

10-301/601: Introduction to Machine Learning Lecture 27 – Random Forests

Henry Chai

7/22/24

Front Matter

- Announcements
 - HW9 released 7/18, due 7/25 at 11:59 PM
 - Please be mindful of your grace day usage!
 - **No lecture or instructor OH on 7/23 (tomorrow)**
- Recommended Readings
 - None

Final Logistics

- Time and place:
 - Friday, 8/2 from TBD to TBD in TBD
- Closed book/notes
 - 1-page cheatsheet allowed, both back and front; can be typeset or handwritten

Final Coverage

- Lectures: 15 – 27 (through today's lecture)
 - Algorithmic Bias
 - Unsupervised Learning: Dimensionality Reduction, Clustering
 - Deep Learning : CNNs, RNNs, Attention and Transformers
 - Reinforcement Learning
 - Learning Theory
 - Ensemble Methods: Boosting, Random Forests
- **The final is *not* cumulative:** pre-midterm content may be referenced but will not be the *primary* focus of any question

Final Preparation

- Review final practice problems, posted to the course website (under [Recitations](#))
- Attend the exam review recitation on 7/25
- Review this year's homeworks
- Consider whether you understand the “Key Takeaways” for each lecture / section
- Write your cheat sheet

Recall: The Netflix Prize

Netfli Prize **COMPLETED**

Home Rules Leaderboard Update Download

Leaderboard

Showing Test Score. [Click here to show quiz score](#)

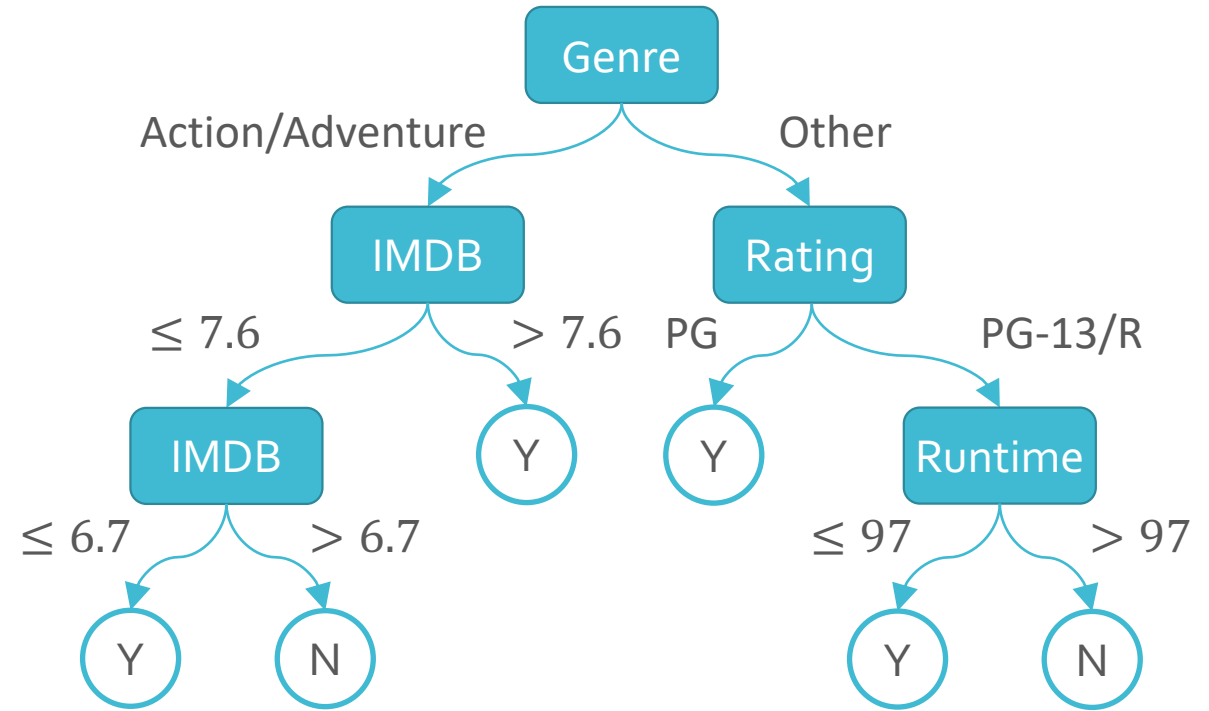
Display top leaders.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

MovieID	Runtime	Genre	Budget	Year	IMDB	Rating	Liked?
1	124	Action	18M	1980	8.7	PG	Y
2	105	Action	30M	1984	7.8	PG	Y
3	103	Comedy	6M	1986	7.8	PG-13	N
4	98	Adventure	16M	1987	8.1	PG	Y
5	128	Comedy	16.4M	1989	8.1	PG	Y
6	120	Comedy	11M	1992	7.6	R	N
7	120	Drama	14.5M	1996	6.7	PG-13	N
8	136	Action	115M	1999	6.5	PG	Y
9	90	Action	90M	2001	6.6	PG-13	Y
10	161	Adventure	100M	2002	7.4	PG	N
11	201	Action	94M	2003	8.9	PG-13	Y
12	94	Comedy	26M	2004	7.2	PG-13	Y
13	157	Biography	100M	2007	7.8	R	N
14	128	Action	110M	2007	7.1	PG-13	N
15	107	Drama	39M	2009	7.1	PG-13	N
16	158	Drama	61M	2012	7.6	PG-13	N
17	169	Adventure	165M	2014	8.6	PG-13	Y
18	100	Biography	9M	2016	6.7	R	N
19	130	Action	180M	2017	7.9	PG-13	Y
20	141	Action	275M	2019	6.5	PG-13	Y

