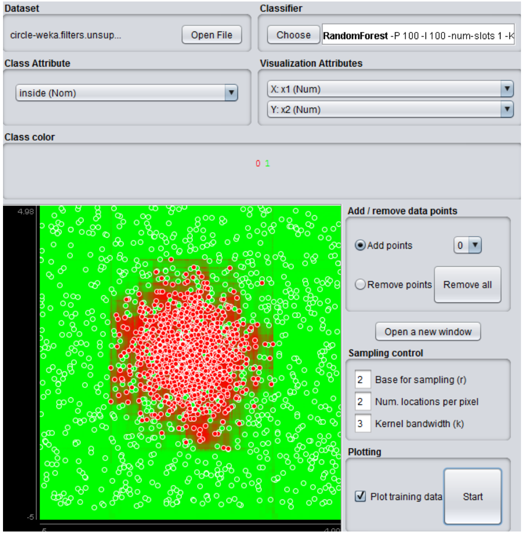
1. Dataset: Circle

- Classifier: Random Forest



Dataset: circle-weka.filters.unsup...

Classifier: RandomForest -P 100 -I 100 -num-slots 1 -K

Class Attribute: inside (Nom)

Visualization Attributes: X: x1 (Num) Y: x2 (Num)

Class color: 0 1

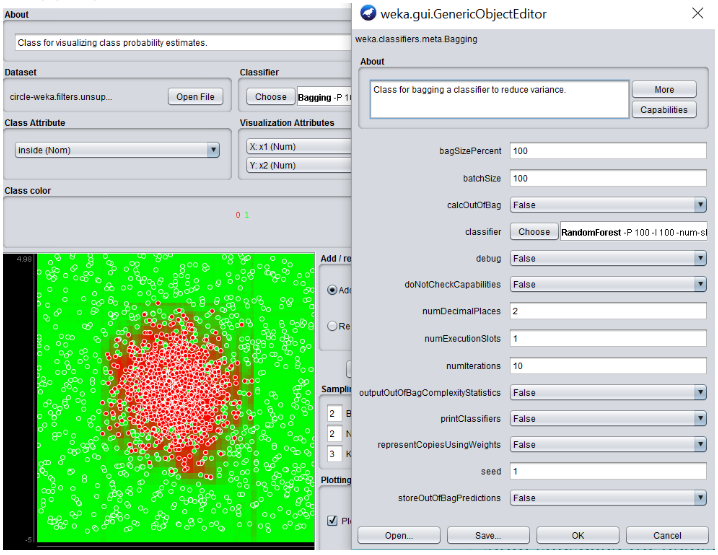
Add / remove data points: ● Add points 0 ○ Remove points

Sampling control:
2 Base for sampling (r)
2 Num. locations per pixel
3 Kernel bandwidth (k)

Plotting: ☑ Plot training data

Random Forest performs basically good.

- Bagging + Classifier: Random Forest



About: Class for visualizing class probability estimates.

Dataset: circle-weka.filters.unsup...

Classifier: Bagging -P 1

Class Attribute: inside (Nom)

Visualization Attributes: X: x1 (Num) Y: x2 (Num)

Class color: 0 1

weka.gui.GenericObjectEditor

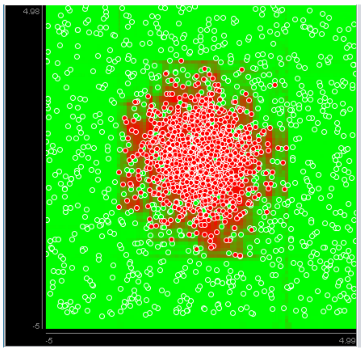
weka.classifiers.meta.Bagging

About: Class for bagging a classifier to reduce variance.

bagSizePercent 100
batchSize 100
calcOutOfBag False
classifier RandomForest -P 100 -I 100 -num-sl
debug False
doNotCheckCapabilities False
numDecimalPlaces 2
numExecutionSlots 1
numIterations 10
outputOutOfBagComplexityStatistics False
printClassifiers False
representCopiesUsingWeights False
seed 1
storeOutOfBagPredictions False

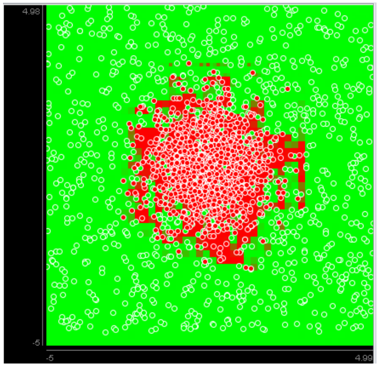
As can be seen, for Random Forest Classifier, adding bagging does not perform better, remaining almost the same.

- AdaBoost M1 + Classifier: Random Forest



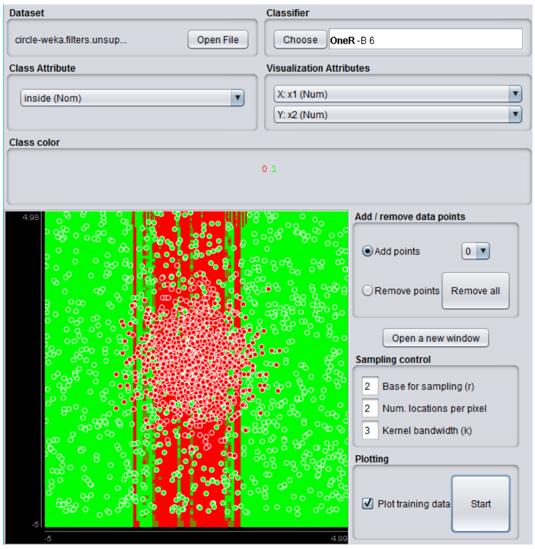
As can be seen, for Random Forest Classifier, adding AdaBoost seems perform a bit better or remaining the same. While, the running speed increases.

