

MIT INSTITUTE FOR DATA,
SYSTEMS, AND SOCIETY



IDSS

Applied Data Science Program

RANDOM FOREST

Munther A. Dahleh

Review

- Learning a Decision Tree
- Feature selection
 - Information Gain
- Statistical learning

Empirical Estimates

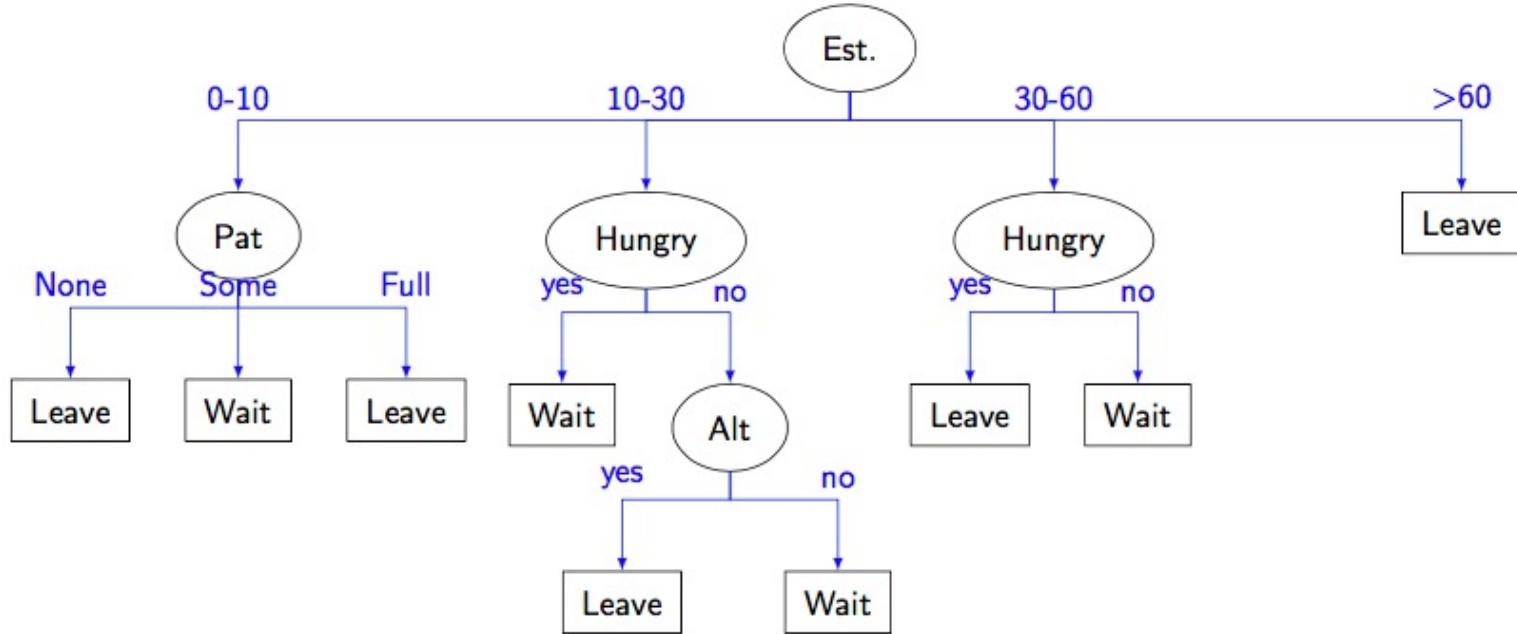
Data Set: $\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \dots, N\}$

Empirical Error of a decision Rule f :

$$R(f) = \frac{1}{N} \sum_i^N \mathbf{I}(f(x_i) \neq y_i)$$

$\mathbf{I}(x) = 1 \text{ if } x \neq 0, \text{ otherwise it is } 0$

Learning a Decision Tree



- How do we pick the feature at each level
- Information Gain (using Entropy)
- Minimize miss-classification

Outline

- Overfitting: Bias-Variance Tradeoff
 - Titanic example
 - Pruning
- Bagging to reduce variance
- Random Forest

Part I: Bias-Variance Tradeoff

Titanic Data Set

891 data point

38% Survival rate

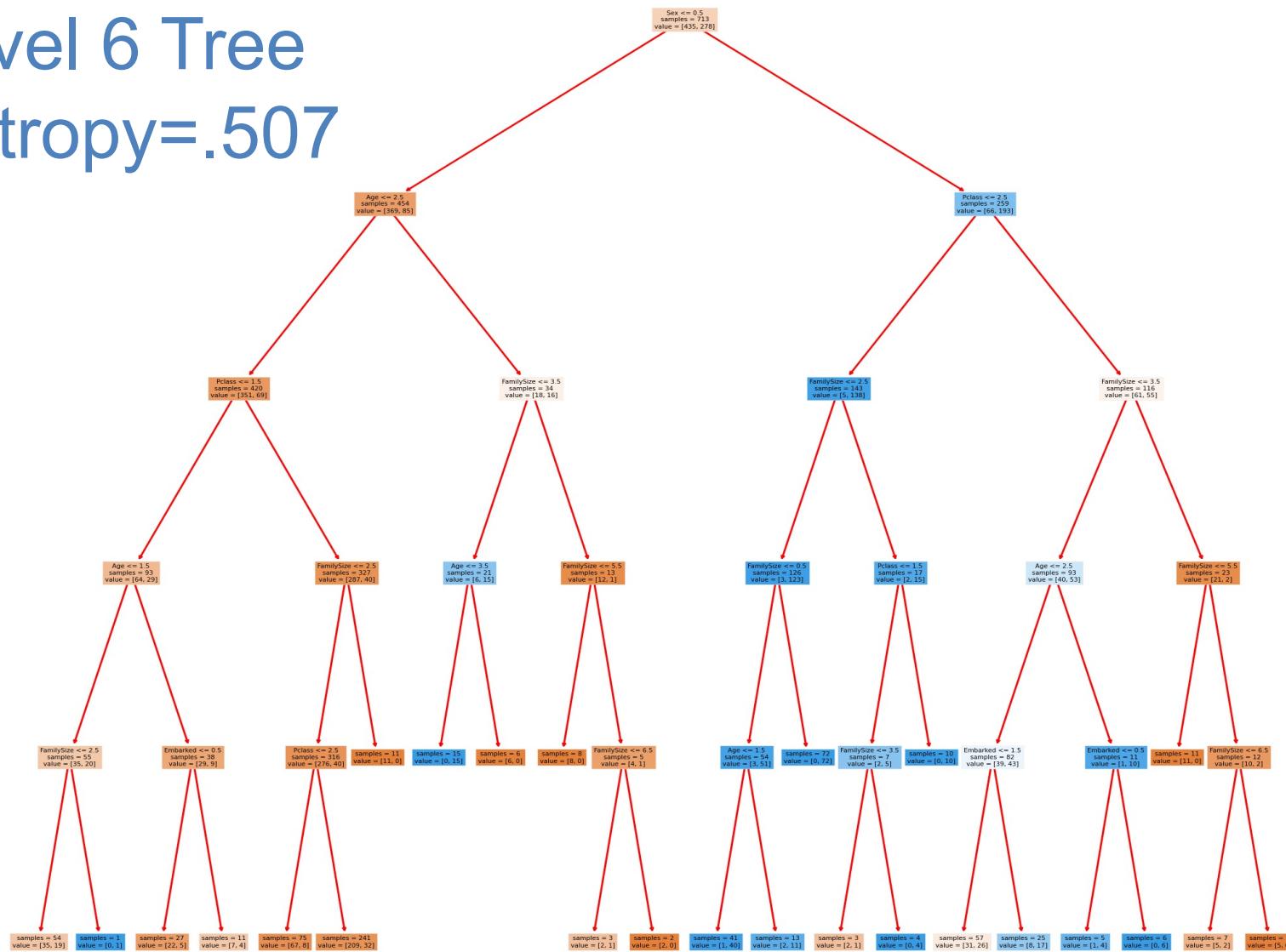
Data Split 8:2

- **Survived:** indicator variable describing whether the person survived the shipwreck.
- **Pclass:** Passenger class. Takes values from {1, 2, 3}.
- **Sex:** The sex of the passenger.
- **Age:** The age group of the passenger. Takes values from {< 13, 13 – 25, 25 – 40, 40 – 65, 65+}.
- **Embarked:** The port from which the passenger embarked on the ship. Takes values from { Cherbourg, Southampton, Queenstown}.
- **FamilySize:** Size of the passenger's family (excluding the passenger) on board.

