CS57300
PURDUE UNIVERSITY

OCTOBER 27, 2021

DATA MINING

PREDICTIVE MODELING: EVALUATION

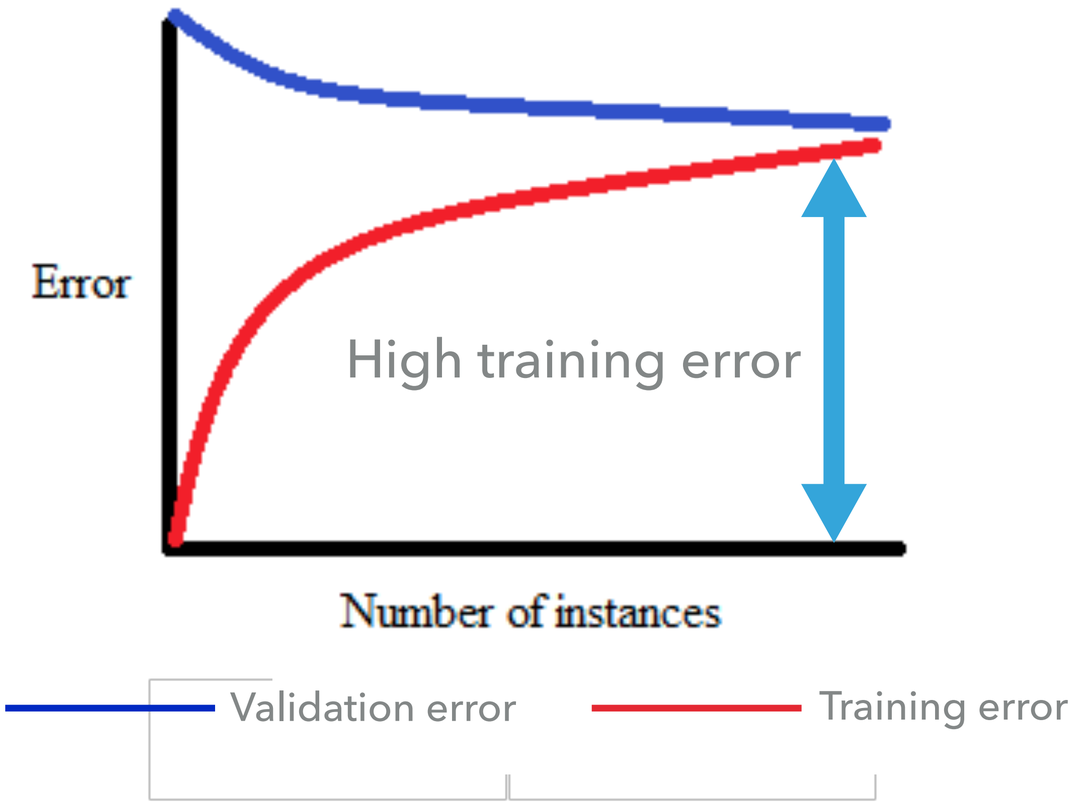
USING CROSS-VALIDATION FOR MODEL SELECTION / TUNING

- Model evaluation
  - Estimate model performance across k-fold cross validation trials
  - Use performance measurement as empirical sampling distribution for model performance
  - Evaluate difference between algorithms with statistical test
- Parameter tuning
  - Decision tree example: Choose threshold for split function with cross validation
    - Repeatedly learn model with different thresholds
    - Pick threshold that shows best cross-validation performance

