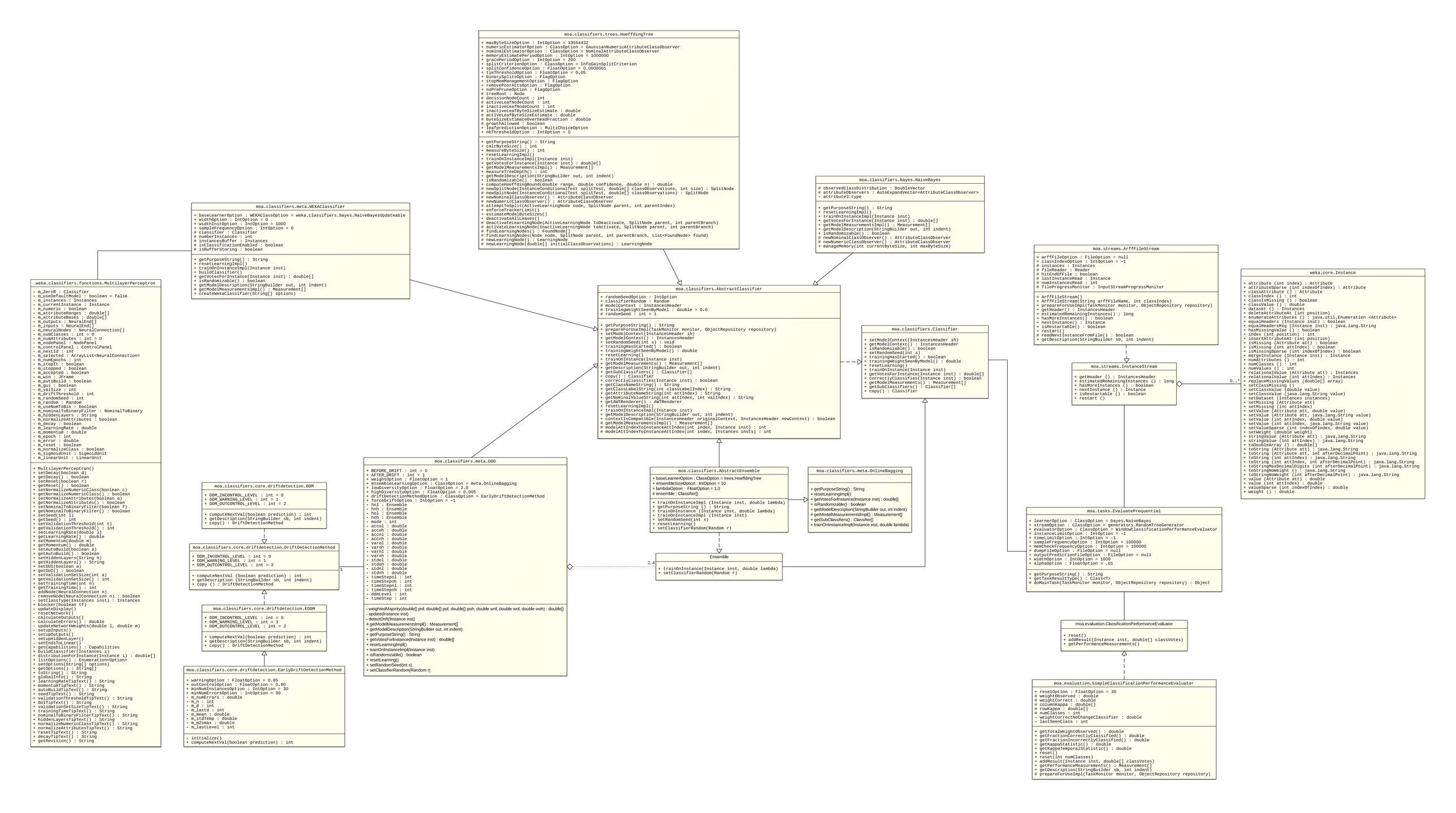
moa.classifiers.trees.HoeffdingTree
+ maxByteSizeOption : IntOption = 33554432
+ numericEstimatorOption : ClassOption = GaussianNumericAttributeClassObserver
+ nominalEstimatorOption : ClassOption = NominalAttributeClassObserver
+ memoryEstimatePeriodOption : IntOption = 1000000
+ gracePeriodOption : IntOption = 200
+ splitCriterionOption : ClassOption = InfoGainSplitCriterion
+ splitConfidenceOption : FloatOption = 0.0000001
+ tieThresholdOption : FloatOption = 0.05
+ binarySplitsOption : FlagOption
+ stopMemManagementOption : FlagOption
+ removePoorAttsOption : FlagOption
+ noPrePruneOption : FlagOption
# treeRoot : Node
# decisionNodeCount : int
# activeLeafNodeCount : int
# inactiveLeafNodeCount : int
# inactiveLeafByteSizeEstimate : double
# activeLeafByteSizeEstimate : double
# byteSizeEstimateOverheadFraction : double
# growthAllowed : boolean
+ leafprediction0ption : MultiChoiceOption
+ nbThresholdOption : IntOption = 0
+ getPurposeString() : String
+ calcByteSize() : int
+ measureByteSize() : int
+ resetLearningImpl()
+ trainOnInstanceImpl(Instance inst)
+ getVotesForInstance(Instance inst) : double[]
+ getModelMeasurementsImpl() : Measurement[]
+ measureTreeDepth() : int
+ getModelDescription(StringBuilder out, int indent)
+ isRandomizable() : boolean
+ computeHoeffdingBound(double range, double confidence, double n) : double
# newSplitNode(InstanceConditionalTest splitTest, double[] classObservations, int size) : SplitNode
# newSplitNode(InstanceConditionalTest splitTest, double[] classObservations) : SplitNode
# newNominalClassObserver() : AttributeClassObserver
# newNumericClassObserver() : AttributeClassObserver
# attemptToSplit(ActiveLearningNode node, SplitNode parent, int parentIndex)
+ enforceTrackerLimit()
+ estimateModelByteSizes()
+ deactivateAllLeaves()
# deactivateLearningNode(ActiveLearningNode toDeactivate, SplitNode parent, int parentBranch)
# activateLearningNode(InactiveLearningNode toActivate, SplitNode parent, int parentBranch)
# findLearningNodes() : FoundNode[]
# findLearningNodes(Node node, SplitNode parent, int parentBranch, List<FoundNode> found)
# newLearningNode() : LearningNode
# newLearningNode(double[] initialClassObservations) : LearningNode
moa.classifiers.bayes.NaiveBayes
observedClassDistribution : DoubleVector
attributeObservers : AutoExpandVector<AttributeClassObserver>
- attribute3:type
+ getPurposeString() : String
+ resetLearningImpl()
+ trainOnInstanceImpl(Instance inst)
+ getVotesForInstance(Instance inst) : double[]
+ getModelMeasurementsImpl() : Measurement[]
+ getModelDescription(StringBuilder out, int indent)
+ isRandomizable() : boolean
# newNominalClassObserver() : AttributeClassObserver
# newNumericClassObserver() : AttributeClassObserver
+ manageMemory(int currentByteSize, int maxByteSize)
moa.classifiers.meta.WEKAClassifier
+ baseLearnerOption : WEKAClassOption = weka.classifiers.bayes.NaiveBayesUpdateable
+ widthOption : IntOption = 0
+ widthInitOption : IntOption = 1000
+ sampleFrequencyOption : IntOption = 0
# classifier : Classifier
# numberInstances : int
# instancesBuffer : Instances
# isClassificationEnabled : boolean
# isBufferStoring : boolean
+ getPurposeString() : String
