TMVA
Cross Validation update

Kim Albertsson
2017-09-01

Agenda

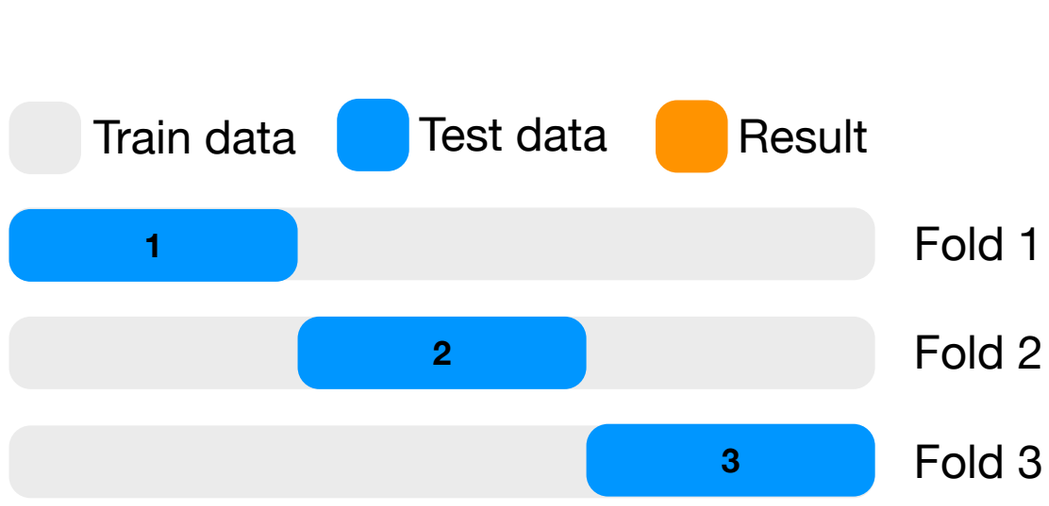
- CV/CE
- Examples with results
- Design Details
  - CrossEvaluation/MethodCrossEvaluation/CvSplit

CV / CE

Cross Validation

- A technique for assessing how the results of a statistical analysis will generalise to an independent data set.

- Allows analysis of small data sets at the cost of increased computation time



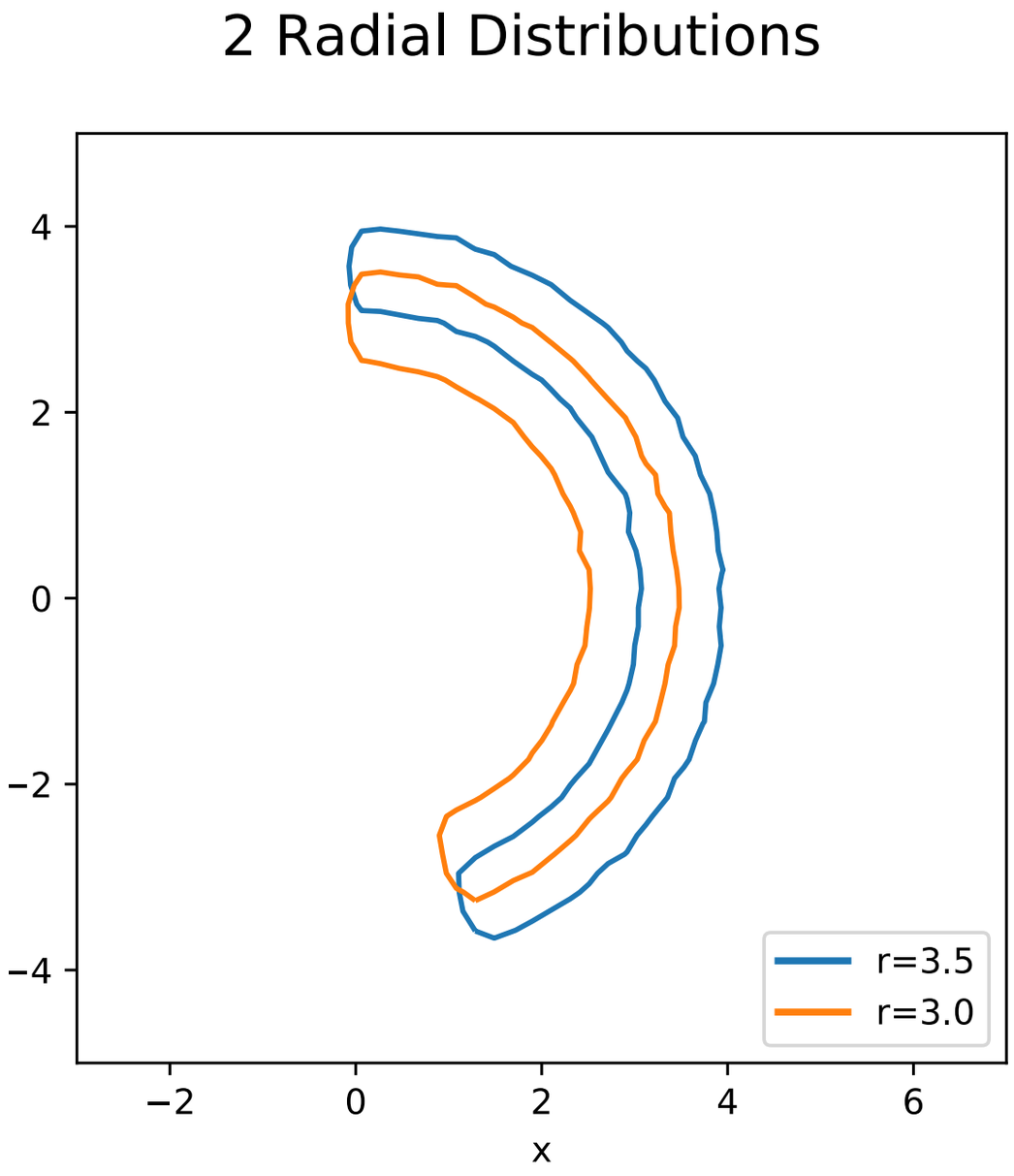
Cross Evaluation

- Builds on cross validation
- One classifier that can evaluate the complete input data set

Examples with Results

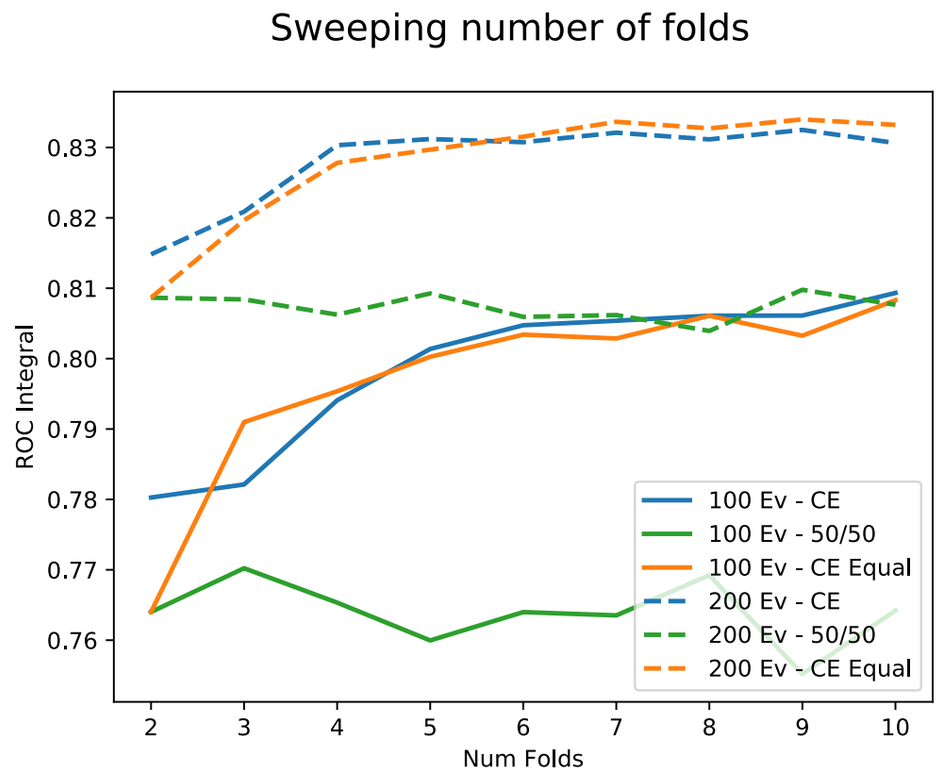
Example - Toy

- Two widened arcs overlapping
- For training we draw few points (~100)
- For evaluation we use many points (~10000)