- LogitBoost + Classifier: Random Forest



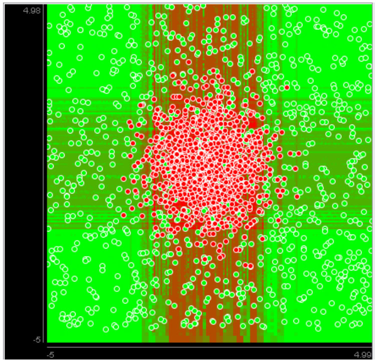
As can be seen, adding LogitBoost performs a bit worse for Random Forest Classifier.

- Classifier: OneR


Dataset
circle-weka.filters.unsup...
Classifier
OneR -B 6
Class Attribute
inside (Nom)
Visualization Attributes
X: x1 (Num)
Y: x2 (Num)
Class color
0 1
Add / remove data points
● Add points 0
○ Remove points
Sampling control
2 Base for sampling (r)
2 Num. locations per pixel
3 Kernel bandwidth (k)
Plotting
☑ Plot training data

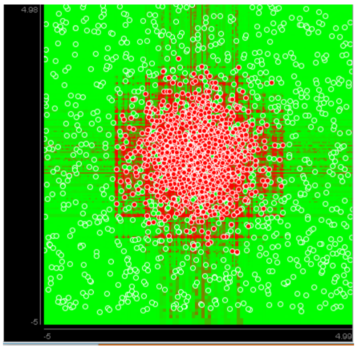
Basically, one vertical bar is selected for OneR.

- Bagging + Classifier: OneR



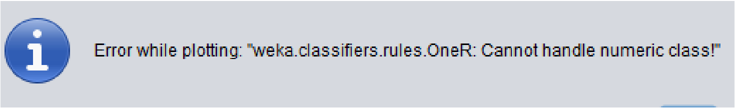
As can be seen, for OneR Classifier, adding bagging looks adding another dimension when classifying. OneR only selects a long band, while Bagging + OneR selects two orthogonal bands with a square in the center, performing better.

- AdaBoost + Classifier: OneR



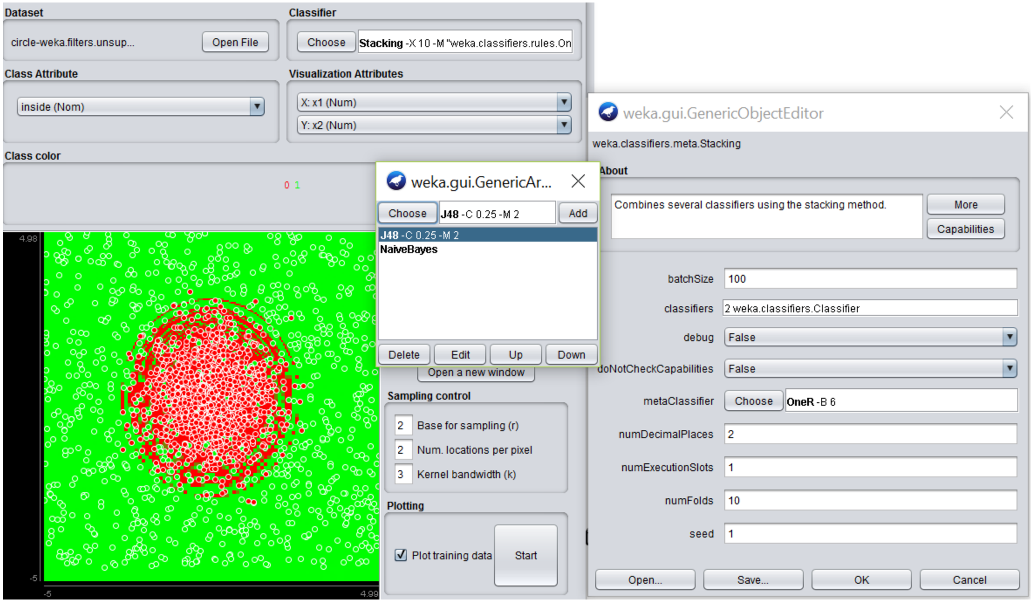
As can be seen, adding AdaBoost to OneR performs much better, and runs more quickly. The bias error of the model OneR decreases a lot.

- LogitBoost + Classifier: OneR



Error while plotting: "weka.classifiers.rules.OneR: Cannot handle numeric class!"

- Stacking: Decision Trees + NaïveBayes + metaclassifier OneR



Dataset: circle-weka.filters.unsup...

Classifier: Stacking -X 10 -M "weka.classifiers.rules.On

Class Attribute: inside (Nom)

Visualization Attributes: X: x1 (Num); Y: x2 (Num)

Class color: 0 1

weka.gui.GenericAr...

J48 -C 0.25 -M 2

NaiveBayes

Sampling control: 2 Base for sampling (r); 2 Num. locations per pixel; 3 Kernel bandwidth (k)

Plotting: ☑ Plot training data

weka.gui.GenericObjectEditor

weka.classifiers.meta.Stacking

About: Combines several classifiers using the stacking method.

batchSize: 100

classifiers: 2 weka.classifiers.Classifier

debug: False

doNotCheckCapabilities: False

metaClassifier: OneR -B 6

numDecimalPlaces: 2

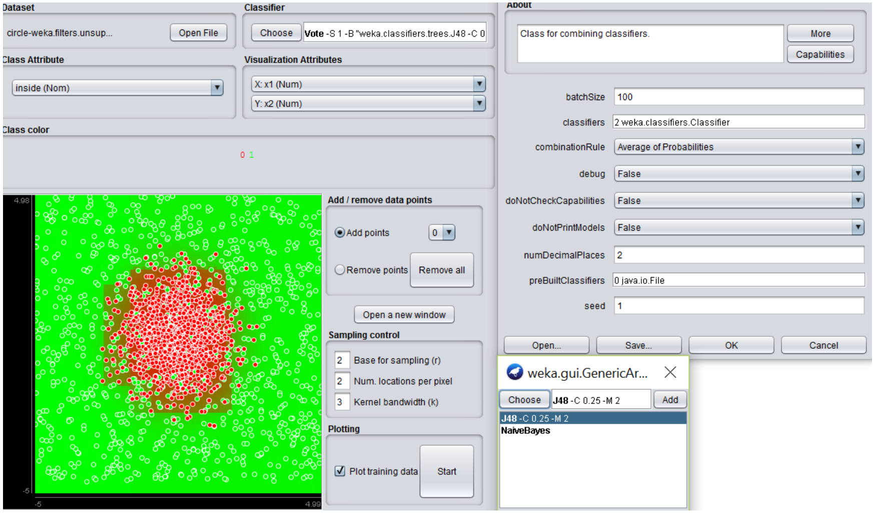
numExecutionSlots: 1

numFolds: 10

seed: 1

As can be seen, it selects an almost exact circle in the center. This performs much better.