+ resetLearningImpl()
+ trainOnInstanceImpl(Instance inst)
+ buildClassifier()
+ getVotesForInstance(Instance inst) : double[]
+ isRandomizable() : boolean
+ getModelDescription(StringBuilder out, int indent)
# getModelMeasurementsImpl() : Measurement[]
+ createWekaClassifier(String[] options)
moa.classifiers.AbstractClassifier
+ randomSeedOption : IntOption
+ classifierRandom : Random
# modelContext : InstancesHeader
# trainingWeightSeenByModel : double = 0.0
# randomSeed : int = 1
+ getPurposeString() : String
+ prepareForUseImpl(TaskMonitor monitor, ObjectRepository repository)
+ setModelContext(InstancesHeader ih)
+ getModelContext() : InstancesHeader
+ setRandomSeed(int s) : void
+ trainingHasStarted() : boolean
+ trainingWeightSeenByModel() : double
+ resetLearning()
+ trainOnInstance(Instance inst)
+ getModelMeasurements() : Measurement[]
+ getDescription(StringBuilder out, int indent)
+ getSubClassifiers() : Classifier[]
+ copy() : Classifier
+ correctlyClassifies(Instance inst) : boolean
+ getClassNameString() : String
+ getClassLabelString(int classLabelIndex) : String
+ getAttributeNameString(int attIndex) : String
+ getNominalValueString(int attIndex, int valIndex) : String
+ getAWTRenderer() : AWTRenderer
+ resetLearningImpl()
+ trainOnInstanceImpl(Instance inst)
+ getModelDescription(StringBuilder out, int indent)
+ contextIsCompatible(InstancesHeader originalContext, InstancesHeader newContext) : boolean
# getModelMeasurementsImpl() : Measurement[]
# modelAttIndexToInstanceAttIndex(int index, Instance inst) : int
# modelAttIndexToInstanceAttIndex(int index, Instances insts) : int
moa.classifiers.Classifier
+ setModelContext(InstancesHeader ih)
+ getModelContext() : InstancesHeader
+ isRandomizable() : boolean
+ setRandomSeed(int s)
+ trainingHasStarted() : boolean
+ trainingWeightSeenByModel() : double
+ resetLearning()
+ trainOnInstance(Instance inst)
+ getVotesForInstance(Instance inst) : double[]
+ correctlyClassifies(Instance inst) : boolean
+ getModelMeasurements() : Measurement[]
+ getSubClassifiers() : Classifier[]
+ copy() : Classifier
moa.streams.ArffFileStream
+ arffFileOption : FileOption = null
+ classIndexOption : IntOption = -1
# instances : Instances
# fileReader : Reader
# hitEndOfFile : boolean
# lastInstanceRead : Instance
# numInstancesRead : int
# fileProgressMonitor : InputStreamProgressMonitor
+ ArffFileStream()
+ ArffFileStream(String arffFileName, int classIndex)
+ prepareForUseImpl(TaskMonitor monitor, ObjectRepository repository)
+ getHeader() : InstancesHeader
+ estimatedRemainingInstances() : long
+ hasMoreInstances() : boolean
+ nextInstance() : Instance
+ isRestartable() : boolean
+ restart()
# readNextInstanceFromFile() : boolean
+ getDescription(StringBuilder sb, int indent)
moa.streams.InstanceStream
+ getHeader () : InstancesHeader
+ estimatedRemainingInstances () : long
+ hasMoreInstances () : boolean
+ nextInstance () : Instance
+ isRestartable () : boolean
+ restart ()
0..*
weka.core.Instance
+ attribute (int index) : Attribute
+ attributeSparse (int indexOfIndex) : Attribute
+ classAttribute () : Attribute
+ classIndex () : int
+ classIsMissing () : boolean