Movie Recommendations

MovieID	Runtime	Genre	Budget	Year	IMDB	Rating	Liked?
1	124	Action	18M	1980	8.7	PG	Y
2	105	Action	30M	1984	7.8	PG	Y
3	103	Comedy	6M	1986	7.8	PG-13	N
4	98	Adventure	16M	1987	8.1	PG	Y
5	128	Comedy	16.4M	1989	8.1	PG	Y
6	120	Comedy	11M	1992	7.6	R	N
7	120	Drama	14.5M	1996	6.7	PG-13	N
8	136	Action	115M	1999	6.5	PG	Y
9	90	Action	90M	2001	6.6	PG-13	Y
10	161	Adventure	100M	2002	7.4	PG	N
11	201	Action	94M	2003	8.9	PG-13	Y
12	94	Comedy	26M	2004	7.2	PG-13	Y
13	157	Biography	100M	2007	7.8	R	N
14	128	Action	110M	2007	7.1	PG-13	N
15	107	Drama	39M	2009	7.1	PG-13	N
16	158	Drama	61M	2012	7.6	PG-13	N
17	169	Adventure	165M	2014	8.6	PG-13	Y
18	100	Biography	9M	2016	6.7	R	N
19	130	Action	180M	2017	7.9	PG-13	Y
20	141	Action	275M	2019	6.5	PG-13	Y