PLOT LEARNING CURVE

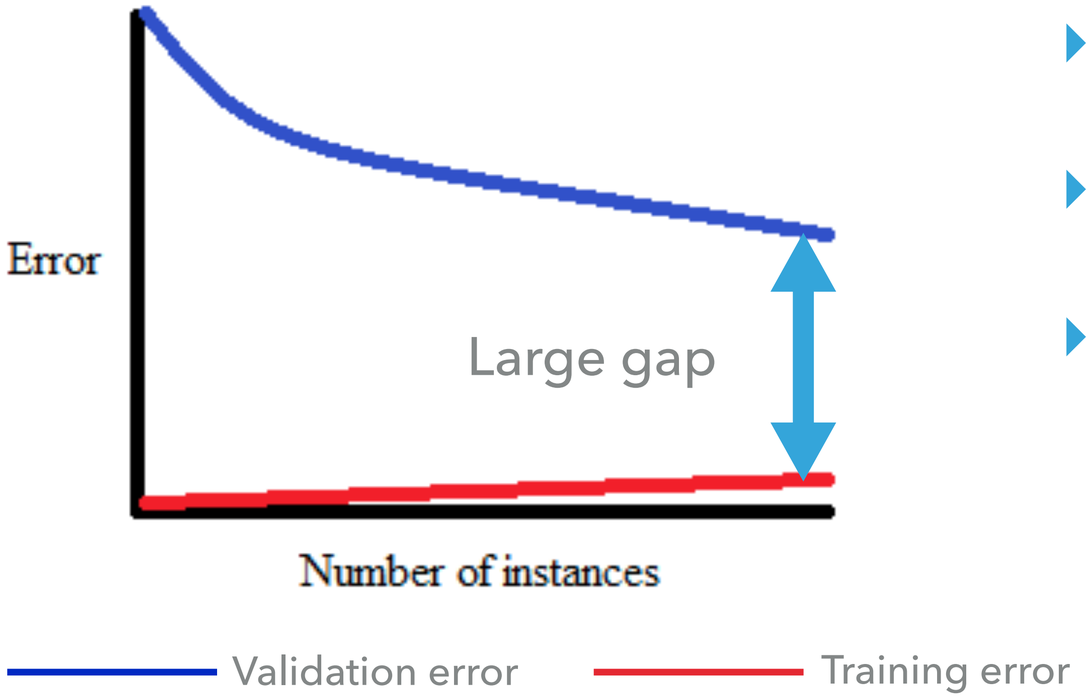
- For a given dataset S, partition it into K folds S_1, S_2, …, S_K
- For frac = [10, 20, …, 100]
  For i = 1:K
  Test set = S_i
  Randomly sample frac% of S_{-i} to construct the training set S_{train}
  Learn model on S_{train} (as a reference, you can estimate the learned model's performance on S_{train}, record it as $perf\_t_{i, frac}$)
  Evaluate model's performance on S_i, record it as $perf\_v_{i, frac}$
- Plot the training set size vs. model performance
  - Given a specific frac, model's performance is captured by the mean and standard errors of [$perf\_v_{1, frac}$, $perf\_v_{2, frac}$, …, $perf\_v_{K, frac}$]