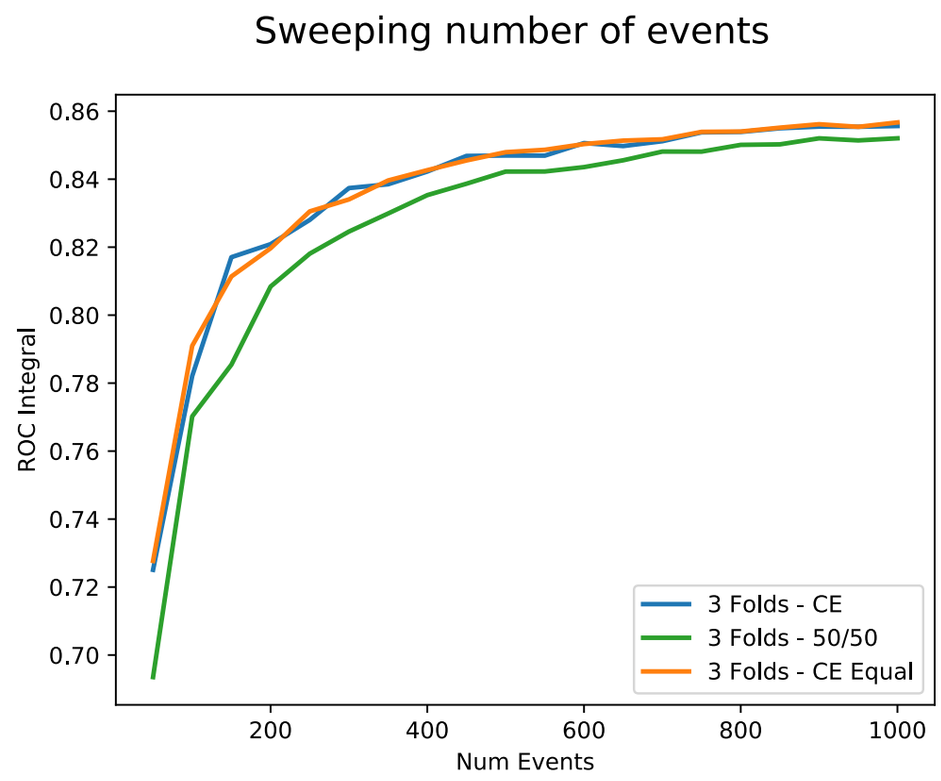
Example - Toy

- 50/50 split performs the same for all values of numFolds
- CE and CE Equal perform about the same



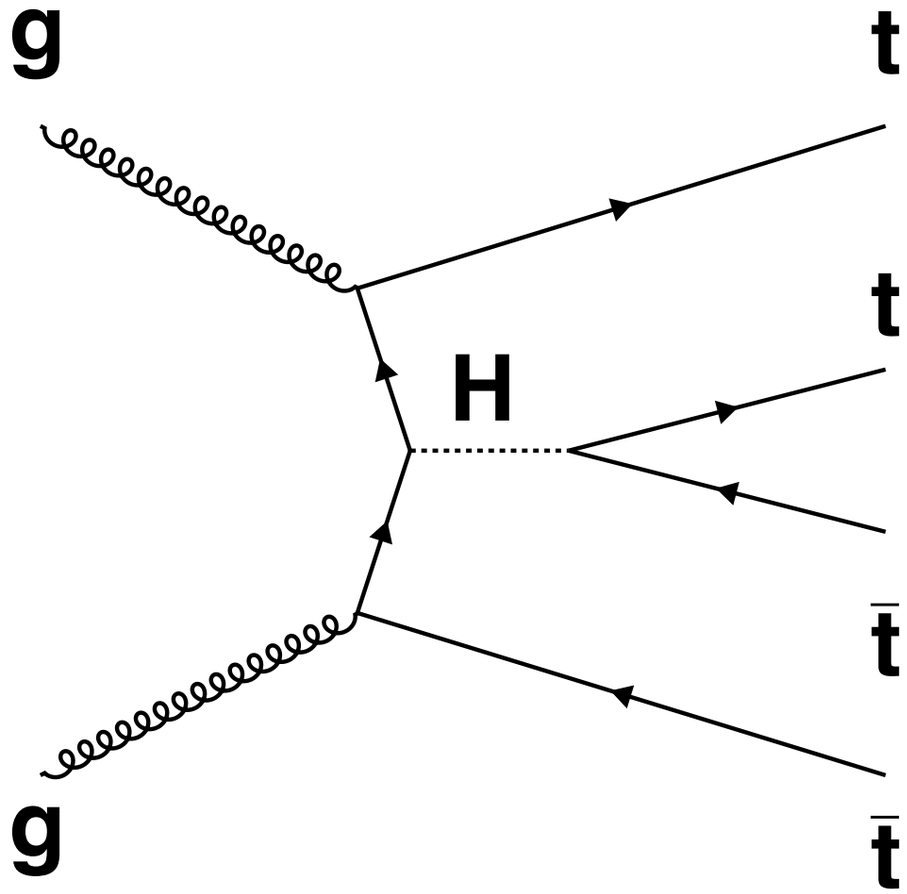
Example - Toy

- 50/50 split performs worst
- CE and CE Equal perform about the same
- All get better with larger training set size



