Data Science

Machine Learning Techniques

Lesson 5: Ensemble Models

Hang Zhang, Ph.D.

October 29, 2018

Lesson Outline

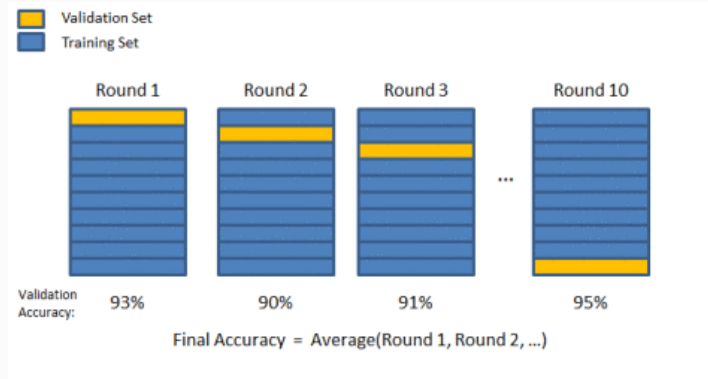
- Capstone Project Teaming Up
- Review:
  - Underfitting and overfitting
  - Decision Trees
- Ensemble Models
  - Random Forest
  - Ensemble of Other Models

Capstone Project Teaming Up

Team Name	Team Members
Datas R Us	Nithya Kannan Sailesh Kalyan **Peter Chang**
DS420_PandaPlayers	**Stacy Dean (leader)** Jim Sullivan Abhishek Gupta Gabrielle George Shri Sundararajan
DS420_astroclass_capstone	**Bhupesh Kumar – Team Lead** Deepika Gupta
DS420_TYF	**Wanqing Ma**
DS420-GRJT	**Ghodsieh Mashouf Roudsari (Team leader)** Judy Tu
DS420_Galileo's_Gala	**Vasuki Subbarao**
DS420_datarangers	Kannan SundaraRajan Rajneesh Venkat Rao Vangalapudi
DS420_Galaxy	John Lehmann **Justin Wilbourne** Sulbha Jain Drew Coogan Anjali Aggarwal

Common Pitfalls in Machine Learning

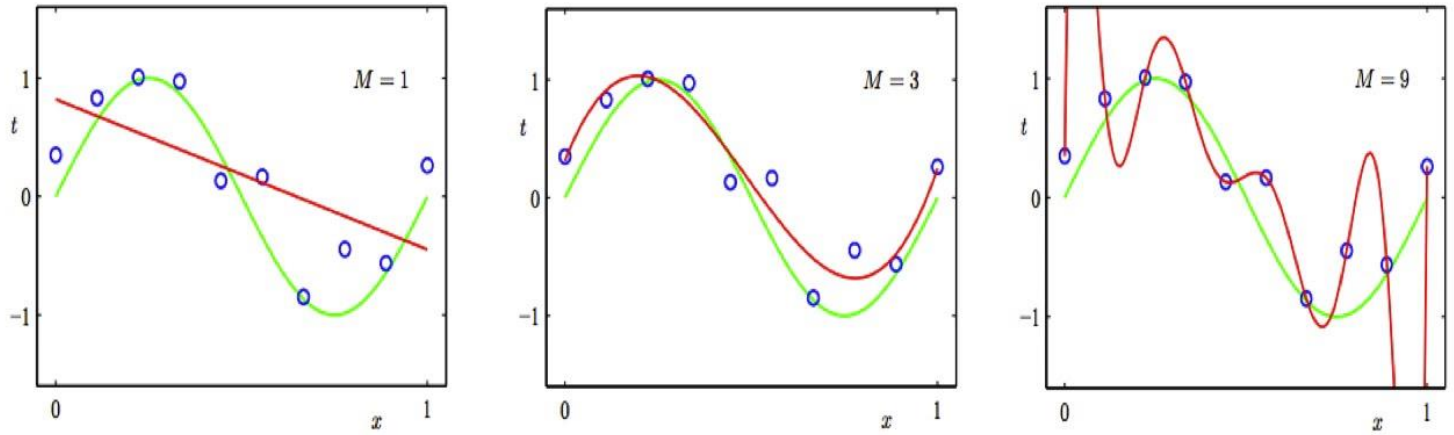
- Overfitting
  - Split the data into training and validation, and only care about the performance on validation
  - Cross validation.