- Vote: Decision Trees + NaiveBayes, Average of Probabilities



Dataset: circle-weka.filters.unsup...

Classifier: Vote -S 1 -B "weka.classifiers.trees.J48 -C 0

Class Attribute: inside (Nom)

Visualization Attributes: X: x1 (Num); Y: x2 (Num)

Class color: 0 1

Add / remove data points: ● Add points 0; ○ Remove points

Sampling control: 2 Base for sampling (r); 2 Num. locations per pixel; 3 Kernel bandwidth (k)

Plotting: ☑ Plot training data

About: Class for combining classifiers.

batchSize: 100

classifiers: 2 weka.classifiers.Classifier

combinationRule: Average of Probabilities

debug: False

doNotCheckCapabilities: False

doNotPrintModels: False

numDecimalPlaces: 2

preBuiltClassifiers: 0 java.io.File

seed: 1

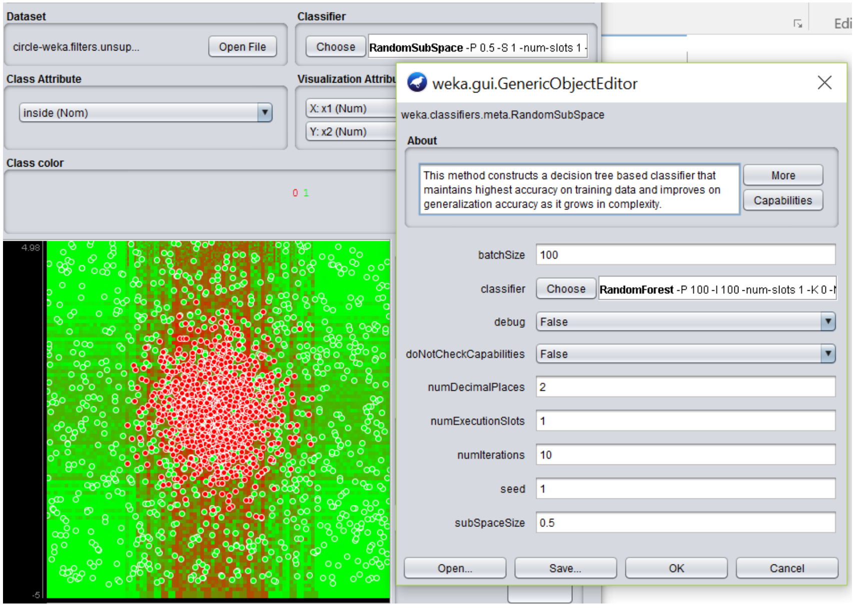
weka.gui.GenericAr...

J48 -C 0.25 -M 2

NaiveBayes

- As can be seen, the Vote result with combination rule of Average of Probabilities is not quite good. The selected areas consist of several overlapping blocks.

- RandomSubSpace: Random Forest



Dataset: circle-weka.filters.unsup... Open File

Classifier: Choose **RandomSubSpace** -P 0.5 -S 1 -num-slots 1 -

Class Attribute: inside (Nom)

Visualization Attributes: X: x1 (Num), Y: x2 (Num)

Class color: 0 1

weka.gui.GenericObjectEditor

weka.classifiers.meta.RandomSubSpace

About

This method constructs a decision tree based classifier that maintains highest accuracy on training data and improves on generalization accuracy as it grows in complexity.

batchSize: 100

classifier: Choose **RandomForest** -P 100 -I 100 -num-slots 1 -K 0 -I

debug: False

doNotCheckCapabilities: False

numDecimalPlaces: 2

numExecutionSlots: 1

numIterations: 10

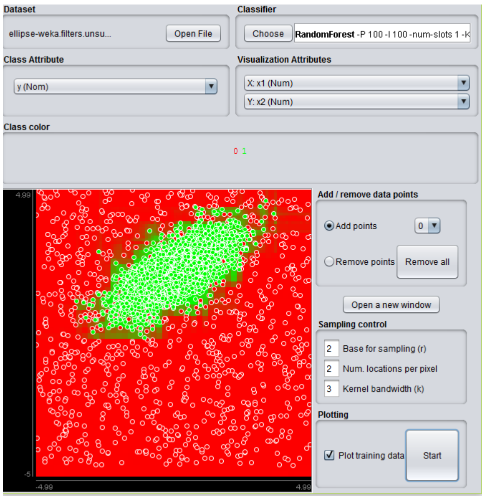
seed: 1

subSpaceSize: 0.5

RandomSubSpace is an attribute bagging. As can be seen, it basically selects a rough band, performs worse than Random Forest itself.

2. Dataset: Ellipse

- Classifier: Random Forest



Dataset: ellipse-weka.filters.unsu... Open File

Classifier: Choose **RandomForest** -P 100 -I 100 -num-slots 1 -K

Class Attribute: y (Nom)

Visualization Attributes: X: x1 (Num), Y: x2 (Num)

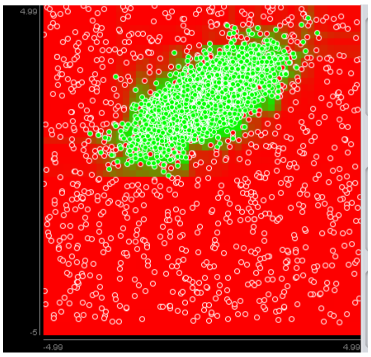
Class color: 0 1

Add / remove data points: ● Add points 0; ○ Remove points

Sampling control:
2 Base for sampling (r)
2 Num. locations per pixel
3 Kernel bandwidth (k)