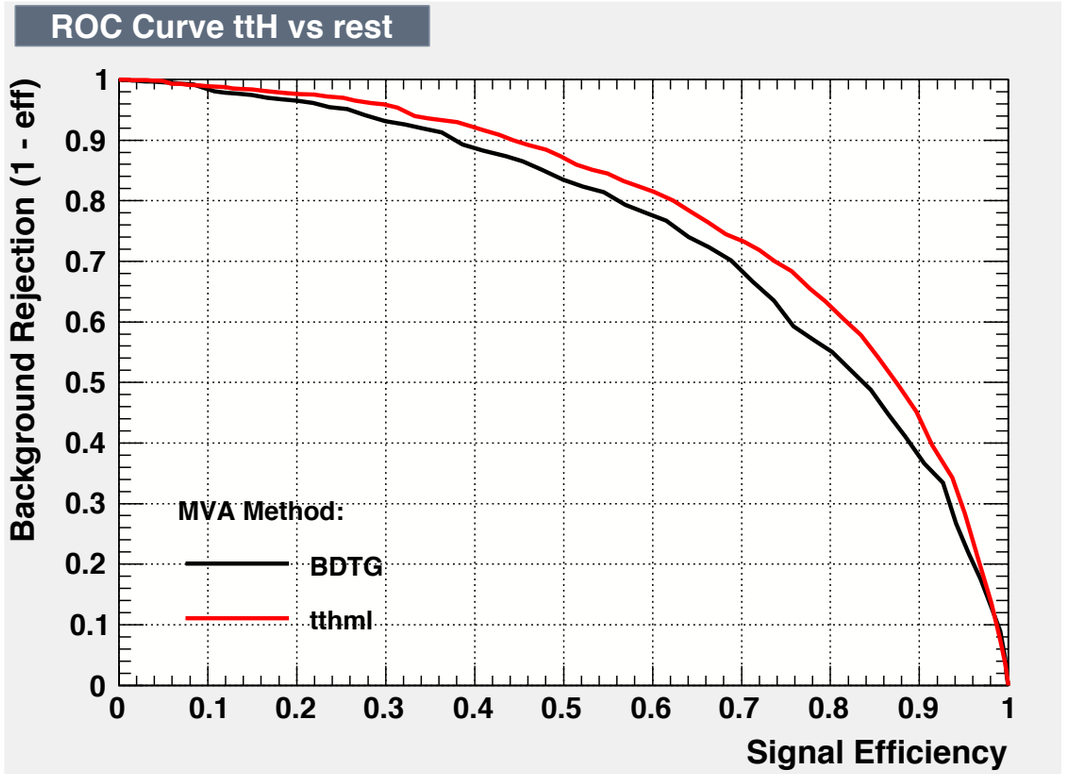
Example - TTHml

- Tested against a real analysis: tthml
- Stefan Gadatsch et. al.
- Searches for Higgs decaying into multiple leptons
- Many channels to identify



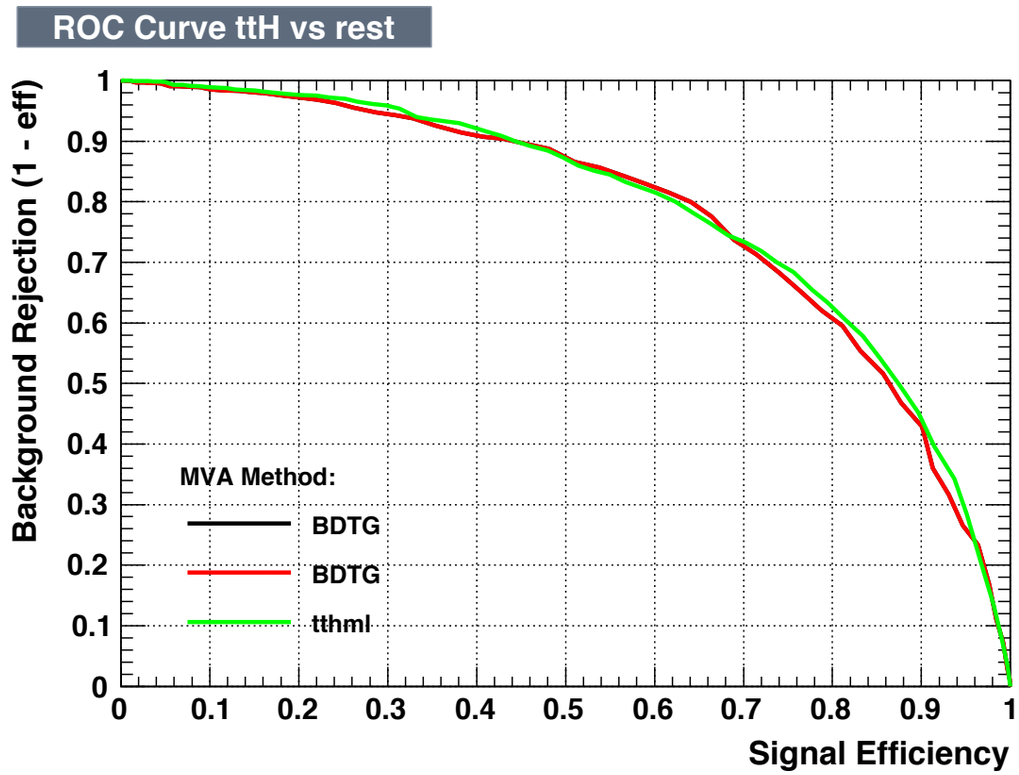
Example - TTHml

- 5-fold Cross Evaluation
- BDT using gradient boosting



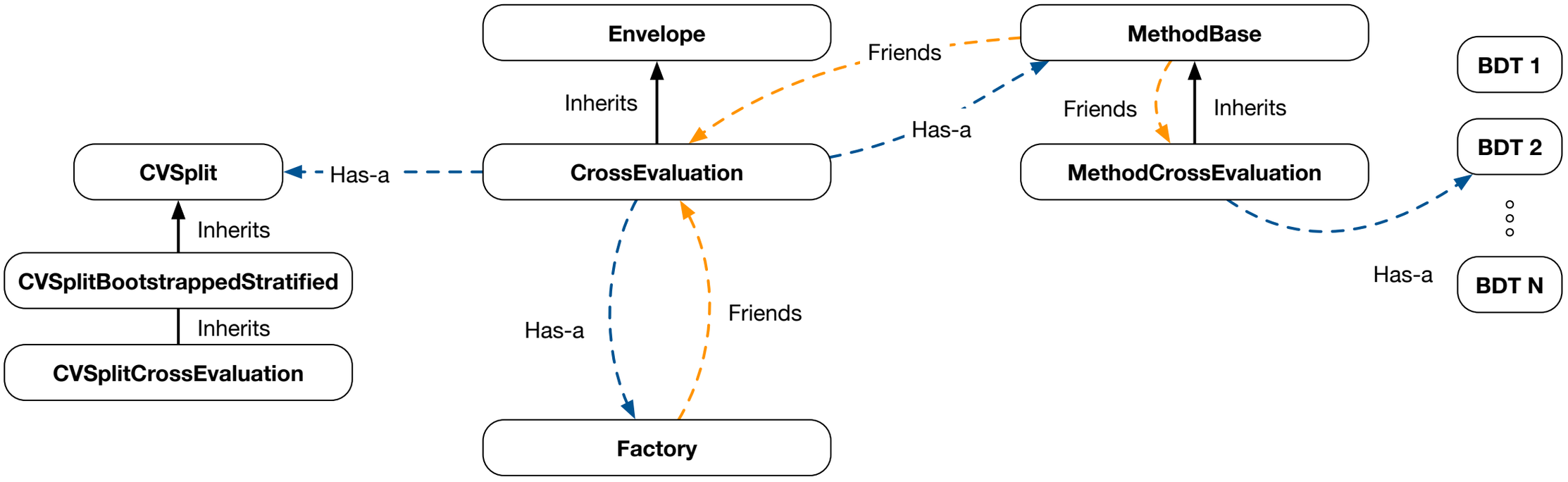
Example - TTHml

- Reference
- One BDT, same parameters as CE
- Data split 50% training, 50% testing