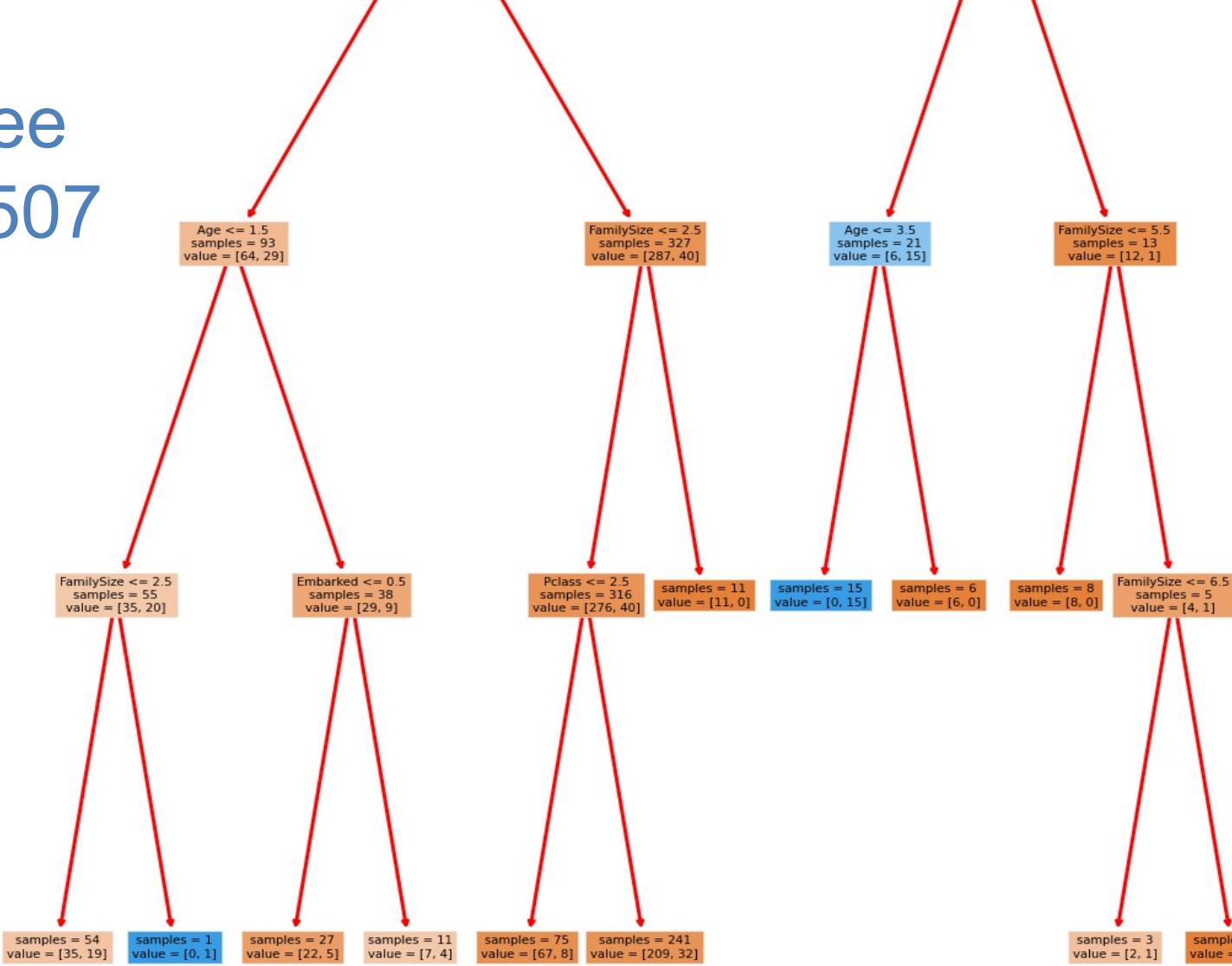
Titanic Example

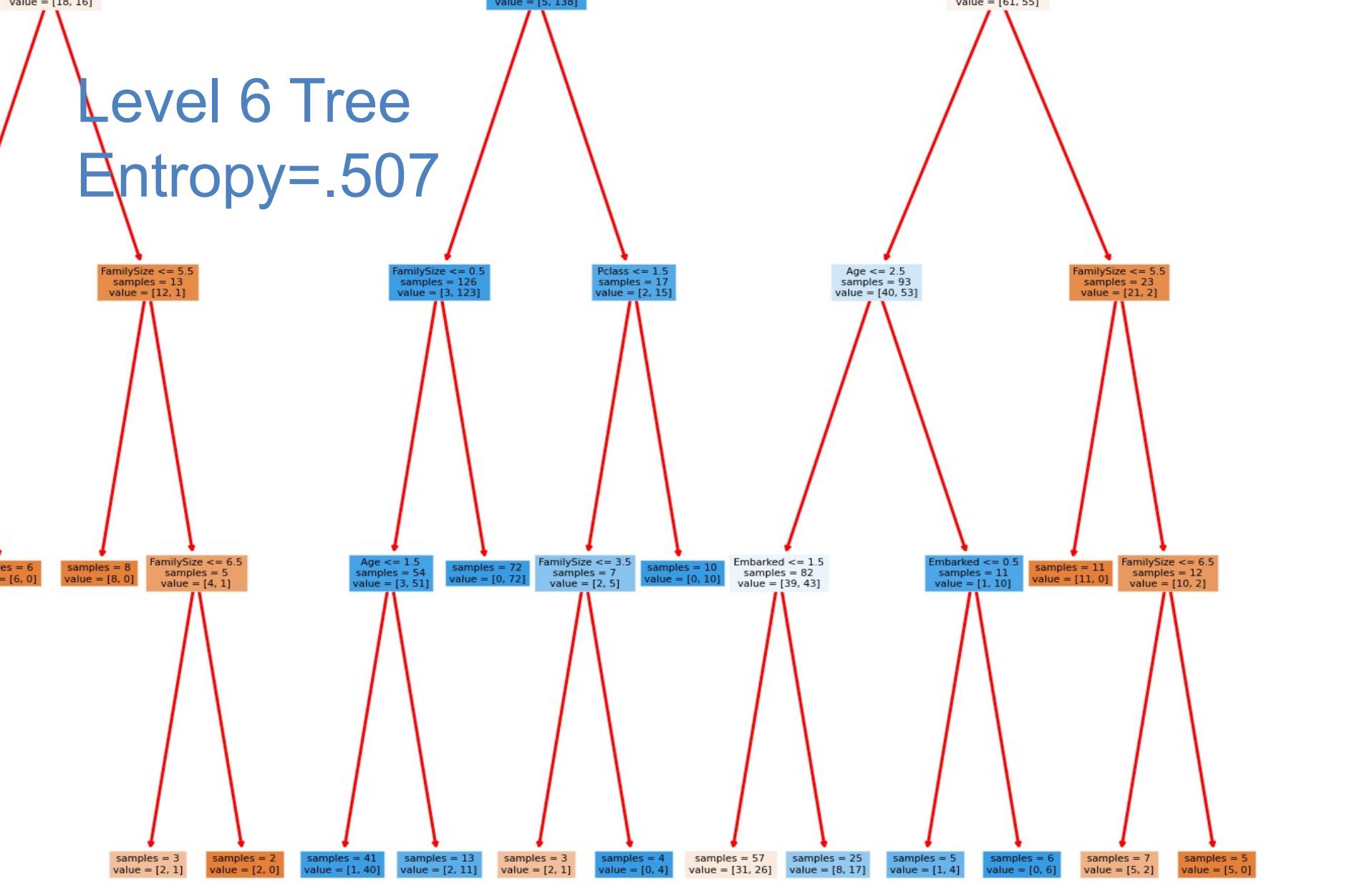
	Survived	Pclass	Sex	Age	Embarked	FamilySize
492	0	1	male	40-65	Southampton	0
141	1	3	female	13-25	Southampton	0
409	0	3	female	25-40	Southampton	4
31	1	1	female	25-40	Cherbourg	1
570	1	2	male	40-65	Southampton	0
593	0	3	female	25-40	Queenstown	2
873	0	3	male	40-65	Southampton	0
399	1	2	female	25-40	Southampton	0
406	0	3	male	40-65	Southampton	0
272	1	2	female	40-65	Southampton	1