- Target leaking:
  - Predicting readmission. You have one binary variable “readmission”, which is your target column. You also have columns “readmission time”, “readmission location”, “readmission reason”.
- Model has good performance on validation, but not applicable

Under- and Over-fitting examples

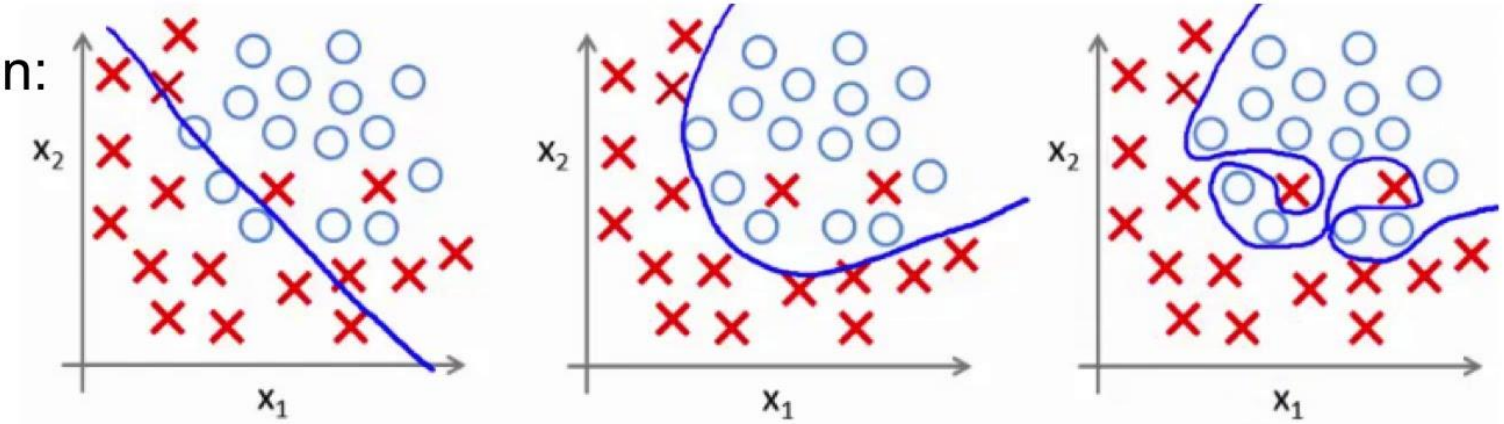
Regression:



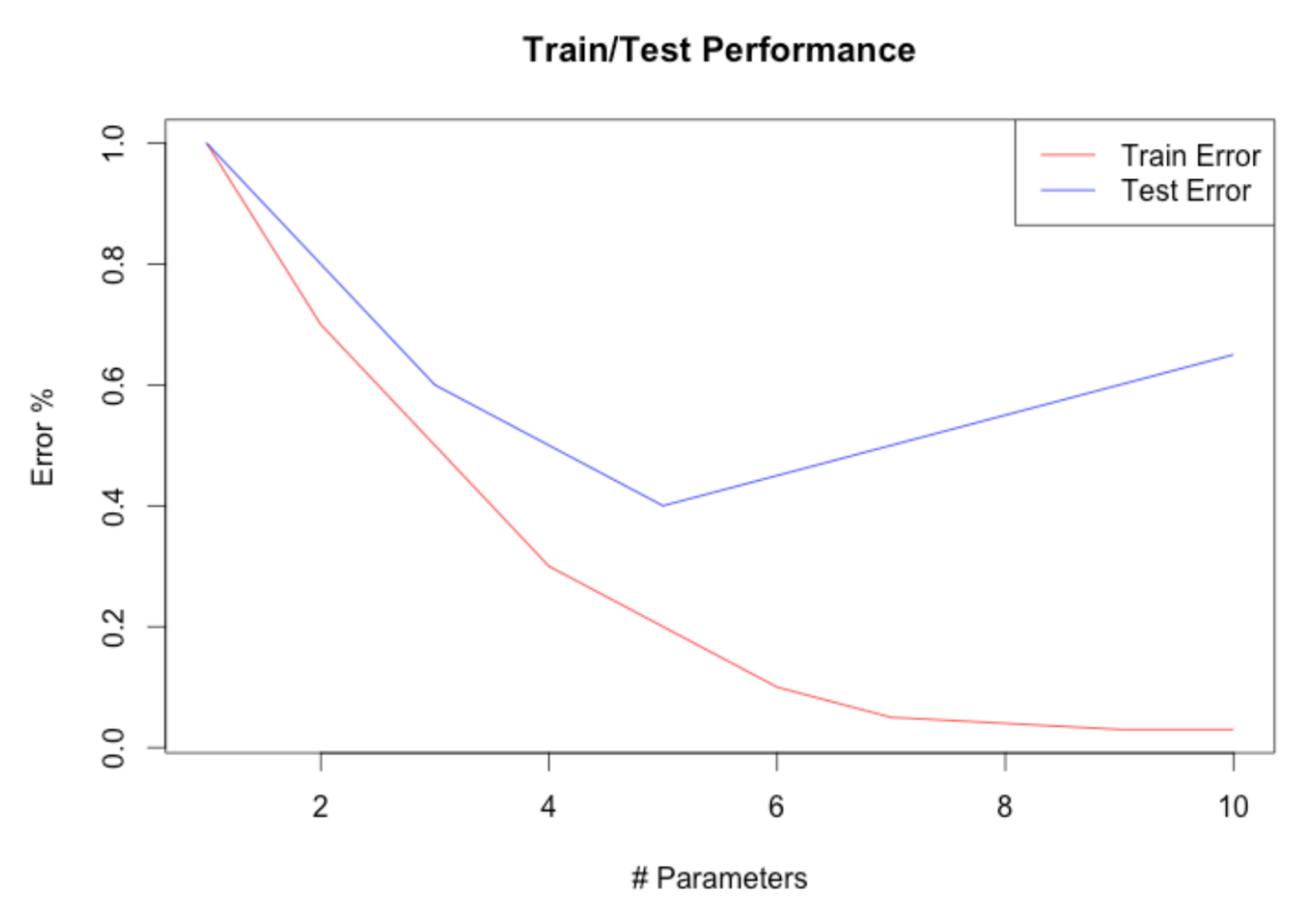
predictor too inflexible:
cannot capture pattern

predictor too flexible:
fits noise in the data

Classification:



Indicators of Underfitting and Overfitting



- Model performs poorly on both training and testing data
  - Underfitting, or
  - Not relevant data
- Model performs well on training, but poorly on testing
  - Overfitting

Reducing Underfitting

- Increase model complexity, for e.g.
  - Increase the number of levels in a decision tree
  - Increase the number of hidden layers in a neural network.
  - Decrease the number of neighbors (k) in k-NN
- Increase the number of features, or create more relevant features
- In iterative training algorithms, iterate long enough so that the objective function has converged.

