Ensemble Methods

Data Mining & Analytics

Prof. Zach Pardos

INFO254/154: Spring ’19

Classifier selection

Features: credit score & income

Target: predicting if a user will be approved for a credit card

Which classifier?

A. Decision Tree
B. Linear regression
C. Neural Network?

Classifier selection

Features: Today’s temperature, last year’s temperature, the temperature two years ago

Target: predicting tomorrow’s temperature

Which classifier?

A. Decision Tree
B. Linear regression
C. Neural Network?

Classifier selection

Features: natural language text of a review, age of reviewer

Target: reviewer’s rating of the restaurant

Which classifier?

A. Decision Tree
B. Linear regression
C. Neural Network?

Ensemble Methods: Intuition

Why choose when you could use them all?

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

Example

- Seeking medical advice from multiple doctors (combination of experts)
- Averaging crowd predictions such as counting jelly beans in jar (crowd wisdom)

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

Example

- Digital communication across a noisy channel

1	1	1	1	1	1	1	1

- Assume **30%** noise (bits will be flipped)
- Error is uniformly distributed

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

Example

- Transmission errors in binary communication

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0

- For two transmissions:
    - What is the probability of both bits being correct?

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

Example

- Transmission errors in binary communication

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0

- For two transmissions:
    - What is the probability of two bits being correct? 0.49 (.7*.7)

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- For three transmissions
    - all three transmissions will be correct: 34.29% of the time ($.7^3$)

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- For three transmissions
  - all three transmissions will be correct: 34.29% of the time ($.7^3$)
  - two transmissions will be correct: 44.1% of the time (3 choose 1) * .30*.70^2

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- For three transmissions
  - all three transmissions will be correct: 34.29% of the time ($.7^3$)
  - two transmissions will be correct: 44.1% of the time (3 choose 1) * .30*.70^2
  - the majority will be correct: what percentage of the time (on average)?

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- For three transmissions
  - all three transmissions will be correct: 34.29% of the time ($.7^3$)
  - two transmissions will be correct: 44.1% of the time (3 choose 1) * .30*.70^2
  - the majority will be correct: **78.38%** (34.29 + 44.09)

Ensemble Methods: Intuition

78.38% > 70.0%!

(multiple transmissions has reduced the error)

Ensemble Methods: Intuition

78.38% > 70.0%!

(multiple transmissions has reduced the error)

Why this works:

- Transmission error is under 50%
- Error is uniformly distributed (almost never the case in ML classifiers)

Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

original	1	1	1	1	1	1	1	1
Transmission 1	1	0	1	1	1	0	1	1
Transmission 2	1	1	1	1	1	1	1	0
Transmission 3	0	1	0	1	1	1	1	0

- What’s the expected average of each column as transmission count goes to Inf?
- Will this eliminate all noise when taking the majority class?
    - Probability all classes will be wrong = (0.3)^n
    - Probability majority is wrong:

      ∑ for i = n/2 to n: (n choose i) * (.30^i)*.70^(n-i)

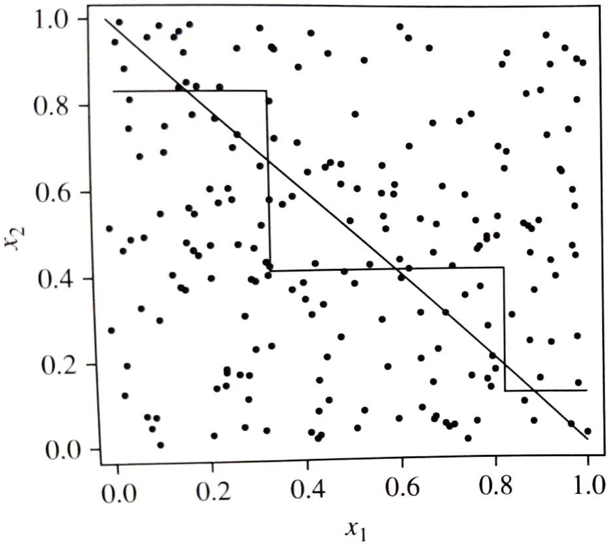
Ensemble Methods: Intuition

Primary ensemble methods: simple combination, bagging, boosting

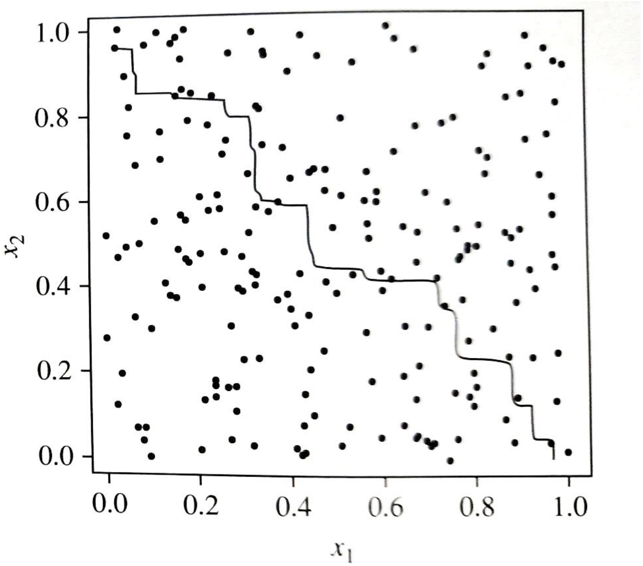
Ensembles achieve higher accuracies by

- Incorporating information from diverse but good predictors/classifiers
- This often creates a combined classifier that better generalizes

linearly separable test set with one DT



classification with a DT ensemble



Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

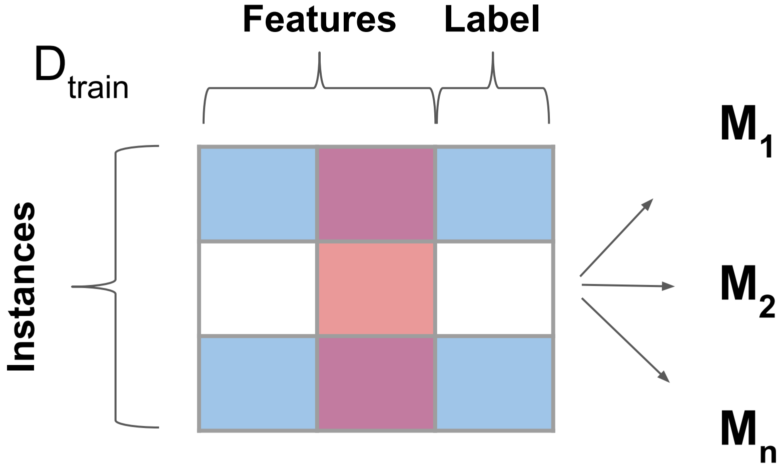
Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (1) Train N different models using a training set, D_{train}



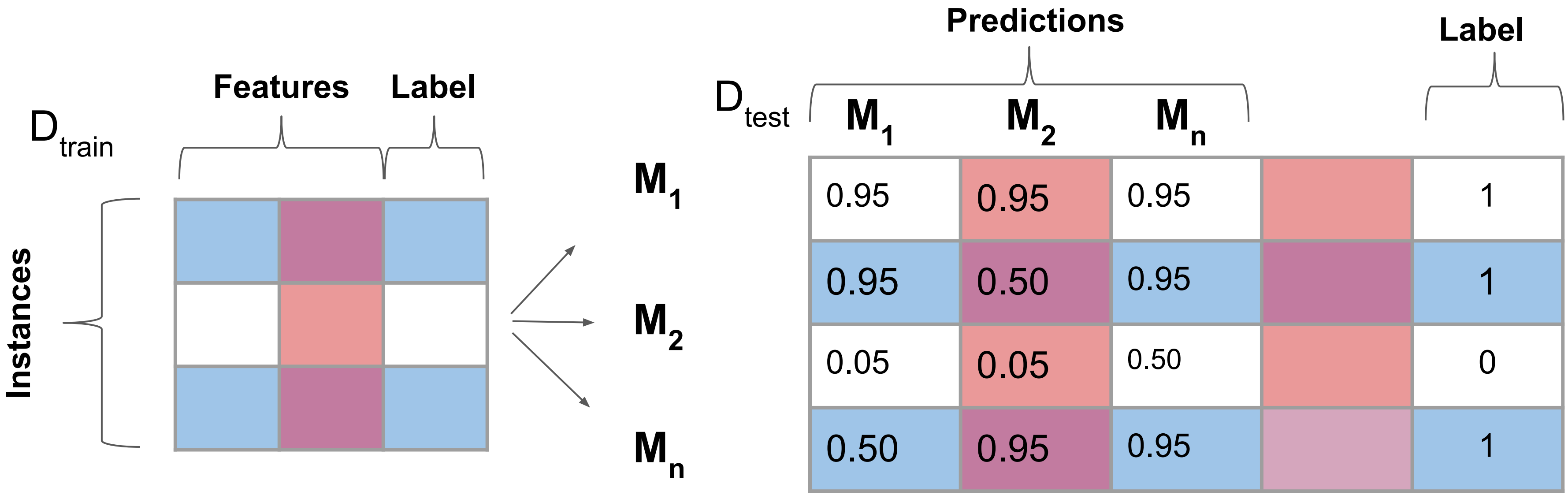
Choice of models can differ in any way including Algorithm and Hyperparameters