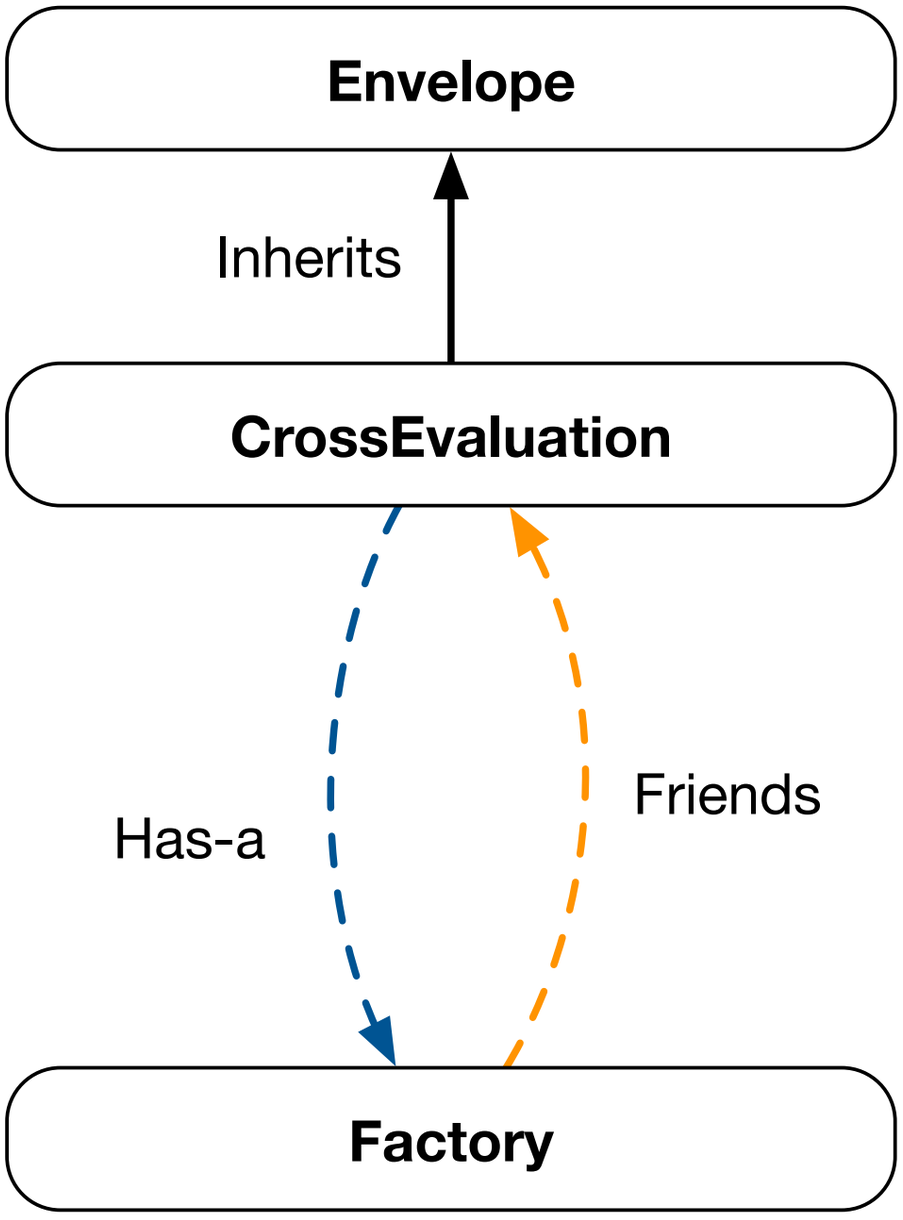
Design Details

Class hierarchy overview



CrossEvaluation

- Builds on Envelope and CrossValidation classes of Omar Zapata and Thomas Stevenson



CrossEvaluation Interface

- Get Factory to get ROC Curves for method

- NumFolds, number of folds to use

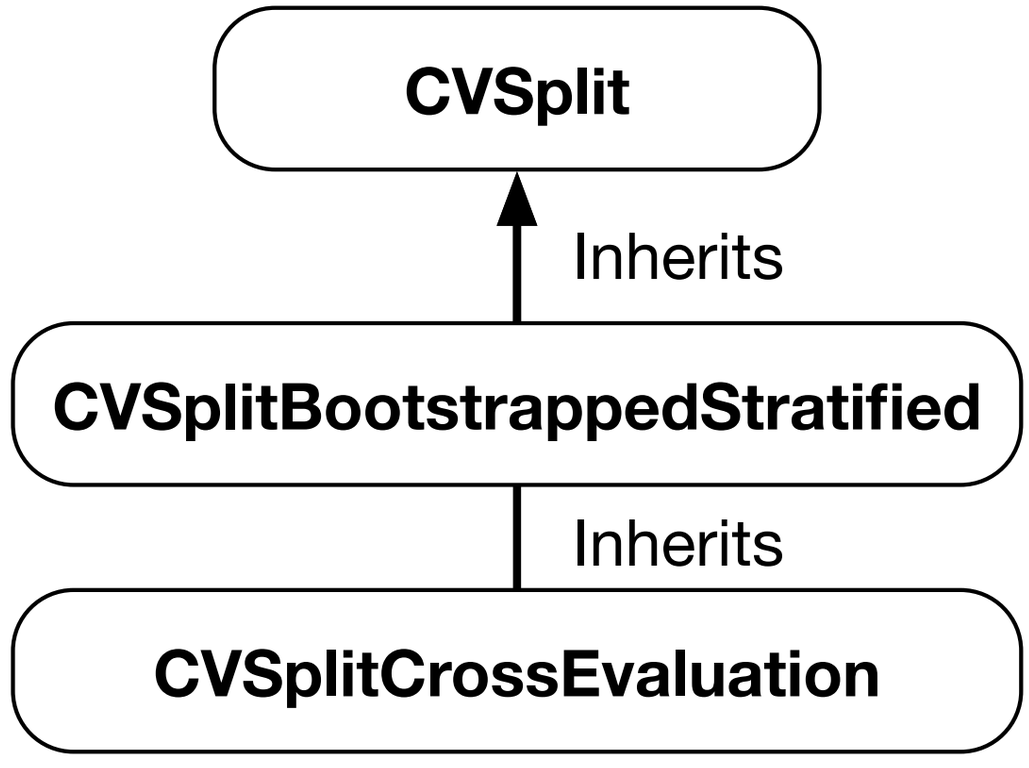
- SplitSpectator, a spectator variable defined in the DataSet. Used for splitting

CrossEvaluation
BookMethod(...)
Evaluate(...)
GetFactory(...)

Options
NumFolds
SplitSpectator

CVSplit

- Used by Dataloader
- Responsible for defining and executing a split
- Stores necessary data internally (this was previously done in dataloader)



Dataloader Interface

- Removed old data loader CV interface
- Replaced by CVSplit
- PrepareTrainingAndTestTree lacked implementation
- SplitSets is now private method of CVSplit

```
Dataloader
PrepareTrainingAndTestTree
        (int foldNum, ...)
MakeKFoldDataSet(...)
PrepareFoldDataSet(...)
SplitSets(...)
MakeKFoldDataSet       (CVSplit & s)
PrepareFoldDataSet     (CVSplit & s)
RecombineKFoldDataSet  (CVSplit & s)
```