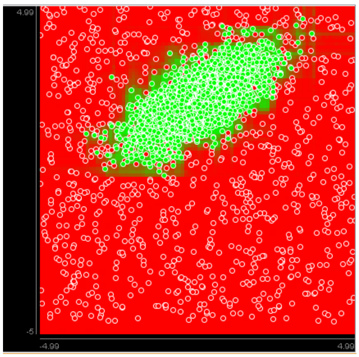
Plotting: ☑ Plot training data

- Bagging + Classifier: Random Forest



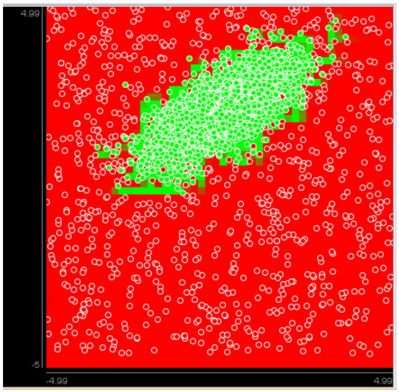
As can be seen, for Random Forest Classifier, adding bagging does not perform better, remaining almost the same. The running time is much longer.

- AdaBoost M1 + Classifier: Random Forest



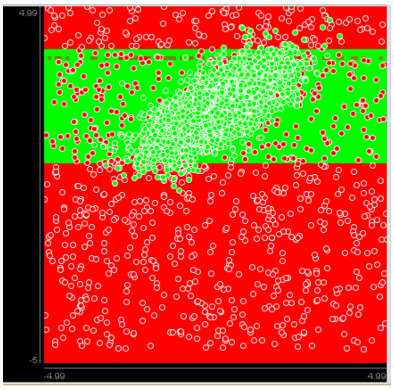
As can be seen, for Random Forest Classifier, adding AdaBoost seems perform a bit better or remaining the same. While, the running speed increases as well.

- LogitBoost + Classifier: Random Forest



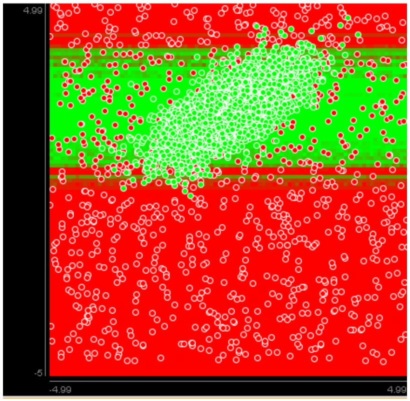
As can be seen, adding LogitBoost performs a bit worse for Random Forest Classifier.

- Classifier: OneR



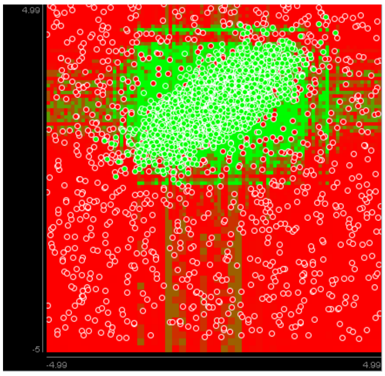
A horizontal bar is selected.

- Bagging + Classifier: OneR



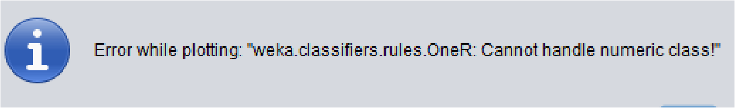
As can be seen, for OneR Classifier, adding bagging looks more fine-grained. OneR only selects a long band, while Bagging + OneR selects not exactly one horizontal bar.

- AdaBoost + Classifier: OneR



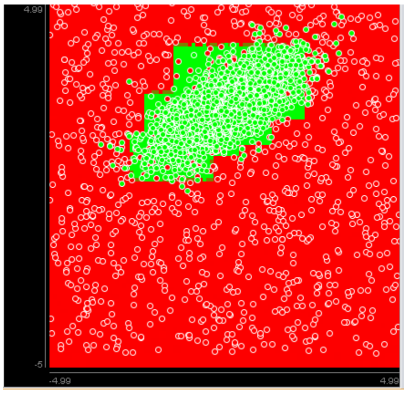
As can be seen, adding AdaBoost to OneR performs better, and runs more quickly. The bias error of the model OneR decreases a lot. The selected area looks like a square.

- LogitBoost + Classifier: OneR



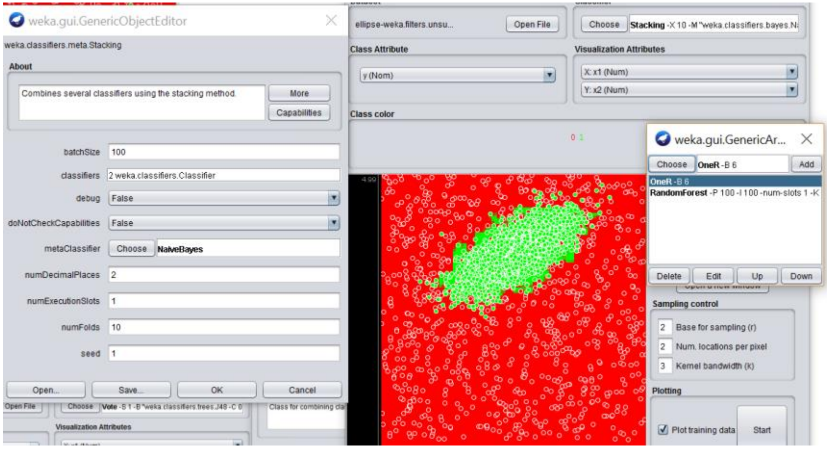
Error while plotting: "weka.classifiers.rules.OneR: Cannot handle numeric class!"

- Stacking: Decision Trees + NaïveBayes + metaclassifier OneR



As can be seen, it selects a ladder-like shape. This performs better.