Level 6 Tree Entropy=.507

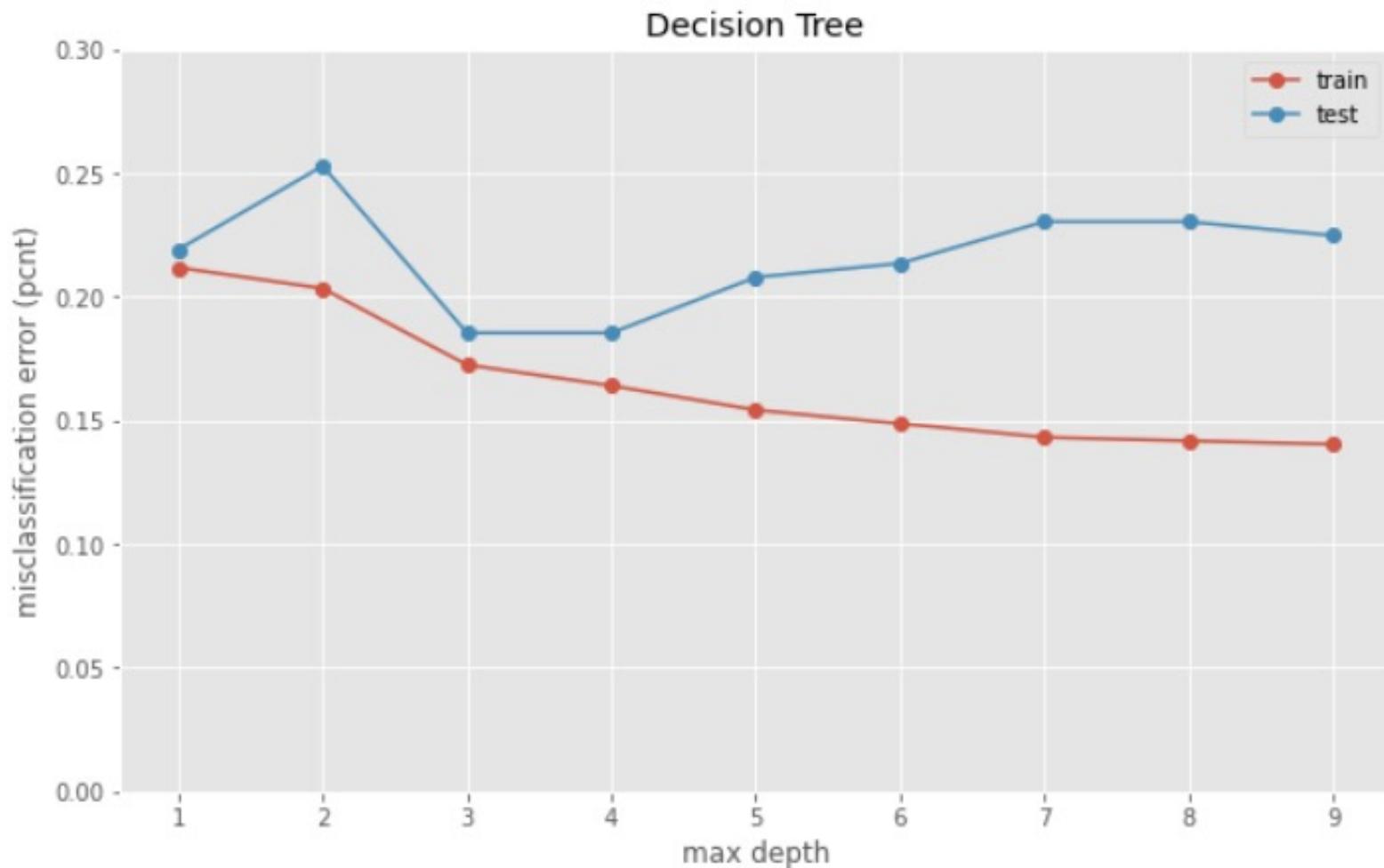


Level 6 Tree Entropy=.507

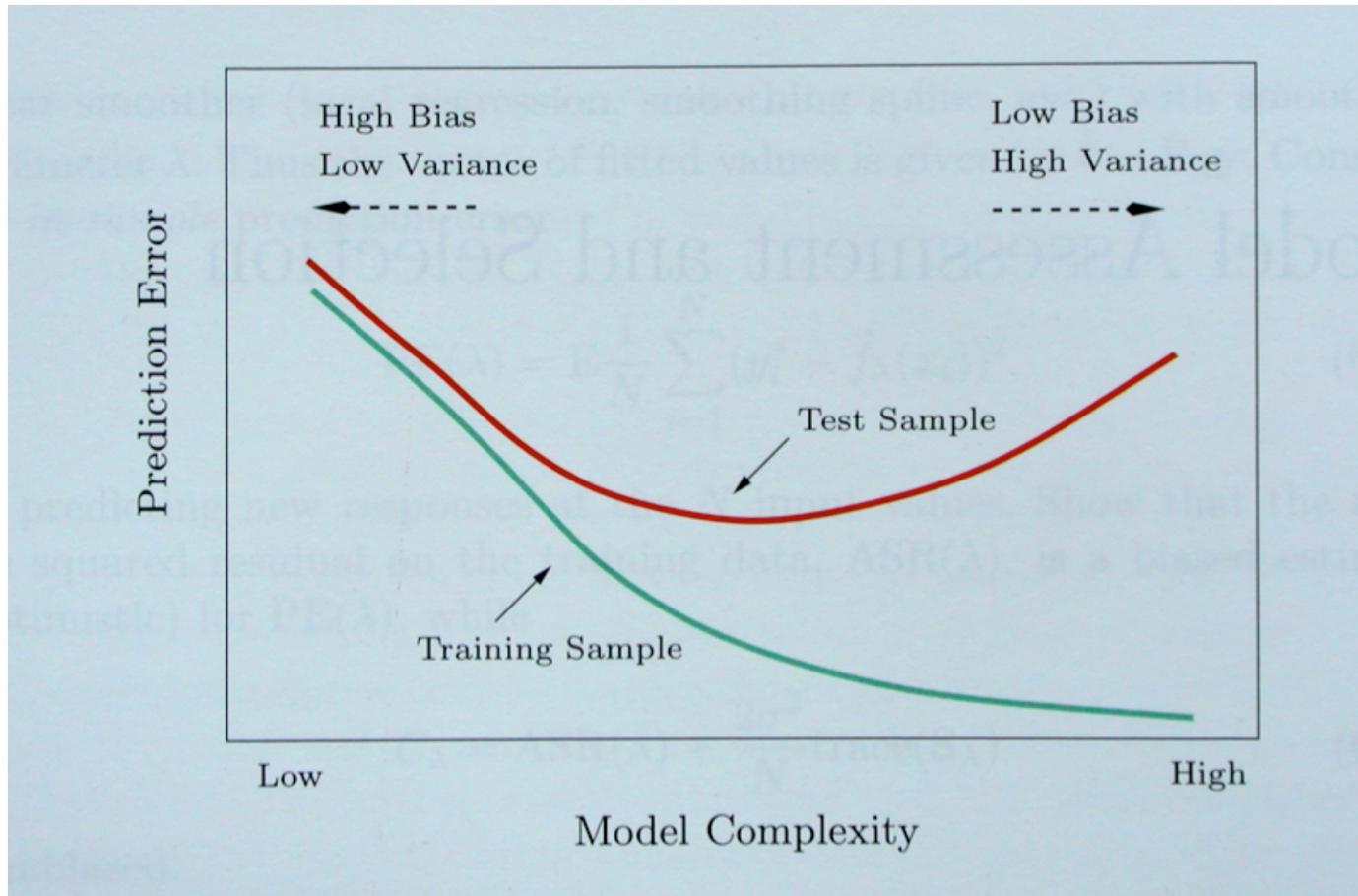




Train vs. Test Results

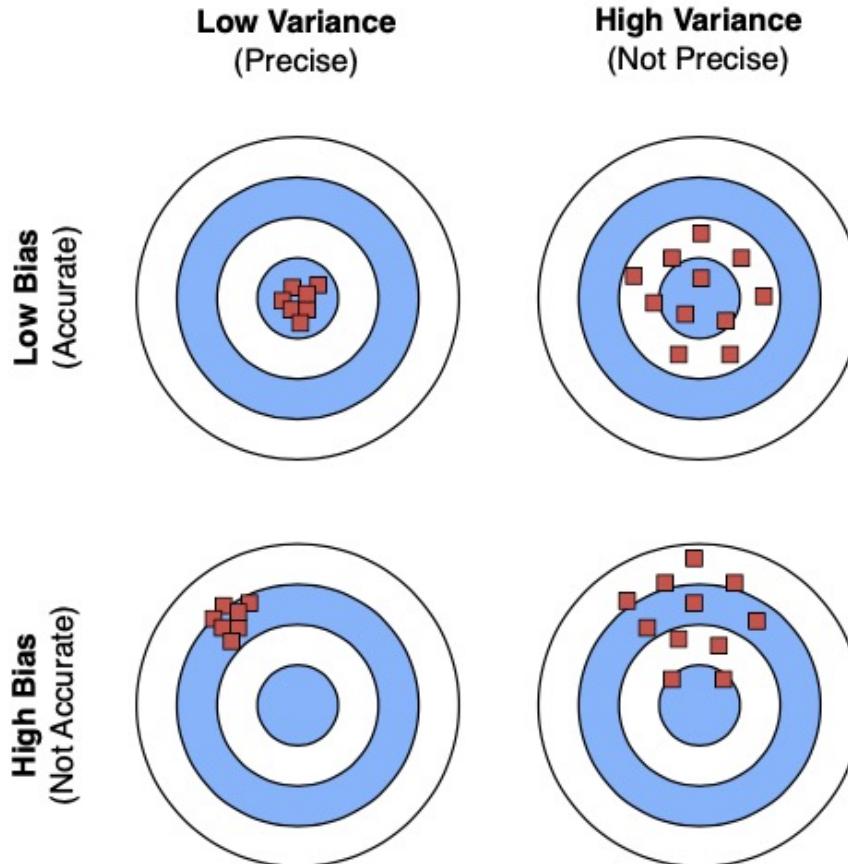


Bias-Variance Tradeoff



Hastie, Tibshirani, Friedman “Elements of Statistical Learning” 2001

Bias-Variance Tradeoff



Quick Fix: Pruning

- DT can be defined to a large level of granularity
 - Not a good way to generalize
- Pruning, Aggregation
 - Use misclassification as a guidance
 - Reduce the depth
 - Eliminate small class

Back to Example

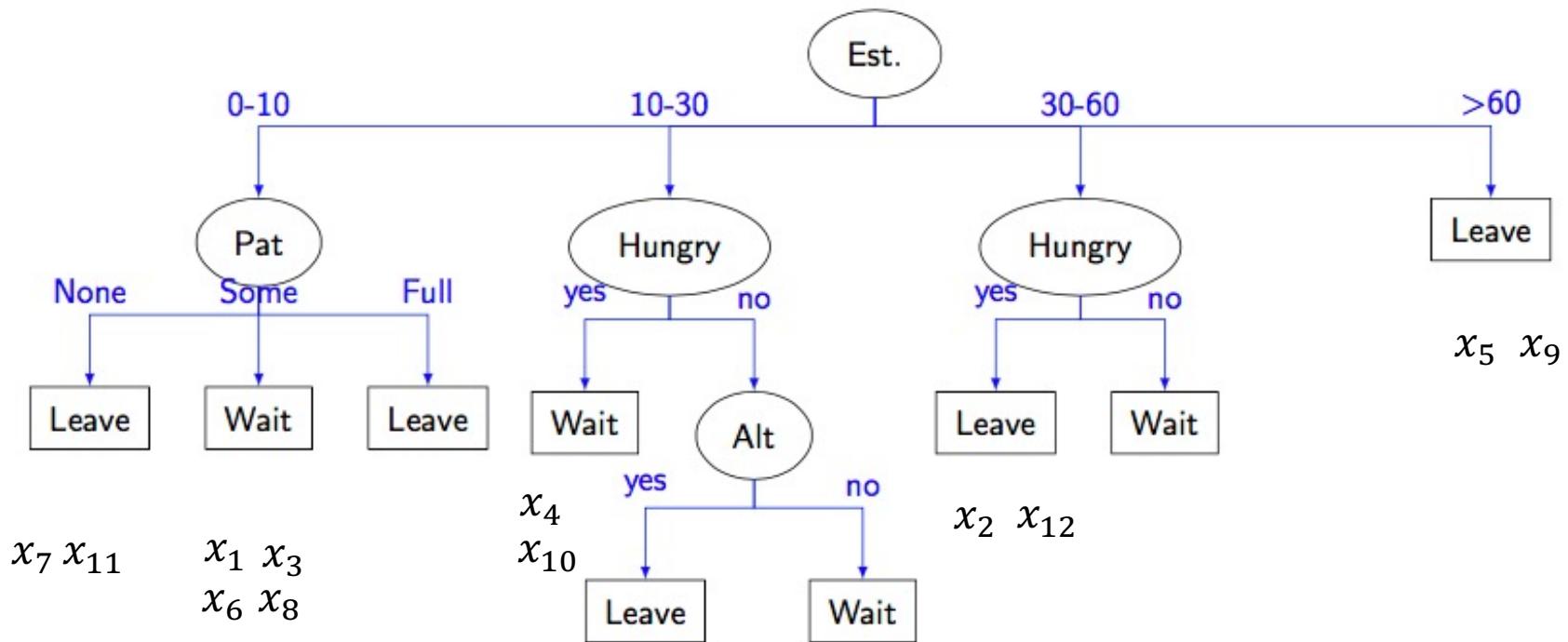
Example	Input Attributes										Goal <i>WillWait</i>
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
x_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	$y_1 = \text{Yes}$
x_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	$y_2 = \text{No}$
x_3	No	Yes	No	No	Some	\$	No	No	Burger	0-10	$y_3 = \text{Yes}$
x_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	$y_4 = \text{Yes}$
x_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	$y_5 = \text{No}$
x_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	$y_6 = \text{Yes}$
x_7	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	$y_7 = \text{No}$
x_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	$y_8 = \text{Yes}$
x_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	$y_9 = \text{No}$
x_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	$y_{10} = \text{No}$
x_{11}	No	No	No	No	None	\$	No	No	Thai	0-10	$y_{11} = \text{No}$
x_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	$y_{12} = \text{Yes}$