DETECTING PROBLEMS WITH LEARNING CURVES



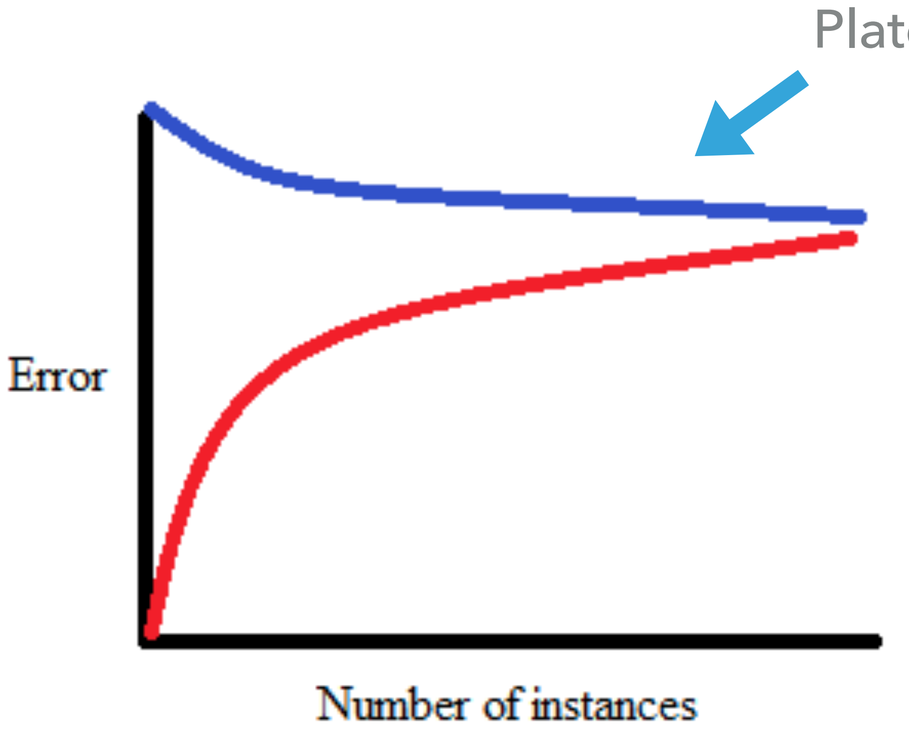
- High bias, low variance
- Underfitting: models are over-simplified

DETECTING PROBLEMS WITH LEARNING CURVES

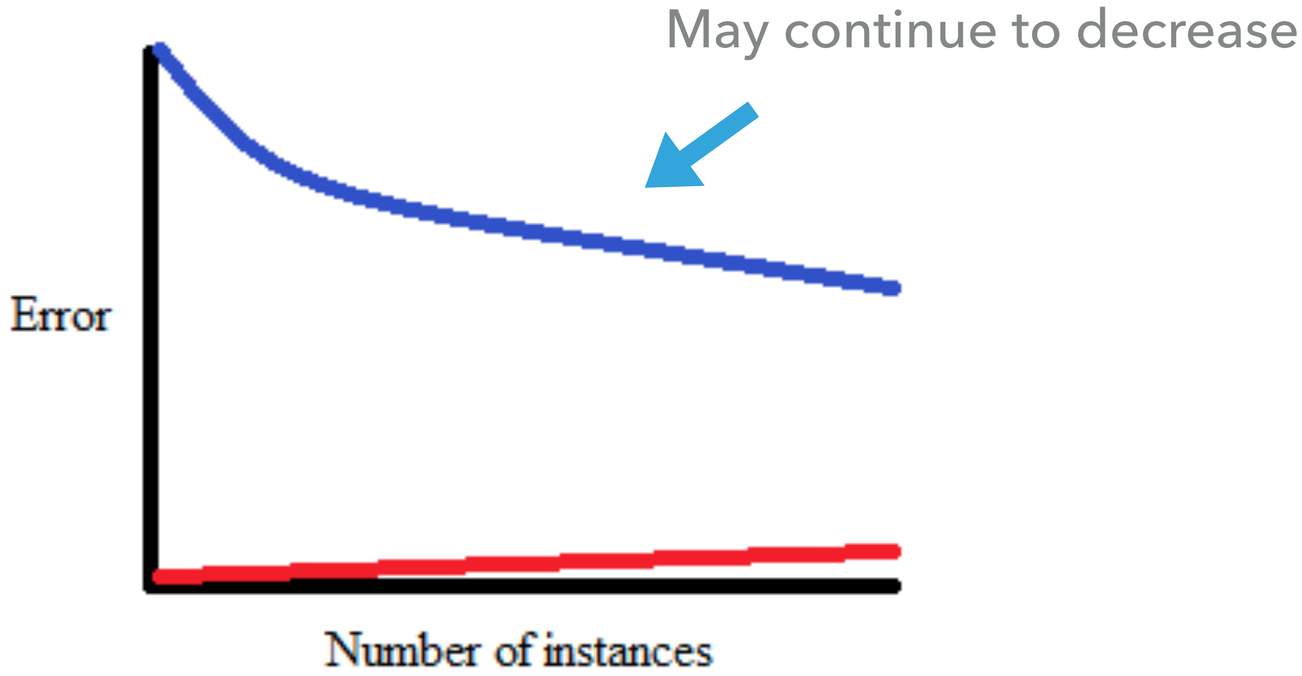


- Low bias, high variance
- Overfitting: models are over-complex
- Consider regularization, adding terms in scoring functions to penalize complexity, etc.