Ensemble Methods

Primary ensemble methods: **simple combination**, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (2) Each model makes a prediction of the label for D_{test}



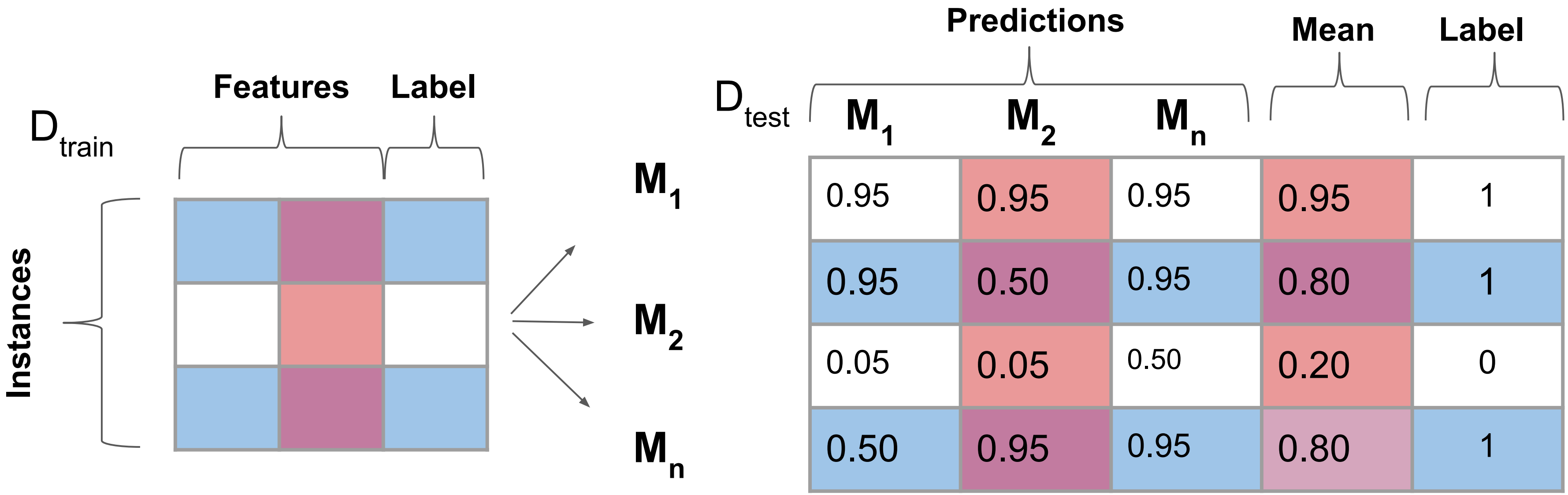
M_1	M_2	M_n		Label
0.95	0.95	0.95		1
0.95	0.50	0.95		1
0.05	0.05	0.50		0
0.50	0.95	0.95		1

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (3) Blend each model's predictions on D_{test} using a combiner



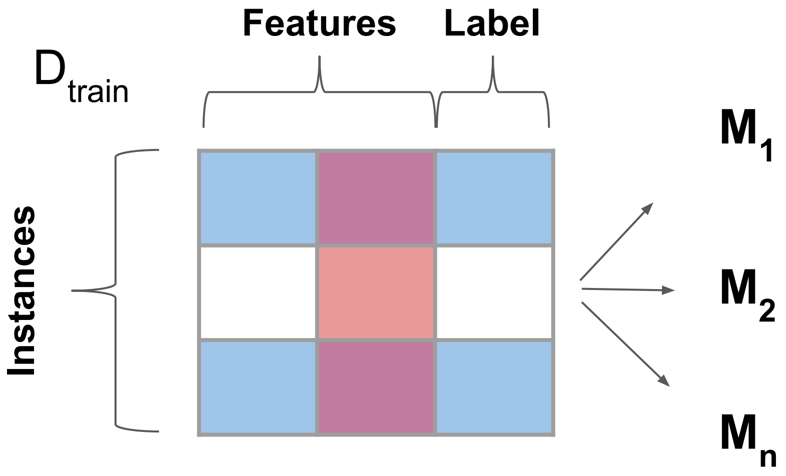
Predictions			Mean	Label
M_1	M_2	M_n		
0.95	0.95	0.95	0.95	1
0.95	0.50	0.95	0.80	1
0.05	0.05	0.50	0.20	0
0.50	0.95	0.95	0.80	1

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (3) Blend each model's predictions on D_{test} using a combiner



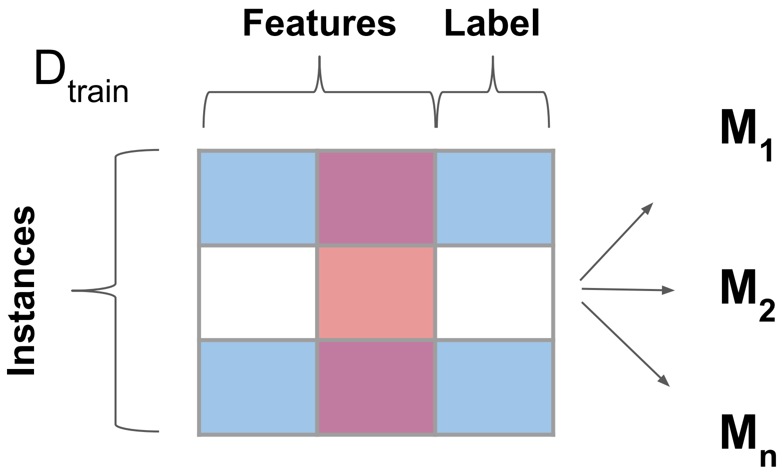
	Predictions			Mean	Label
D_{test}	M_1	M_2	M_n		
	0.95	0.95	0.95	0.95	1
	0.95	0.50	0.95	0.80	1
	0.05	0.05	0.50	0.20	0
	0.50	0.95	0.95	0.80	1
RMSE:	0.2537	0.2537	0.2537		

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (3) Blend each model's predictions on D_{test} using a combiner



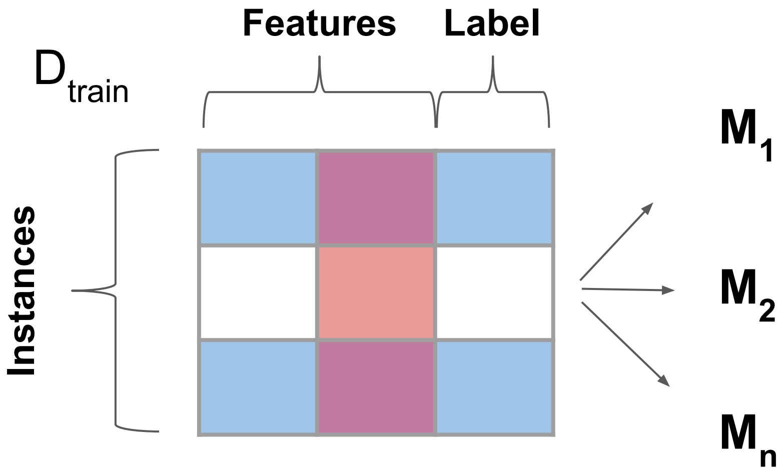
	Predictions			Mean	Label
D_{test}	M_1	M_2	M_n		
	0.95	0.95	0.95	0.95	1
	0.95	0.50	0.95	0.80	1
	0.05	0.05	0.50	0.20	0
	0.50	0.95	0.95	0.80	1
RMSE:	0.2537	0.2537	0.2537	0.1750	