+ classValue () : double
+ dataset () : Instances
+ deleteAttributeAt (int position)
+ enumerateAttributes () : java.util.Enumeration <Attribute>
+ equalHeaders (Instance inst) : boolean
+ equalHeadersMsg (Instance inst) : java.lang.String
+ hasMissingValue () : boolean
+ index (int position) : int
+ insertAttributeAt (int position)
+ isMissing (Attribute att) : boolean
+ isMissing (int attIndex) : boolean
+ isMissingSparse (int indexOfIndex) : boolean
+ mergeInstance (Instance inst) : Instance
+ numAttributes () : int
+ numClasses () : int
+ numValues () : int
+ relationalValue (Attribute att) : Instances
+ relationalValue (int attIndex) : Instances
+ replaceMissingValues (double[] array)
+ setClassMissing ()
+ setClassValue (double value)
+ setClassValue (java.lang.String value)
+ setDataset (Instances instances)
+ setMissing (Attribute att)
+ setMissing (int attIndex)
+ setValue (Attribute att, double value)
+ setValue (Attribute att, java.lang.String value)
+ setValue (int attIndex, double value)
+ setValue (int attIndex, java.lang.String value)
+ setValueSparse (int indexOfIndex, double value)
+ setWeight (double weight)
+ stringValue (Attribute att) : java.lang.String
+ stringValue (int attIndex) : java.lang.String
+ toDoubleArray () : double[]
+ toString (Attribute att) : java.lang.String
+ toString (Attribute att, int afterDecimalPoint) : java.lang.String
+ toString (int attIndex) : java.lang.String
+ toString (int attIndex, int afterDecimalPoint) : java.lang.String
+ toStringMaxDecimalDigits (int afterDecimalPoint) : java.lang.String
+ toStringNoWeight () : java.lang.String
+ toStringNoWeight (int afterDecimalPoint) : java.lang.String
+ value (Attribute att) : double
+ value (int attIndex) : double
+ valueSparse (int indexOfIndex) : double
+ weight () : double
weka.classifiers.functions.MultilayerPerceptron
- m_ZeroR : Classifier
- m_useDefaultModel : boolean = false
- m_instances : Instances
- m_currentInstance : Instance
- m_numeric : boolean
- m_attributeRanges : double[]
- m_attributeBases : double[]
- m_outputs : NeuralEnd[]
- m_inputs : NeuralEnd[]
- m_neuralNodes : NeuralConnection[]
- m_numClasses : int = 0
- m_numAttributes : int = 0
- m_nodePanel : NodePanel
- m_controlPanel : ControlPanel
- m_nextId : int
- m_selected : ArrayList<NeuralConnection>
- m_numEpochs : int
- m_stopIt : boolean
- m_stopped : boolean
- m_accepted : boolean
- m_win : JFrame
- m_autoBuild : boolean
- m_gui : boolean
- m_valSize : int
- m_driftThreshold : int
- m_randomSeed : int
- m_random : Random
- m_useNomToBin : boolean
- m_nominalToBinaryFilter : NominalToBinary
- m_hiddenLayers : String
- m_normalizeAttributes : boolean
- m_decay : boolean
- m_learningRate : double
- m_momentum : double
- m_epoch : int
- m_error : double
- m_reset : boolean
- m_normalizeClass : boolean
- m_sigmoidUnit : SigmoidUnit
- m_linearUnit : LinearUnit
+ MultilayerPerceptron()
+ setDecay(boolean d)
+ getDecay() : boolean
+ setReset(boolean r)
+ getReset() : boolean
+ setNormalizeNumericClass(boolean c)
+ getNormalizeNumericClass() : boolean
+ setNormalizeAttributes(boolean a)
+ getNormalizeAttributes() : boolean
+ setNominalToBinaryFilter(boolean f)
+ getNominalToBinaryFilter() : boolean