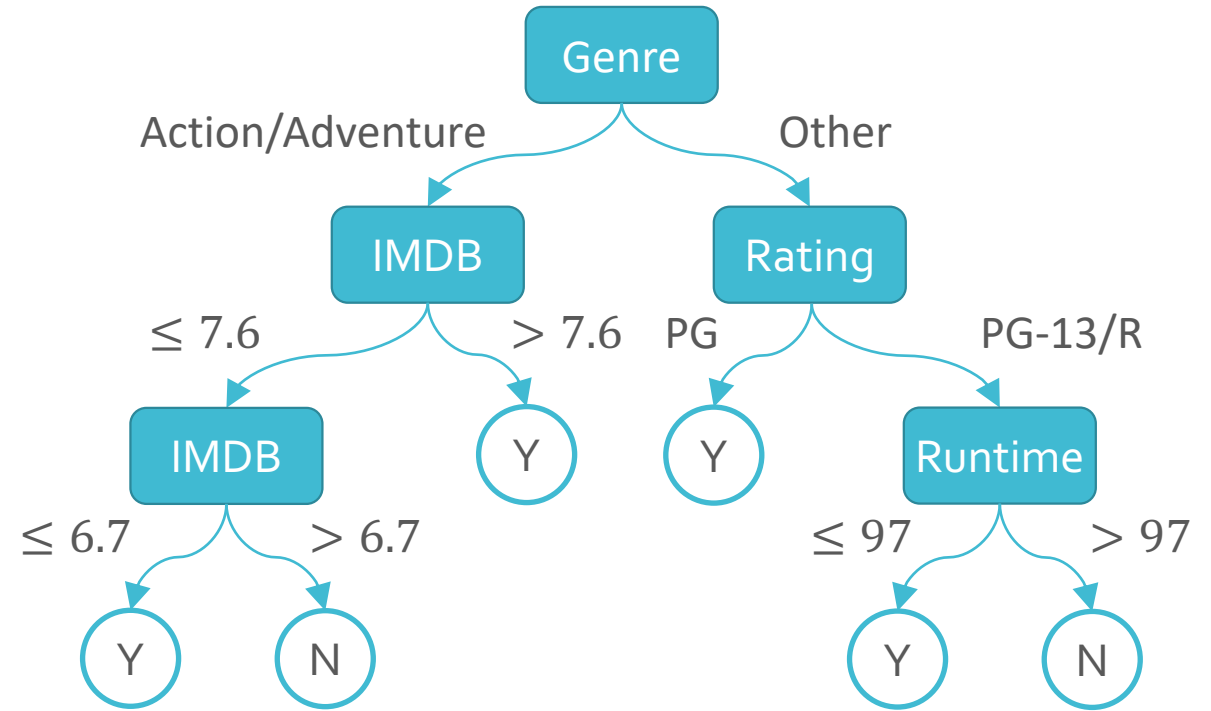


Decision Trees

Decision Trees: Pros & Cons

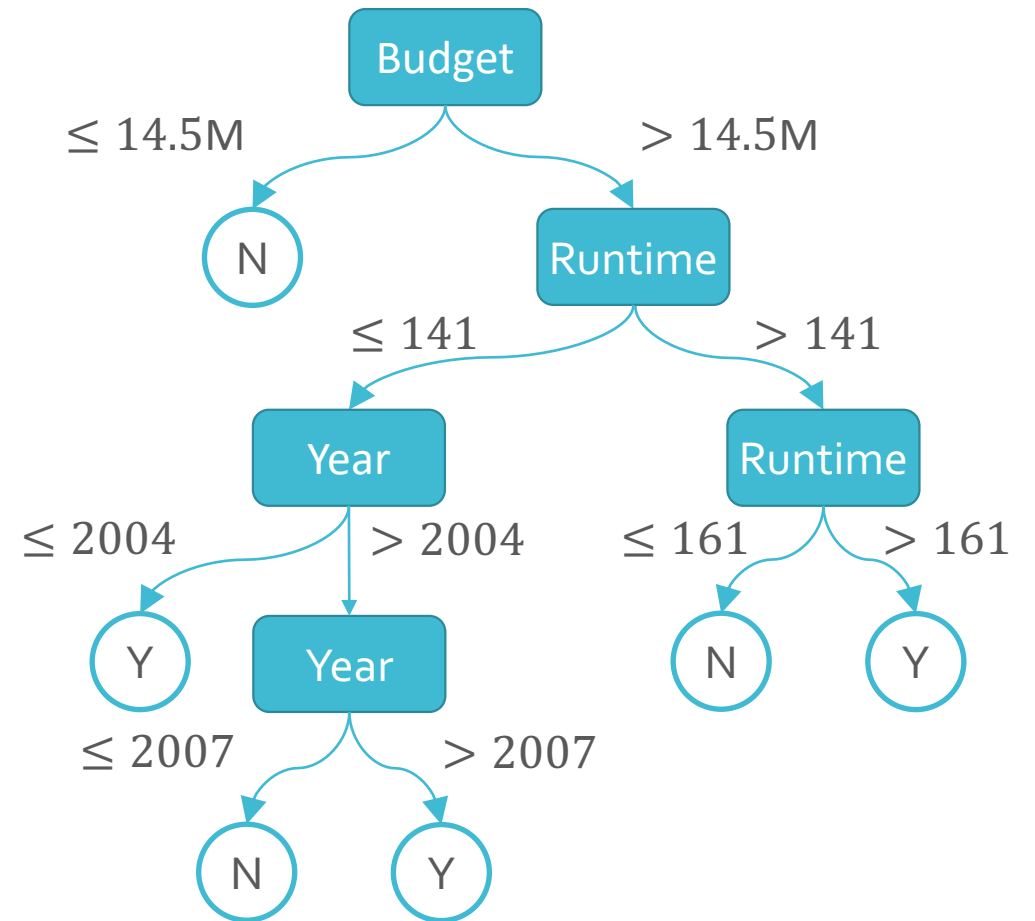
- Pros
 - Interpretable
 - Efficient (computational cost and storage)
 - Can be used for classification and regression tasks
 - Compatible with categorical and real-valued features
- Cons
 - Learned greedily: each split only considers the immediate impact on the splitting criterion
 - Not guaranteed to find the smallest (fewest number of splits) tree that achieves a training error rate of 0.
 - Prone to overfit
 - Limited expressivity (especially short trees, i.e., stumps)
 - Can be addressed via boosting
 - Highly variable
 - Can be addressed via bagging → random forests

MovieID	Runtime	Genre	Budget	Year	IMDB	Rating	Liked?
1	124	Action	18M	1980	8.7	PG	Y
2	105	Action	30M	1984	7.8	PG	Y
3	103	Comedy	6M	1986	7.8	PG-13	N
4	98	Adventure	16M	1987	8.1	PG	Y
5	128	Comedy	16.4M	1989	8.1	PG	Y
6	120	Comedy	11M	1992	7.6	R	N
7	120	Drama	14.5M	1996	6.7	PG-13	N
8	136	Action	115M	1999	6.5	PG	Y
9	90	Action	90M	2001	6.6	PG-13	Y
10	161	Adventure	100M	2002	7.4	PG	N
11	201	Action	94M	2003	8.9	PG-13	Y
12	94	Comedy	26M	2004	7.2	PG-13	Y
13	157	Biography	100M	2007	7.8	R	N
14	128	Action	110M	2007	7.1	PG-13	N
15	107	Drama	39M	2009	7.1	PG-13	N
16	158	Drama	61M	2012	7.6	PG-13	N
17	169	Adventure	165M	2014	8.6	PG-13	Y
18	100	Biography	9M	2016	6.7	R	N
19	130	Action	180M	2017	7.9	PG-13	Y
20	141	Action	275M	2019	6.5	PG-13	Y

