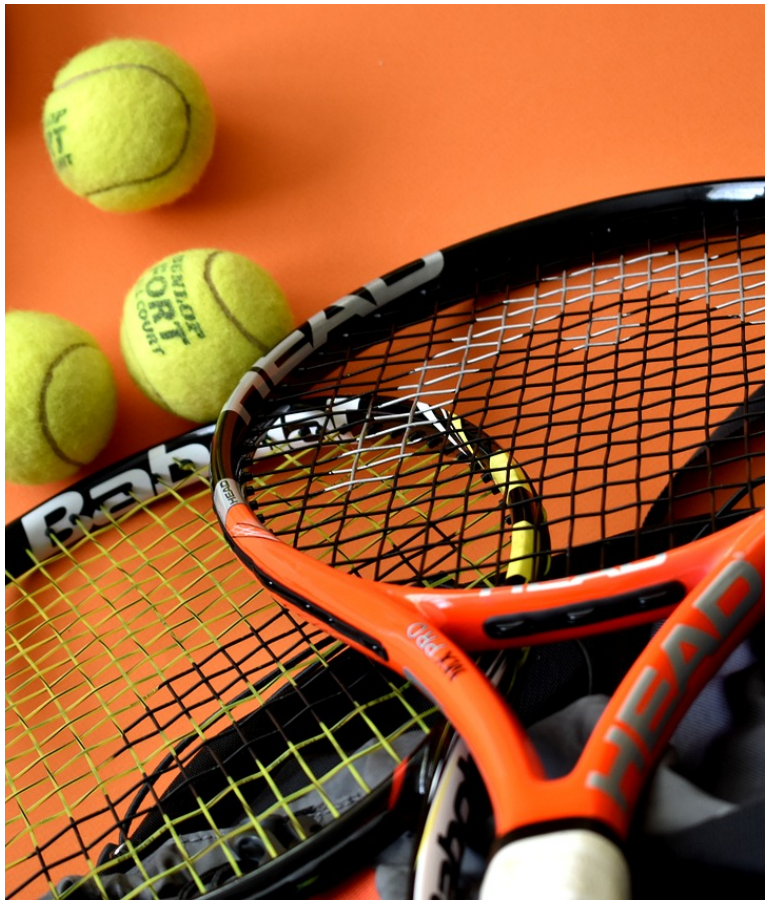
Decision Trees & Random Forests

Decision and Regression Trees


HEAD
HEAD

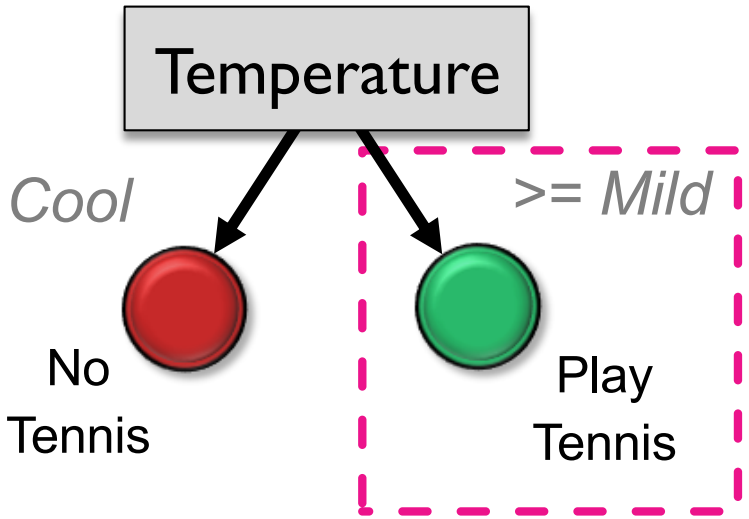
Day	Outlook	Temp	Humidity	Wind	Play Tennis?
01	Sunny	Hot	High	Weak	No
02	Sunny	Hot	High	Strong	No
03	Overcast	Hot	High	Weak	Yes
04	Rain	Mild	High	Weak	Yes
05	Rain	Cool	Nml	Weak	Yes
06	Rain	Cool	Nml	Strong	No
07	Overcast	Cool	Nml	Strong	Yes
08	Sunny	Mild	High	Weak	No
09	Sunny	Cool	Nml	Weak	Yes
10	Rain	Mild	Nml	Weak	Yes
11	Sunny	Mild	Nml	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Nml	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp	Humidity	Wind	Play Tennis?
01	Sunny	Hot	High	Weak	No
02	Sunny	Hot	High	Strong	No
03	**Overcast**	**Hot**	**High**	**Weak**	**Yes**
04	**Rain**	**Mild**	**High**	**Weak**	**Yes**
05	**Rain**	**Cool**	**Normal**	**Weak**	**Yes**
06	Rain	Cool	Normal	Strong	No
07	**Overcast**	**Cool**	**Normal**	**Strong**	**Yes**
08	Sunny	Mild	High	Weak	No
09	**Sunny**	**Cool**	**Normal**	**Weak**	**Yes**
10	**Rain**	**Mild**	**Normal**	**Weak**	**Yes**
11	**Sunny**	**Mild**	**Normal**	**Strong**	**Yes**
12	**Overcast**	**Mild**	**High**	**Strong**	**Yes**
13	**Overcast**	**Hot**	**Normal**	**Weak**	**Yes**
14	Rain	Mild	High	Strong	No

Our choice to play tennis depends on the weather

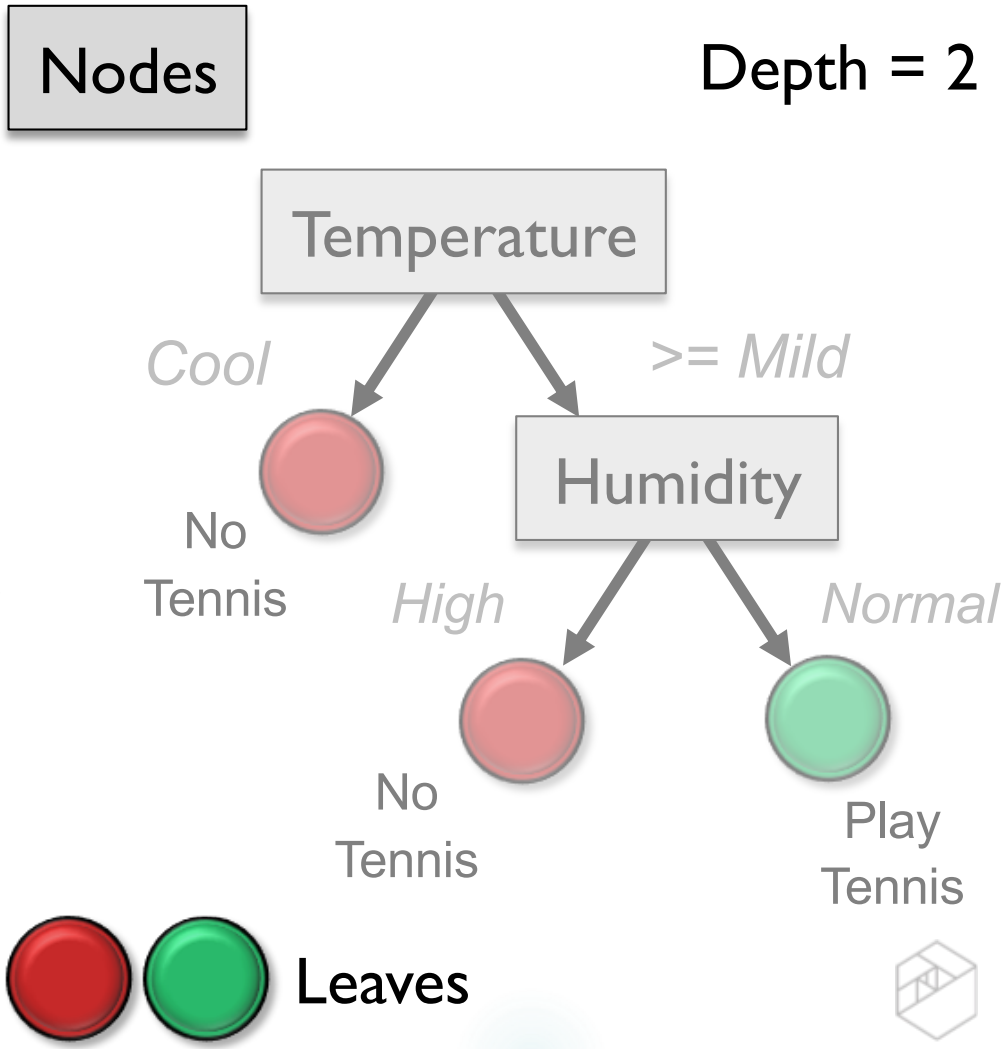
Decision Trees

- Predict whether or not we play tennis from temperature, humidity, wind, etc.
- Segment data based on features to predict results



Decision Trees

- Predict whether or not we play tennis from temperature, humidity, wind, etc.
- Segment data based on features to predict results
- Trees that predict **categorical** results are **decision trees**

