
Factory
BDT
Factory
MLP
Factory
BDT
TFHandler_BDT
Factory
TFHandler_MLP
```

```
: [dataset1] : Evaluation of BDT on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: 0.268 sec
: Test method: MLP for Classification performance
:
: [dataset1] : Evaluation of MLP on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: 0.0293 sec
: Test method: BDT for Classification performance
:
: [dataset2] : Evaluation of BDT on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: 0.218 sec
: Test method: MLP for Classification performance
:
: [dataset2] : Evaluation of MLP on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: 0.0383 sec
: Evaluate classifier: BDT
:
: [dataset1] : Loop over test events and fill histograms with classifier response...
:
: Variable        Mean        RMS   [        Min        Max ]
: -------------------------------------------------------------
:     var1: 0.00077102     1.6695   [    -5.8991     4.7639 ]
:     var2: -0.0063164     1.5765   [    -5.2454     4.8300 ]
:     var3:  -0.010870     1.7365   [    -5.3563     4.6430 ]
: -------------------------------------------------------------
: Evaluate classifier: MLP
:
: Variable        Mean        RMS   [        Min        Max ]
: -------------------------------------------------------------
:     var1:   0.066774    0.35913   [    -1.2024     1.0914 ]
:     var2:   0.079492    0.36669   [    -1.1391     1.2044 ]
:     var3:   0.079125    0.37282   [    -1.0685     1.0783 ]
: -------------------------------------------------------------
```

After:

BDT : : [dataset1]: Evaluation of BDT on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: [1;31m0.28 sec[0m

Factory : : Test method: MLP for Classification performance
:

MLP : : [dataset1]: Evaluation of MLP on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: [1;31m0.0661 sec[0m

Factory : : Test method: BDT for Classification performance
:

BDT : : [dataset2]: Evaluation of BDT on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: [1;31m0.365 sec[0m

Factory : : Test method: MLP for Classification performance
:

MLP : : [dataset2]: Evaluation of MLP on testing sample (10000 events)
: Elapsed time for evaluation of 10000 events: [1;31m0.0346 sec[0m

Factory : : Evaluate classifier: BDT
:

BDT : : [dataset1]: Loop over test events and fill histograms with classifier response...
:

TFHandler_BDT : ----------

Variable	Mean	RMS	Min	Max
var1	0.00077102	1.6695	-5.8991	4.7639
var2	-0.0063164	1.5765	-5.2454	4.8300
var3	-0.010870	1.7365	-5.3563	4.6430

: ----------

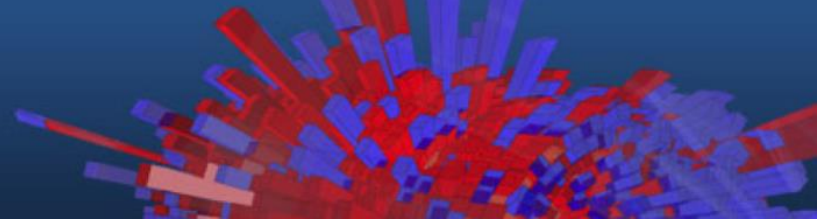
Factory : : Evaluate classifier: MLP
:

TFHandler_MLP : ----------

Variable	Mean	RMS	Min	Max
var1	0.066774	0.35913	-1.2024	1.0914
var2	0.079492	0.36669	-1.1391	1.2044
var3	0.079125	0.37282	-1.0685	1.0783

: ----------

Result:



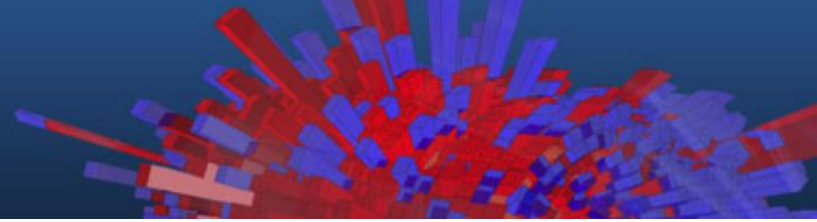
Done:

- Cleaner TMVA Output
- More user-friendly TMVA Output
    - Using HTML

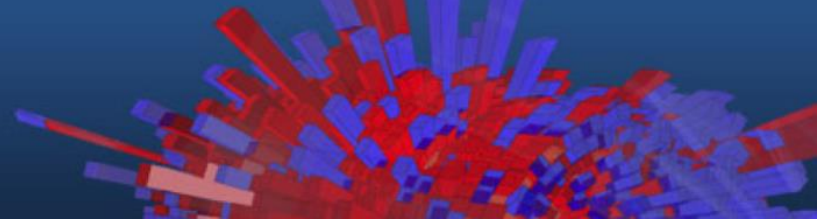
Can be done better:

- Even cleaner TMVA Output
- Even more creative TMVA Output
    - Using HTML

Notebooks:



- http://nbviewer.jupyter.org/github/salbulena/friday-rehearsal/blob/master/clean%20TMVA%20output%20notebook.ipynb
- http://nbviewer.jupyter.org/github/iml-wg/tmvatutorials/blob/master/TMVA_DataLoader.ipynb
- http://nbviewer.jupyter.org/github/salbulena/friday-rehearsal/blob/master/HTML%20notebook.ipynb



Thank you!