- Stacking: OneR + Random Forest metaclassifier: NaiveBayes



weka.classifiers.meta.Stacking

About

Combines several classifiers using the stacking method.

batchSize: 100
classifiers: 2 weka.classifiers.Classifier
debug: False
doNotCheckCapabilities: False
metaClassifier: NaiveBayes
numDecimalPlaces: 2
numExecutionSlots: 1
numFolds: 10
seed: 1

ellipse-weka.filters.unsu...
Stacking -X 10 -M "weka.classifiers.bayes.N
y (Nom)
X: x1 (Num)
Y: x2 (Num)

OneR -B 6
RandomForest -P 100 -I 100 -num-slots 1 -K

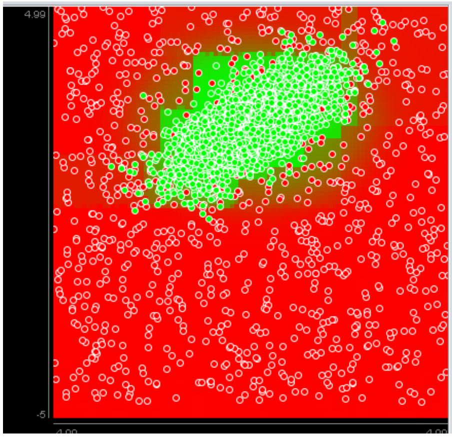
Vote -S 1 -B "weka.classifiers.trees.J48 -C 0

Sampling control
2 Base for sampling (r)
2 Num. locations per pixel
3 Kernel bandwidth (k)

Plotting
☑ Plot training data

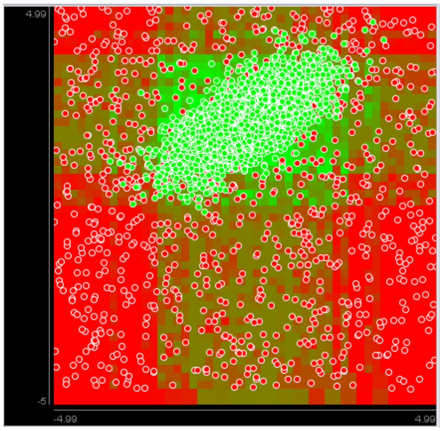
The above stacking of different classifiers performs better than the previous stacking.

- Vote: Decision Trees + NaiveBayes, Average of Probabilities



As can be seen, the Vote result with combination rule of Average of Probabilities is not quite good. The selected areas consist of several overlapping blocks, similar to stacking with majority voting rule.

- RandomSubSpace: Random Forest

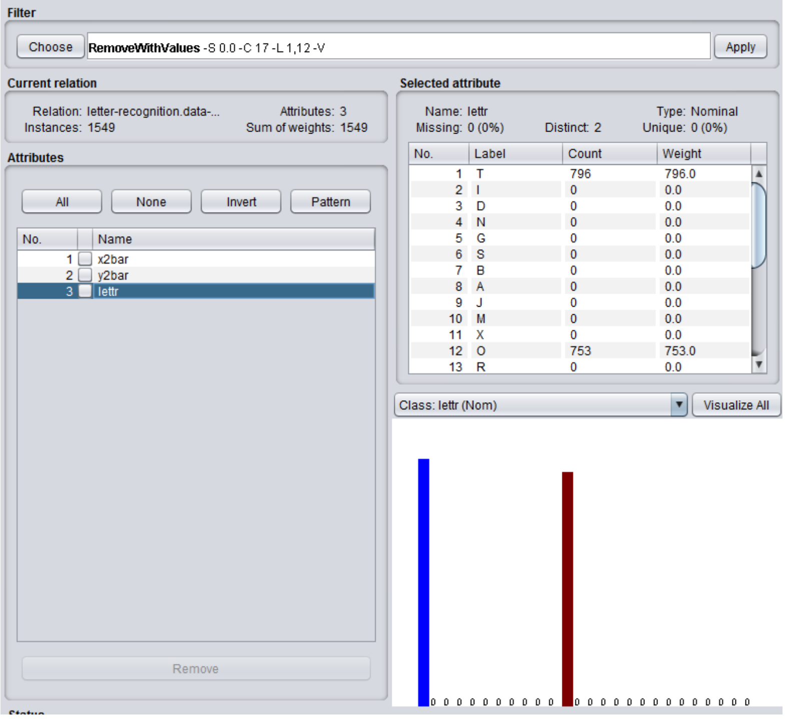


RandomSubSpace is an attribute bagging. As can be seen, it basically selects a rough square, performing much worse than Random Forest itself.

3. Dataset: letter-recognition.data (OCR)

Preprocessing:

I Only select letter “T” and “O”, and select two attributes “x2bar” and “y2bar”. The two attributes look easier to distinguish for the two letters.



Filter

RemoveWithValues -S 0.0 -C 17 -L 1,12 -V

Current relation

Relation: letter-recognition.data-... Attributes: 3
Instances: 1549 Sum of weights: 1549

Attributes

No.	Name
1	x2bar
2	y2bar
3	lettr

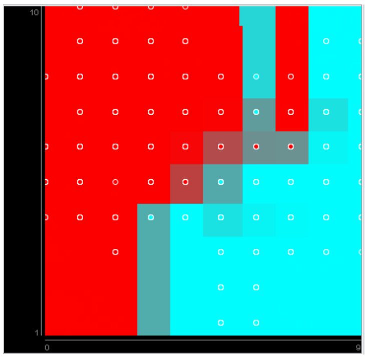
Selected attribute

Name: lettr Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	T	796	796.0
2	I	0	0.0
3	D	0	0.0
4	N	0	0.0
5	G	0	0.0
6	S	0	0.0
7	B	0	0.0
8	A	0	0.0
9	J	0	0.0
10	M	0	0.0
11	X	0	0.0
12	O	753	753.0
13	R	0	0.0

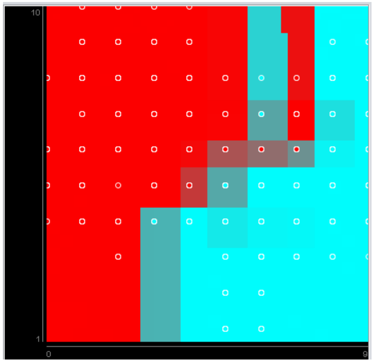
Class: lettr (Nom)

- Classifier: Random Forest



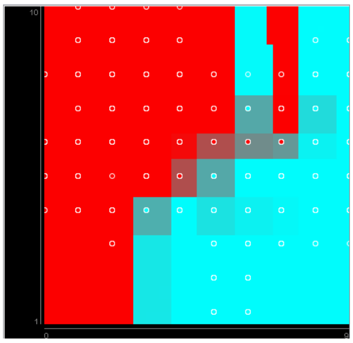
Random Forest performs basically good. Red points are almost in red part, and blue points are almost in blue part.