1.	Alternate: whether there is a suitable alternative restaurant nearby.
2.	Bar: whether the restaurant has a comfortable bar area to wait in.
3.	Fri/Sat: true on Fridays and Saturdays.
4.	Hungry: whether we are hungry.
5.	Patrons: how many people are in the restaurant (values are None, Some, and Full).
6.	Price: the restaurant's price range (\$, \$\$, \$\$\$).
7.	Raining: whether it is raining outside.
8.	Reservation: whether we made a reservation.
9.	Type: the kind of restaurant (French, Italian, Thai or Burger).
10.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

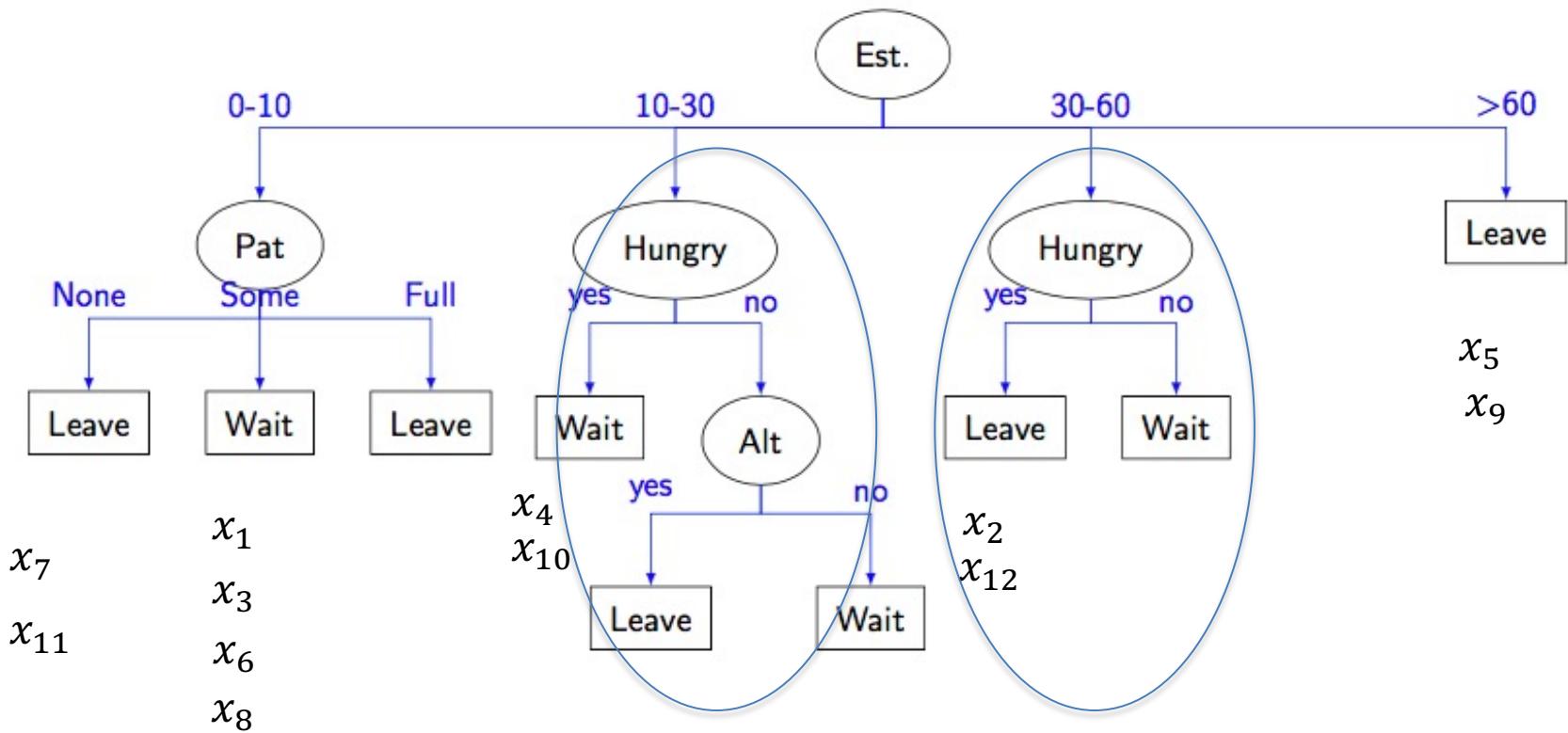
Attributes:

[from: Russell & Norvig]

How does Pruning work here?

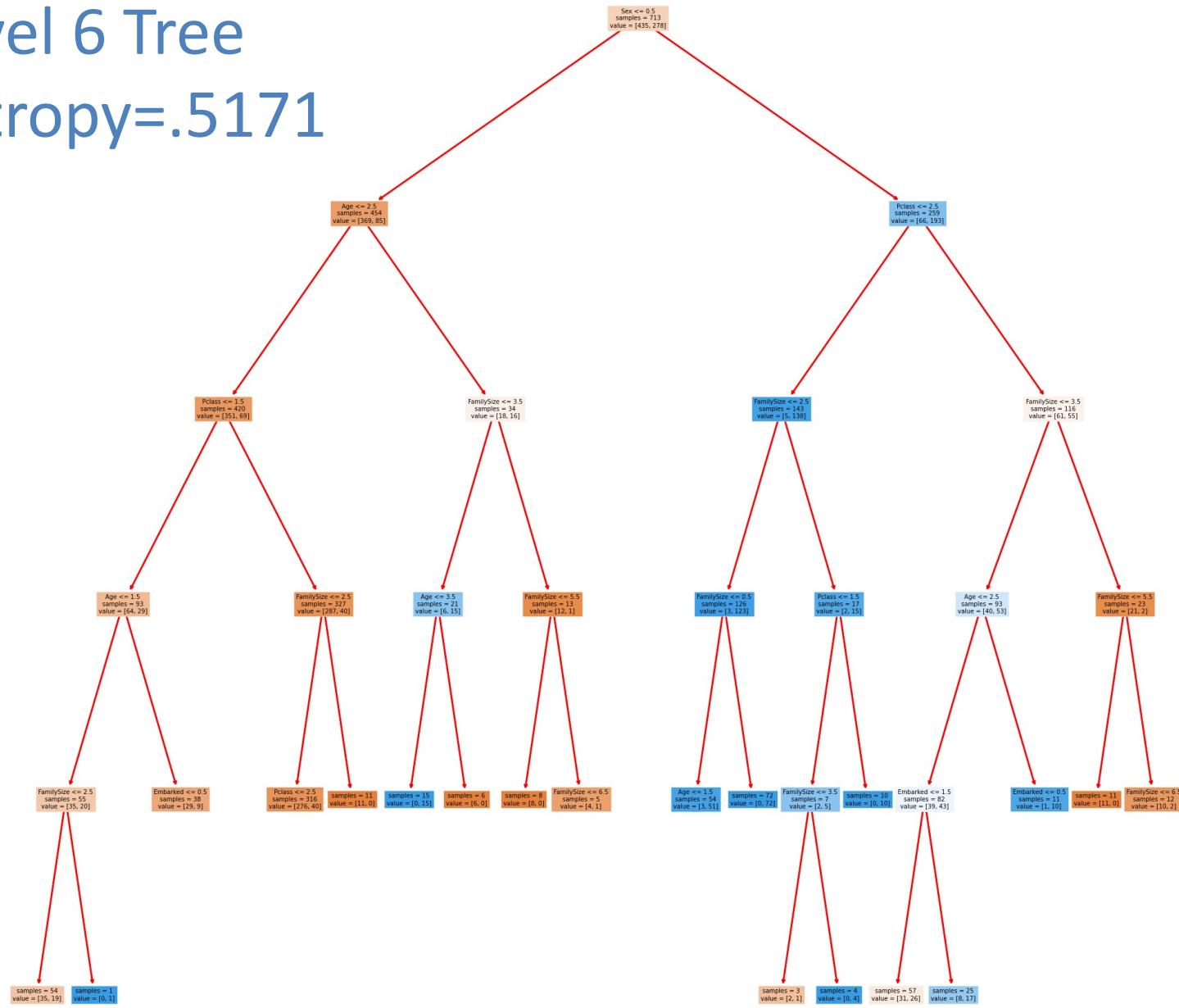


How does Pruning work here?



Level 6 Tree

Entropy=.5171



Summary

- Create a tree with maximum depth
 - Either until every leaf is a single data point
 - Or use all features
- Pick a subtree (a node and all leaves)
- Aggregate that leaves all the way to the node
- Compute new error
 - Misclassification
 - Entropy

Summary

- Pruning is expensive
- Pruning is counter-intuitive
 - Fixing a bad model
- How about directly building a better model

Part II: Bagging

Ensemble Learning/Random Forest

- Ensemble Methods are the key idea behind Random Forests
- Motivated by averaging techniques
- You can reduce the variance if you average a number of independent RVs
- How?

History

- Tin Kam Ho (Random Subspace Methods)



- Breiman and Cutler (Registered Random Forest as a trademark: They combine Bagging with random feature selection



- Amit and Geman: did the same independently



Ensemble Learning: Basic idea

Bagging= Bootstrap + Aggregation

Ensemble Learning: Bootstrap

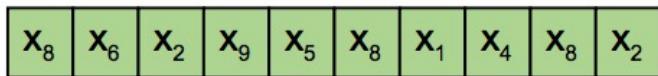
- Sample data with replacement
 - Create many data sets
- Data not sampled is used for cross validation
- Data sets are correlated (or dependent)
 - Reasonably uncorrelated for large sample

Bootstrap Sample

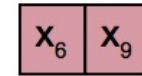
Original Dataset



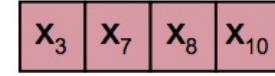
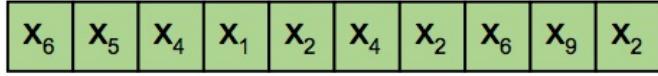
Bootstrap 1



Bootstrap 2



Bootstrap 3



Training Sets

Test Sets

Size of Test Set

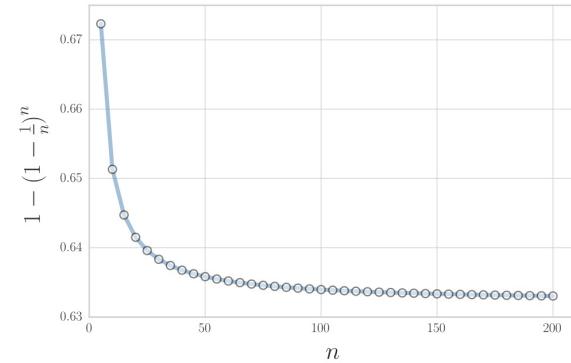
- Assume that we have n data points
- What's the probability that a specific data point is not selected in n samples with replacement?