+ setSeed(int i)
+ getSeed() : int
+ setValidationThreshold(int t)
+ getValidationThreshold() : int
+ setLearningRate(double l)
+ getLearningRate() : double
+ setMomentum(double m)
+ getMomentum() : double
+ setAutoBuild(boolean a)
+ getAutoBuild() : boolean
+ setHiddenLayers(String h)
+ getHiddenLayers() : String
+ setGUI(boolean a)
+ getGUI() : boolean
+ setValidationSetSize(int a)
+ getValidationSetSize() : int
+ setTrainingTime(int n)
+ getTrainingTime() : int
- addNode(NeuralConnection n)
- removeNode(NeuralConnection n) : boolean
- setClassType(Instances inst) : Instances
+ blocker(boolean tf)
- updateDisplay()
- resetNetwork()
- calculateOutputs()
- calculateErrors() : double
- updateNetworkWeights(double l, double m)
- setupInputs()
- setupOutputs()
- setupHiddenLayer()
- setEndsToLinear()
+ getCapabilities() : Capabilities
+ buildClassifier(Instances i)
+ distributionForInstance(Instance i) : double[]
+ listOptions() : Enumeration<Option>
+ setOptions(String[] options)
+ getOptions() : String[]
+ toString() : String
+ globalInfo() : String
+ learningRateTipText() : String
+ momentumTipText() : String
+ autoBuildTipText() : String
+ seedTipText() : String
+ validationThresholdTipText() : String
+ GUITipText() : String
+ validationSetSizeTipText() : String
+ trainingTimeTipText() : String
+ nominalToBinaryFilterTipText() : String
+ hiddenLayersTipText() : String
+ normalizeNumericClassTipText() : String
+ normalizeAttributesTipText() : String
+ resetTipText() : String
+ decayTipText() : String
+ getRevision() : String
moa.classifiers.meta.DDD
+ BEFORE_DRIFT : int = 0
+ AFTER_DRIFT : int = 1
+ weightOption : FloatOption = 1
+ ensembleLearningOption : ClassOption = meta.OnlineBagging
+ lowDiversityOption : FloatOption = 0.4
+ highDiversityOption : FloatOption = 0.005
+ driftDetectionMethodOption : ClassOption = EarlyDriftDetectionMethod
+ forceDriftOption : IntOption = -1
- hnl : Ensemble
- hnh : Ensemble
- hol : Ensemble
- hoh : Ensemble
- mode : int
- accol : double
- accoh : double
- accnl : double
- accnh : double
- varol : double
- varoh : double
- varnl : double
- varnh : double
- stdol : double
- stdoh : double
- stdnl : double
- stdnh : double
- timeStepol : int
- timeStepoh : int
- timeStepnl : int
- timeStepnh : int
- ddmLevel : int
- timeStep : int
- weightedMajority(double[] pnl, double[] pol, double[] poh, double wnl, double wol, double woh) : double[]
- update(Instance inst)
- detectDrift(Instance inst)
getModelMeasurementsImpl() : Measurement[]
+ getModelDescription(StringBuilder out, int indent)
+ getPurposeString() : String
+ getVotesForInstance(Instance inst) : double[]
+ resetLearningImpl()
+ trainOnInstanceImpl(Instance inst)
+ isRandomizable() : boolean
+ resetLearning()
+ setRandomSeed(int s)
+ setClassifierRandom(Random r)
moa.classifiers.AbstractEnsemble
+ baseLearnerOption : ClassOption = trees.HoeffdingTree
+ ensembleSizeOption : IntOption = 10
+ lambdaOption : FloatOption = 1.0
# ensemble : Classifier[]
+ trainOnInstanceImpl (Instance inst, double lambda)
+ getPurposeString () : String
+ trainOnInstance (Instance inst, double lambda)
+ trainOnInstanceImpl (Instance inst)
+ setRandomSeed(int s)
+ resetLearning()