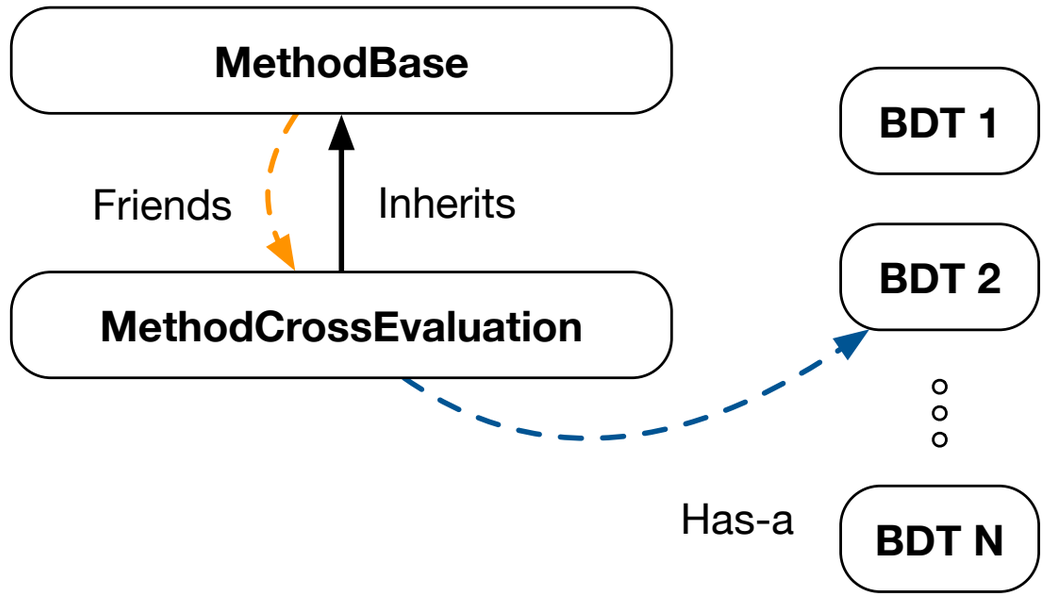
CVSplit Interface

- MakeKFoldDataSet shuffles data around to prepare for CE
- PrepareFoldDataSet injects the fold data into the DataSet
- RecombineKFoldDataSet joins splits and puts them as training and testing sets

```
CVSplit
MakeKFoldDataSet
    (DataSetInfo & dsi)
PrepareFoldDataSet
    (DataSetInfo & dsi,
     UInt_t foldNumber,
     ETreeType tt)
RecombineKFoldDataSet
    (DataSetInfo & dsi,
     ETreeType tt)
```

MethodCrossEvaluation

- Used for application phase
- Stores one method per fold internally
- Manages splitting and distribution to the correct method

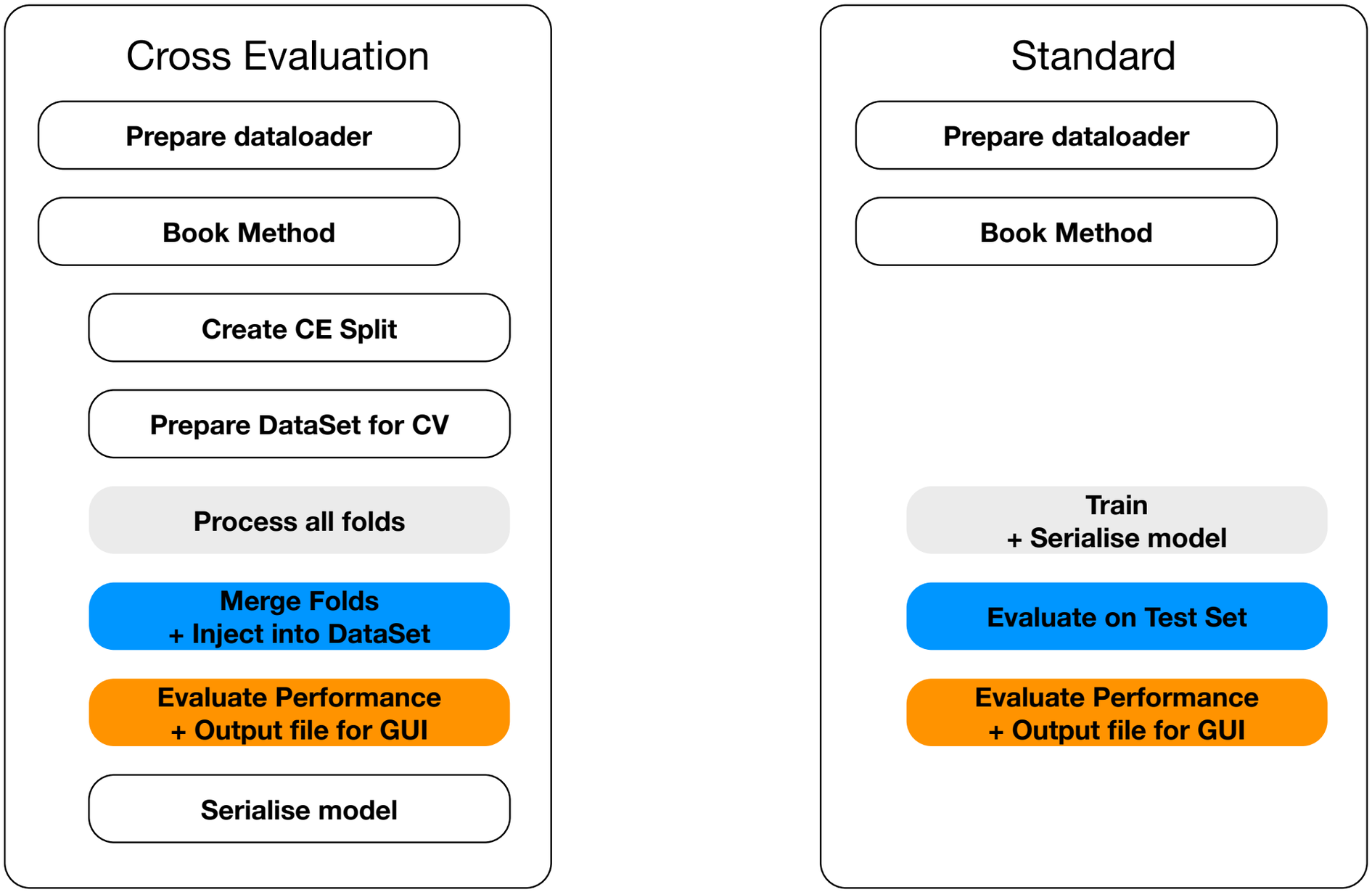


MethodCrossEvaluation Interface

- Minimal implementation of a Method
- Should only be instantiated by CrossEvaluation or Reader
- Supports only reading from xml
- Greyed out methods lack implementation or are disabled

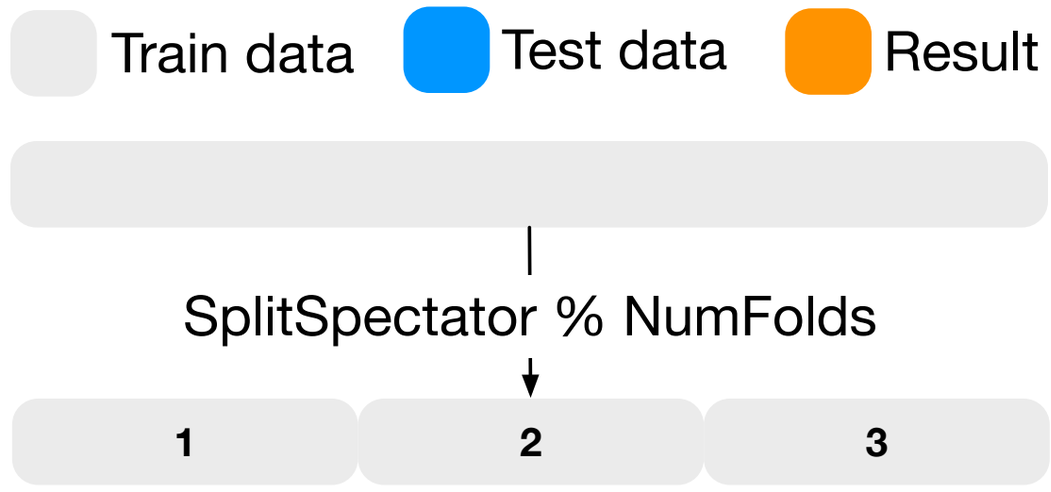
MethodCrossEvaluation
Train
Reset
AddWeightsXMLTo
ReadWeightsFromStream
ReadWeightsFromXML
WriteMonitoringHistosToFile
GetMvaValue
GetMulticlassValues
GetRegressionValues
DeclareOptions
ProcessOptions
MakeClassSpecific
MakeClassSpecificHeader
GetHelpMessage
CreateRanking
HasAnalysisType
Init
DeclareCompatibilityOptions
InstantiateMethodFromXML