DETECTING PROBLEMS WITH LEARNING CURVE



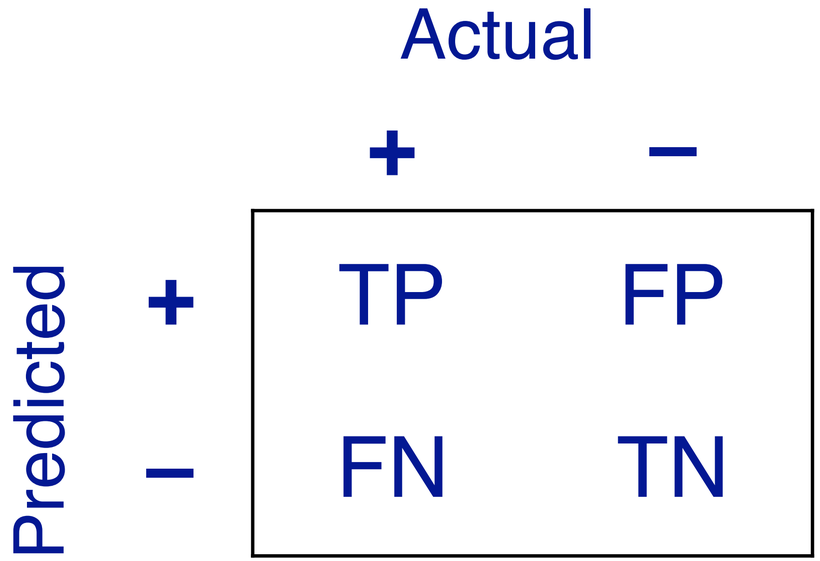
More training data won't help



More training data may help

BEYOND ACCURACY: CONTINGENCY TABLE SCORE FUNCTIONS

- True positive (**TP**):
  positive prediction that is correct
- True negative (**TN**):
  negative prediction that is correct
- False positive (**FP**):
  positive prediction that is incorrect
- False negative (**FN**):
  negative prediction that is incorrect



BEYOND ACCURACY

- Accuracy = (TP + TN) / (TP + TN + FP + FN) *% predictions that are correct*
- Misclassification = (FP+FN) / (TP+TN+FP+FN) *% predictions that are incorrect*
- Recall/Sensitivity = TP / (TP + FN) *% positive instances that are predicted positive*
- Precision = TP / (TP + FP) *% positive predictions that are correct*
- Specificity = TN / (TN + FP) *% negative instances that are predicted negative*
- F1 = 2 (P·R) / (P+R) *%harmonic mean of precision and recall*

MORE SCORING FUNCTIONS FOR PROBABILISTIC CLASSIFIERS

- Absolute loss: $\frac{1}{n}\sum_{i=1}^{n}|p(y_i=t_i)-1.0|$ where t is true label
- Squared loss: $\frac{1}{n}\sum_{i=1}^{n}[p(y_i=t_i)-1.0]^2$ where t is true label
- Likelihood/conditional likelihood: $\prod_{i=1}^{n}p(y_i=t_i)$ where t is true label

ROC CURVES

- Receiver Operating Characteristic (ROC) curve
- Plots the true positive rate (sensitivity) against the false positive rate (1-specificity) for different classification thresholds