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Step (3) Blend each model's predictions on D_{test} using a combiner



D_{test}	Predictions			Mean	Label
	M_1	M_2	M_n		
	0.95	0.95	0.95	0.95	1
	0.95	0.50	0.95	0.80	1
	0.05	0.05	0.50	0.20	0
	0.50	0.95	0.95	0.80	1
RMSE:	0.2537	0.2537	0.2537	0.1750	

Ensemble Methods

Primary ensemble methods: **simple combination**, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Max (confidence) is the prediction furthest from 0.50

	Predictions			**Mean**	**Max**	**Min**	**Label**
D_{test}	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_n}$				
	0.95	0.95	0.95	0.95			1
	0.95	0.50	0.95	0.80			1
	0.05	0.05	0.50	0.20			0
	0.50	0.95	0.95	0.80			1
RMSE:	0.2537	0.2537	0.2537	0.1750			

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Max (confidence) is the prediction furthest from 0.50

	Predictions			Mean	Max	Min	Label
D_{test}	M_1	M_2	M_n				
	0.95	0.95	0.95	0.95	0.95	0.95	1
	0.95	0.50	0.95	0.80	0.95	0.50	1
	0.05	0.05	0.50	0.20	0.05	0.50	0
	0.50	0.95	0.95	0.80	0.95	0.50	1
RMSE:	0.2537	0.2537	0.2537	0.1750	0.05	0.4337	

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

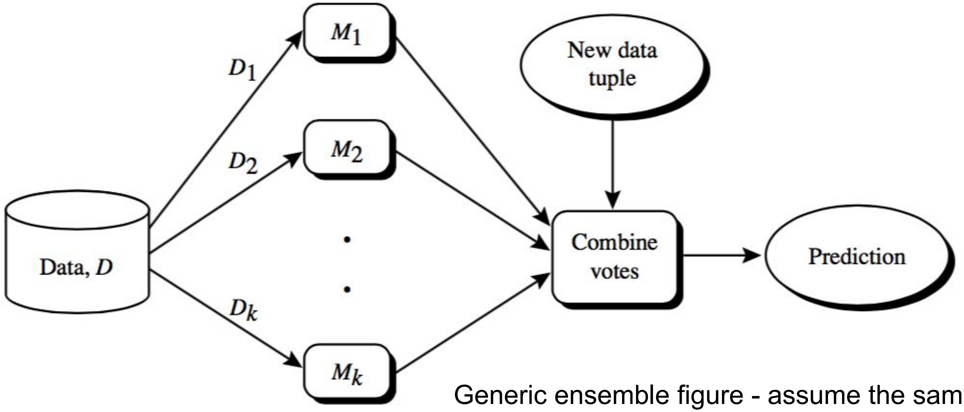
Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Max (confidence) is the prediction furthest from 0.50

	Predictions			**Mean**	**Max**	**Min**	**Label**
D_{test}	$\mathbf{M_1}$	$\mathbf{M_2}$	$\mathbf{M_n}$				
	0.95	0.95	0.95	0.95	0.95	0.95	1
	0.95	0.50	0.95	0.80	0.95	0.50	1
	0.05	0.05	0.50	0.20	0.05	0.50	0
	0.50	0.95	0.95	0.80	0.95	0.50	1
RMSE:	0.2537	0.2537	0.2537	0.1750	0.05	0.4337	

Simple combination: summary

- A composite model
- Combines a series of models with an non-trainable combiner with the aim of creating an improved classification model
- Especially effective when models are good (low bias) but differ from one another (not highly correlated)



Generic ensemble figure - assume the same *D* for all models with the simple combiner.

Han, Kamber, Pei (2011), Sec 8.6.1

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Diversifying a model by bootstrapping the training set and averaging the predictions (aka bootstrapped aggregation)

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Diversifying a model by bootstrapping the training set and averaging the predictions (aka bootstrapped aggregation)

- Each Model in the ensemble is trained on a bootstrapped sample of the data
- More robust to noisy data & overfitting
- Easy source of model diversity
- When the label is a class, the majority vote is used instead of averaging

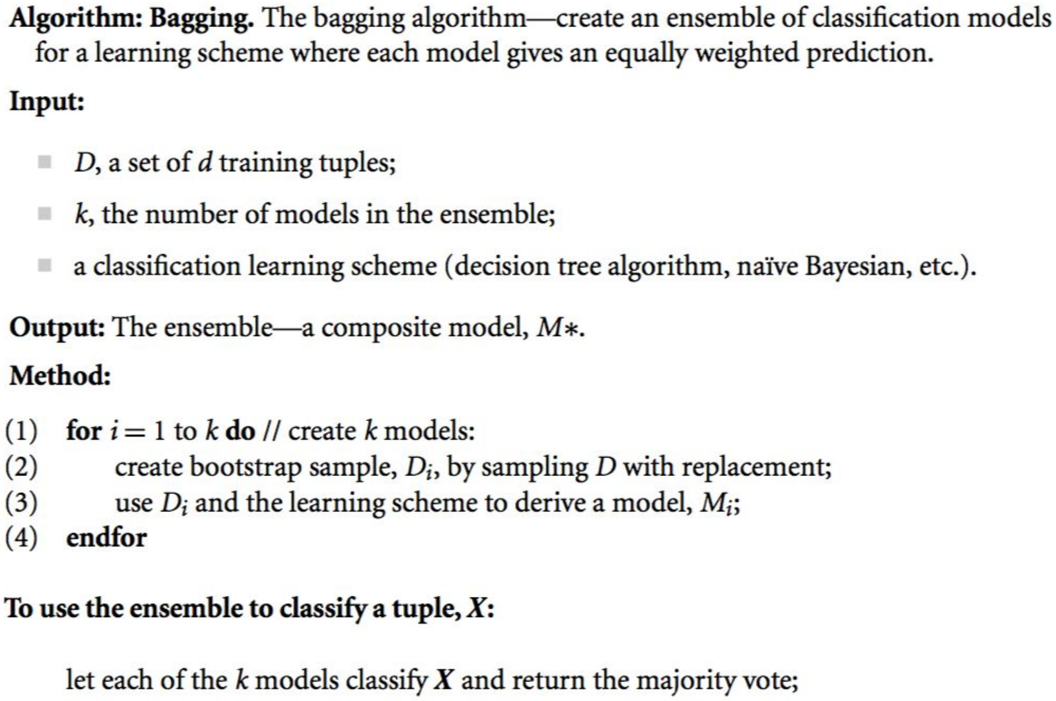
Han, Kamber, Pei (2011), Sec 8.6.2

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Diversifying a model by bootstrapping the training set and taking the majority vote (aka bootstrapped aggregation)

- Each Model in the ensemble is trained on a bootstrapped sample of the data
- More robust to noisy data & overfitting
- Easy source of model diversity
- When the label is a class, the majority vote is used instead of averaging



Algorithm: Bagging. The bagging algorithm—create an ensemble of classification models for a learning scheme where each model gives an equally weighted prediction.

Input:

- D, a set of d training tuples;
- k, the number of models in the ensemble;
- a classification learning scheme (decision tree algorithm, naïve Bayesian, etc.).

Output: The ensemble—a composite model, $M*$.