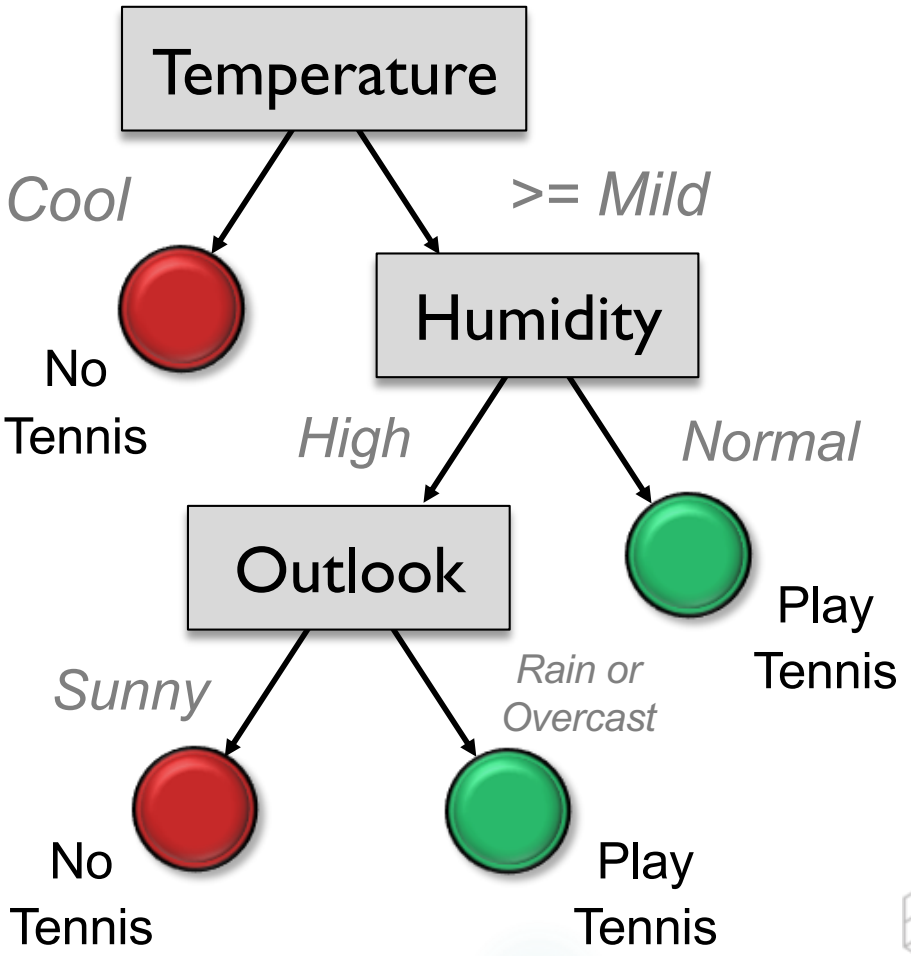


Predictions

Once a decision tree is trained, predict by following split decisions

Example:

- *Mild*
- *High humidity*
- *Overcast*

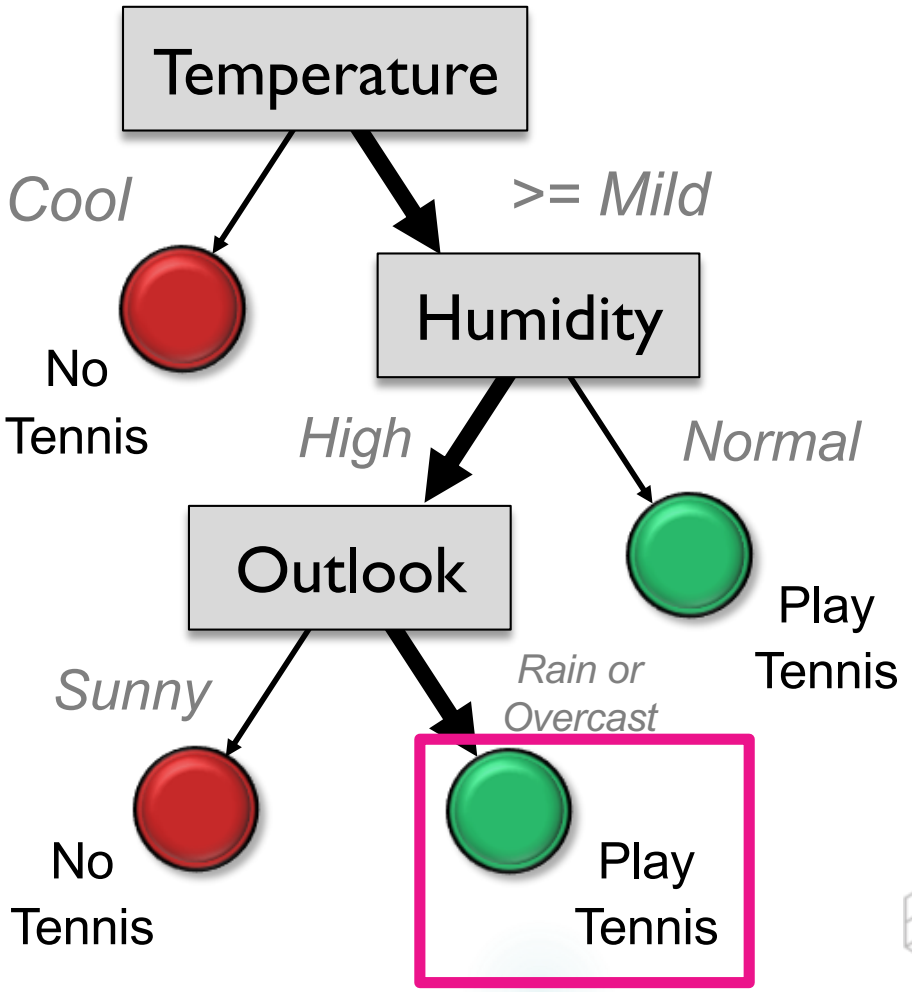


Predictions

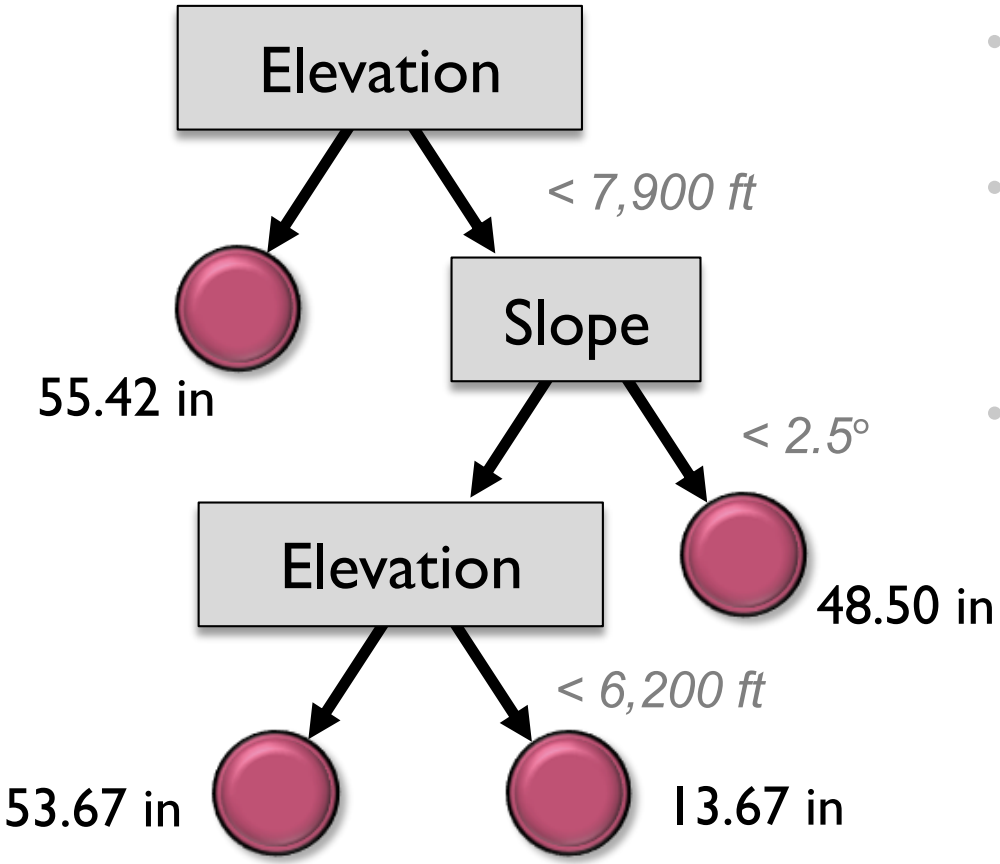
Once a decision tree is trained, predict by following split decisions

Example:

- *Mild*
- *High humidity*
- *Overcast*

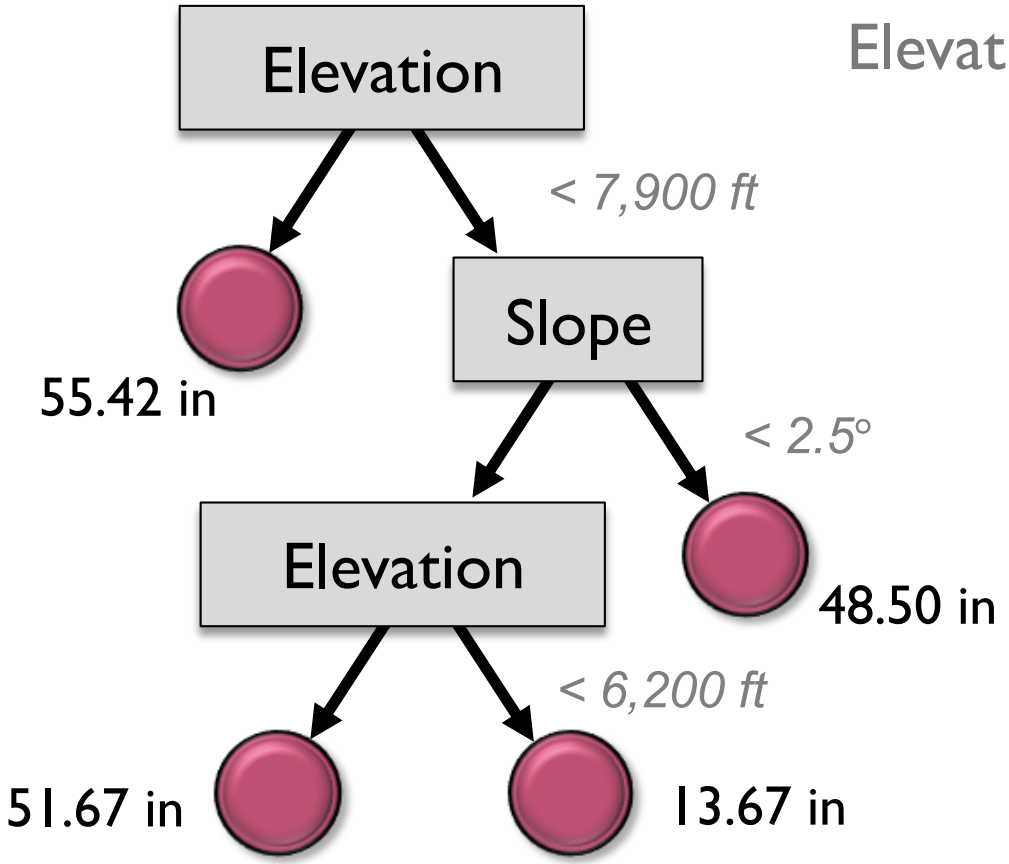


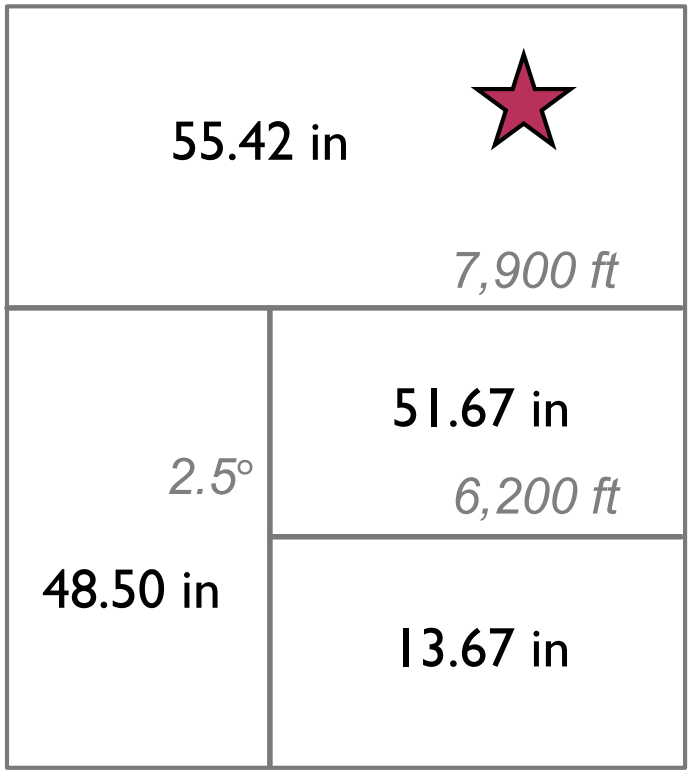
Regression Trees



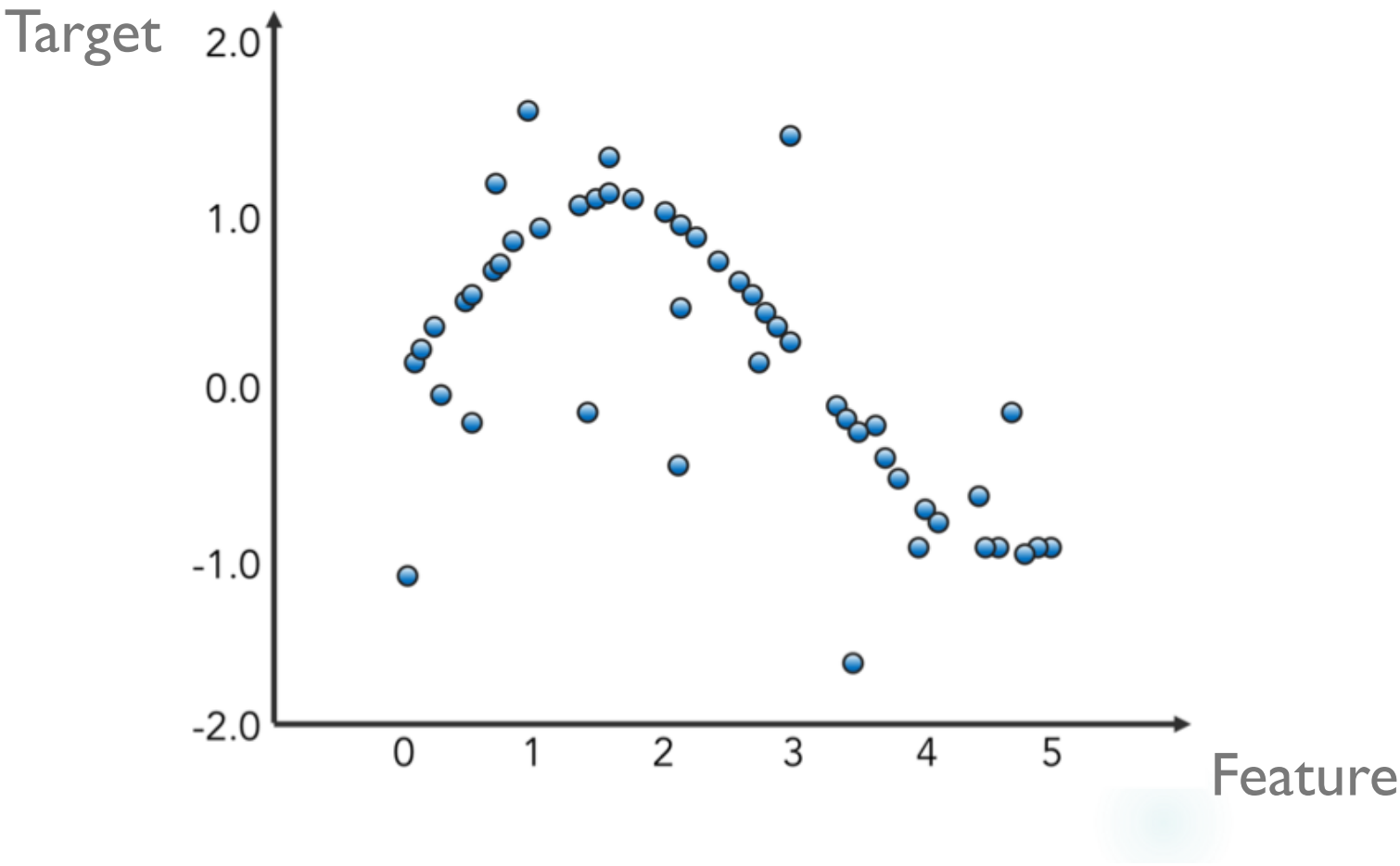
- Predict continuous values
- **Example**: *Himalayas*
  Slope, Elevation → Precipitation
- Values at leaves are averages of members