$$\left(1 - \frac{1}{n}\right)^n$$

- If n is large then this probability is:

$$\frac{1}{e} = 0.368$$

- Provides a reasonable percentage for cross validation to estimate error



Multiple Classifiers

- Build a classifier for each "new" data set

$$\hat{y}_i = f_i(x) \text{ with error } \epsilon_i$$

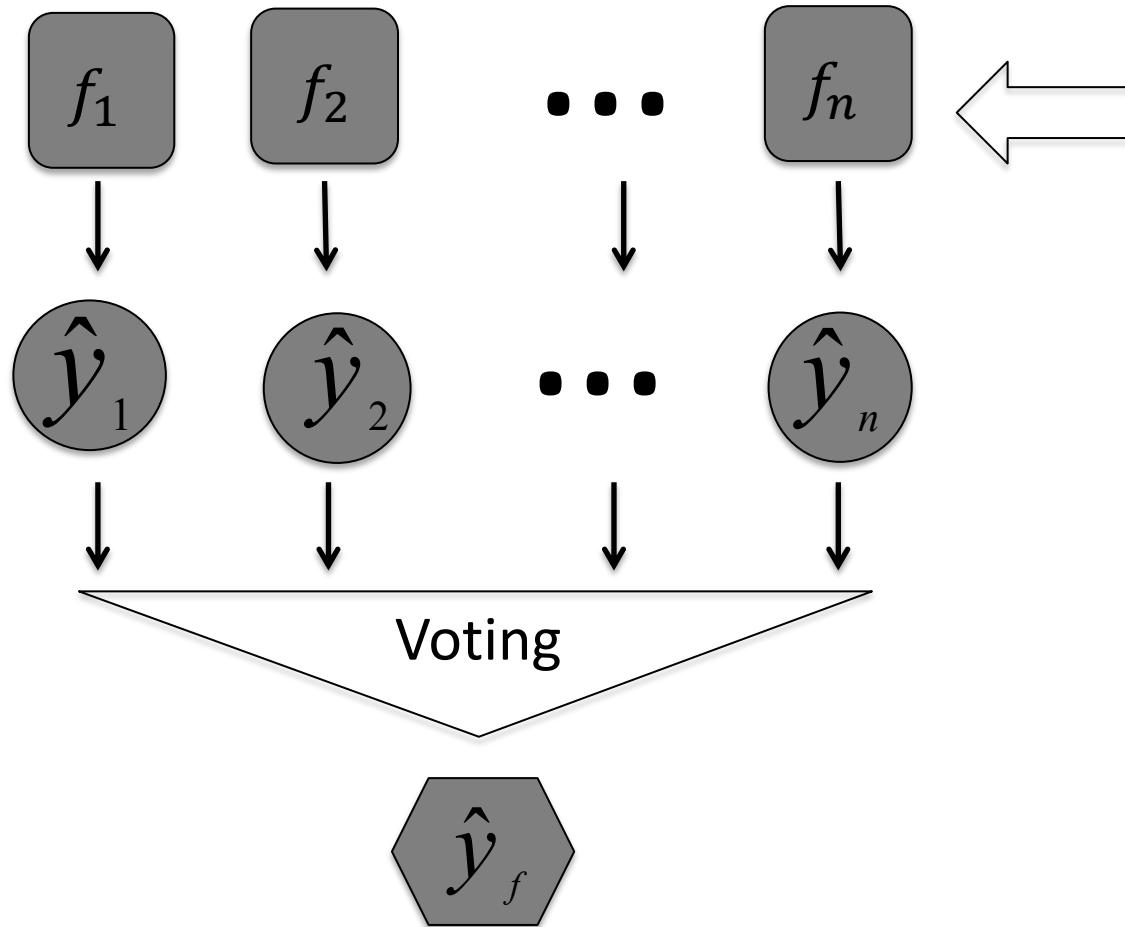
- If the n is large, then the outcome of these classifiers are "**not**" too dependent

Aggregation

- Multiple classifiers, decision trees
- Voting between choices (aggregation)
- Reduce the variance (generalization) of classification

Majority Classifier (Aggregate)

Classification
models



Final Prediction

Majority Classifier (aggregate)

- Voting:

$$\{\hat{y}_f\} = f(x) \equiv \text{majority}(f_1(x), f_2(x), \dots, f_l(x))$$

- How does this estimator perform?
- You can also combine using weighted sum

Analysis of voting

- Assume each classifier has error

$$\epsilon_i = P(f_i(x) \neq \hat{y}_i) < 0.5$$

- Assume classifiers are independent
- Error in aggregation: More than half of the classifiers are wrong

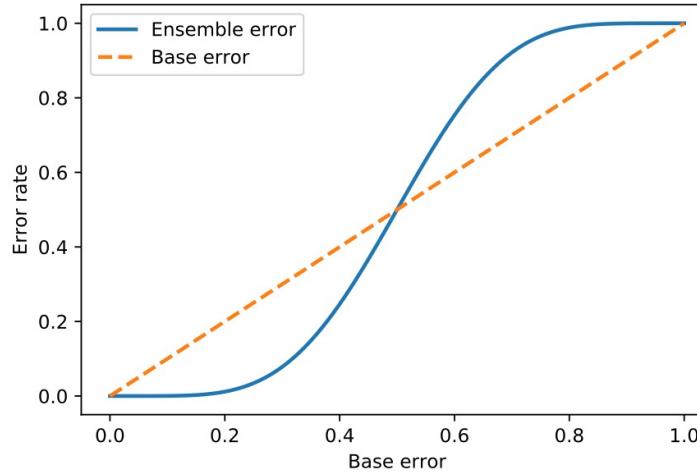
Analysis of Voting

- Uniform error

$$\epsilon = P(f_i(x) \neq \hat{y}_i) = 0.25$$

- Then $P(f(x) \neq \hat{y}_f) = \sum_{\{k \geq \frac{l}{2}\}}^l \binom{l}{k} \epsilon^k (1 - \epsilon)^{l-k}$

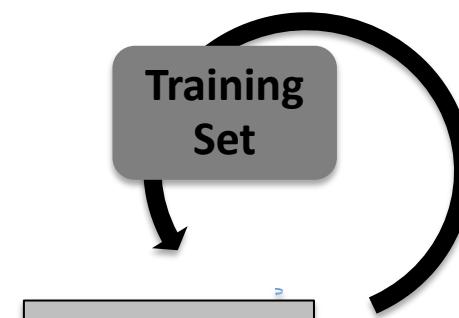
- *Improvement!*



Majority Classifier (Bagging)

Bootstrap
samples

T_1



Classification
models

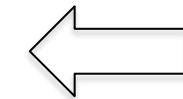
f_1

f_2

...

T_l

f_l



Predictions

\hat{y}_1

\hat{y}_2

...

\hat{y}_l

Voting

Final Prediction

Summary: Bagging

- Bagging= Bootstrap + Aggregation
- Bootstrap: sample with replacement
- Build a tree classifier with each sample
- Aggregate through majority voting

Illustrative Example: Waiting at a restaurant

Example	Input Attributes										Goal <i>WillWait</i>
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
x_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0–10	$y_1 = \text{Yes}$
x_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30–60	$y_2 = \text{No}$
x_3	No	Yes	No	No	Some	\$	No	No	Burger	0–10	$y_3 = \text{Yes}$
x_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10–30	$y_4 = \text{Yes}$
x_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	$y_5 = \text{No}$
x_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0–10	$y_6 = \text{Yes}$
x_7	No	Yes	No	No	None	\$	Yes	No	Burger	0–10	$y_7 = \text{No}$
x_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0–10	$y_8 = \text{Yes}$
x_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	$y_9 = \text{No}$
x_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10–30	$y_{10} = \text{No}$
x_{11}	No	No	No	No	None	\$	No	No	Thai	0–10	$y_{11} = \text{No}$
x_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30–60	$y_{12} = \text{Yes}$

1.	Alternate: whether there is a suitable alternative restaurant nearby.
2.	Bar: whether the restaurant has a comfortable bar area to wait in.
3.	Fri/Sat: true on Fridays and Saturdays.
4.	Hungry: whether we are hungry.
5.	Patrons: how many people are in the restaurant (values are None, Some, and Full).
6.	Price: the restaurant's price range (\$, \$\$, \$\$\$).
7.	Raining: whether it is raining outside.
8.	Reservation: whether we made a reservation.
9.	Type: the kind of restaurant (French, Italian, Thai or Burger).
10.	WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Attributes:



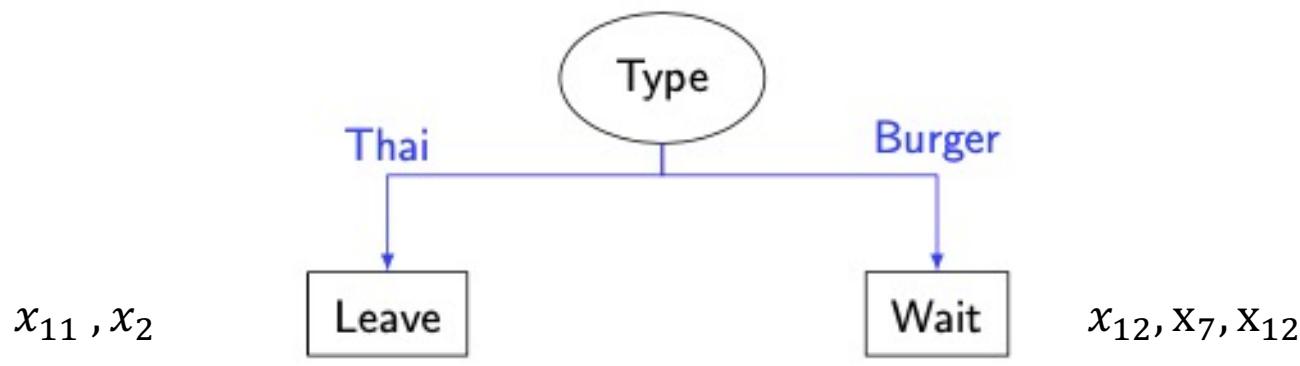
Institute of
Technology

[from: Russell & Norvig]