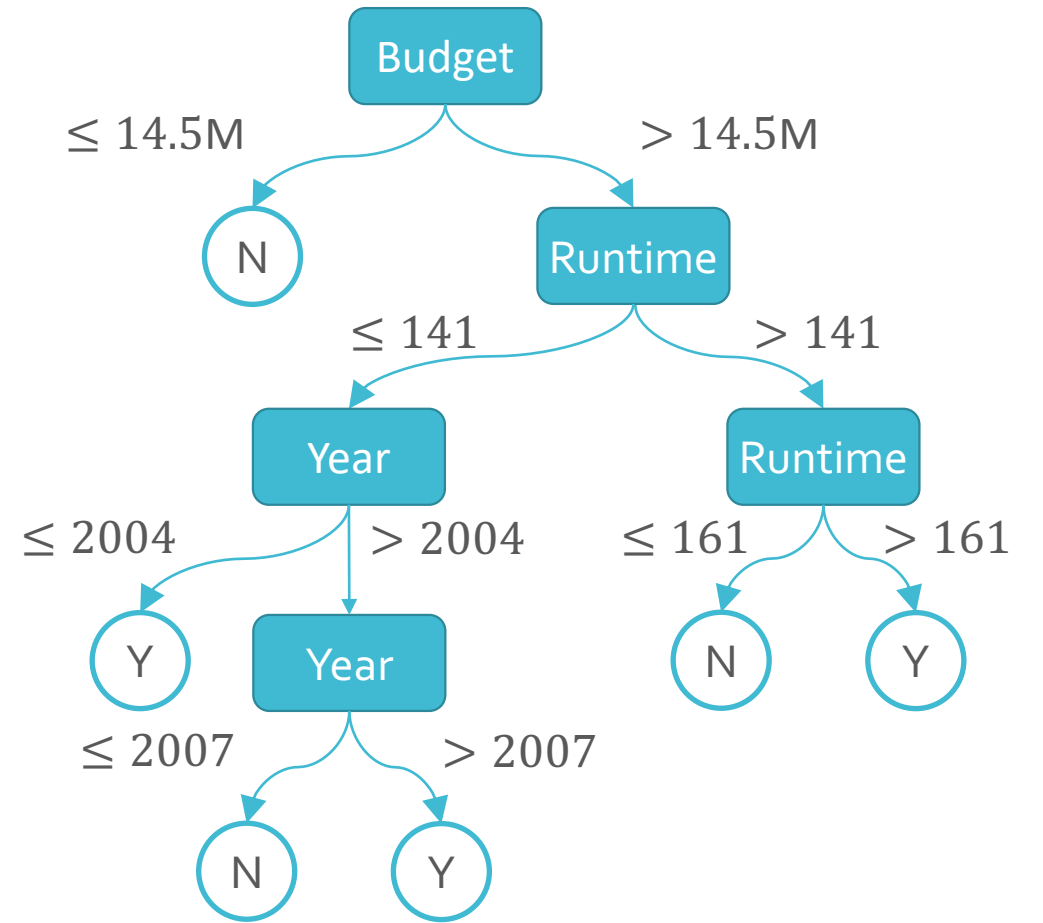
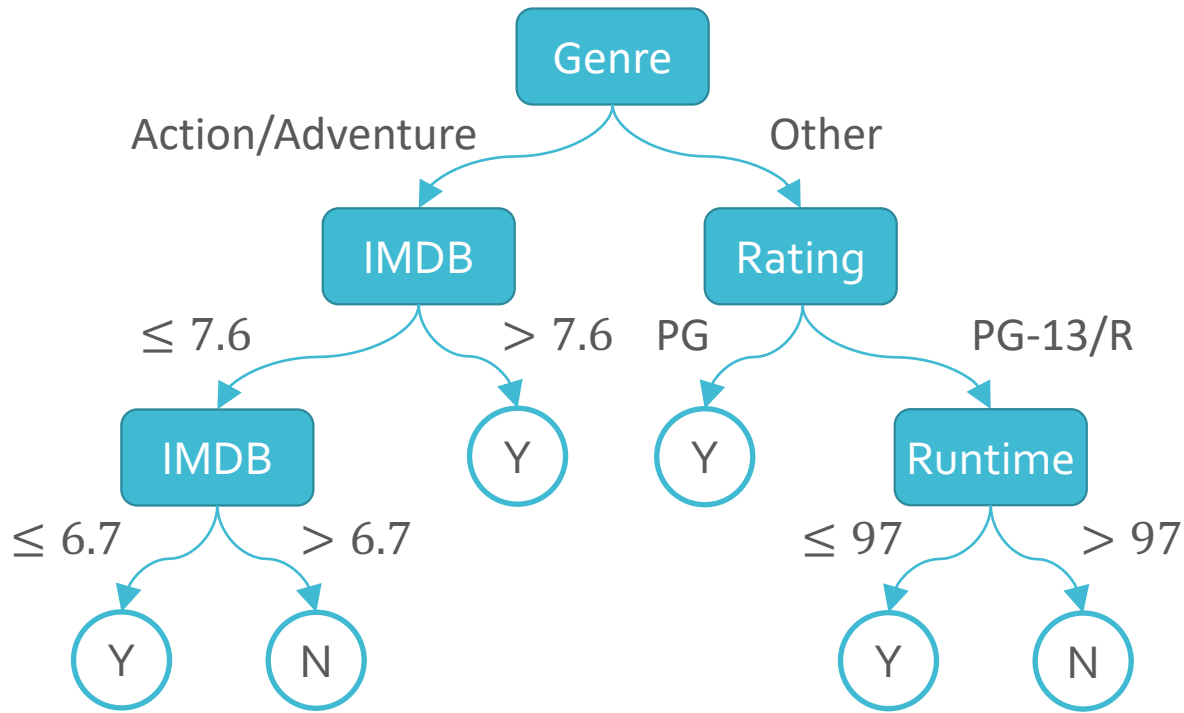


Decision Trees

MovieID	Runtime	Genre	Budget	Year	IMDB	Rating	Liked?
1	124	Action	18M	1980	8.7	PG	Y
2	105	Action	30M	1984	7.8	PG	Y
3	103	Comedy	6M	1986	7.8	PG-13	N
4	98	Adventure	16M	1987	8.1	PG	Y
5	128	Comedy	16.4M	1989	8.1	PG	Y
6	120	Comedy	11M	1992	7.6	R	N
7	120	Drama	14.5M	1996	6.7	PG-13	N
8	136	Action	115M	1999	6.5	PG	Y
9	90	Action	90M	2001	6.6	PG-13	Y
10	161	Adventure	100M	2002	7.4	PG	N
11	201	Action	94M	2003	8.9	PG-13	Y
12	94	Comedy	26M	2004	7.2	PG-13	Y
13	157	Biography	100M	2007	7.8	R	N
14	128	Action	110M	2007	7.1	PG-13	N
15	107	Drama	39M	2009	7.1	PG-13	N
16	158	Drama	61M	2012	7.6	PG-13	Y
17	169	Adventure	165M	2014	8.6	PG-13	Y
18	100	Biography	9M	2016	6.7	R	N
19	130	Action	180M	2017	7.9	PG-13	Y
20	141	Action	275M	2019	6.5	PG-13	Y



Decision Trees



Decision Trees

Random Forests

- Combines the prediction of many diverse decision trees to reduce their variability
- If B independent random variables $x^{(1)}, x^{(2)}, \dots, x^{(B)}$ all have variance σ^2 , then the variance of $\frac{1}{B} \sum_{b=1}^B x^{(b)}$ is $\frac{\sigma^2}{B}$
- Random forests = bagging + split-feature randomization
= bootstrap aggregating + split-feature randomization

Random Forests

- Combines the prediction of many diverse decision trees to reduce their variability
- If B independent random variables $x^{(1)}, x^{(2)}, \dots, x^{(B)}$ all have variance σ^2 , then the variance of $\frac{1}{B} \sum_{b=1}^B x^{(b)}$ is $\frac{\sigma^2}{B}$
- Random forests = bagging + split-feature randomization
= bootstrap aggregating + split-feature randomization

Aggregating

- How can we combine multiple decision trees, $\{t_1, t_2, \dots, t_B\}$, to arrive at a single prediction?
- Regression - average the predictions:

$$\bar{t}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B t_b(\mathbf{x})$$

- Classification - plurality (or majority) vote; for binary labels encoded as $\{-1, +1\}$:

$$\bar{t}(\mathbf{x}) = \text{sign} \left(\frac{1}{B} \sum_{b=1}^B t_b(\mathbf{x}) \right)$$

Random Forests

- Combines the prediction of many **diverse** decision trees to reduce their variability
- If B independent random variables $x^{(1)}, x^{(2)}, \dots, x^{(B)}$ all have variance σ^2 , then the variance of $\frac{1}{B} \sum_{b=1}^B x^{(b)}$ is $\frac{\sigma^2}{B}$
- Random forests = bagging + split-feature randomization
= bootstrap aggregating + split-feature randomization

Bootstrapping

- Insight: one way of generating different decision trees is by changing the training data set
- Issue: often, we only have one fixed set of training data
- Idea: resample the data multiple times *with replacement*

MovieID	...
1	...
2	...
3	...
⋮	⋮
19	...
20	...