- Bagging + Classifier: Random Forest



As can be seen, for Random Forest Classifier, adding bagging performs the same. The values are all integers, so it should be easier to classify. The running time is much longer.

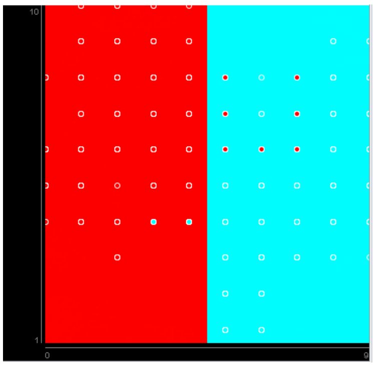
- AdaBoost M1 + Classifier: Random Forest



As can be seen, for Random Forest Classifier, adding AdaBoost performs the same. The values are all integers, so it should be easier to classify. While, the running time increases a lot.

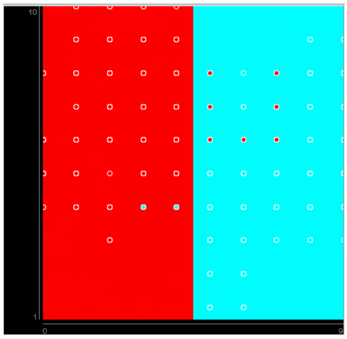
- LogitBoost + Classifier: Random Forest
  The running time is much longer, so I stop doing this one. Similarly, the result should remain almost the same as above.

- Classifier: OneR



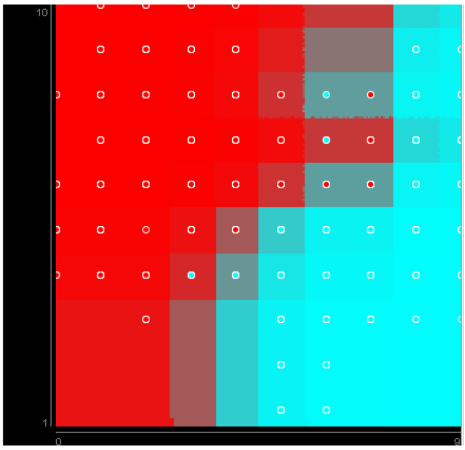
Basically, one vertical parting line is generated with OneR. There are several error points.

- Bagging + Classifier: OneR



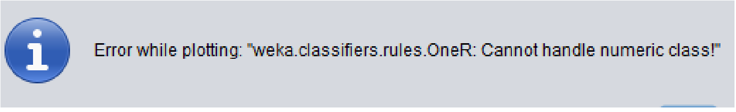
As can be seen, for OneR Classifier, adding bagging performs exactly the same.

- AdaBoost + Classifier: OneR



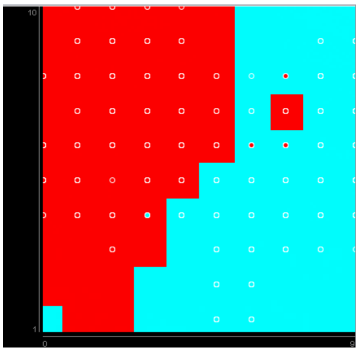
As can be seen, adding AdaBoost to OneR performs much better. The bias error of the model OneR decreases a lot. There are 4 to 5 points that are incorrectly classified.

- LogitBoost + Classifier: OneR



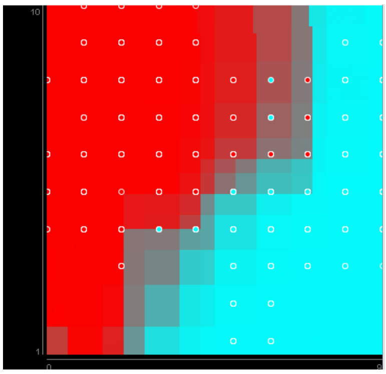
Error while plotting: "weka.classifiers.rules.OneR: Cannot handle numeric class!"

- Stacking: Decision Trees + NaïveBayes + metaclassifier OneR



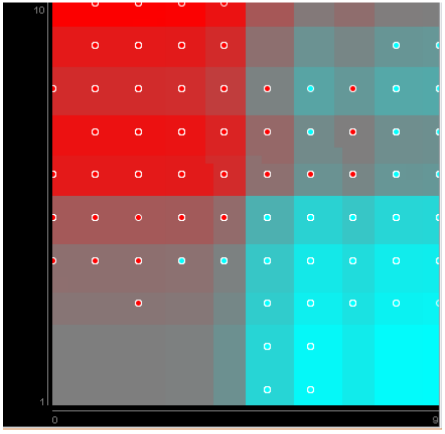
As can be seen, there are 4 points that are incorrectly classified with majority voting.

- Vote: Decision Trees + NaiveBayes, Average of Probabilities



As can be seen, the Vote result with combination rule of Average of Probabilities is basically good with some overlapping shadow. The selected areas consist of several overlapping blocks. There are 3 to 4 points that are incorrectly classified.

- RandomSubSpace: Random Forest



RandomSubSpace is an attribute bagging. As can be seen, it basically separates the whole into 4 parts including 2 shadow areas. It performs worse than Random Forest itself.

4. Dataset: names_ethnea_genni_country_sample

Preprocessing:

I only select PubCountry attribute as class attribute, and select Portugal and HongKong two countries. I select Last Name as the only attribute for classification.