MIT INSTITUTE FOR DATA,
SYSTEMS, AND SOCIETY

Restaurant example

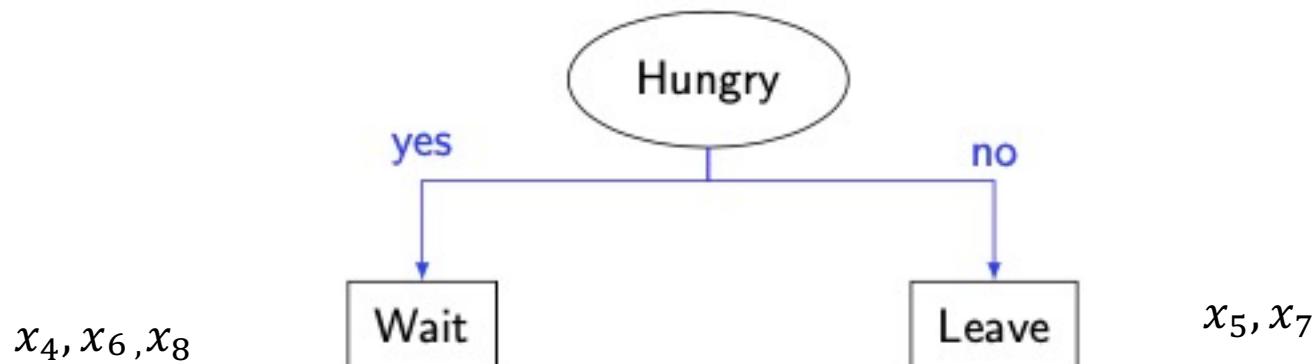
Tree 1: $x_{11}, x_{12}, x_7, x_2, x_{12}$



Loss: 0.55098

Restaurant example

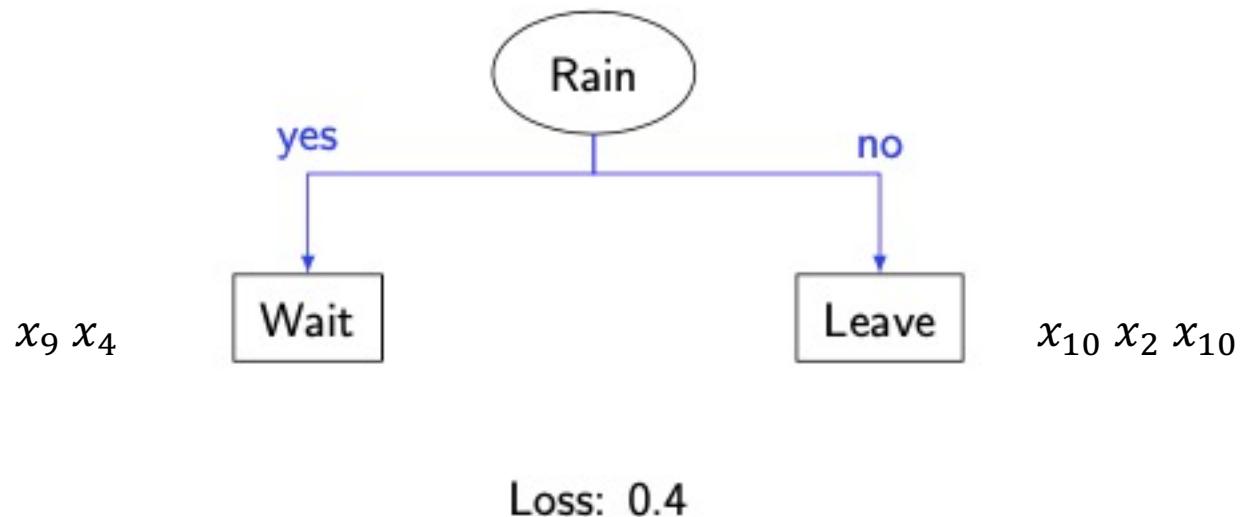
Tree 2: x_5, x_4, x_6, x_8, x_7



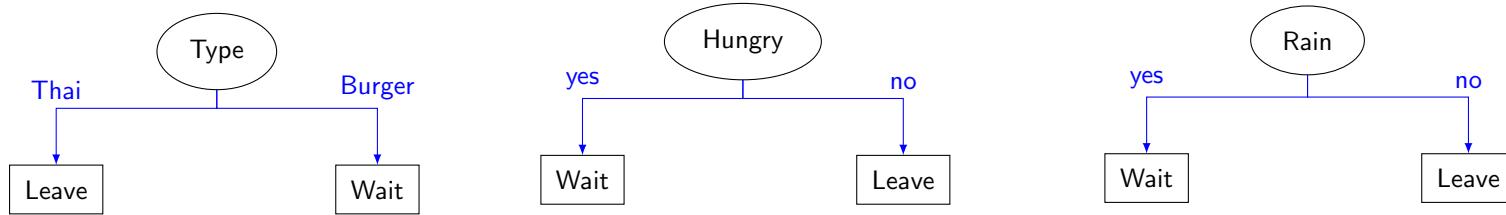
Loss: 0.

Restaurant Example

Tree 3: $x_9, x_{10}, x_4, x_2, x_{10}$



What's the Aggregate Classifier



$$x_1: 1, 1, 0 \rightarrow 1$$

$$x_2: 0, 0, 1 \rightarrow 0$$

$$x_3: 1, 0, 0 \rightarrow 0$$

$$x_4: 0, 1, 1 \rightarrow 1$$

$$x_5: 1, 0, 0 \rightarrow 0$$

$$x_6: 0, 1, 1 \rightarrow 1$$

$$x_7: 1, 0, 1 \rightarrow 1$$

$$x_8: 0, 1, 1 \rightarrow 1$$

$$x_9: 1, 0, 1 \rightarrow 1$$

$$x_{10}: 0, 1, 0 \rightarrow 0$$

$$x_{11}: 0, 0, 0 \rightarrow 0$$

$$x_{12}: 1, 1, 0 \rightarrow 1$$

Example	Input Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	
x_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	$WillWait$
x_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	$y_1 = Yes$
x_3	No	Yes	No	No	Some	\$	No	No	Burger	0-10	$y_2 = No$
x_4	Yes	No	Yes	Yes	Full	\$	Yes	No	Thai	10-30	$y_3 = Yes$
x_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	$y_4 = Yes$
x_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	$y_5 = No$
x_7	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	$y_6 = Yes$
x_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	$y_7 = No$
x_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	$y_8 = Yes$
x_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	$y_9 = No$
x_{11}	No	No	No	No	None	\$	No	No	Thai	0-10	$y_{10} = No$
x_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	$y_{11} = No$
											$y_{12} = Yes$

1. Alternate: whether there is a suitable alternative restaurant nearby.
2. Bar: whether the restaurant has a comfortable bar area to wait in.
3. Fri/Sat: true on Fridays and Saturdays.
4. Hungry: whether we are hungry.
5. Patrons: how many people are in the restaurant (values are None, Some, and Full).
6. Price: the restaurant's price range (\$, \$\$, \$\$\$).
7. Raining: whether it is raining outside.
8. Reservation: whether we made a reservation.
9. Type: the kind of restaurant (French, Italian, Thai or Burger).
10. WaitEstimate: the wait estimated by the host (0-10 minutes, 10-30, 30-60, >60).

Attributes:

[from: Russell & Norvig]

Drawback: We lost the tree!

Random Forest: Part III

Random Forest

- What if the samples are not independent
- You can increase the independence through sampling the features at each node
- Benefit: better generalization
- *Downside: less interpretable, less powerful*

Random Forest

Bootstrap

T_1

Feature
sample

set 1

f_1

\hat{y}_1

Training
Set

T_2

Set 2

f_2

\hat{y}_2

...

T_l

Set l

...