Training data

MovieID	...
1	...
1	...
1	...
⋮	⋮
14	...
19	...

Bootstrapped
Sample 1

MovieID	...
4	...
4	...
5	...
⋮	⋮
16	...
16	...

Bootstrapped
Sample 2

...

...

Bootstrapping

- Idea: resample the data multiple times *with replacement*
 - Each bootstrapped sample has the same number of data points as the original data set
 - Duplicated points cause different decision trees to focus on different parts of the input space

MovieID	...
1	...
2	...
3	...
⋮	⋮
19	...
20	...

Training data

MovieID	...
1	...
1	...
1	...
⋮	⋮
14	...
19	...

Bootstrapped
Sample 1

MovieID	...
4	...
4	...
5	...
⋮	⋮
16	...
16	...

Bootstrapped
Sample 2

...

...

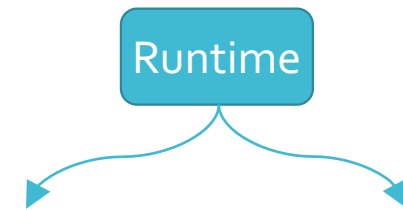
Split-feature Randomization

- Issue: decision trees trained on bootstrapped samples still behave similarly
- Idea: in addition to sampling the data points (i.e., the rows), also sample the features (i.e., the columns)
- Each time a split is being considered, limit the possible features to a randomly sampled subset

Runtime	Genre	Budget	Year	IMDB	Rating
---------	-------	--------	------	------	--------

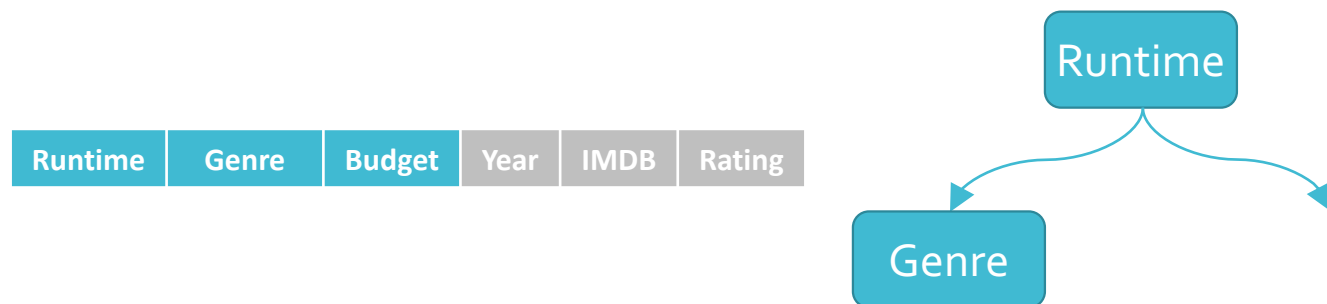
Split-feature Randomization

- Issue: decision trees trained on bootstrapped samples still behave similarly
- Idea: in addition to sampling the data points (i.e., the rows), also sample the features (i.e., the columns)
- Each time a split is being considered, limit the possible features to a randomly sampled subset



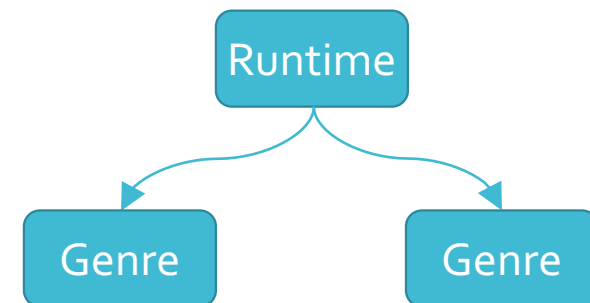
Split-feature Randomization

- Issue: decision trees trained on bootstrapped samples still behave similarly
- Idea: in addition to sampling the data points (i.e., the rows), also sample the features (i.e., the columns)
- Each time a split is being considered, limit the possible features to a randomly sampled subset



Split-feature Randomization

- Issue: decision trees trained on bootstrapped samples still behave similarly
- Idea: in addition to sampling the data points (i.e., the rows), also sample the features (i.e., the columns)
- Each time a split is being considered, limit the possible features to a randomly sampled subset