+ setClassifierRandom(Random r)
moa.classifiers.meta.OnlineBagging
+ getPurposeString() : String
+ resetLearningImpl()
+ getVotesForInstance(Instance inst) : double[]
+ isRandomizable() : boolean
+ getModelDescription(StringBuilder out, int indent)
# getModelMeasurementsImpl() : Measurement[]
+ getSubClassifiers() : Classifier[]
+ trainOnInstanceImpl(Instance inst, double lambda)
Ensemble
+ trainOnInstance(Instance inst, double lambda)
+ setClassifierRandom(Random r)
2..4
moa.classifiers.core.driftdetection.DDM
+ DDM_INCONTROL_LEVEL : int = 0
+ DDM_WARNING_LEVEL : int = 1
+ DDM_OUTCONTROL_LEVEL : int = 2
+ computeNextVal(boolean prediction) : int
+ getDescription(StringBuilder sb, int indent)
+ copy() : DriftDetectionMethod
moa.classifiers.core.driftdetection.DriftDetectionMethod
+ DDM_INCONTROL_LEVEL : int = 0
+ DDM_WARNING_LEVEL : int = 1
+ DDM_OUTCONTROL_LEVEL : int = 2
+ computeNextVal (boolean prediction) : int
+ getDescription (StringBuilder sb, int indent)
+ copy () : DriftDetectionMethod
1
moa.classifiers.core.driftdetection.EDDM
+ DDM_INCONTROL_LEVEL : int = 0
+ DDM_WARNING_LEVEL : int = 1
+ DDM_OUTCONTROL_LEVEL : int = 2
+ computeNextVal(boolean prediction) : int
+ getDescription(StringBuilder sb, int indent)
+ copy() : DriftDetectionMethod
moa.classifiers.core.driftdetection.EarlyDriftDetectionMethod
+ warningOption : FloatOption = 0.95
+ outControlOption : FloatOption = 0.90
+ minNumInstancesOption : IntOption = 30
+ minNumErrorsOption : IntOption = 30
- m_numErrors : double
- m_n : int
- m_d : int
- m_lastd : int
- m_mean : double
- m_stdTemp : double
- m_m2smax : double
- m_lastLevel : int
- initialize()
+ computeNextVal(boolean prediction) : int
moa.tasks.EvaluatePrequential
+ learnerOption : ClassOption = bayes.NaiveBayes
+ streamOption : ClassOption = generators.RandomTreeGenerator
+ evaluatorOption : ClassOption = WindowClassificationPerformanceEvaluator
+ instanceLimitOption : IntOption = -1
+ timeLimitOption : IntOption = -1
+ sampleFrequencyOption : IntOption = 100000
+ memCheckFrequencyOption : IntOption = 100000
+ dumpFileOption : FileOption = null
+ outputPredictionFileOption : FileOption = null
+ widthOption : IntOption = 1000
+ alphaOption : FloatOption = .01
+ getPurposeString() : String
+ getTaskResultType() : Class<?>
# doMainTask(TaskMonitor monitor, ObjectRepository repository) : Object
moa.evaluation.ClassificationPerformanceEvaluator
+ reset()
+ addResult(Instance inst, double[] classVotes)
+ getPerformanceMeasurements()
moa.evaluation.SimpleClassificationPerformanceEvaluator
+ resetOption : FloatOption = 30
# weightObserved : double
# weightCorrect : double
# columnKappa : double[]
# rowKappa : double[]
# numClasses : int
- weightCorrectNoChangeClassifier : double
- lastSeenClass : int
+ getTotalWeightObserved() : double
+ getFractionCorrectlyClassified() : double
+ getFractionIncorrectlyClassified() : double
+ getKappaStatistic() : double
+ getKappaTemporalStatistic() : double
+ reset()
+ reset(int numClasses)
+ addResult(Instance inst, double[] classVotes)
+ getPerformanceMeasurements() : Measurement[]
+ getDescription(StringBuilder sb, int indent)
# prepareForUseImpl(TaskMonitor monitor, ObjectRepository repository)