Execution flow train/test/evaluate



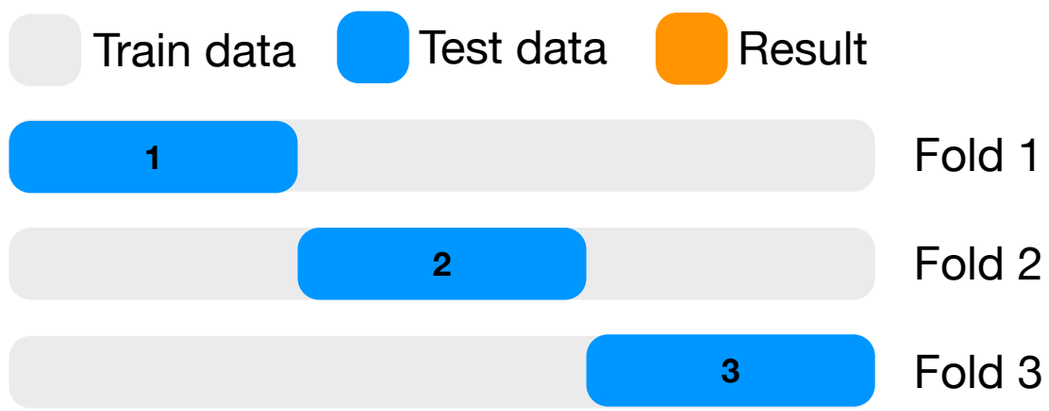
Prepare DataSet for CV

- Uses CVSplitCrossEvaluation
- Folds are stored in split
- Important! CE uses train data. All original test data will be ignored



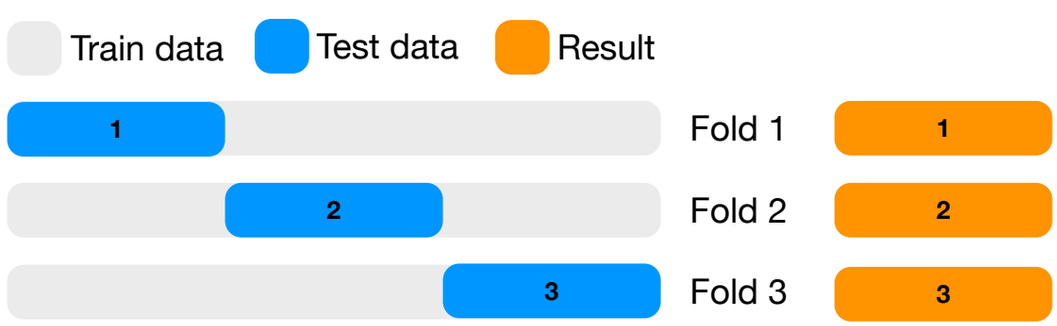
Process all folds

- CVSplit::PrepareFoldDataSet()
- Instantiate new method with given parameters (from CE::BookMethod)
- Train method



Process all folds

- Results stored in internally
- (CE::StoreResults(…))