Trees Make Local Predictions

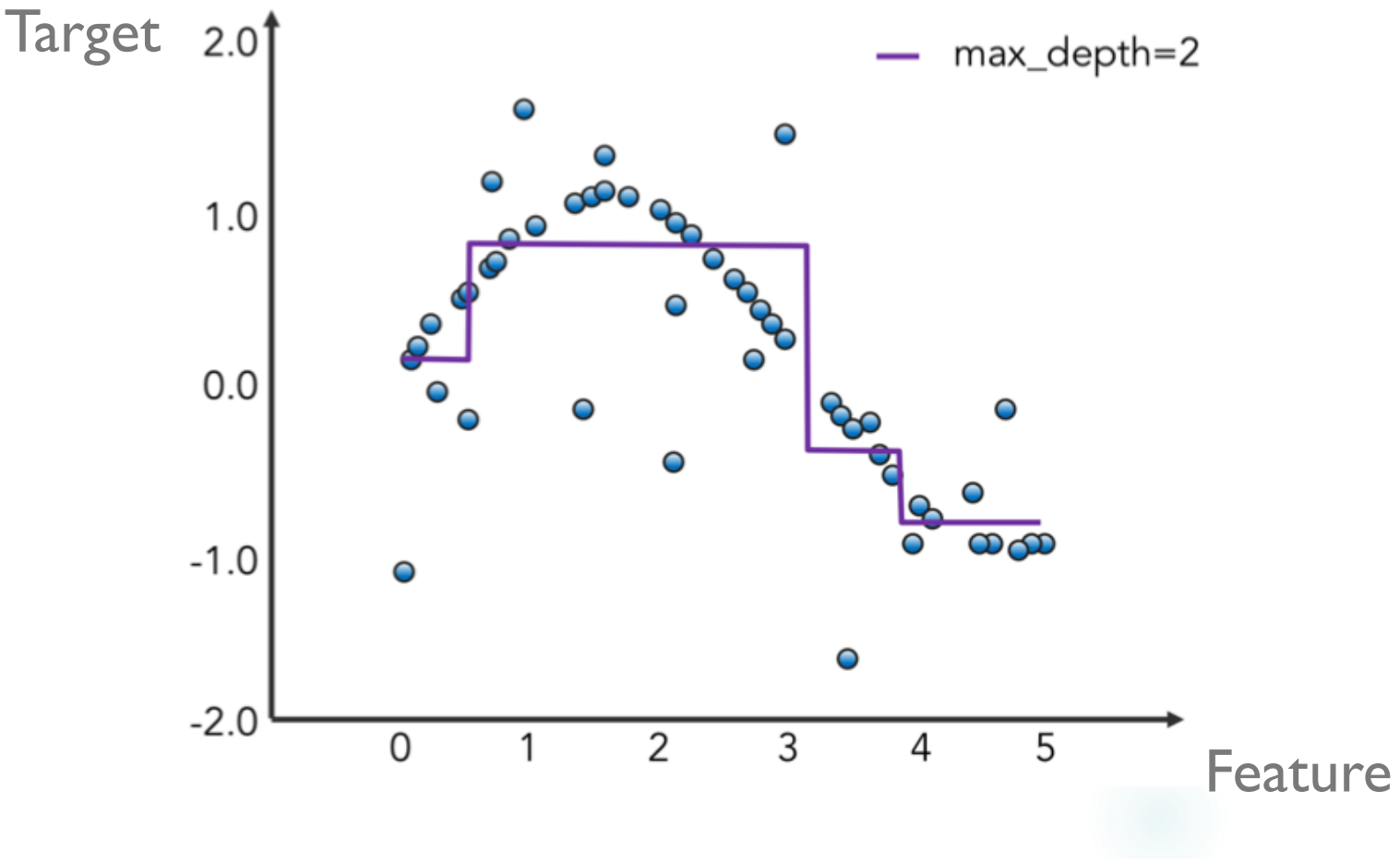




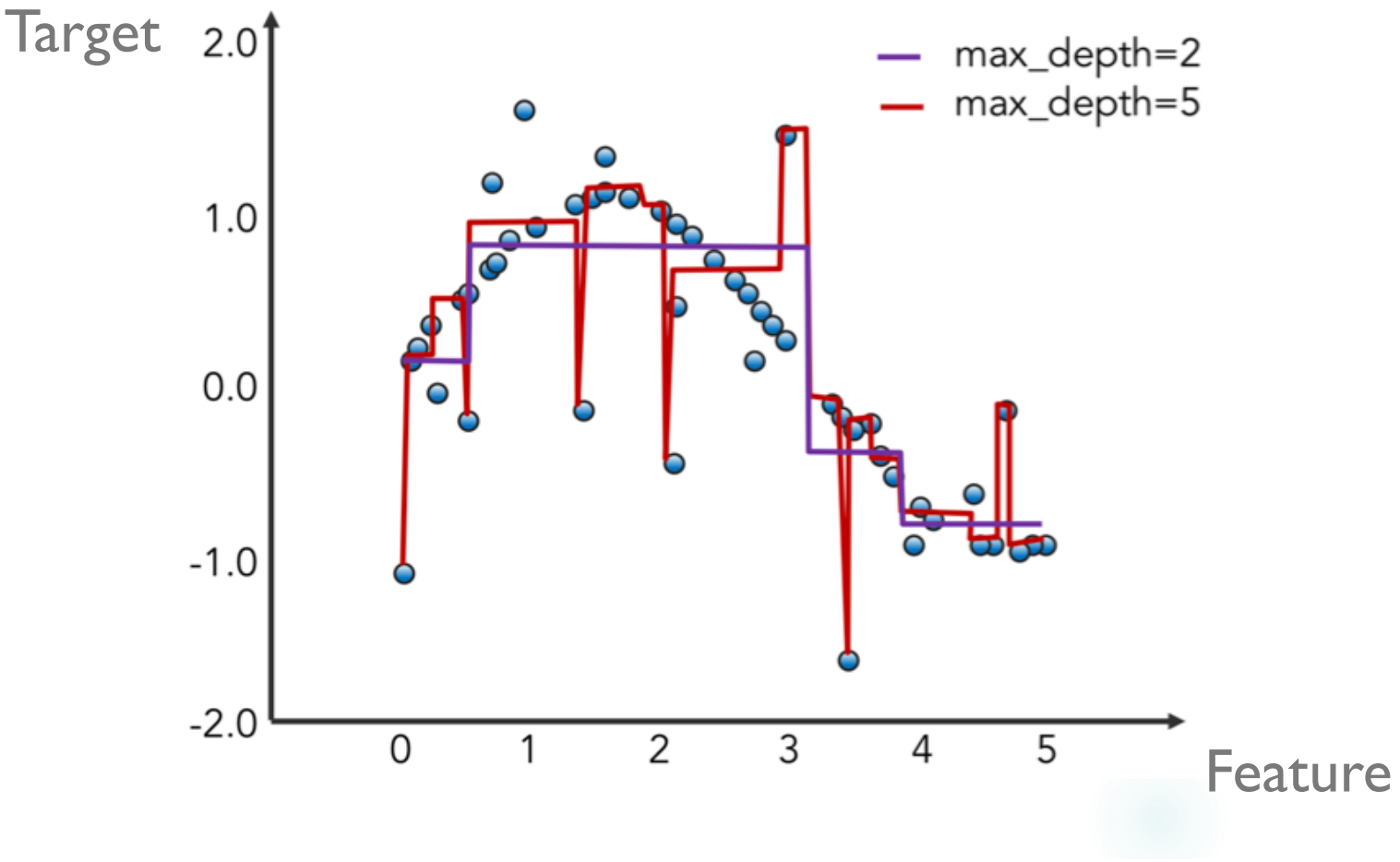
Regression Trees



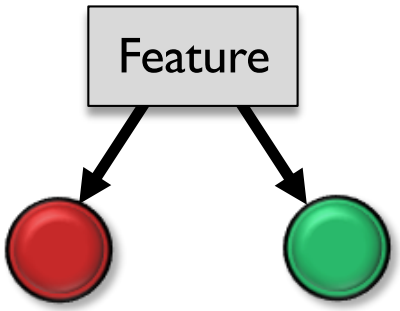
Regression Trees



Regression Trees

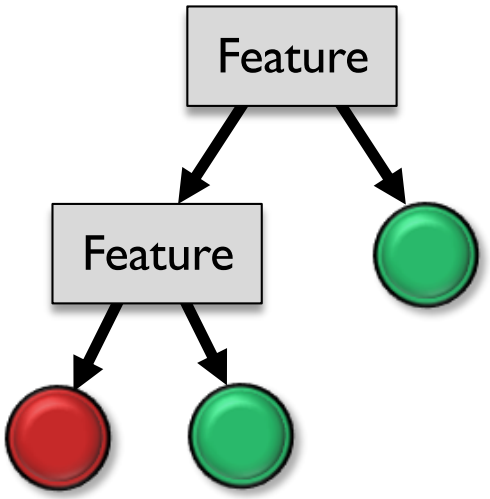


Building a Decision Tree



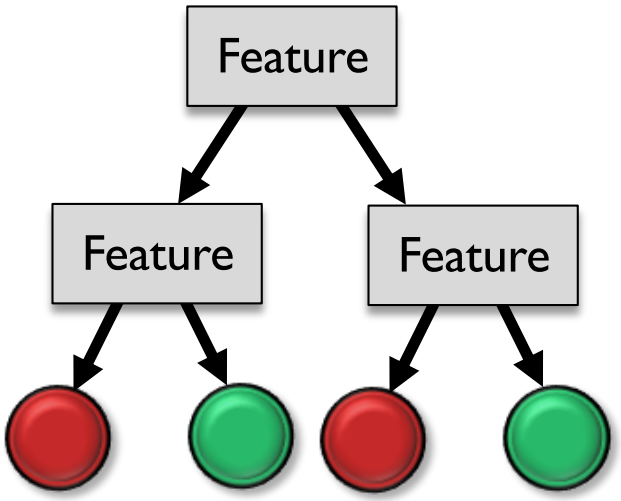
- Select a feature and split data into binary tree

Building a Decision Tree



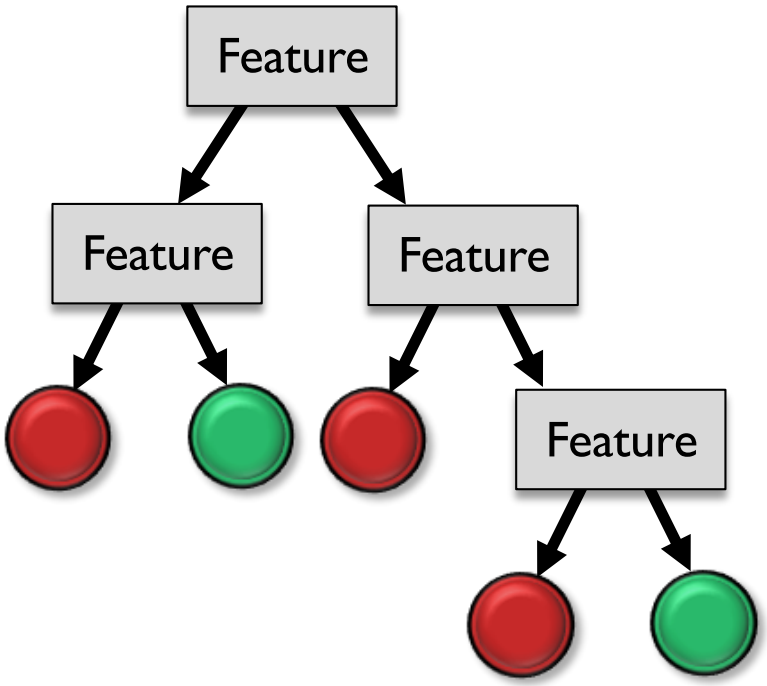
- Select a feature and split data into binary tree
- Continue splitting on available features

Building a Decision Tree



- Select a feature and split data into binary tree
- Continue splitting on available features

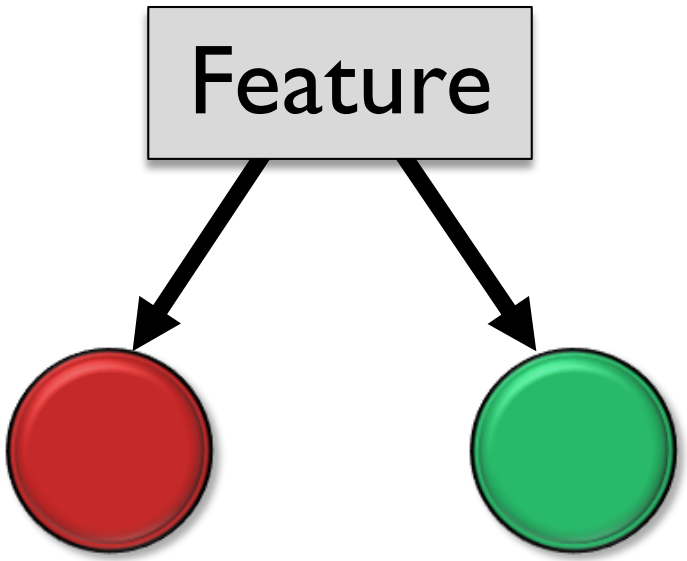
How Long to Continue Splitting?