f_l

\hat{y}_l

Voting

\hat{y}_f

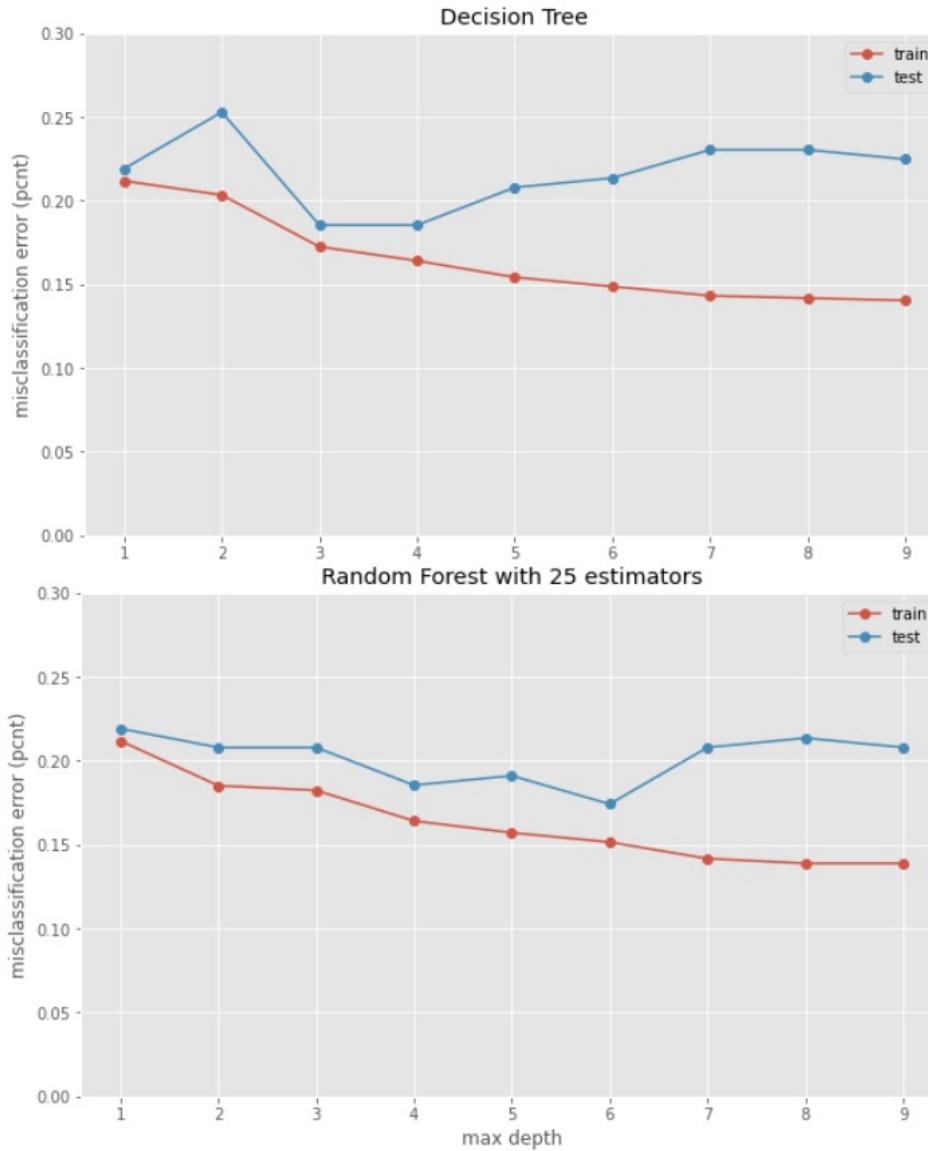


Massachusetts
Institute of
Technology

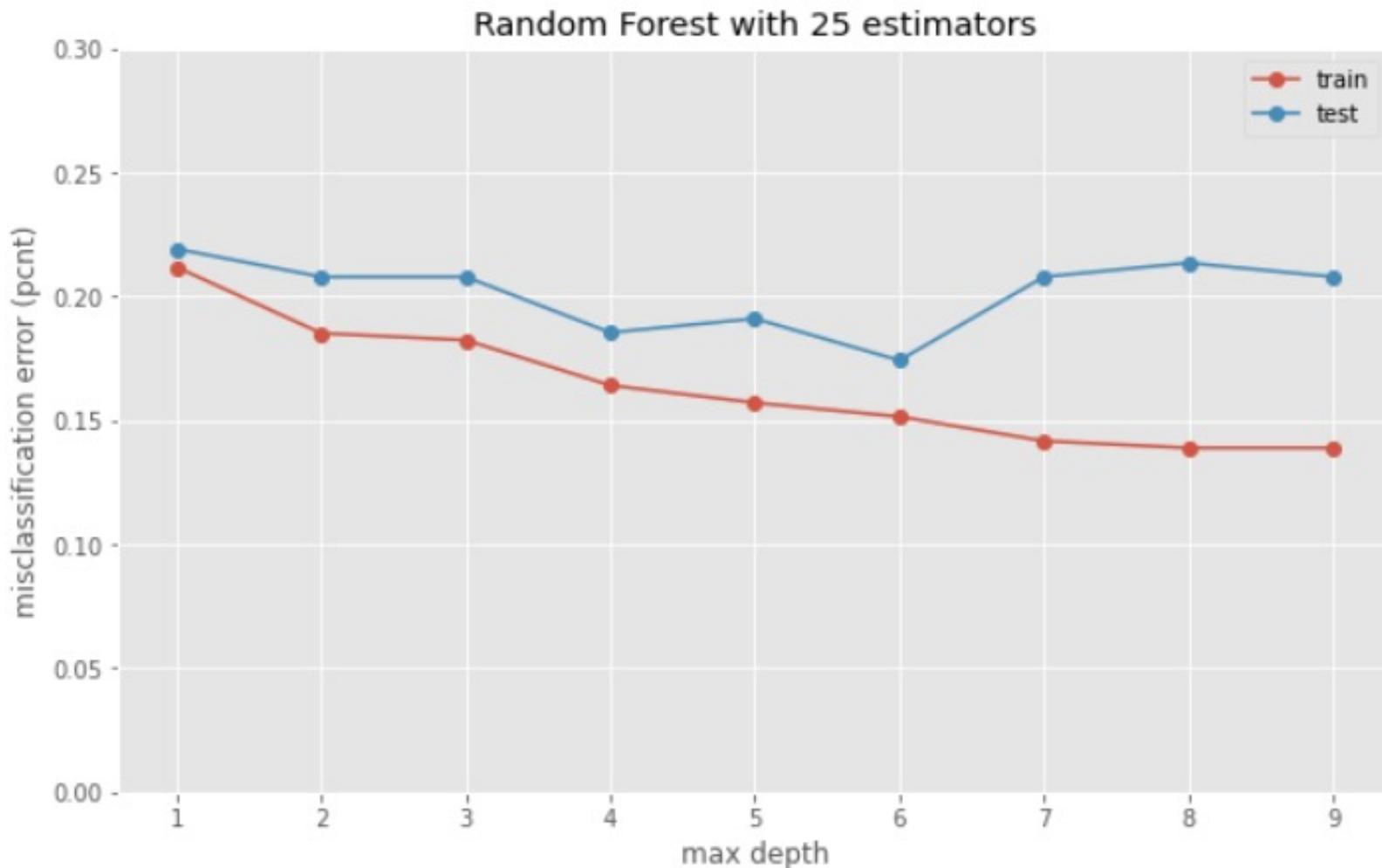
More Elaborate

- Can we diversify further?
- Select a random set of feature at each tree level
- Continue with previous process
- More diverse, difficult to interpret

DT vs Random Forest



Random Forest—Titanic data

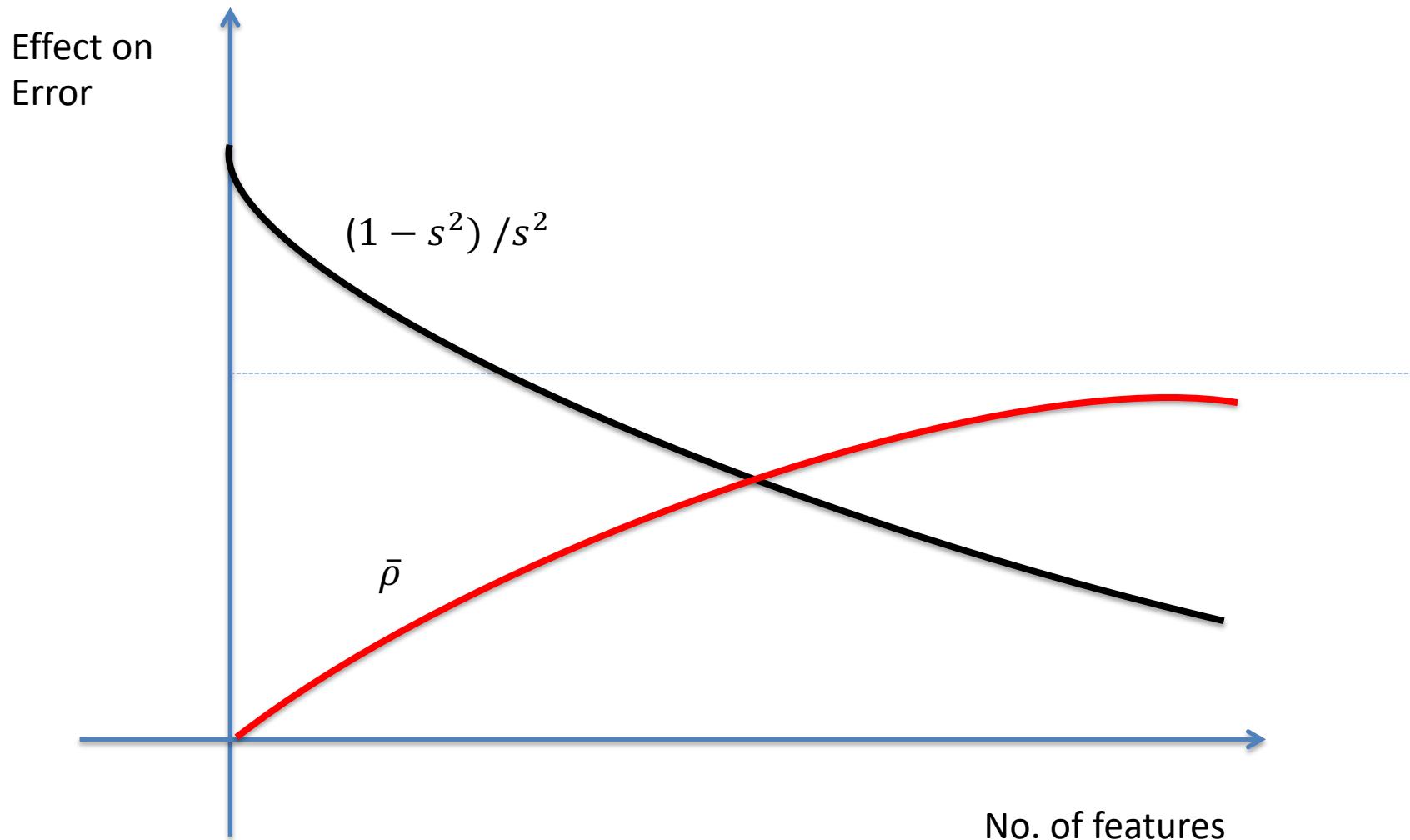


Generalization Error of Random Forests

$$Error \leq \bar{\rho} (1 - s^2) / s^2$$

- $\bar{\rho}$: Correlation between classifiers
- s : is a measure of the strength of the classifier (1-error)
- These two terms are connected!