Method:

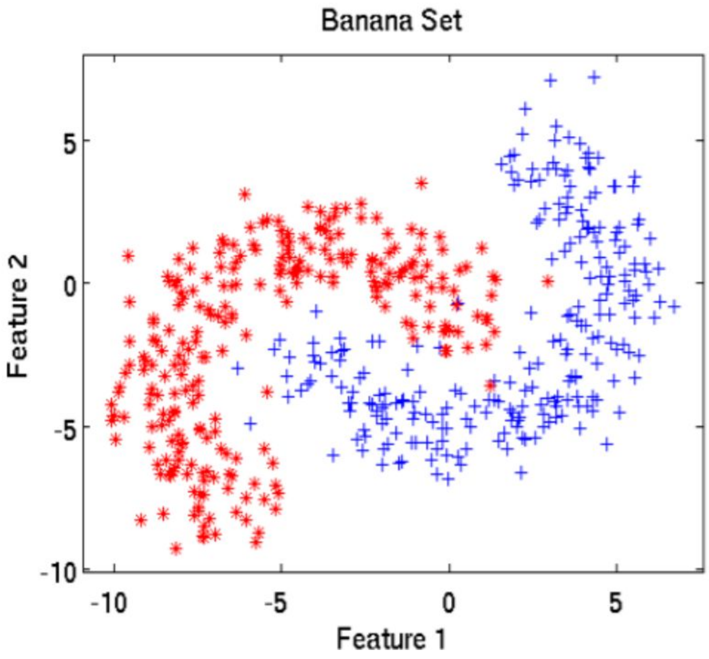
```
for $i = 1$ to $k$ do // create $k$ models:
    create bootstrap sample, $D_i$, by sampling $D$ with replacement;
    use $D_i$ and the learning scheme to derive a model, $M_i$;
endfor
```

To use the ensemble to classify a tuple, X:

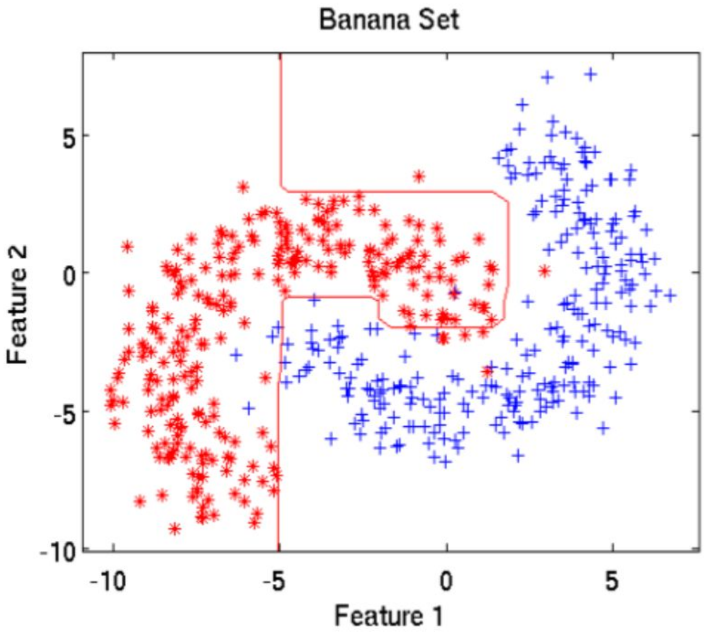
let each of the k models classify $\boldsymbol{X}$ and return the majority vote;

Han, Kamber, Pei (2011), Sec 8.6.2

Bagging DT example



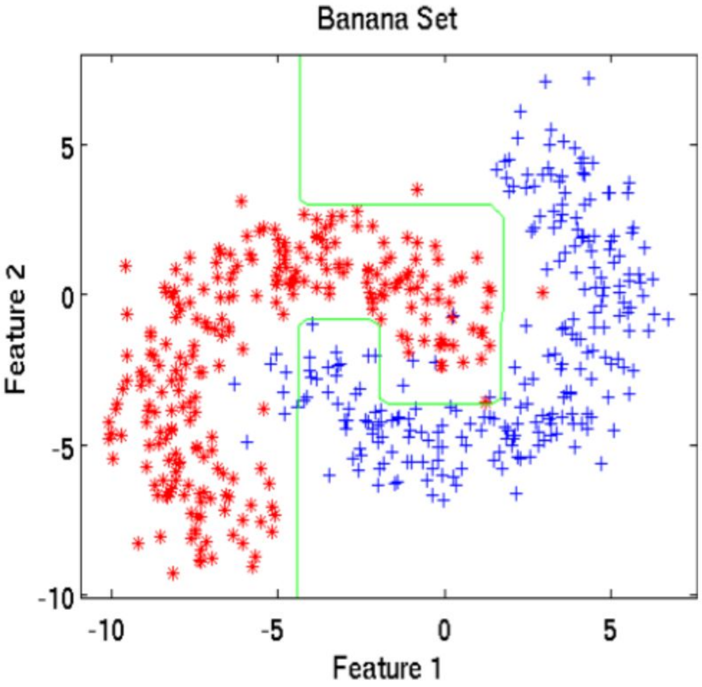
Training Data



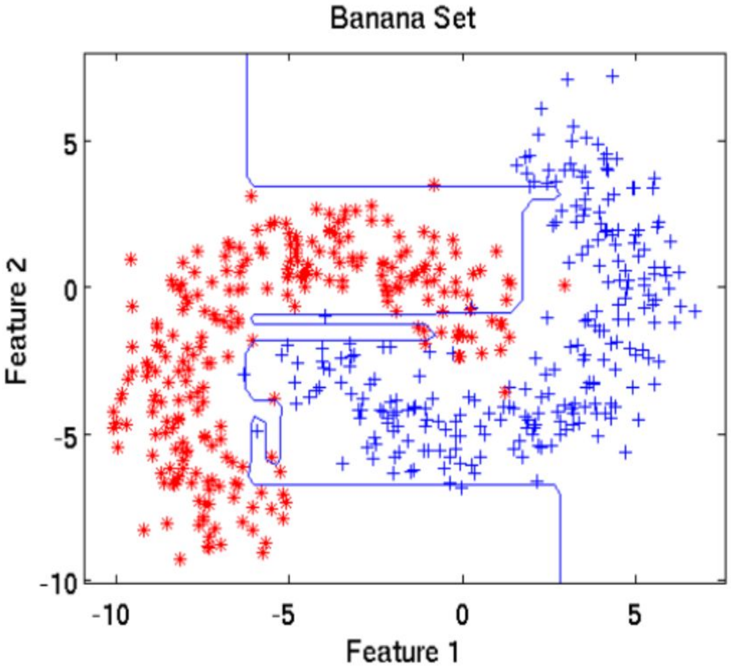
1 Decision Tree

*(https://www.cs.rit.edu/~rlaz/prec20092/slides/Bagging_and_Boosting.pdf)

Bagging DT example



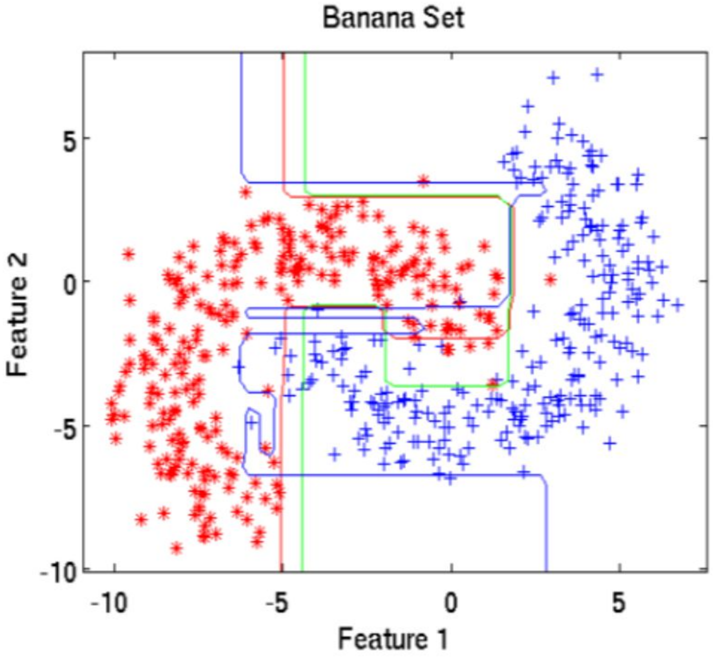
2 Decision Trees



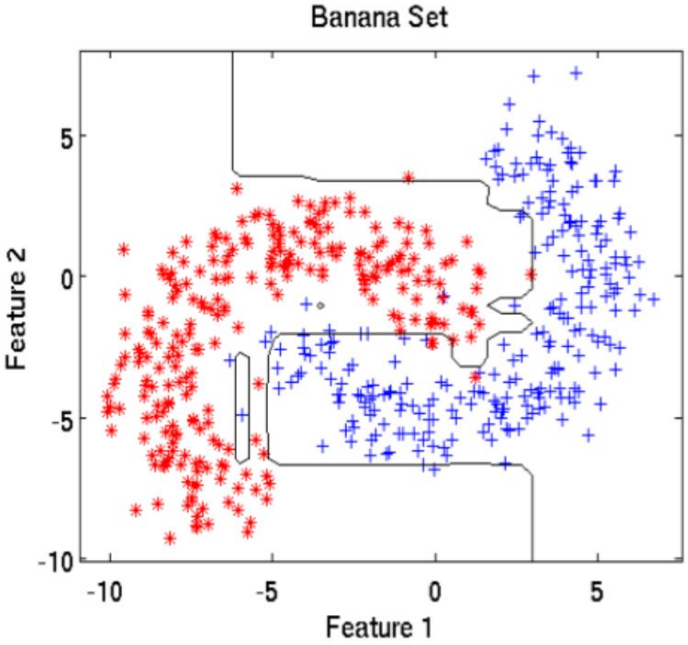
3 Decision Trees

*(https://www.cs.rit.edu/~rlaz/prec20092/slides/Bagging_and_Boosting.pdf)