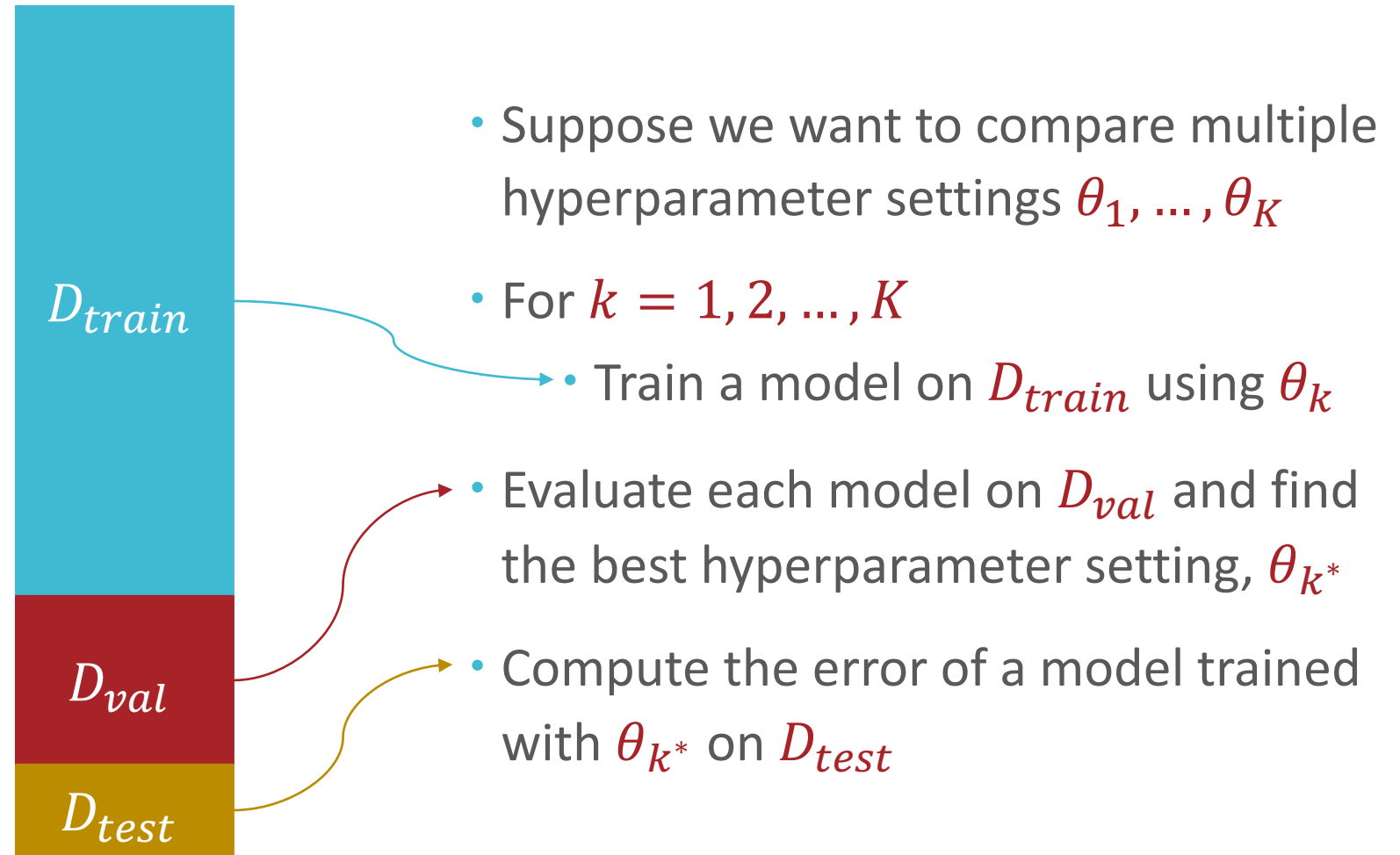
Random Forests

- Input: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N, B, \rho$
- For $b = 1, 2, \dots, B$
 - Create a dataset, \mathcal{D}_b , by sampling N points from the original training data \mathcal{D} **with replacement**
 - Learn a decision tree, t_b , using \mathcal{D}_b and the ID3 algorithm **with split-feature randomization**, sampling ρ features for each split
- Output: $\bar{t} = f(t_1, \dots, t_B)$, the aggregated hypothesis

How can we set B and ρ ?

- Input: $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N, B, \rho$
- For $b = 1, 2, \dots, B$
 - Create a dataset, \mathcal{D}_b , by sampling N points from the original training data \mathcal{D} **with replacement**
 - Learn a decision tree, t_b , using \mathcal{D}_b and the ID3 algorithm **with split-feature randomization**, sampling ρ features for each split
- Output: $\bar{t} = f(t_1, \dots, t_B)$, the aggregated hypothesis

Recall: Validation Sets



Out-of-bag Error

- For each training point, $\mathbf{x}^{(n)}$, there are some decision trees which $\mathbf{x}^{(n)}$ was not used to train (roughly B/e trees or 37%)
 - Let these be $t^{(-n)} = \{t_1^{(-n)}, t_2^{(-n)}, \dots, t_{N-n}^{(-n)}\}$
- Compute an aggregated prediction for each $\mathbf{x}^{(n)}$ using the trees in $t^{(-n)}$, $\bar{t}^{(-n)}(\mathbf{x}^{(n)})$
- Compute the out-of-bag (OOB) error, e.g., for regression

$$E_{OOB} = \frac{1}{N} \sum_{n=1}^N (\bar{t}^{(-n)}(\mathbf{x}^{(n)}) - y^{(n)})^2$$

Out-of-bag Error

- For each training point, $\mathbf{x}^{(n)}$, there are some decision trees which $\mathbf{x}^{(n)}$ was not used to train (roughly B/e trees or 37%)
 - Let these be $t^{(-n)} = \{t_1^{(-n)}, t_2^{(-n)}, \dots, t_{N-n}^{(-n)}\}$
- Compute an aggregated prediction for each $\mathbf{x}^{(n)}$ using the trees in $t^{(-n)}$, $\bar{t}^{(-n)}(\mathbf{x}^{(n)})$

- Compute the out-of-bag (OOB) error, e.g., for classification

$$E_{OOB} = \frac{1}{N} \sum_{n=1}^N \mathbb{1}(\bar{t}^{(-n)}(\mathbf{x}^{(n)}) \neq y^{(n)})$$

- E_{OOB} can be used for hyperparameter optimization!