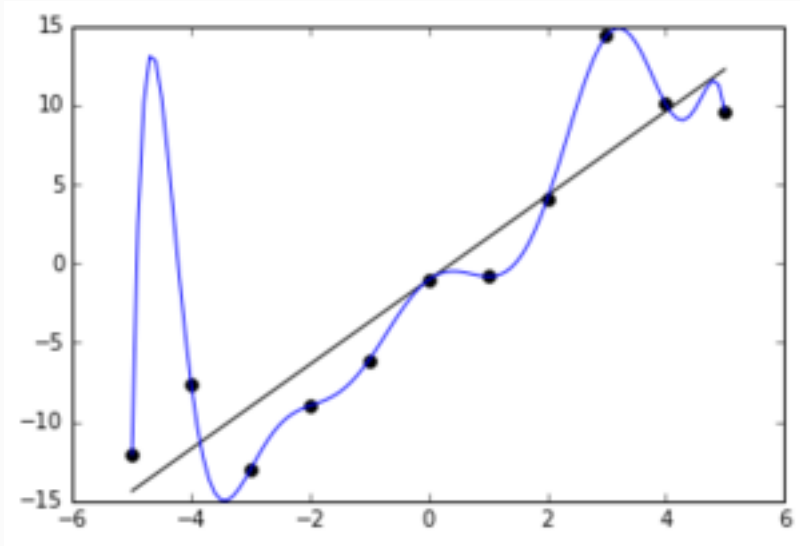
Reducing Overfitting

- Decrease model complexity, for e.g.
  - Prune a decision tree
  - Reduce the number of hidden layers in a neural network.
  - Increase the number of neighbors (k) in k-NN
- Decrease the number of features
  - More aggressive feature selection
- Regularization (control feature complexity)
  - Penalize high weights.
  - L-1 regularization (LASSO) very efficient at pushing weights of non-informative features to 0.
- Gather more training data if possible
- In iterative training algorithms, stop training earlier to prevent "memorization" of training data

Regularization: A Popular Way of Controlling Overfitting

- Loss Function of Training
  - You can almost always increase the complexity of f_θ to reduce SSE
  - Increase the risk of overfitting
- Add regularization to control overfitting
  - L1 (LASSO) or L2 (Ridge regression) regularization

$$SSE = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$$



$$LOSS = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda_1 \sum_{k=1}^{m} |\theta_k| + \lambda_2 \sum_{k=1}^{m} \theta_k^2$$

$\lambda_1, \lambda_2 \geq 0$

$\lambda_1 = 0, \lambda_2 > 0$: Ridge regression

$\lambda_2 = 0, \lambda_1 > 0$: LASSO

$\lambda_1, \lambda_2 > 0$: Elastic net

What to remember about supervised model

- Try simple models first
- Better to have smart features and simple models than simple features and smart models
- Use increasingly powerful models with more training data

Why Esemble?

- Think about a patient with some complicated disease
  - A group (panel) of doctors are involved in diagnosis
  - Each doctor may diagnose based on a specific set of data, and/or on his own specific domain expertise (model)
  - The final diagnosis is made by majority voting, weighted average (some doctors might be more experienced, their diagnosis take higher weights than others)
- Benefits of ensemble models:
  - Usually performs better than each individual model
  - Reduce the variance in the predictions, generalize better than individual models
  - Make the process of building the machine learning solutions more scalable

Different Ways of Ensembling

- Bagging:
  - Each model is trained on a subset of observations and/or features independently
- Boosting:
  - Model i+1 is trained on a sampled subset of observations, where observations that are not classified corrected by model i have higher probability of being sampled
- Different ways of making the final decision from the decisions of multiple models to be ensembled:
  - Simple average
  - Weighted average
    - Based on performance of each model (Random Forest, Boosted Decision Tree)
    - Weights are determined by another machine learning model

Random Forest (Decision Forests)

Ensemble of multiple independently trained decision trees

- Each tree is trained using a sample of observations and a sample of independent variables
  - Think about three doctors diagnosing heart disease. One doctor is trained by just looking at ECG, one doctor is a Chinese medicine doctor who is trained only by only touching the pulse, and one doctor is trained by looking at the ultrasound image
  - Each doctor is trained on data of different patients (there might be overlapping among the sets of patients)

Advantages of Random Forest:

- Significantly better performance than individual trees
- Automatic Feature Selection
- Less risk of overfitting
- Can be parallelized easily (training of multiple doctors can happen at the same time independently)