P(Y)	True class
0.94	+
0.84	-
0.67	+
0.58	+
0.51	+
0.42	+
0.16	-
0.1	+
0.07	-

P(Y)	True class	Predict class
0.94	+	+
0.84	-	-
0.67	+	-
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

TPR = 1/4
FPR = 0/5

P(Y)	True class	Predict class
0.94	+	+
0.84	-	+
0.67	+	-
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

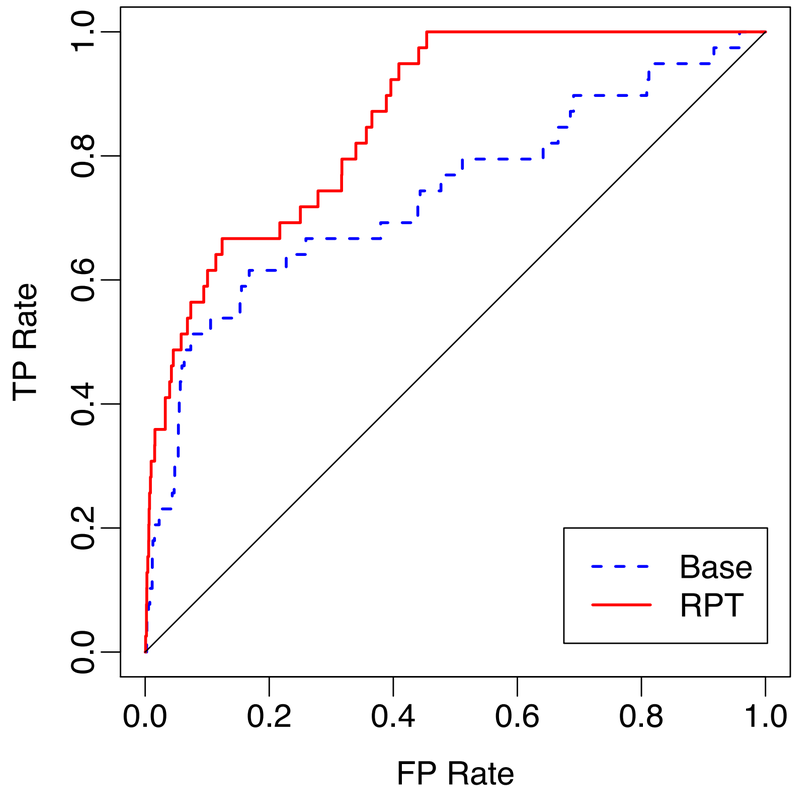
TPR = 1/4
FPR = 1/5

P(Y)	True class	Predict class
0.94	+	+
0.84	-	+
0.67	+	+
0.58	-	-
0.51	+	-
0.42	+	-
0.16	-	-
0.1	-	-
0.07	-	-

TPR = 2/4
FPR = 1/5

AUC

- Evaluates performance over varying costs and class distributions
  - Can summarize with area under the curve (AUC)
  - AUC of 0.5 is random
  - AUC of 1.0 is perfect



ENSEMBLE METHODS

THE NETFLIX PRIZE STORY

- Predictive learning tasks (i.e., supervised learning task)
  - Training data is a set of users and the ratings these users have given to movies (on a five-star scale)
  - Task: construct a predictive model that given a user and an unrated movie, predict the user's rating on that movie as 1, 2, 3, 4, or 5 stars
  - Evaluation criteria: RMSE (root mean square error) $= \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}}$
- Launched in 2006, $1 million prize for a 10% improvement of Netflix's classifier then (RMSE = 0.9514)

IN THREE WEEKS…

- More than 40 teams had outperformed the Netflix classifier…
- The best team showed about 5% improvement
- But the improvement slowed down for a while

Leaderboard

Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54
Elliptic Chaos	0.9179	3.52
datcracker	0.9183	3.48
Foreseer	0.9214	3.15
bsdfish	0.9229	3.00
Three Blind Mice	0.9234	2.94
Bocsimacko	0.9238	2.90
Remco	0.9252	2.75
karmatics	0.9301	2.24
Chapelator	0.9314	2.10
Flmod	0.9325	1.99
mthrox	0.9328	1.96

AFTER A YEAR…

- The 2007 progressive prize was awarded to the team KorBell, with a RMSE of 0.8712, representing a 8.43% improvement…

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

WHAT MODELS DID TEAMS USE?