Tradeoffs



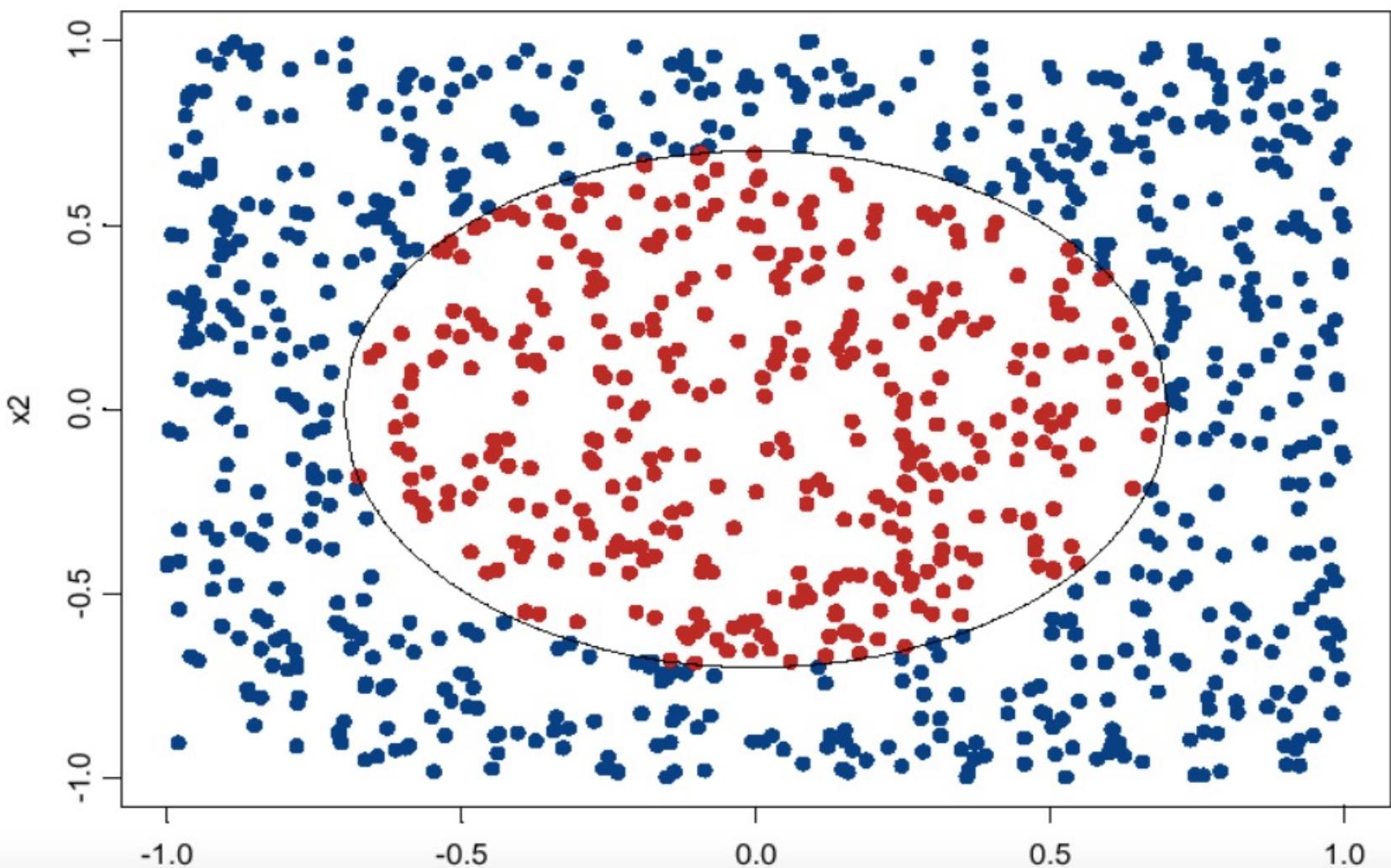
Classification and Regression Trees (CART)

- What if the features are numerical?
- At each level, you construct a linear classifier of the form:

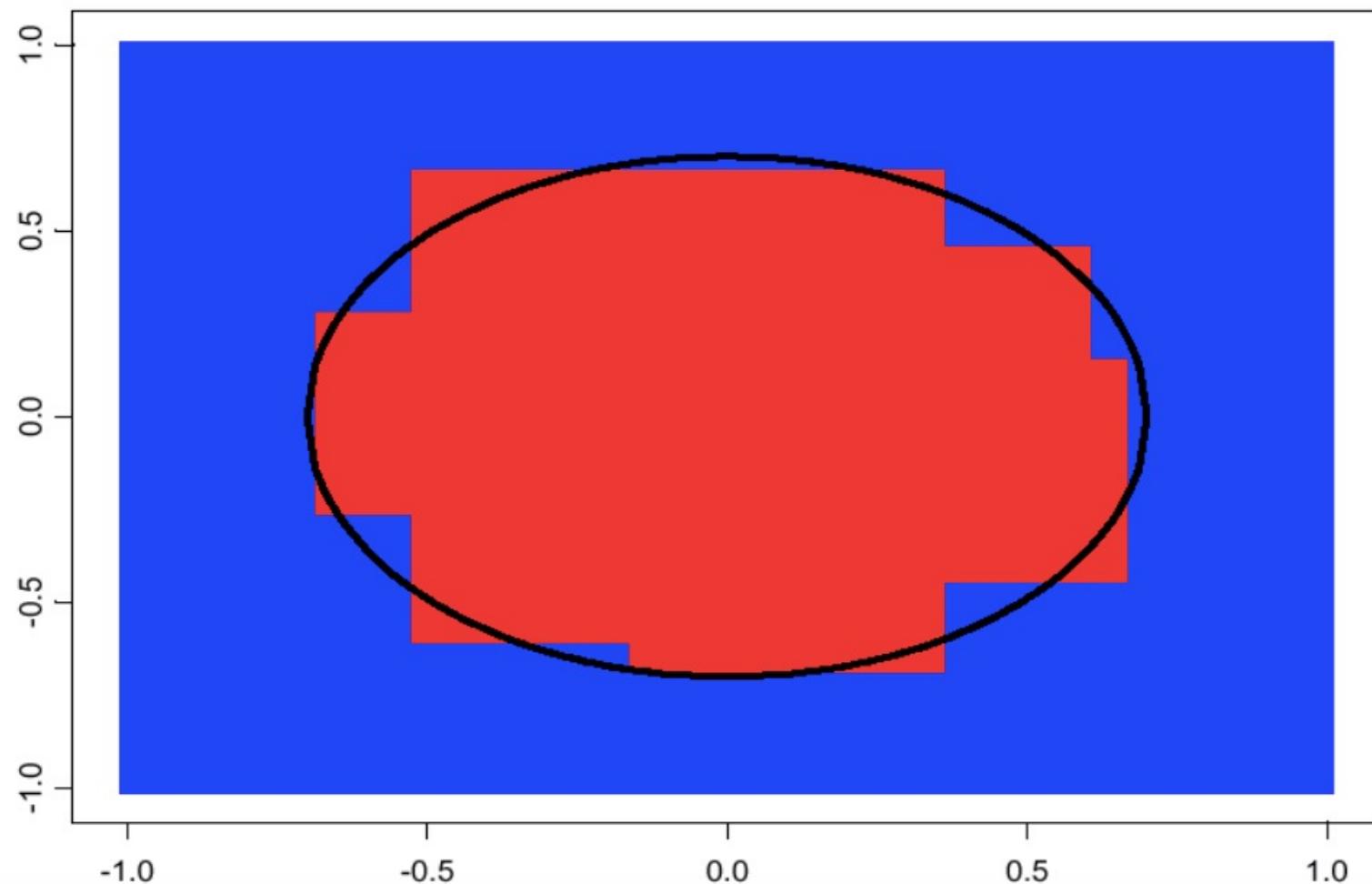
$$a' x + b \geq 0$$

- Continue the same way!

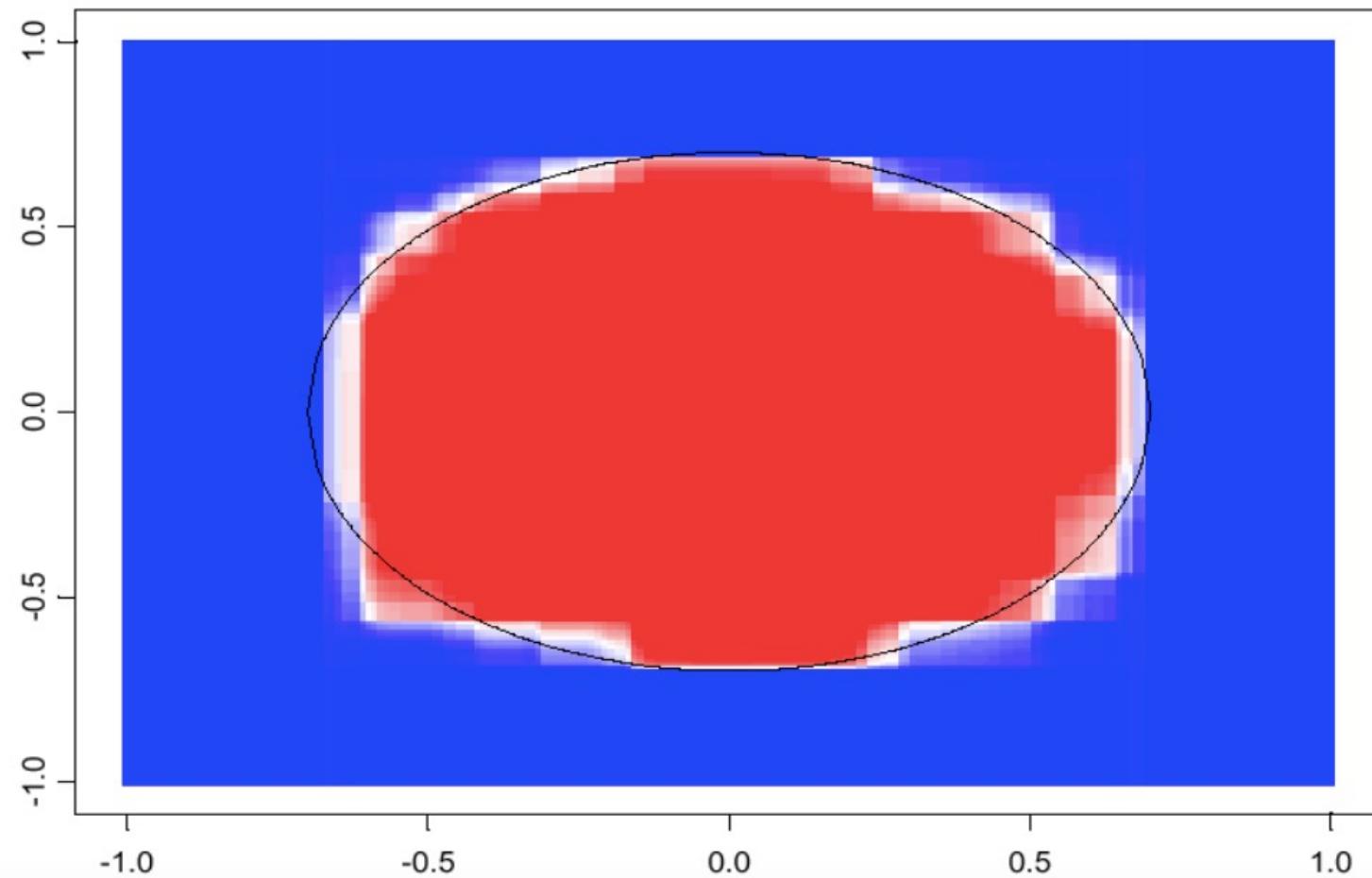
Random Forest



One outcome



Effect of Bagging



Summary

- Overfitting increases variance on test data
- Reduce overfitting by
 - Pruning
 - Bagging
 - Random Forests
- Regression trees

Discussion



Thank You