Out-of-bag Error

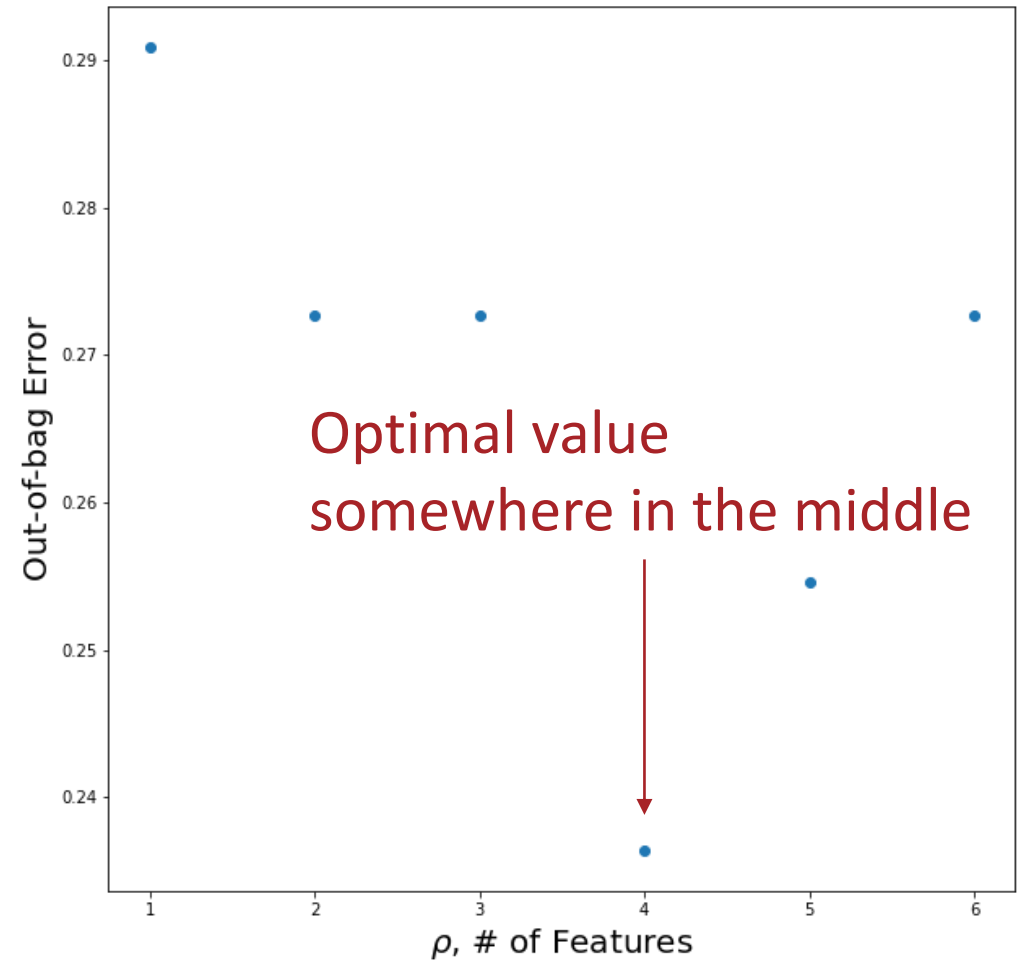
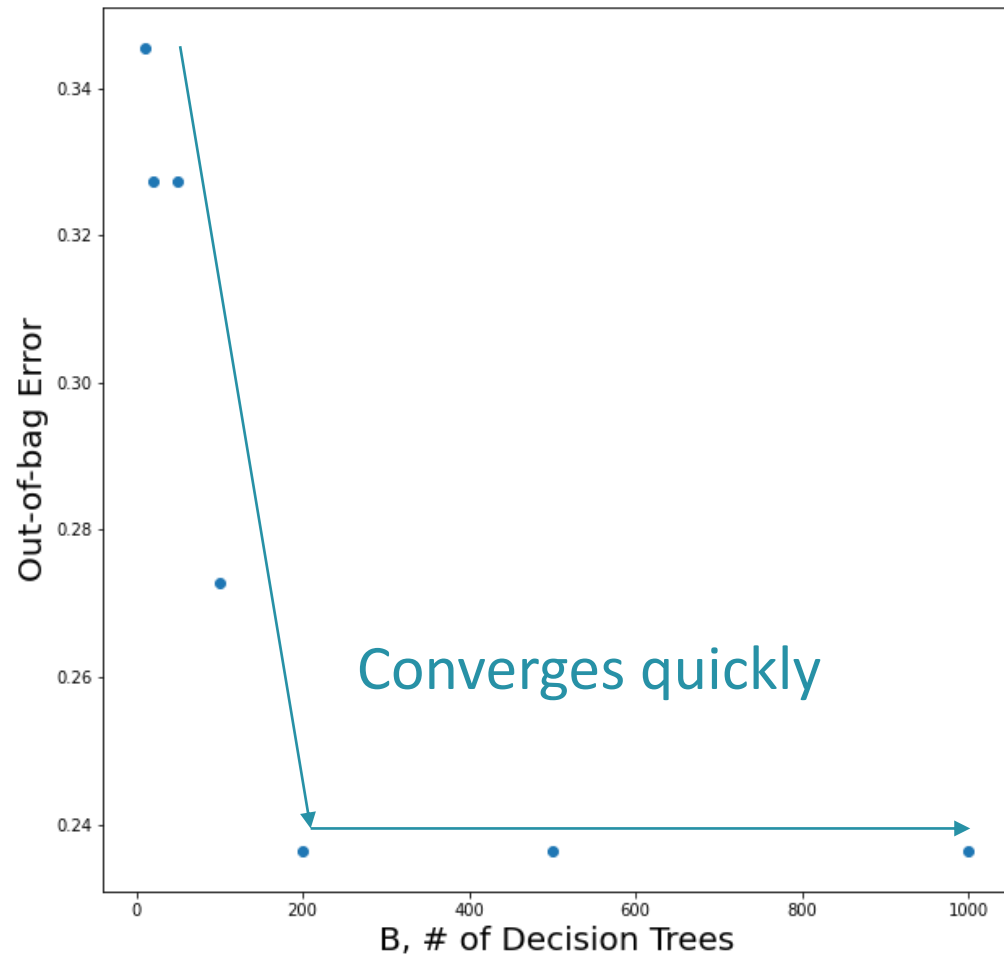