Split training data until:

- Leaf nodes are pure – only one class per leaf
- Minimum leaf size or maximum depth reached
- Performance metric achieved

How to Form Each Split?



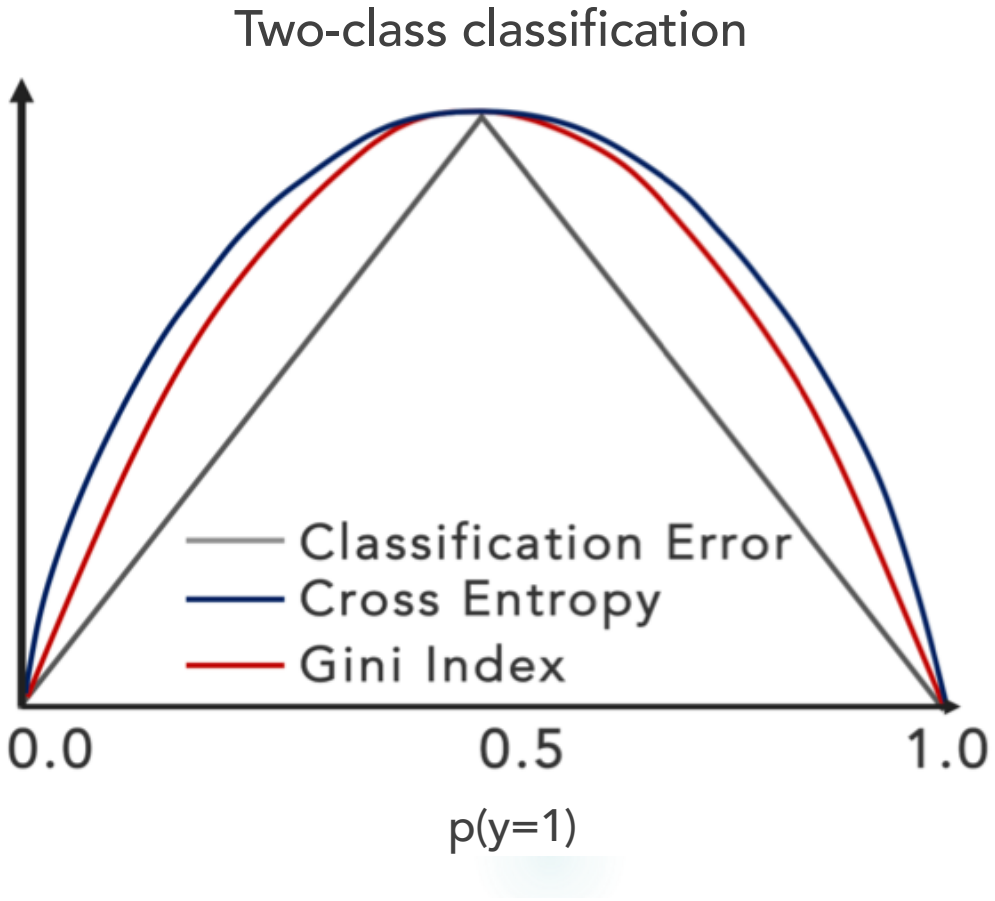
- Use **greedy approach**: find best split at each step without considering later splits
- What is the “best” split?

Information Gain and Impurity

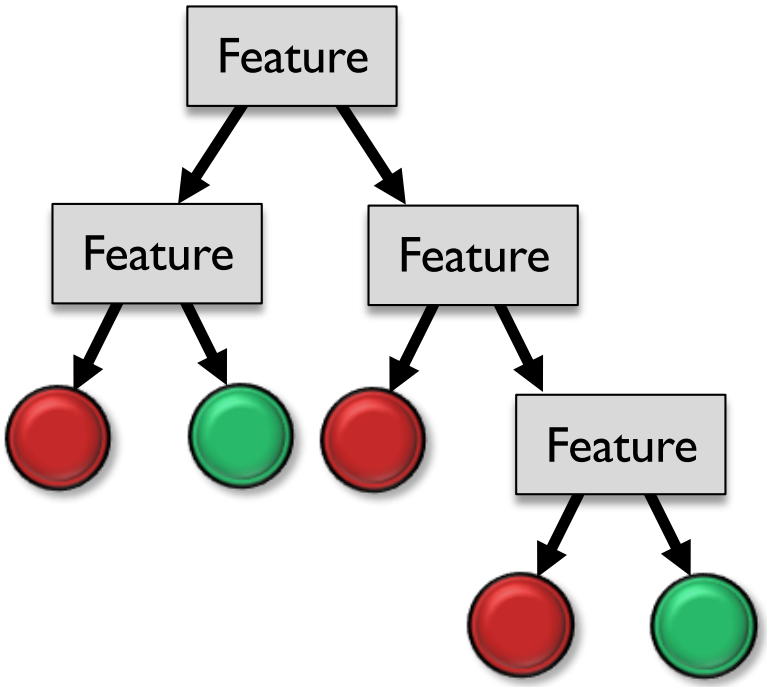
- Choose feature and cutpoint to maximize information gain (reduce Gini impurity)

$$G(\boldsymbol{p}) = \sum_{i=1}^{J} p_i(1-p_i)$$

$$= 1 - \sum_{i=1}^{J} p_i^2$$



Decision Trees Tend to Overfit

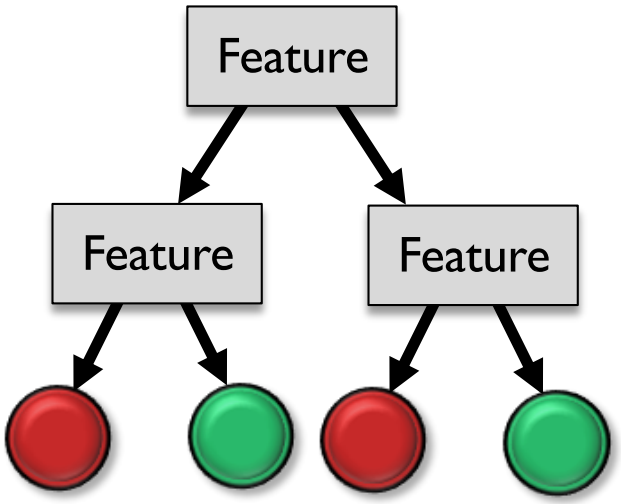


- High variance
- Small changes in data greatly affect prediction

SOLUTION

Prune Trees

Decision Trees Tend to Overfit

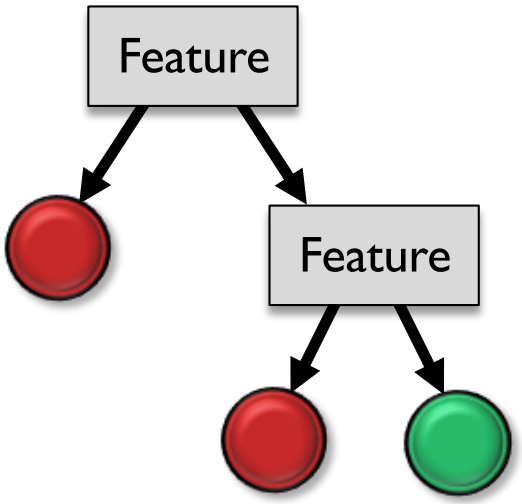


- High variance
- Small changes in data greatly affect prediction

SOLUTION

Prune Trees

Decision Trees Tend to Overfit

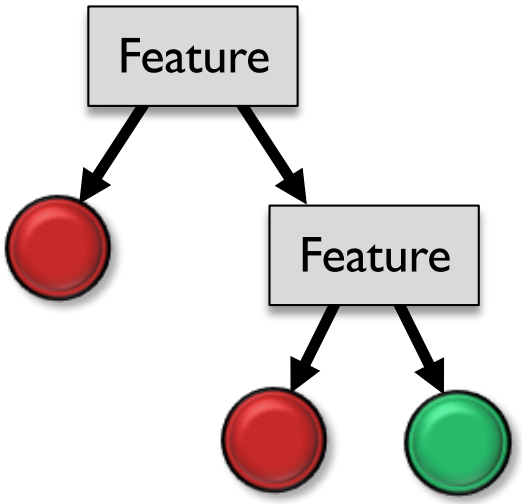


- High variance
- Small changes in data greatly affect prediction

SOLUTION

Prune Trees

Pruning Decision Trees



- Which leaves should be pruned?