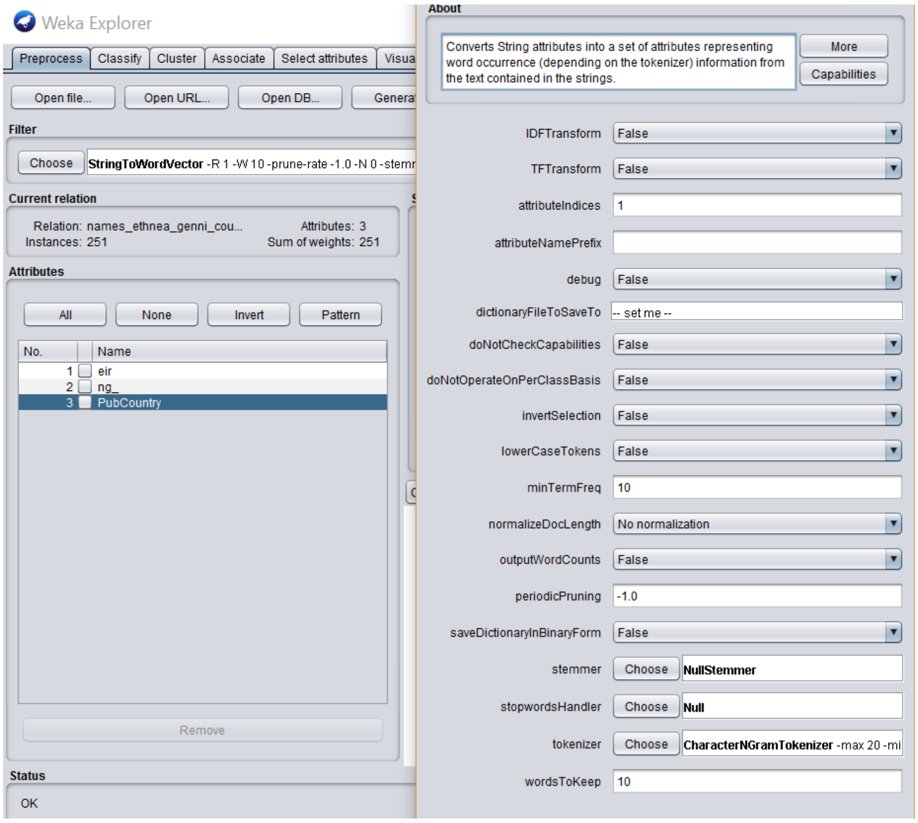
Filter: For First attribute, NominalToString -> StringToWordVector

CharacterNGramTokenizer -> set NGramMinSize and NGramMaxSize

Set minTermFreq to 10.

Set WordsToKeep to 10.

Select “ng_” and “eir”.



StringToWordVector -R 1 -W 10 -prune-rate -1.0 -N 0 -stemr

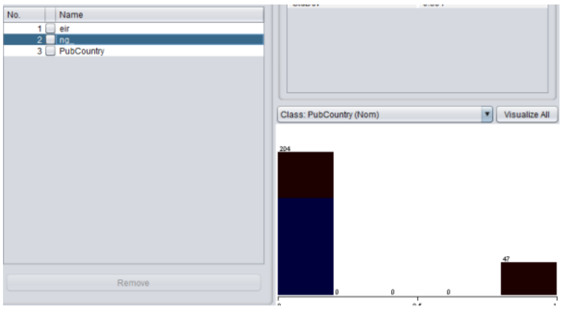
Relation: names_ethnea_genni_cou... Attributes: 3
Instances: 251 Sum of weights: 251

No.	Name
1	eir
2	ng_
3	PubCountry

OK

Converts String attributes into a set of attributes representing word occurrence (depending on the tokenizer) information from the text contained in the strings.

IDFTransform	False
TFTransform	False
attributeIndices	1
attributeNamePrefix	
debug	False
dictionaryFileToSaveTo	-- set me --
doNotCheckCapabilities	False
doNotOperateOnPerClassBasis	False
invertSelection	False
lowerCaseTokens	False
minTermFreq	10
normalizeDocLength	No normalization
outputWordCounts	False
periodicPruning	-1.0
saveDictionaryInBinaryForm	False
stemmer	NullStemmer
stopwordsHandler	Null
tokenizer	CharacterNGramTokenizer -max 20 -mi
wordsToKeep	10

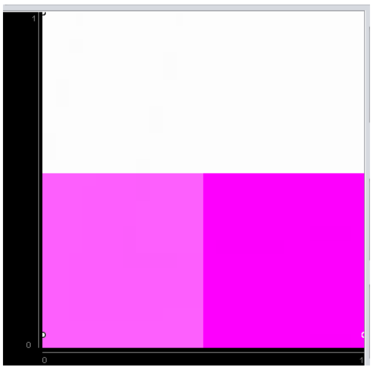


No.	Name
1	eir
2	ng_
3	PubCountry

Class: PubCountry (Nom)

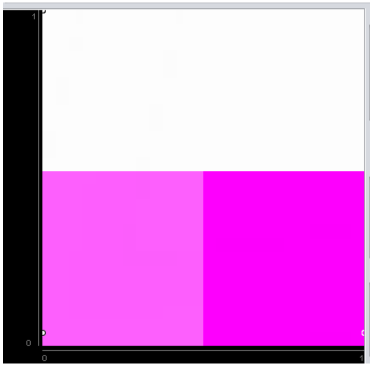
For “eir” and “ng_”, instances only have numeric values “0” and “1”.

- Classifier: Random Forest



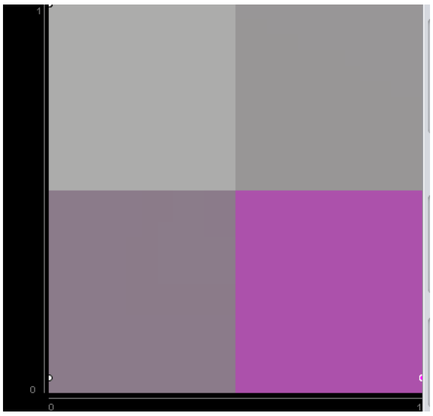
The two points are located at the left and right edges. There is not much difference between the color of the two blocks. It is not a good classification.