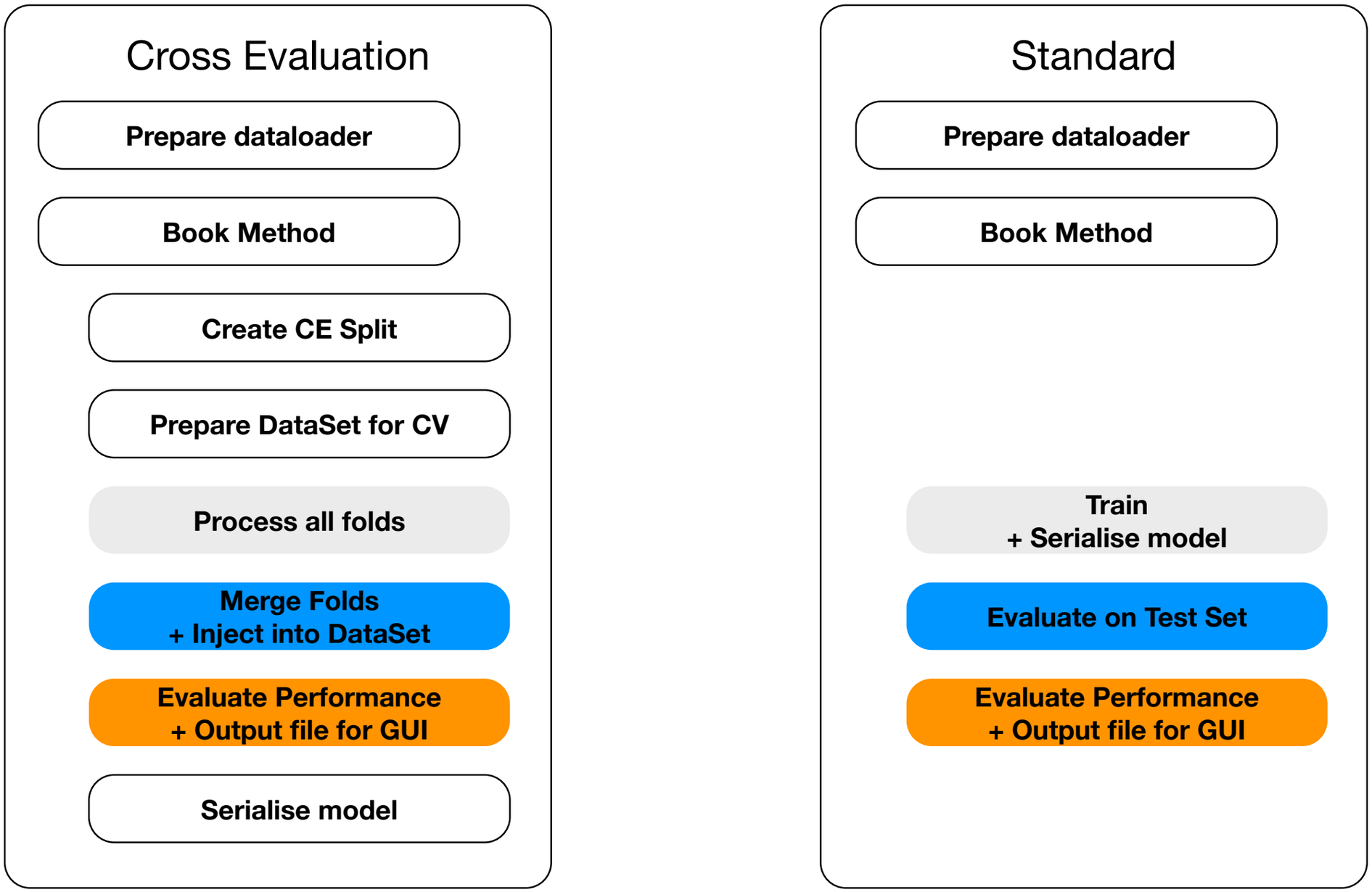
Merge folds

- Reassemble training and test data in DataSet
- Put stored results into DataSet

Evaluate performance

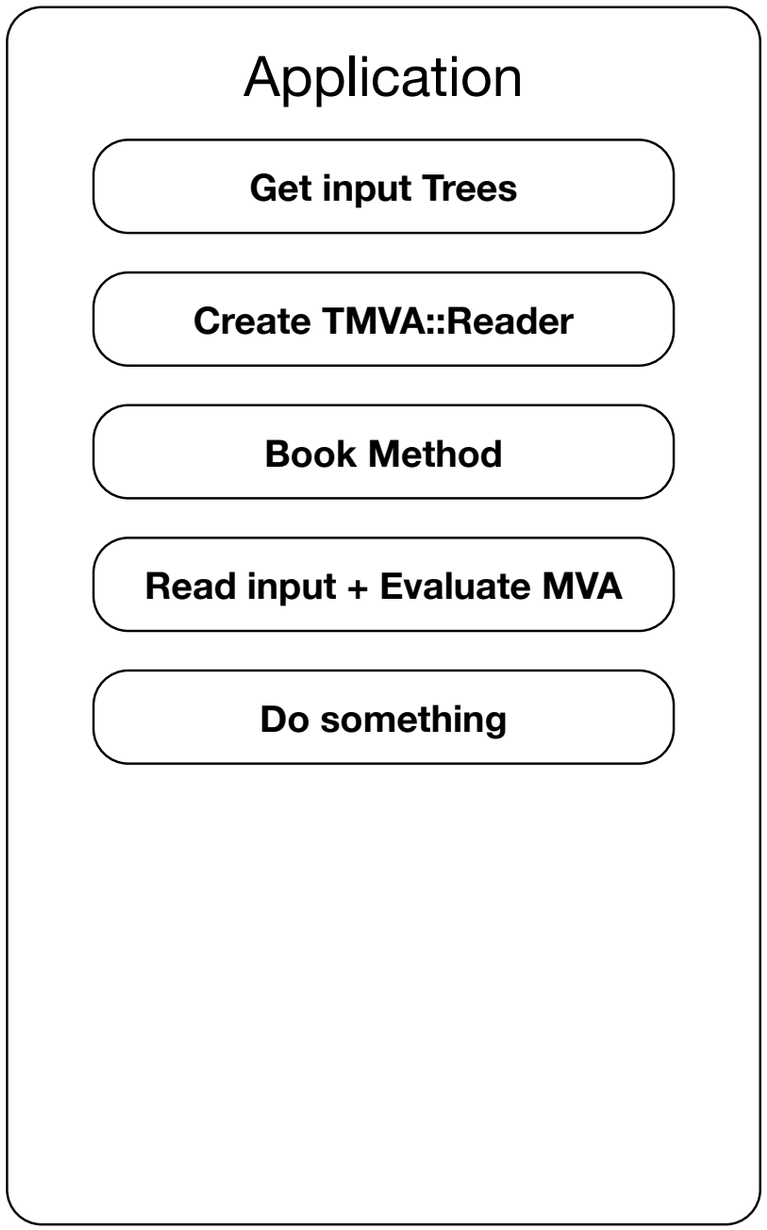
- DataSet is now prepared for performance evaluation
- Instantiate new method and run Factory::EvaluateAllMethods
- Make sure to write output data if required

Execution flow train/test/evaluate



Execution flow application

- Identical flow for standard usage and cross evaluation



MethodCrossEvaluation Interface

- Minimal implementation of a Method
- Should only be instantiated by CrossEvaluation or Reader
- Supports only reading from xml
- Greyed out methods lack implementation or are disabled

MethodCrossEvaluation
Train
Reset
AddWeightsXMLTo
ReadWeightsFromStream
ReadWeightsFromXML
WriteMonitoringHistosToFile
GetMvaValue
GetMulticlassValues
GetRegressionValues
DeclareOptions
ProcessOptions
MakeClassSpecific
MakeClassSpecificHeader
GetHelpMessage
CreateRanking
HasAnalysisType
Init
DeclareCompatibilityOptions
InstantiateMethodFromXML

MethodCrossEvaluation Interface

- Minimal implementation of a Method
- Should only be instantiated by CrossEvaluation or Reader
- Supports only reading from xml
- Greyed out methods lack implementation or are disabled

Used for BookMethod

MethodCrossEvaluation

```
Train
Reset
AddWeightsXMLTo
ReadWeightsFromStream
ReadWeightsFromXML
WriteMonitoringHistosToFile
GetMvaValue
GetMulticlassValues
GetRegressionValues
DeclareOptions
ProcessOptions
MakeClassSpecific
MakeClassSpecificHeader
GetHelpMessage
CreateRanking
HasAnalysisType
Init
DeclareCompatibilityOptions
InstantiateMethodFromXML
```

MethodCrossEvaluation Interface

- Minimal implementation of a Method
- Should only be instantiated by CrossEvaluation or Reader
- Supports only reading from xml
- Greyed out methods lack implementation or are disabled

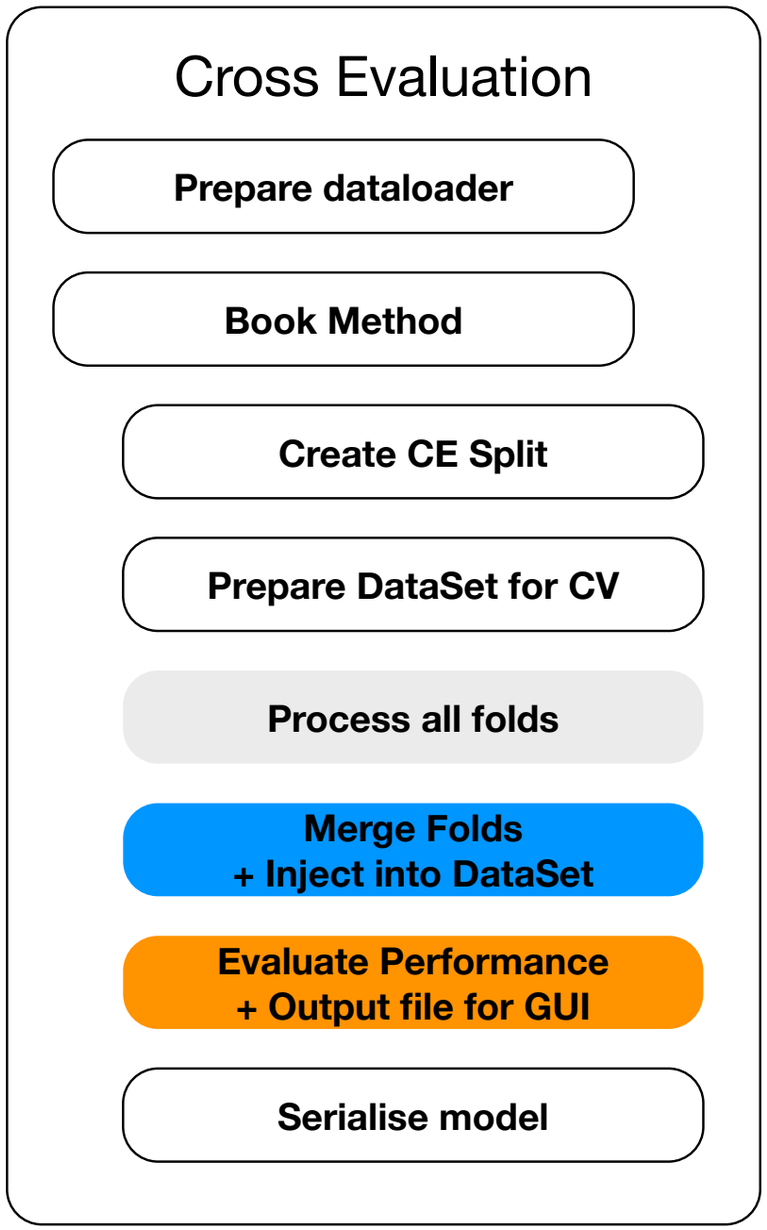
Used for EvaluateMVA

MethodCrossEvaluation

```
Train
Reset
AddWeightsXMLTo
ReadWeightsFromStream
ReadWeightsFromXML
WriteMonitoringHistosToFile
GetMvaValue
GetMulticlassValues
GetRegressionValues
DeclareOptions
ProcessOptions
MakeClassSpecific
MakeClassSpecificHeader
GetHelpMessage
CreateRanking
HasAnalysisType
Init
DeclareCompatibilityOptions
InstantiateMethodFromXML
```