- Rookies
  - *"Thanks to Paul Harrison's collaboration, a simple **mix** of our solutions improved our result from 6.31 to 6.75"*

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

WHAT MODELS DID TEAMS USE?

- Arek Paterek
  - *"My approach is to **combine the results of many methods** (also two-way interactions between them) using linear regression on the test set."*

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

WHAT MODELS DID TEAMS USE?

- U of Toronto
  - *"When the predictions of **multiple** RBM models and **multiple** SVD models are linearly combined, we achieve an error rate that is well over 6% better than the score of Netflix's own system."*

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

WHAT MODELS DID TEAMS USE?

- When Gravity and Dinosaurs Unite
  - *"Our common team **blends** the result of team Gravity and team Dinosaur Planet"*
  - The team's name says that already…

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

WHAT MODELS DID TEAMS USE?

- KorBell
  - *"Our final solution consists of **blending 107 individual results**"*

--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

FINAL OUTCOME OF THE NETFLIX PRIZE

- In 2009, the $1 million grand prize was finally won by the team Bellkor's Pragmatic Chaos (RMSE=0.8567, a 10.06% improvement)
- The Bellkor's Pragmatic Chaos team is expanded from KorBell by adding new members, including an entire competing team
- The second place finisher "the Emsemble" is also formed as a collection of previous independent teams…

THE LESSON

- It might be too difficult to construct a single model that optimizes performance
- So maybe the solution is to combine the results of different models

REVISITING BIAS-VARIANCE TRADEOFF

- Why is it difficult for a single model to optimize performance?
- Suppose we have a population of data {(x, y)}, our training sample D is randomly sampled from this population
- The model we construct using D is f(x; D); notice this model varies with D
- For a new data point (x*, y*), we apply the learned model on it to make prediction f(x*; D)
- The expected prediction error is $E_D[(f(x^*; D)-y^*)^2]$

REVISITING BIAS-VARIANCE TRADEOFF

- Denote $\overline{f(x^*)} = E_D[f(x^*; D)]$ as the "average prediction"

REVISITING BIAS-VARIANCE TRADEOFF

- Denote $\overline{f(x^*)} = E_D[f(x^*; D)]$ as the "average prediction"
- Then, we have:

$$E_D[(f(x^*; D) - y)^2] = E_D[(f(x^*; D) - \overline{f(x^*)} + \overline{f(x^*)} - y^*)^2]$$