- Bagging + Classifier: Random Forest



As can be seen, for Random Forest Classifier, adding bagging performs the same. The running time is much longer. It is not a good classification either.

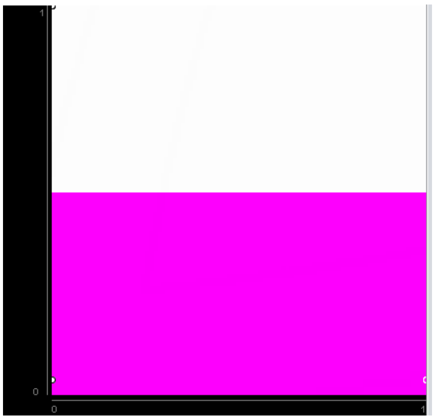
- AdaBoost M1 + Classifier: Random Forest



As can be seen, for Random Forest Classifier, adding AdaBoost produces 4 blocks. But it is not a good classification either. The two points are located at the left and right edges. There is not much difference between the color of the two blocks. The running time is much longer.

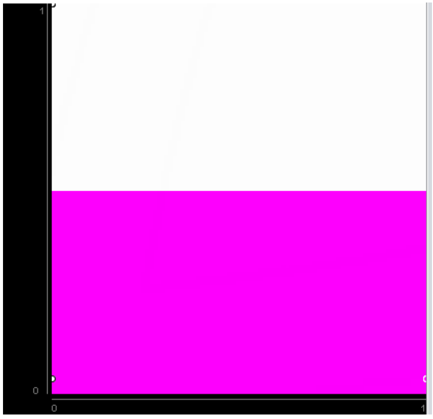
- LogitBoost + Classifier: Random Forest
  The running time is much longer, so I stop doing this one. Similarly, the result should remain almost the same as above.

- Classifier: OneR



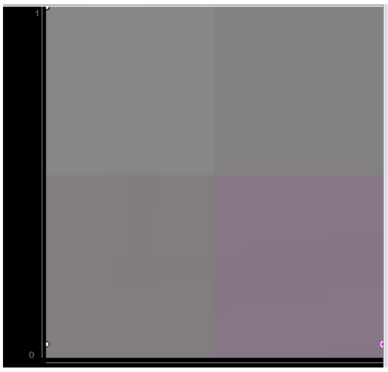
Basically, one vertical parting line is generated with OneR. The two points are not separated at all.

- Bagging + Classifier: OneR



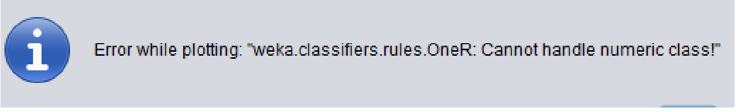
As can be seen, for OneR Classifier, adding bagging performs exactly the same.

- AdaBoost + Classifier: OneR



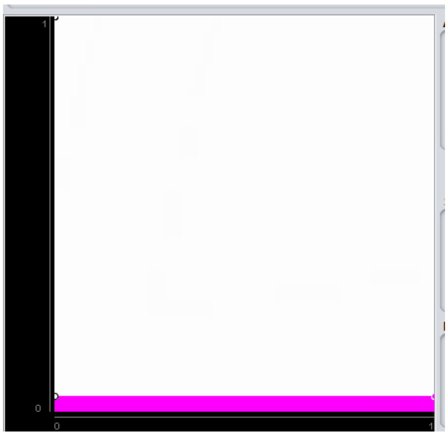
As can be seen, adding AdaBoost to OneR performs better. The two points are separated in two different blocks. But there is not much difference between the colors of the two blocks.

- LogitBoost + Classifier: OneR



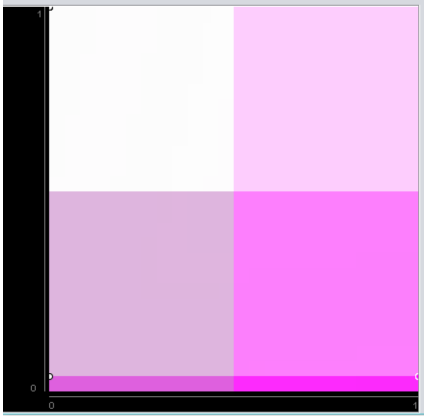
Error while plotting: "weka.classifiers.rules.OneR: Cannot handle numeric class!"

- Stacking: Decision Trees + NaïveBayes + metaclassifier OneR



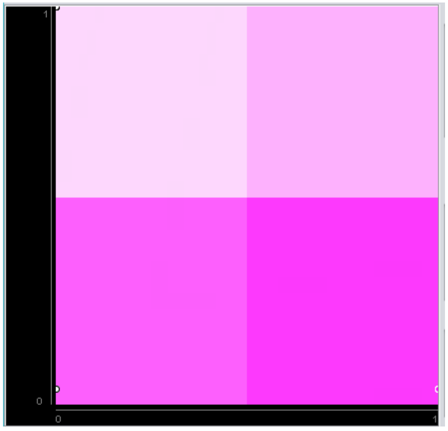
The two points are not separated at all.

- Vote: Decision Trees + NaiveBayes, Average of Probabilities



The two points are separated in two different blocks of different colors. But there is not much difference between the colors. It is not a really good classification.

- RandomSubSpace: Random Forest



RandomSubSpace is an attribute bagging. The two points are separated in two different blocks of different colors. But there is not much difference between the colors. It is not a really good classification.

5 General Conclusion

In general, Random Forest performs better than OneR, consuming longer time as well. (Random Forest: Bagging + random subspace for decision trees)

Bias is the model error, and variance represents training set error (whether representative or not).

For bagging, it performs better together with unstable classifiers than the classifiers themselves. Those classifiers include decision trees. Generally, bagging will decrease the variance error in some cases, however, bagging will not change the performance in other cases.

Boosting (not work for regression) probably performs better when adding to some classifiers. It will decrease bias error in most cases.

LogitBoost is additive logistic regression. Hard to explain.

Stacking method is combining models of different types. The default in Weka is majority voting. However, for voting with rule of average probability, some shadow areas will be generated based on the majority voting result.