SOLUTION

Prune such that classification errors are minimized

Decision/Regression Trees

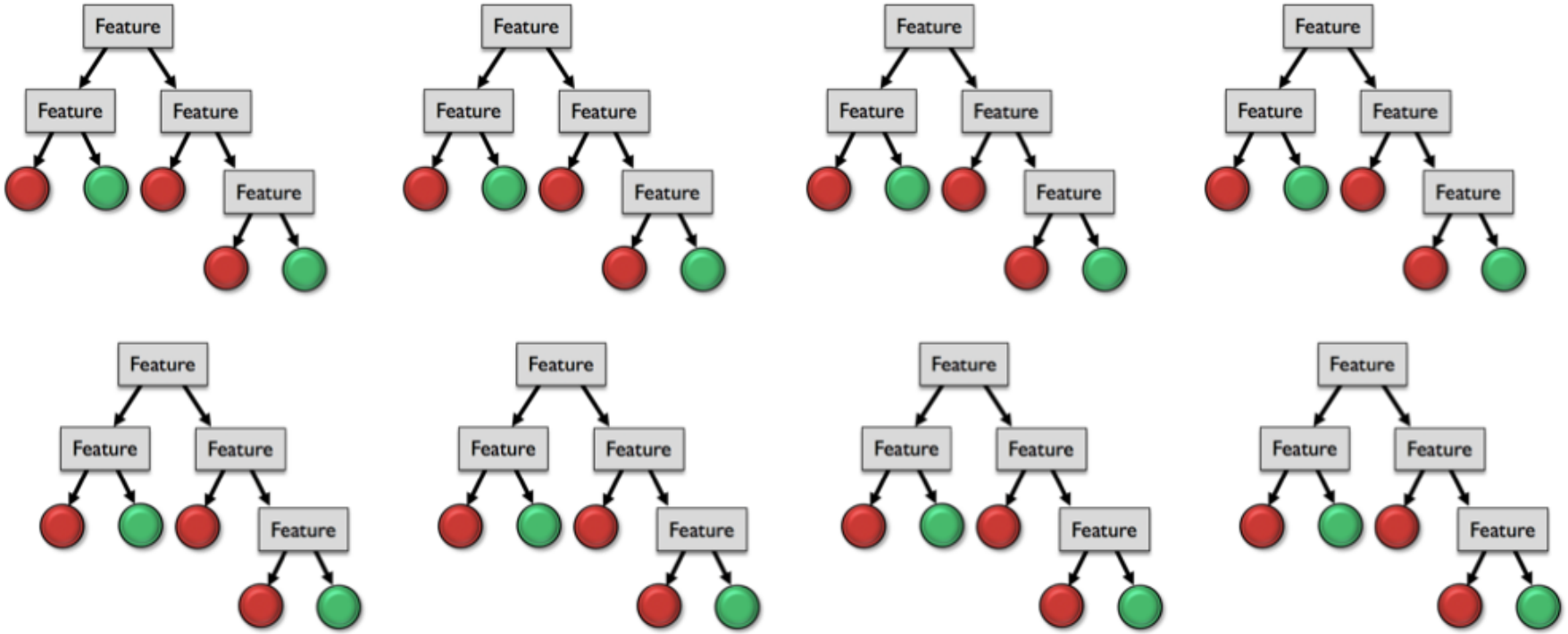
ADVANTAGES

- Easy to interpret
- “If… then… else” logic
- Any data type, handles missings
- No scaling required

DISADVANTAGES

- OVERFITING
- Difficult to generalize to unseen data

Overcome Variance Issue by Ensembling



Bagging

Bagging: **B**ootstrap **Agg**regat**ing**

Ensembling: Improve robustness by combining base learners

BAGGING STEPS

1. Generate new datasets by sampling from training set uniformly with replacement
2. Fit a tree to each new bootstrapped dataset
3. Make predictions by aggregating across trees

Step 1: Bootstraping

Sample training data uniformly with replacement

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

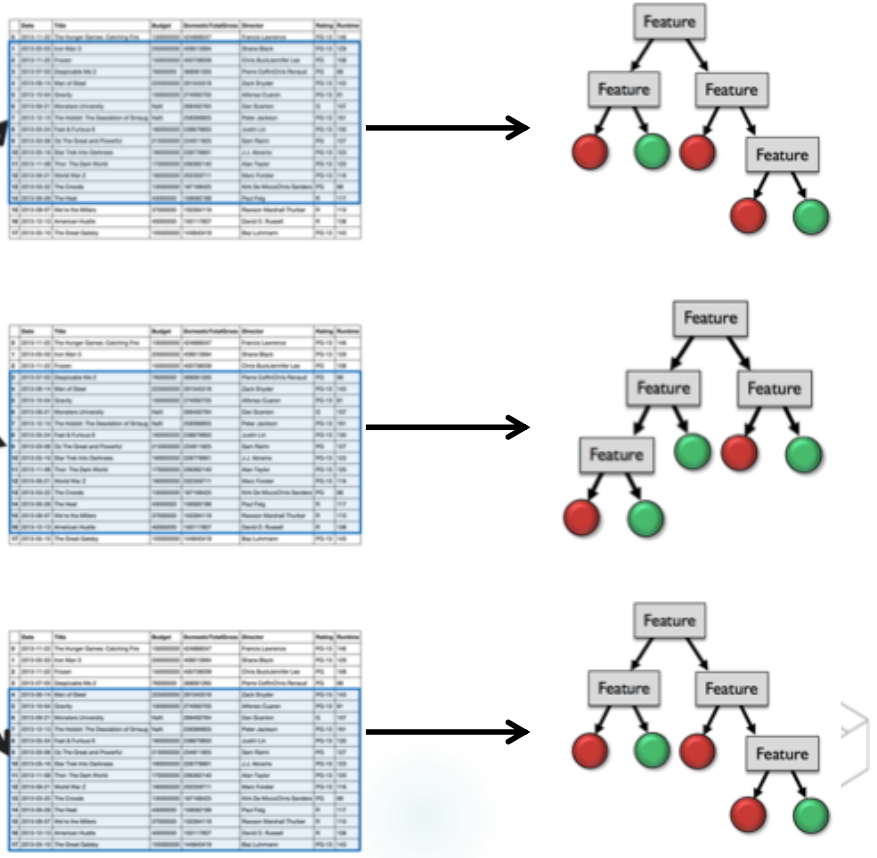
Step 1: Bootstraping

Sample training data uniformly with replacement

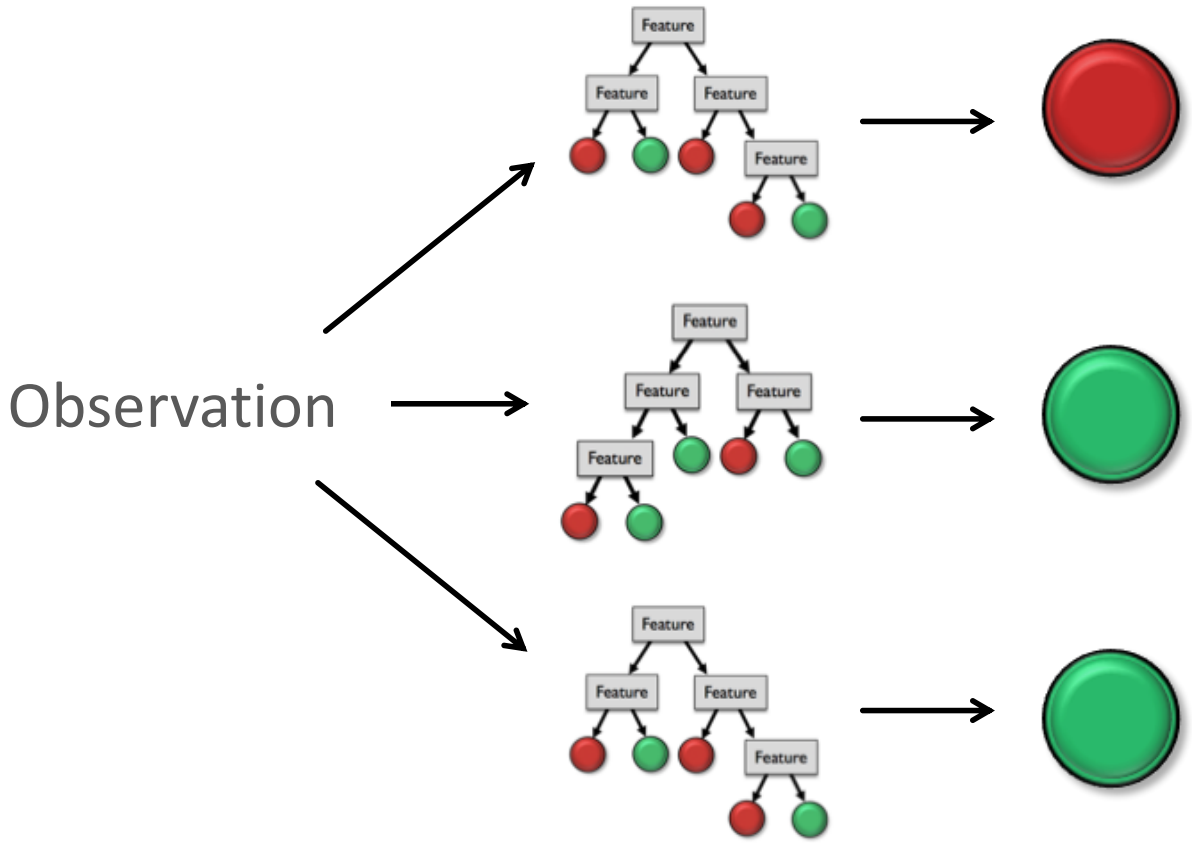
	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143