Bagging DT example



3 Decision Trees with the decision boundary overlaid



Final Result

*(https://www.cs.rit.edu/~rlaz/prec20092/slides/Bagging_and_Boosting.pdf)

Random Forests

- An ensemble of decision trees
- Each decision tree is trained on a bootstrapped sample of the data
- Each decision tree is trained on a random subset of the features from the sample
- Comparable accuracy to Adaboost but more robust
- Accuracy depends on the strength of independent classifiers and diversity in error among them
- It’s fast; Each tree can be trained and used for testing in parallel
- Can operate in regression (continuous value prediction) or classification mode
- Predictions of the trained trees are averaged (regression) or majority vote is used (classification)

Invented by Leo Breiman, the same inventor of CART decision trees (using info gain)

Random Forests: Hyper-parameters

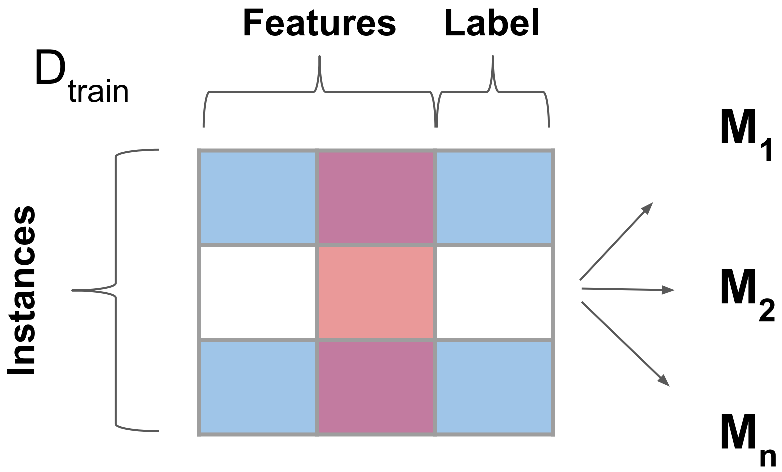
- Number of Trees
- Percentage of features sampled
- Percentage of total rows sampled
- Max Depth
- Min Leaf

Ensemble Methods

Primary ensemble methods: simple combination, bagging, boosting

Combining predictions through simple averaging or other non-trainable combiner (mean, min, max)

Is simple combination a form of bagging?



D_{test}

Predictions			Mean	Label
M_1	M_2	M_n		
0.95	0.95	0.95	0.95	1
0.95	0.50	0.95	0.80	1
0.05	0.05	0.50	0.20	0
0.50	0.95	0.95	0.80	1

Ensemble Methods

Primary ensemble methods: simple combination, bagging, **boosting**

Resampling the data based on classifier error

Ensemble Methods

Primary ensemble methods: simple combination, bagging, **boosting**

Resampling the data based on classifier error

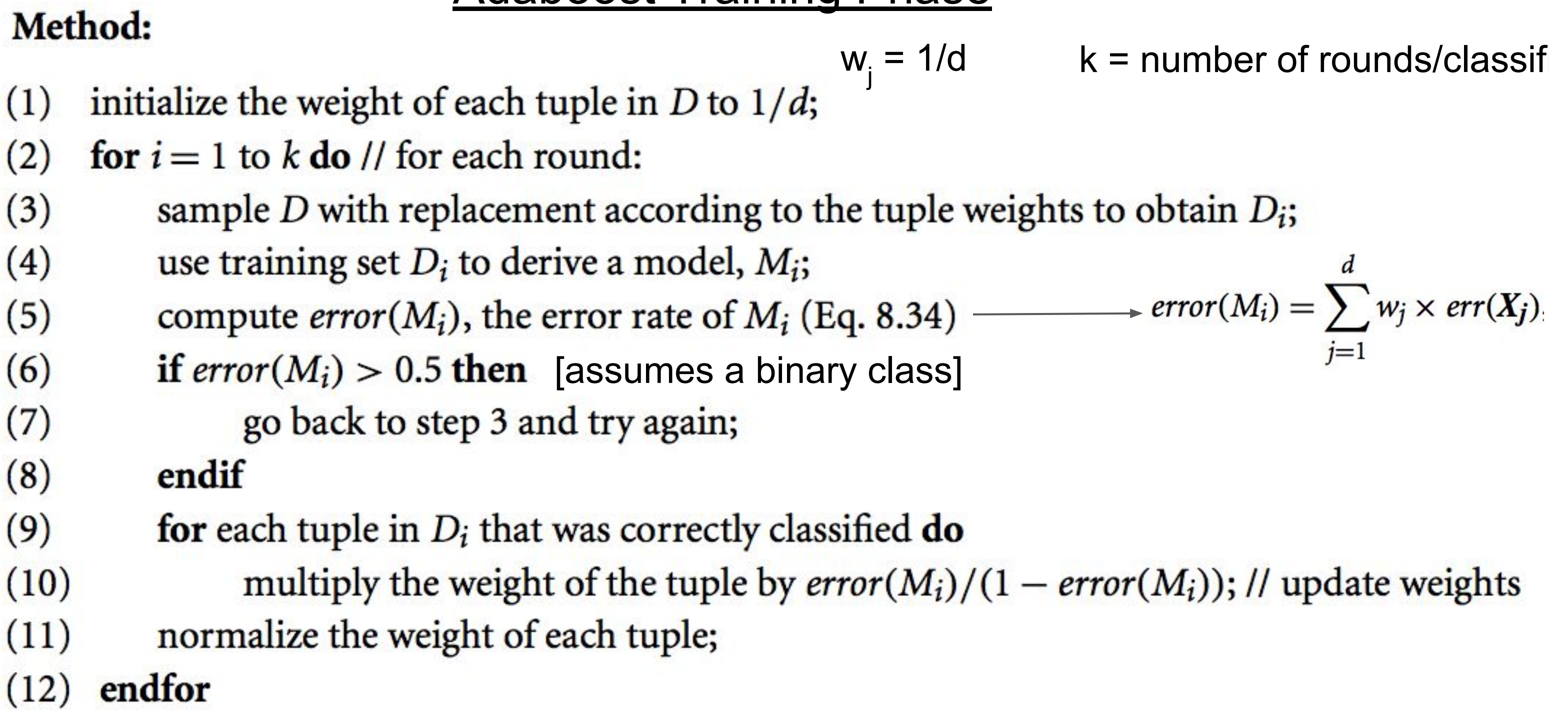
- Tuples (data) are “weighted”
- Each classifier is iteratively learned and the tuples that were misclassified by the previous model are favored for re-sampling in the next iteration
- Final classification is based on the weighted combined vote of all models where weight is based on its accuracy on the training set

Ensemble Methods

Primary ensemble methods: simple combination, bagging, **boosting**

Resampling the data based on classifier error

Adaboost Training Phase



$w_j = 1/d$ k = number of rounds/classifiers

Method:

```
(1)  initialize the weight of each tuple in $D$ to $1/d$;
(2)  **for** $i = 1$ to $k$ **do** // for each round:
(3)      sample $D$ with replacement according to the tuple weights to obtain $D_i$;
(4)      use training set $D_i$ to derive a model, $M_i$;
(5)      compute $error(M_i)$, the error rate of $M_i$ (Eq. 8.34) ⟶ $error(M_i) = \sum_{j=1}^{d} w_j \times err(\mathbf{X}_j)$;
(6)      **if** $error(M_i) > 0.5$ **then**  [assumes a binary class]
(7)          go back to step 3 and try again;
(8)      **endif**
(9)      **for** each tuple in $D_i$ that was correctly classified **do**
(10)         multiply the weight of the tuple by $error(M_i)/(1 - error(M_i))$; // update weights
(11)     normalize the weight of each tuple;
(12) **endfor**
```