- Suppose we want to compare different numbers of trees in our random forest B_1, \dots, B_K
- For $k = 1, 2, \dots, K$
 - Train a random forest on D_{train} with B_k trees
 - Compute E_{OOB} for each random forest and find the best number of trees, B_{k^*}
- Evaluate the random forest with B_{k^*} trees on D_{test}

Out-of-bag Error



- Suppose we want to compare different numbers of trees in our random forest B_1, \dots, B_K
- For $k = 1, 2, \dots, K$
 - Train a random forest on D_{train} with B_k trees
 - Compute E_{OOB} for each random forest and find the best number of trees, B_{k^*}
- Evaluate the random forest with B_{k^*} trees on D_{test}

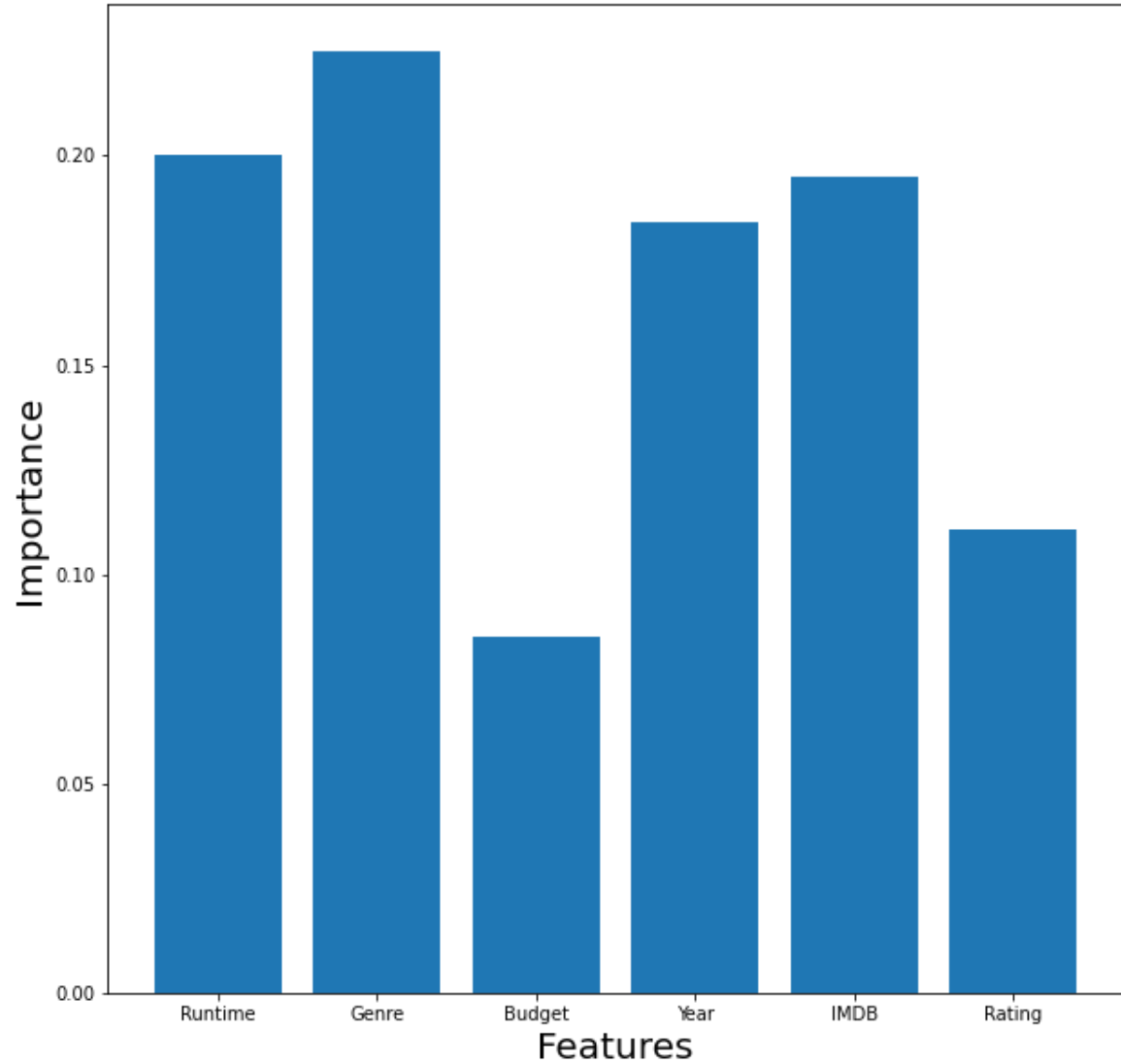


Setting Hyperparameters

Feature Importance

- Some of the interpretability of decision trees gets lost when switching to random forests
- Random forests allow for the computation of “feature importance”, a way of ranking features based on how useful they are at predicting the target
- Initialize each feature’s importance to zero
- Each time a feature is chosen to be split on, add the reduction in Gini impurity (weighted by the number of data points in the split) to its importance

Feature Importance



Key Takeaways

- Ensemble methods employ a “wisdom of crowds” philosophy
 - Can reduce the variance of high variance methods
- Random forests = bagging + split-feature randomization
 - Aggregate multiple decision trees together
 - Bootstrapping and split-feature randomization increase diversity in the decision trees
 - Use out-of-bag errors for hyperparameter optimization