Step 2: Fit tree to each

	Date	Title	Budget	DomesticTotalGross	Director	Rating	Runtime
0	2013-11-22	The Hunger Games: Catching Fire	130000000	424668047	Francis Lawrence	PG-13	146
1	2013-05-03	Iron Man 3	200000000	409013994	Shane Black	PG-13	129
2	2013-11-22	Frozen	150000000	400738009	Chris BuckJennifer Lee	PG	108
3	2013-07-03	Despicable Me 2	76000000	368061265	Pierre CoffinChris Renaud	PG	98
4	2013-06-14	Man of Steel	225000000	291045518	Zack Snyder	PG-13	143
5	2013-10-04	Gravity	100000000	274092705	Alfonso Cuaron	PG-13	91
6	2013-06-21	Monsters University	NaN	268492764	Dan Scanlon	G	107
7	2013-12-13	The Hobbit: The Desolation of Smaug	NaN	258366855	Peter Jackson	PG-13	161
8	2013-05-24	Fast & Furious 6	160000000	238679850	Justin Lin	PG-13	130
9	2013-03-08	Oz The Great and Powerful	215000000	234911825	Sam Raimi	PG	127
10	2013-05-16	Star Trek Into Darkness	190000000	228778661	J.J. Abrams	PG-13	123
11	2013-11-08	Thor: The Dark World	170000000	206362140	Alan Taylor	PG-13	120
12	2013-06-21	World War Z	190000000	202359711	Marc Forster	PG-13	116
13	2013-03-22	The Croods	135000000	187168425	Kirk De MiccoChris Sanders	PG	98
14	2013-06-28	The Heat	43000000	159582188	Paul Feig	R	117
15	2013-08-07	We're the Millers	37000000	150394119	Rawson Marshall Thurber	R	110
16	2013-12-13	American Hustle	40000000	150117807	David O. Russell	R	138
17	2013-05-10	The Great Gatsby	105000000	144840419	Baz Luhrmann	PG-13	143



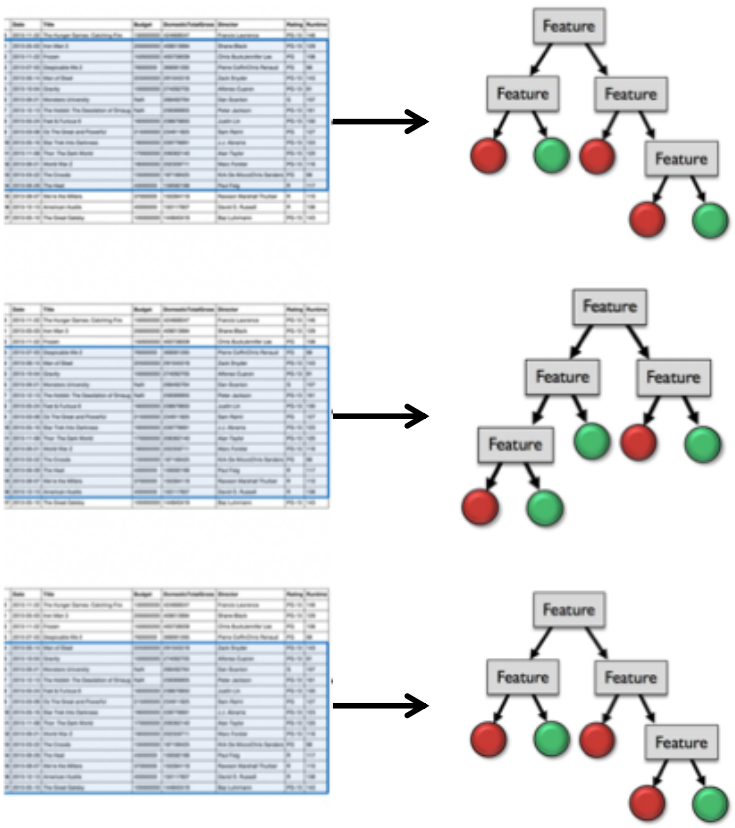
Step 3: Aggregate results



Classification:
Vote for final prediction

Regression:
Average value

Bagging



- Including more trees reduces variance without greatly increasing bias
- Corrects decision tree overfitting
- Can yield correlated trees if some features especially strong predictors

SOLUTION

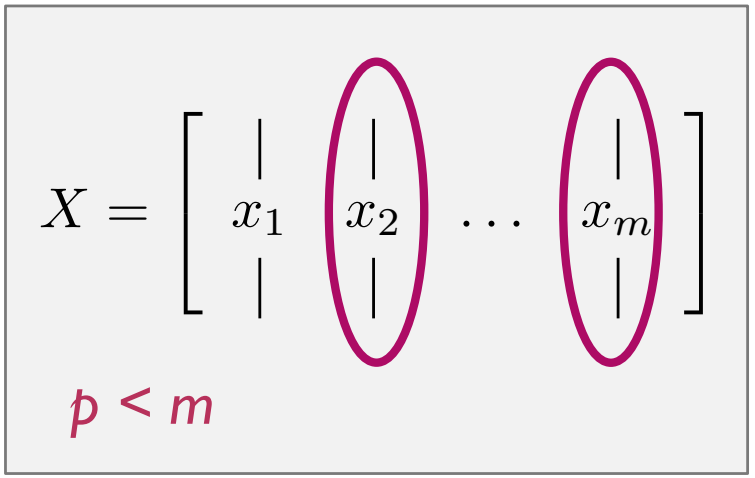
Random Forests

Random Forests

Bagging with more randomness!

Feature Bagging

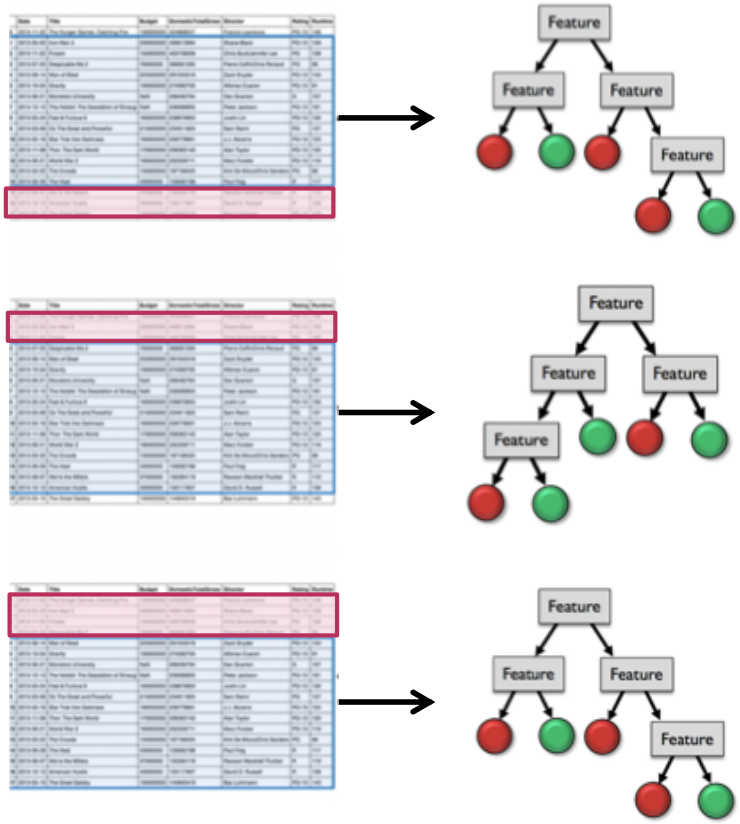
- Introduce randomness when building each tree
- At each split, only consider a subset of the *m* features
- Each feature now has opportunity to influence outcome
- Trees now decorrelated



Random Forest Rough Guidelines

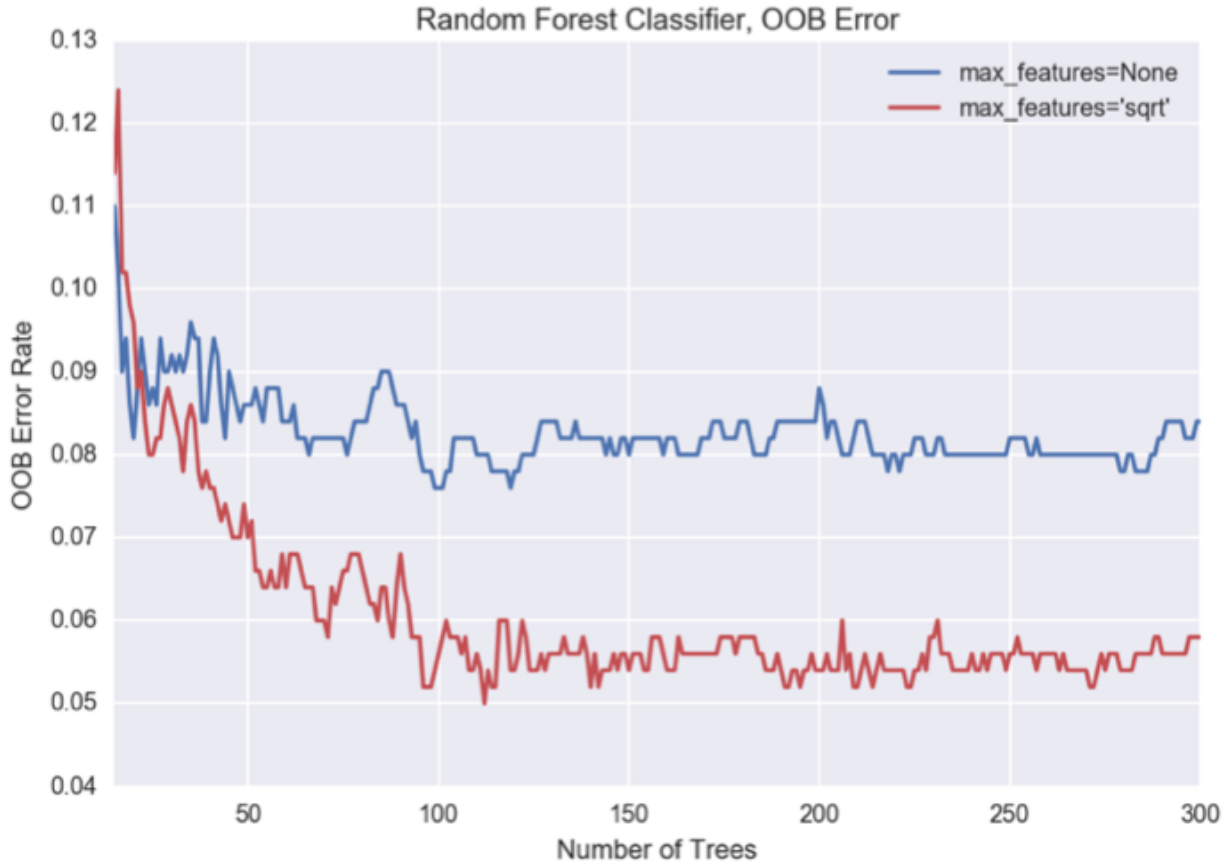
- Typically comprised of 50 – X00 trees
- Each tree is grown deep without pruning
- At each split, typically consider about
  - $\sqrt{m}$ features for classification
  - $m/3$ features for regression