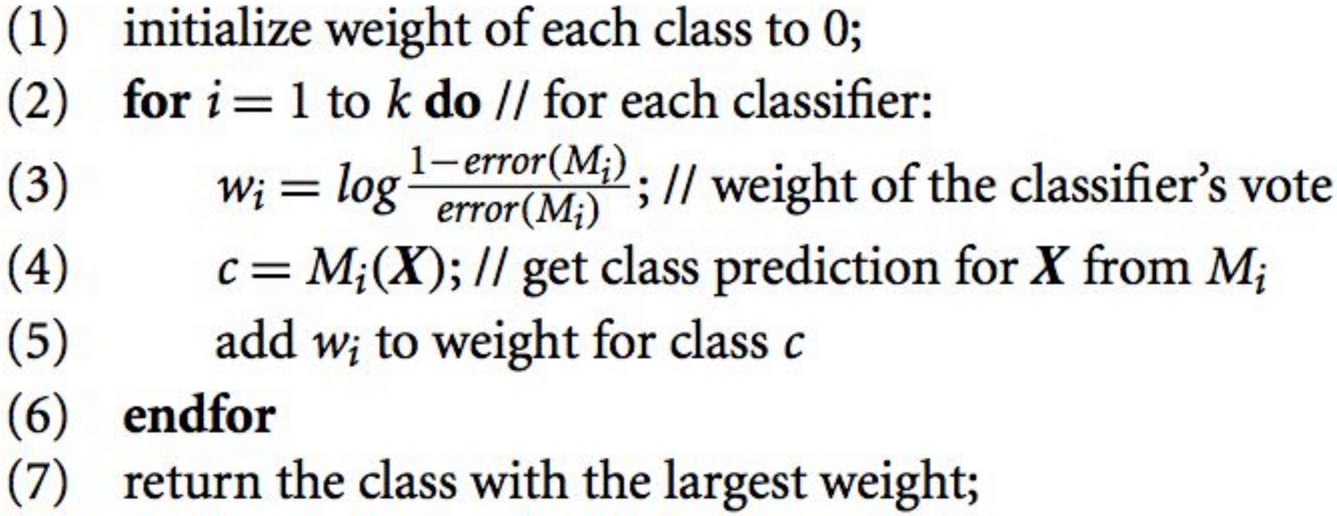
Ensemble Methods

Primary ensemble methods: simple combination, bagging, **boosting**

Resampling the data based on classifier error

Adaboost Testing Phase

For every tuple in the test set:

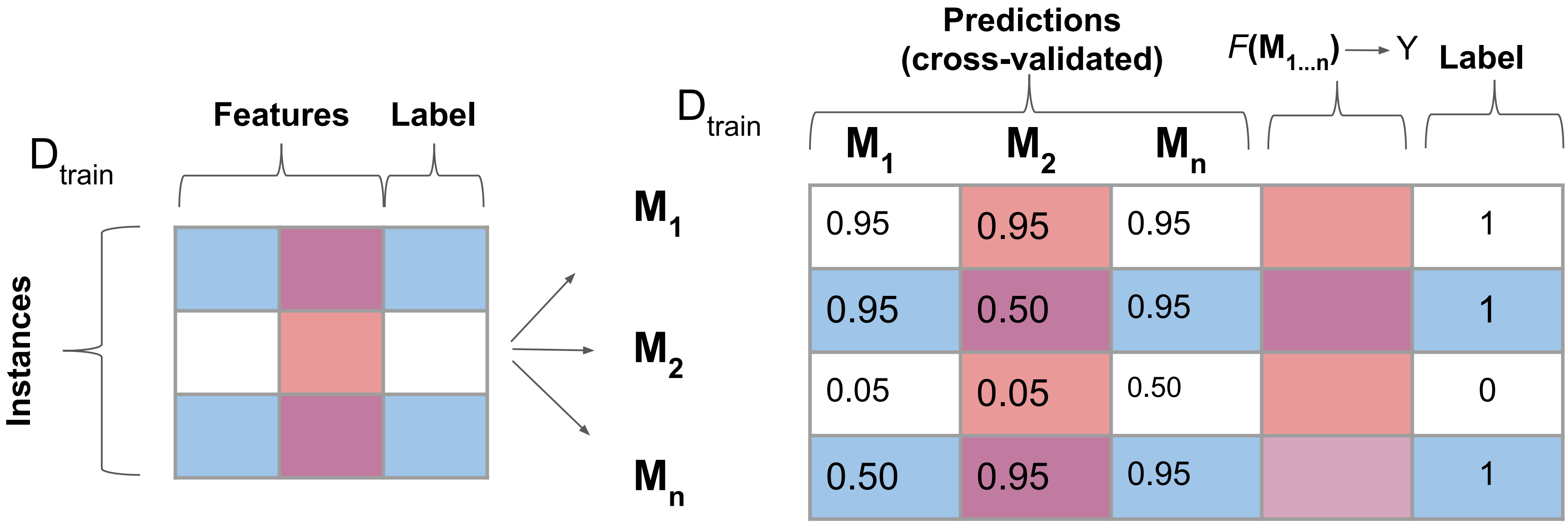


```
initialize weight of each class to 0;
for $i = 1$ to $k$ do // for each classifier:
    $w_i = log\frac{1-error(M_i)}{error(M_i)}$; // weight of the classifier's vote
    $c = M_i(\boldsymbol{X})$; // get class prediction for $\boldsymbol{X}$ from $M_i$
    add $w_i$ to weight for class $c$
endfor
return the class with the largest weight;
```

Ensemble Methods

Primary ensemble methods: blending, bagging, boosting, **stacking & blending**

Any arbitrary supervised learning technique can be used to train an ensemble using your set of predictions as the features

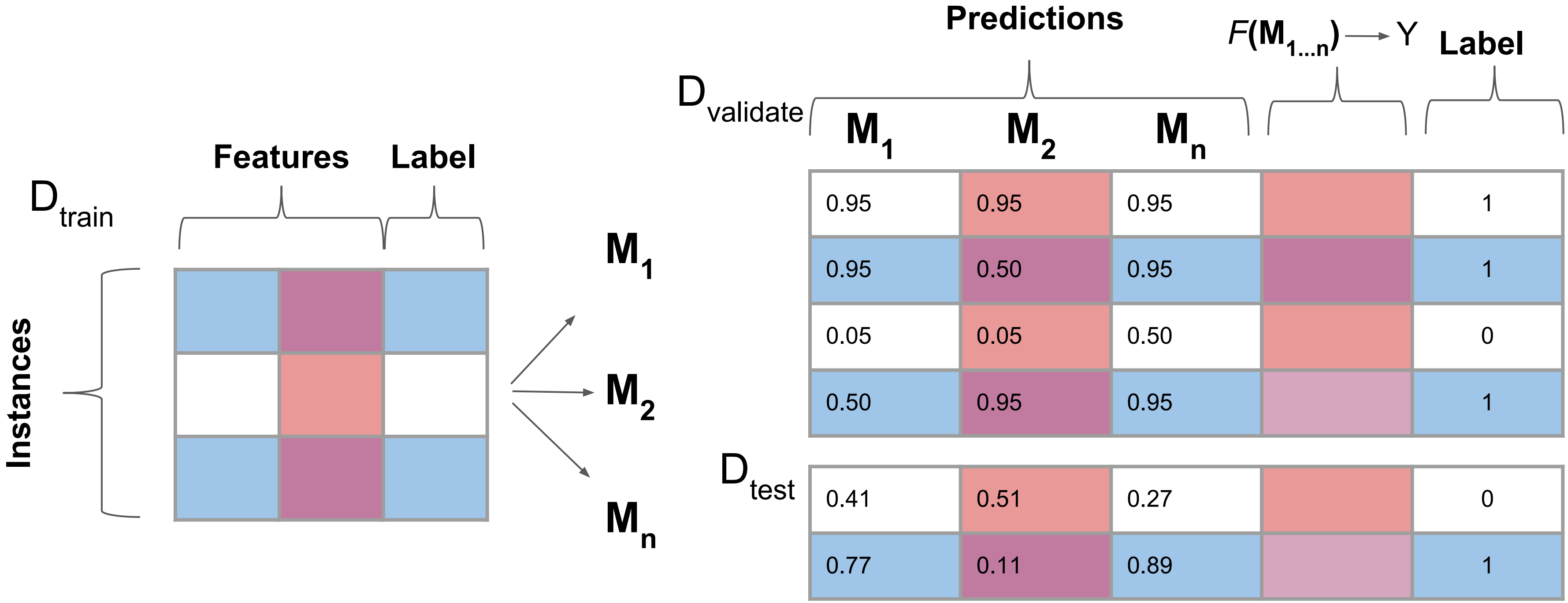


M_1	M_2	M_n	$F(M_{1...n}) \rightarrow Y$	Label
0.95	0.95	0.95		1
0.95	0.50	0.95		1
0.05	0.05	0.50		0
0.50	0.95	0.95		1