Serialisation

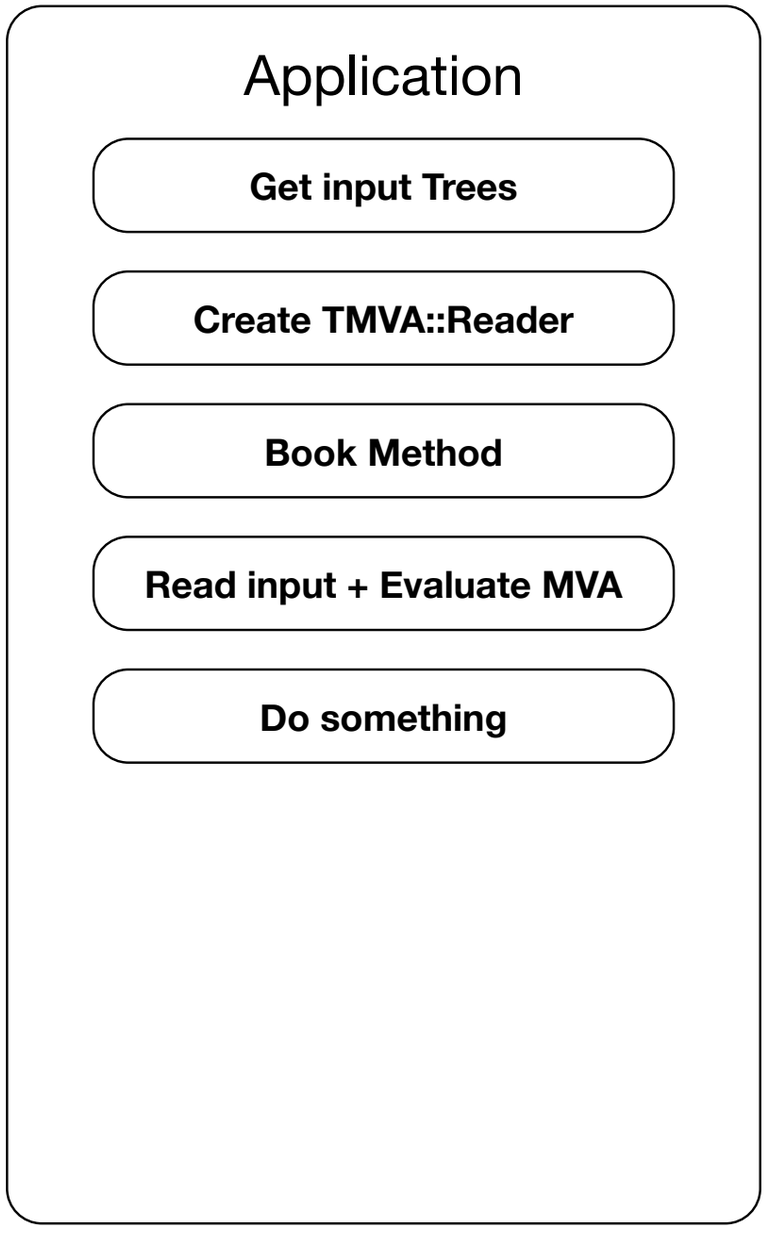
- MethodCrossEvaluation keeps a number of other methods
- Assumes these are serialised already
- Reads them in and emits them again inside its own xml
- Adds auxiliary information, numFolds, SplitSpectator etc.



Deserialisation

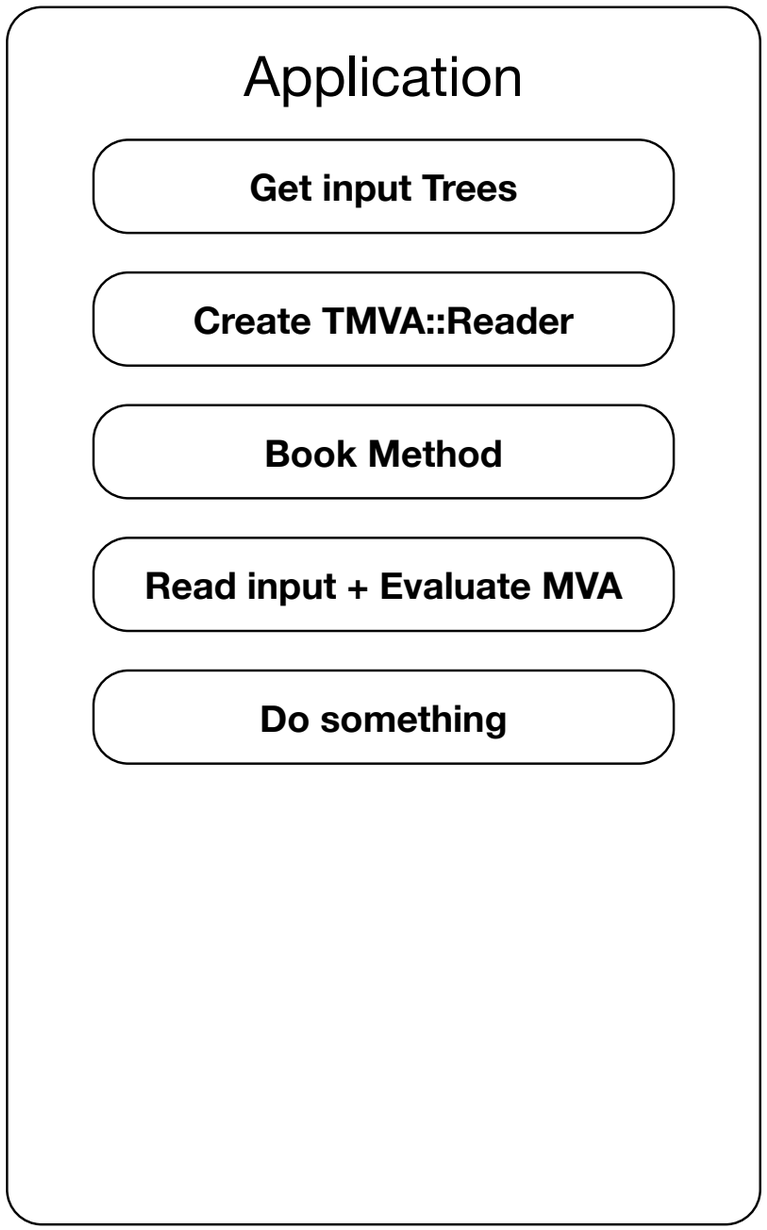
- Instantiates methods from included xml
- Distributes events to the correct method using “cross evaluation formula”

  specVar % numFolds



Deserialisation

- That’s it!



Thanks