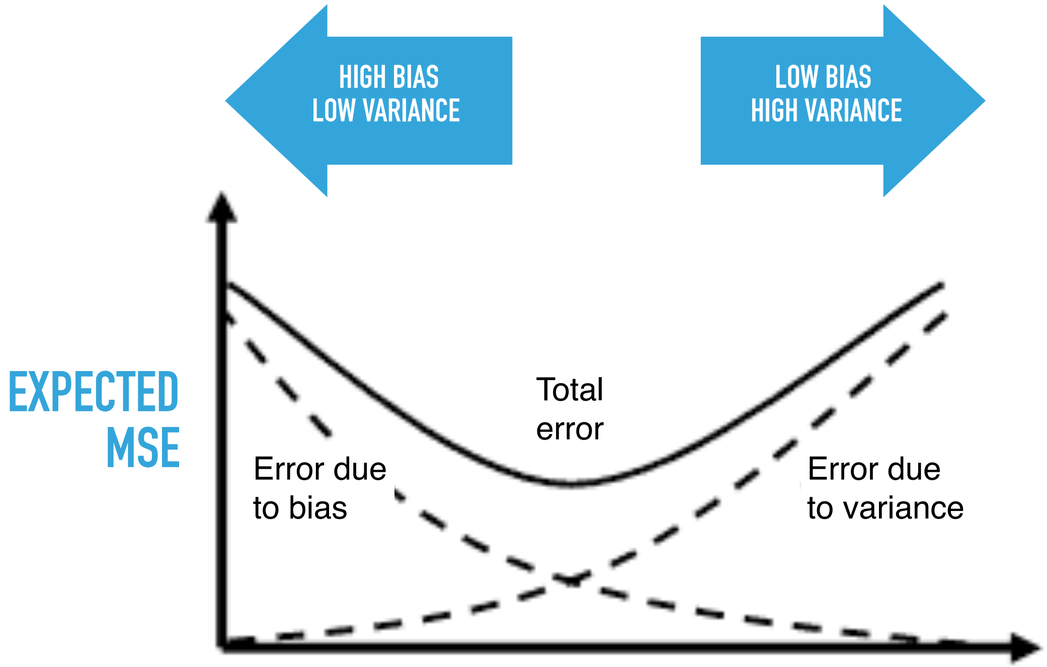
$$= E_D[(f(x^*; D) - \overline{f(x^*)})^2] + E_D[(\overline{f(x^*)} - y^*)^2] + 2E_D[(f(x^*; D) - \overline{f(x^*)})(\overline{f(x^*)} - y^*)]$$

$$= \underbrace{E_D[(f(x^*; D) - \overline{f(x^*)})^2]}_{\text{Variance}} + \underbrace{(\overline{f(x^*)} - y^*)^2}_{\text{Bias}^2}$$

FINDINGS

- **Bias**
  - Often related to size of model space
  - High bias indicates a poor match between model and concept
  - More complex models tend to have lower bias
- **Variance**
  - Often related to size of dataset (relative to the complexity of the model)
  - More complex models tend to have high variance
  - When data is large enough to estimate parameters well then models have lower variance

BIAS/VARIANCE TRADEOFF FOR LEARNING A SINGLE MODEL



Bias-variance tradeoff:
increasing the size of the model space can **reduce bias** of the learned model, but that also tends to **increase variance**...

and *decreasing* the model space tends to **reduce variance** but also **increase bias**

HOW ABOUT BLENDING MULTIPLE MODELS?

- Suppose there are N *independent* predictors $f_1(x; D), f_2(x; D), \ldots, f_N(x; D)$
- The "blended" predictor is $f(x; D) = \frac{1}{N} \sum_{i=1}^{N} f_i(x; D)$
- At data point (x*, y*), say all individual prediction has a bias of *b* and a variance of σ^2, and we have $\overline{f(x^*)} = E_D[f(x^*; D)] = \frac{1}{N} \sum_{i=1}^{N} \overline{f_i(x^*)}$

HOW ABOUT BLENDING MULTIPLE MODELS?

- Suppose there are N *independent* predictors $f_1(x; D), f_2(x; D), \ldots, f_N(x; D)$
- The "blended" predictor is $f(x; D) = \frac{1}{N} \sum_{i=1}^{N} f_i(x; D)$
- At data point (x*, y*), say all individual prediction has a bias of *b* and a variance of σ^2, and we have $\overline{f(x^*)} = E_D[f(x^*; D)] = \frac{1}{N} \sum_{i=1}^{N} \overline{f_i(x^*)}$
- Bias of $f(x^*; D)$: $\overline{f(x^*)} - y^* = \frac{1}{N} \sum_{i=1}^{N} (\overline{f_i(x^*)} - y^*) = b$
- Variance of $f(x^*; D)$: $\frac{1}{N^2} \sum_{i=1}^{N} Var(f_i(x^*; D)) = \frac{\sigma^2}{N}$

Variance decreases!