Ensemble Methods

Primary ensemble methods: blending, bagging, boosting, **stacking & blending**

Blending: Make predictions on a test and validation set. Train a blending model based on the validation set



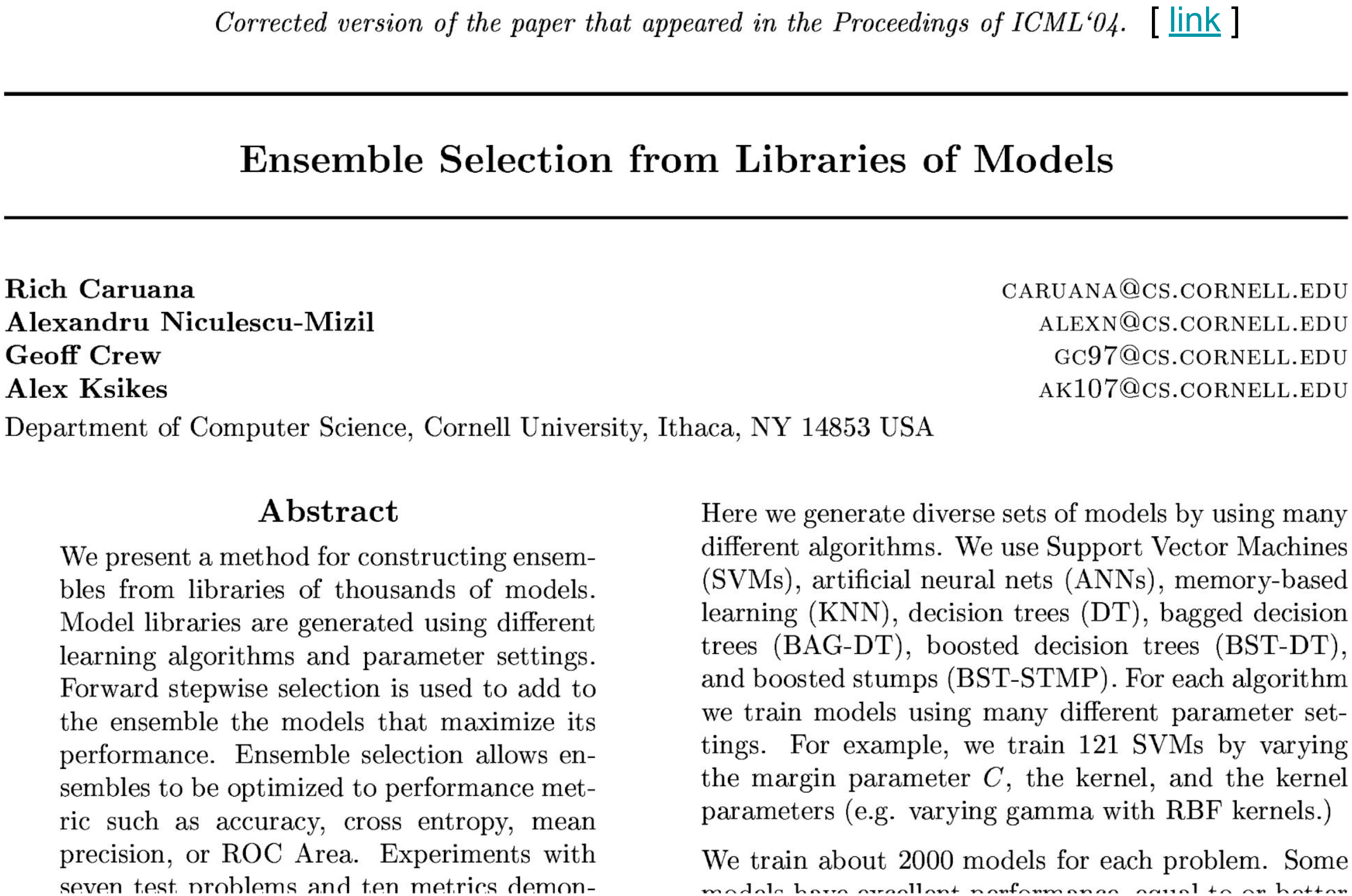
$D_{validate}$

M_1	M_2	M_n	$F(\mathbf{M}_{1...n}) \rightarrow Y$	Label
0.95	0.95	0.95		1
0.95	0.50	0.95		1
0.05	0.05	0.50		0
0.50	0.95	0.95		1

D_{test}

M_1	M_2	M_n	$F(\mathbf{M}_{1...n}) \rightarrow Y$	Label
0.41	0.51	0.27		0
0.77	0.11	0.89		1

Ensemble Selection (Caruana et al., 2004)



Corrected version of the paper that appeared in the Proceedings of ICML'04. [link]

Ensemble Selection from Libraries of Models

Rich Caruana CARUANA@CS.CORNELL.EDU
Alexandru Niculescu-Mizil ALEXN@CS.CORNELL.EDU
Geoff Crew GC97@CS.CORNELL.EDU
Alex Ksikes AK107@CS.CORNELL.EDU
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

Abstract

We present a method for constructing ensembles from libraries of thousands of models. Model libraries are generated using different learning algorithms and parameter settings. Forward stepwise selection is used to add to the ensemble the models that maximize its performance. Ensemble selection allows ensembles to be optimized to performance metric such as accuracy, cross entropy, mean precision, or ROC Area. Experiments with seven test problems and ten metrics demon-

Here we generate diverse sets of models by using many different algorithms. We use Support Vector Machines (SVMs), artificial neural nets (ANNs), memory-based learning (KNN), decision trees (DT), bagged decision trees (BAG-DT), boosted decision trees (BST-DT), and boosted stumps (BST-STMP). For each algorithm we train models using many different parameter settings. For example, we train 121 SVMs by varying the margin parameter C, the kernel, and the kernel parameters (e.g. varying gamma with RBF kernels.)

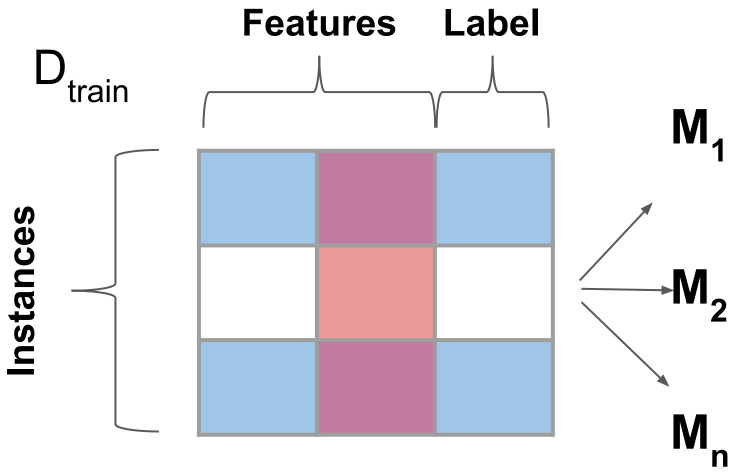
We train about 2000 models for each problem. Some models have excellent performance, equal to or better

Was this a type of ensemble method is this: blending, stacking, or boosting?