Out-of-Bag Error



- Measure tree error on samples not included in bootstrap
- Validate number of trees in forest by monitoring OOB

Out-of-Bag Error



Feature Importance

- Rank features as being more or less useful to model
- Helps with model interpretability
- In sklearn, average Gini importance
- Note: Feature importance does not tell you *how/why* a feature is important

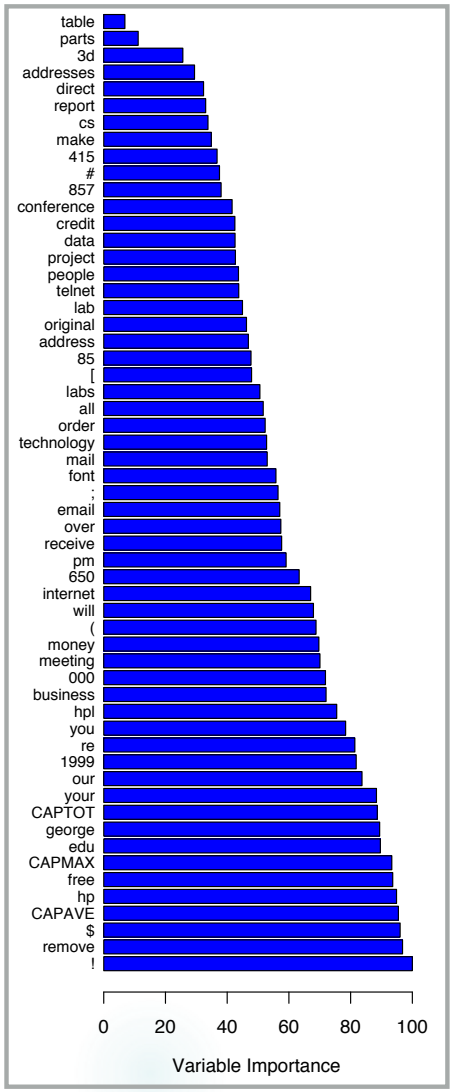
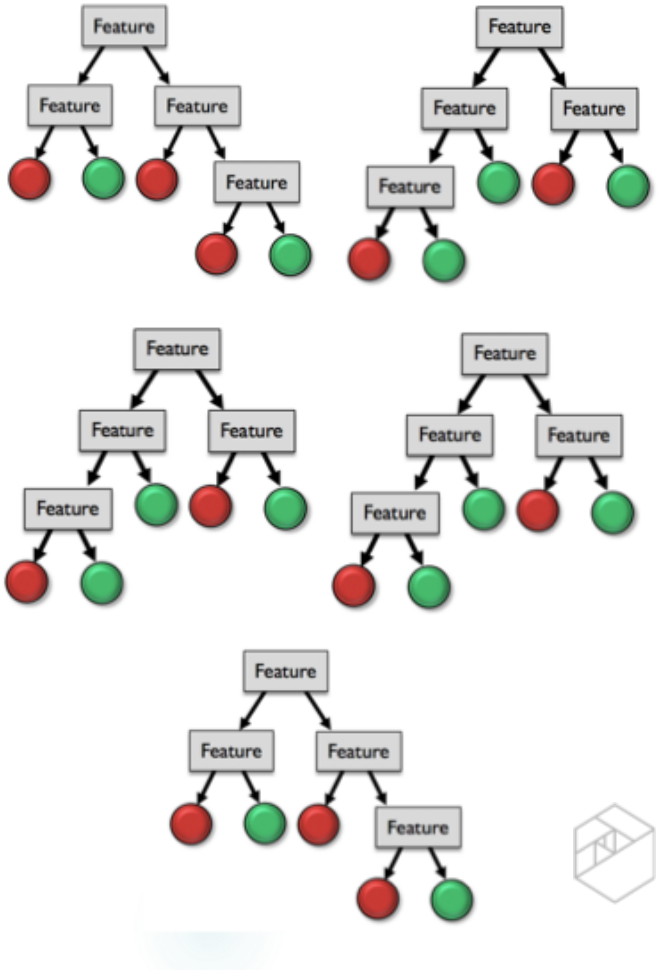


Figure: Hastie 2009

Extra Trees

- **Ext**remely **Ra**ndomized **Trees**
- Instead of bootstrapping, grow trees by selecting ***random cutpoint*** for random subset of features at each split
- Aggregate across trees to predict



Random Forests

ADVANTAGES

- Any data type, handles missings
- No scaling required
- Few tuning parameters
- Can handle "small n, large m"
- Well suited for parallelization
- "Good for folks with a tight deadlines"

DISADVANTAGES

- Less interpretable than decision trees
- Poor with categorical variables with many categories
- Other algorithms may outperform (e.g. XGBoost)

Questions?

Appendix

Bias-Variance Tradeoff

- Synthetic dataset including from $m = 50$ features
- Considering all 50 features at each split leads to higher variance
- Considering a smaller number of features yields model bias

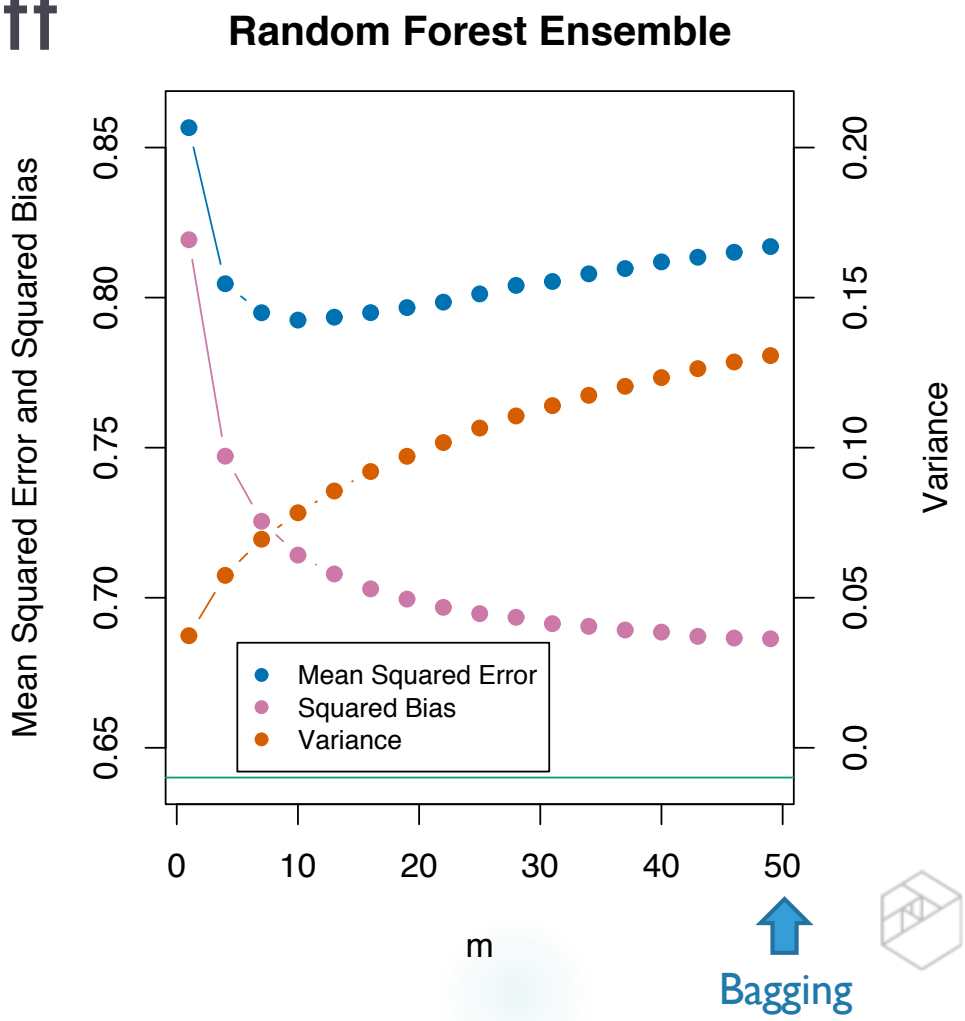


Figure: Hastie 2009

RandomForest: sklearn Syntax

Import the class containing the classification method

```
from sklearn.ensemble import RandomForestClassifier
```

Create an instance of the class

```
RC = RandomForestClassifier(n_estimators=100, max_features=10)
```

Fit the instance on the data and make new classifications on test data

```
RC.fit(X_train, y_train)
y_predict = RC.predict(X_test)
```

Tune parameters with cross-validation.

Use `RandomForestRegressor` for regression.