Ensemble Methods

Primary ensemble methods: blending, bagging, boosting, **stacking & blending**

Stacking: same as blending but uses *cross-validation*



Predictions (cross-validated)

D_{train}

fold	M_1	M_2	M_n	$F(M_{1...n}) \rightarrow Y$	Label
1	0.95	0.95	0.95		1
2	0.95	0.50	0.95		1
3	0.05	0.05	0.50		0
4	0.50	0.95	0.95		1

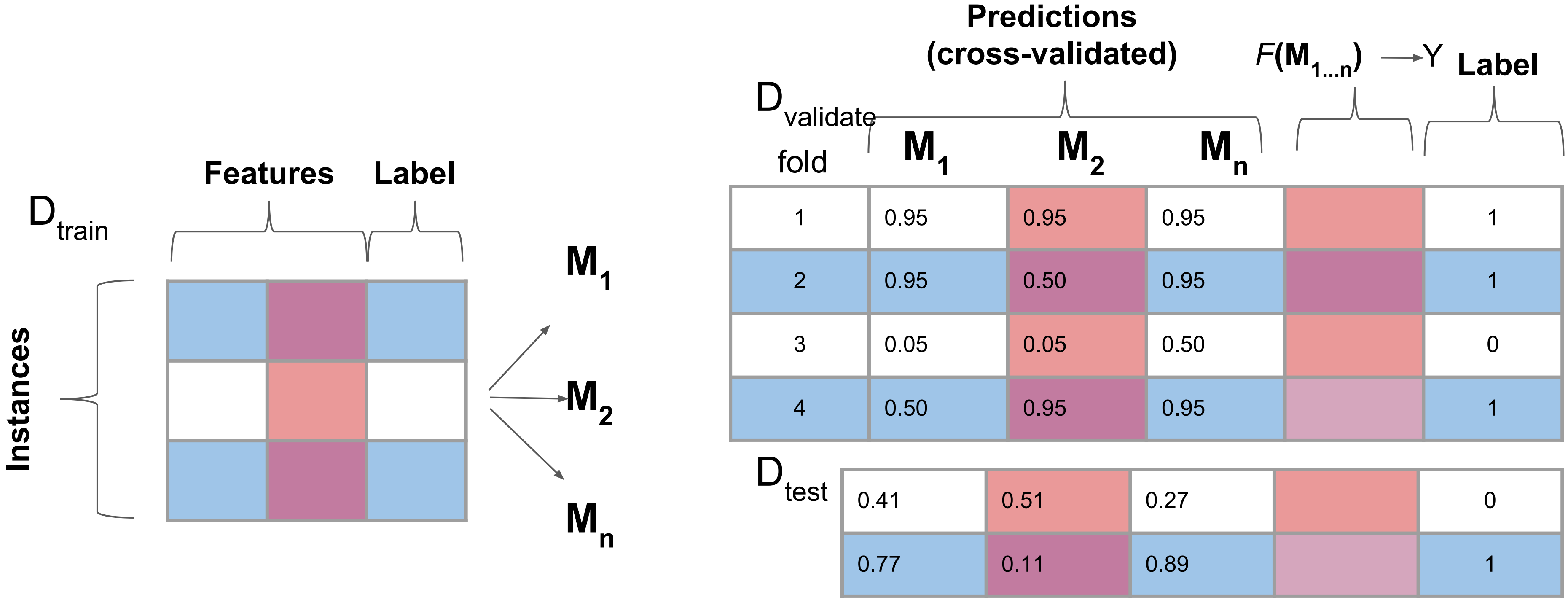
D_{test}

M_1	M_2	M_n	$F(M_{1...n}) \rightarrow Y$	Label
0.41	0.51	0.27		0
0.77	0.11	0.89		1

Ensemble Methods

Primary ensemble methods: blending, bagging, boosting, **stacking & blending**

Blending vs. Stacking

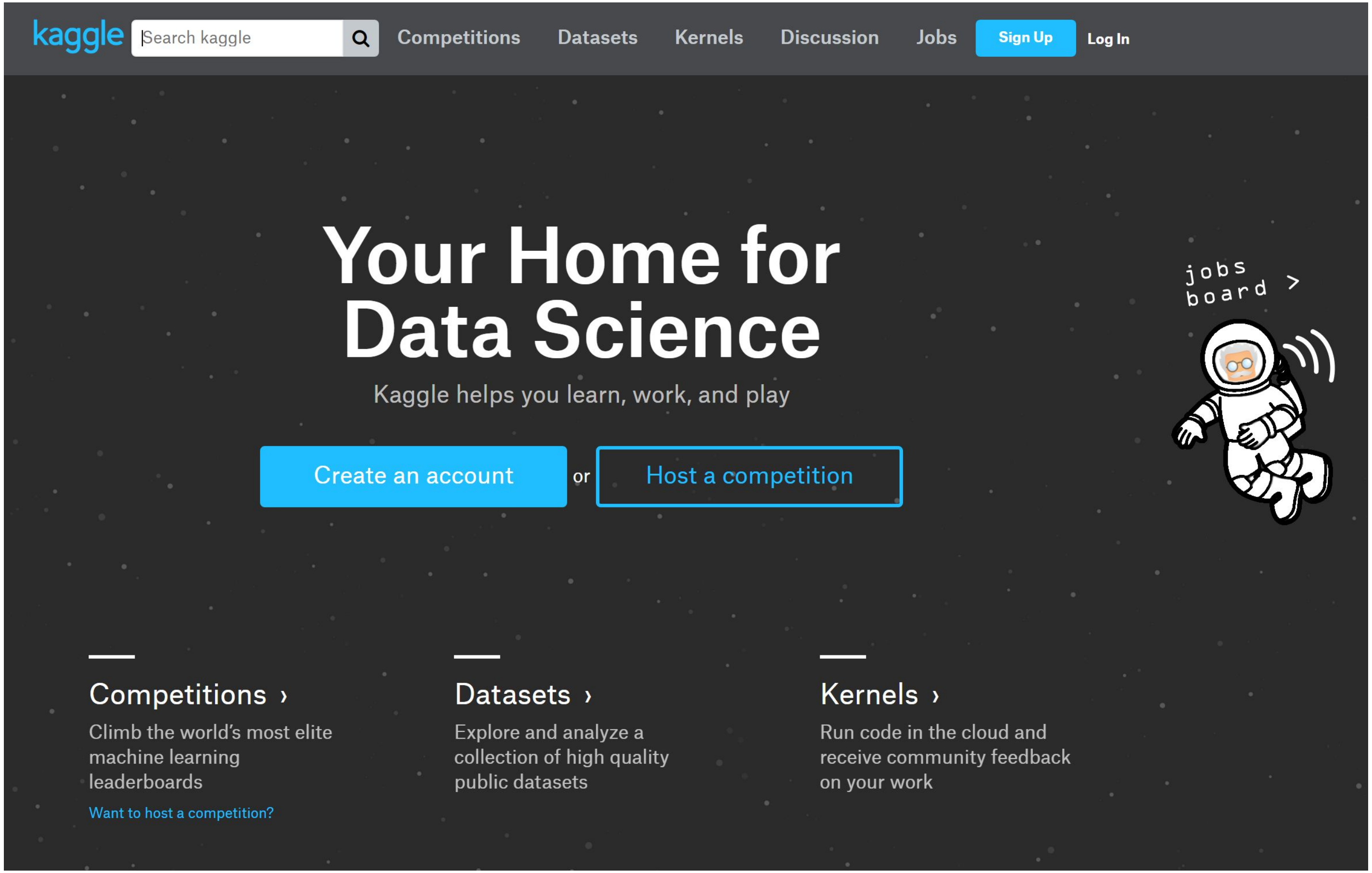


fold	M_1	M_2	M_n	$F(M_{1...n}) \rightarrow Y$	Label
1	0.95	0.95	0.95		1
2	0.95	0.50	0.95		1
3	0.05	0.05	0.50		0
4	0.50	0.95	0.95		1

D_{test}

M_1	M_2	M_n	$F(M_{1...n}) \rightarrow Y$	Label
0.41	0.51	0.27		0
0.77	0.11	0.89		1

Next lab: Kaggle



Your Home for Data Science

Kaggle helps you learn, work, and play

Competitions ›

Climb the world's most elite machine learning leaderboards

Want to host a competition?

Datasets ›

Explore and analyze a collection of high quality public datasets

Kernels ›

Run code in the cloud and receive community feedback on your work

Ensembling guide