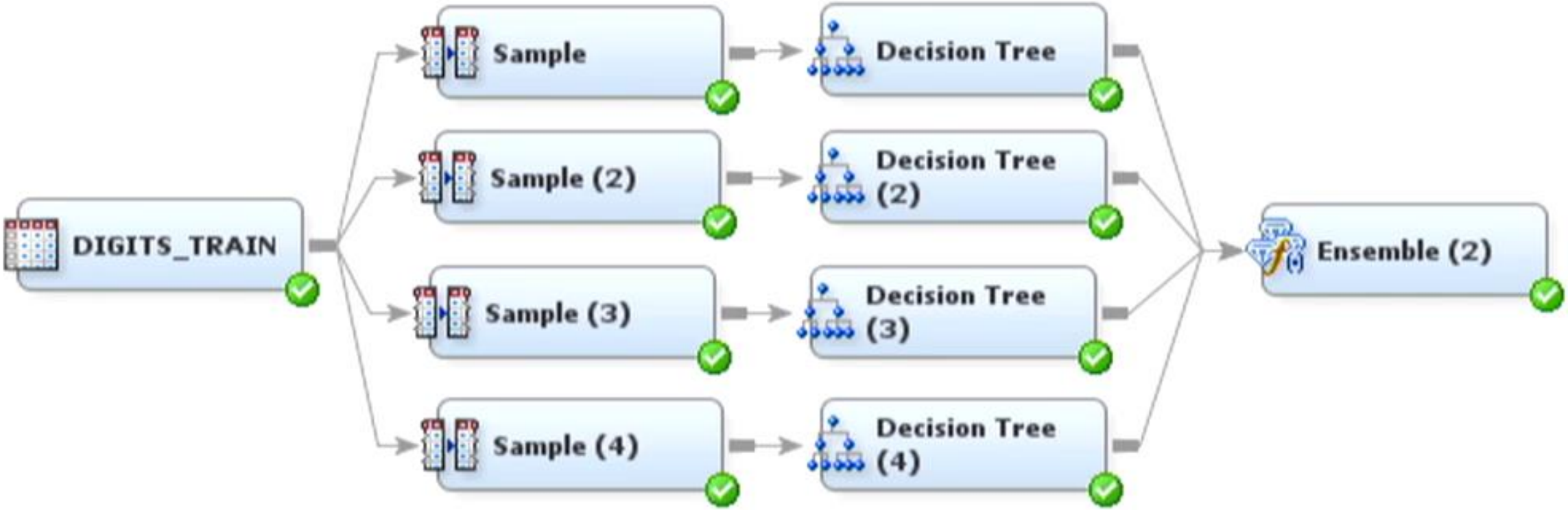
Disadvantages:

- Less interpretability than decision trees
- In some algorithms, data is copied in order to train each tree. Has higher requirement in memory space than individual trees.

Random Forests

- Combination of decision trees and bagging concepts
- A large number of decision trees is trained, each on a different bagging sample
- At each split, only a random number of the original variables is available (i.e. small selection of columns)
- Data points are classified by majority voting of the individual trees

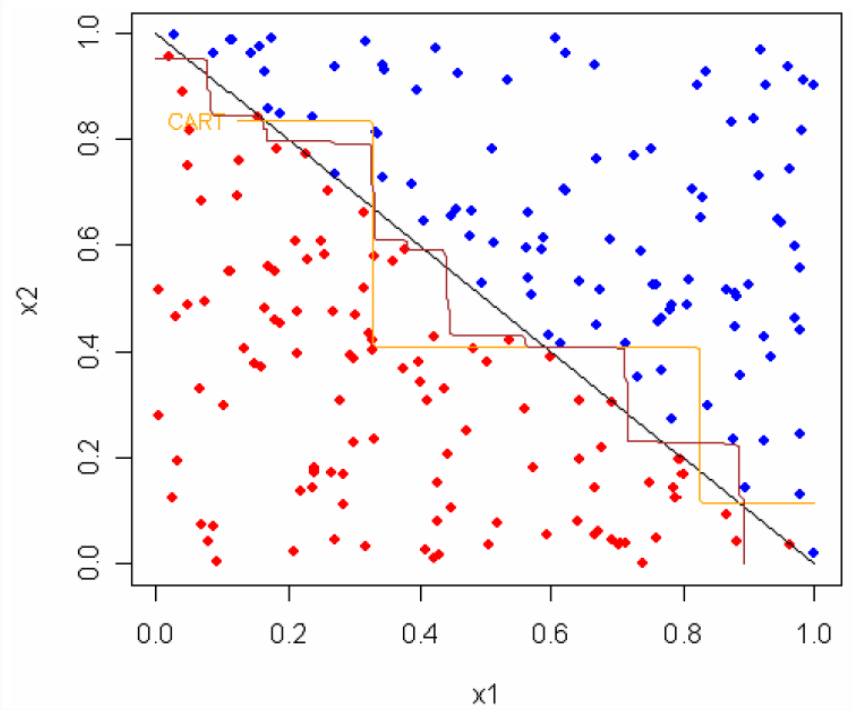
Random Forests



Bagging: reduces variance – Example 1

- Two categories of samples: blue, red
- Two predictors: x1 and x2

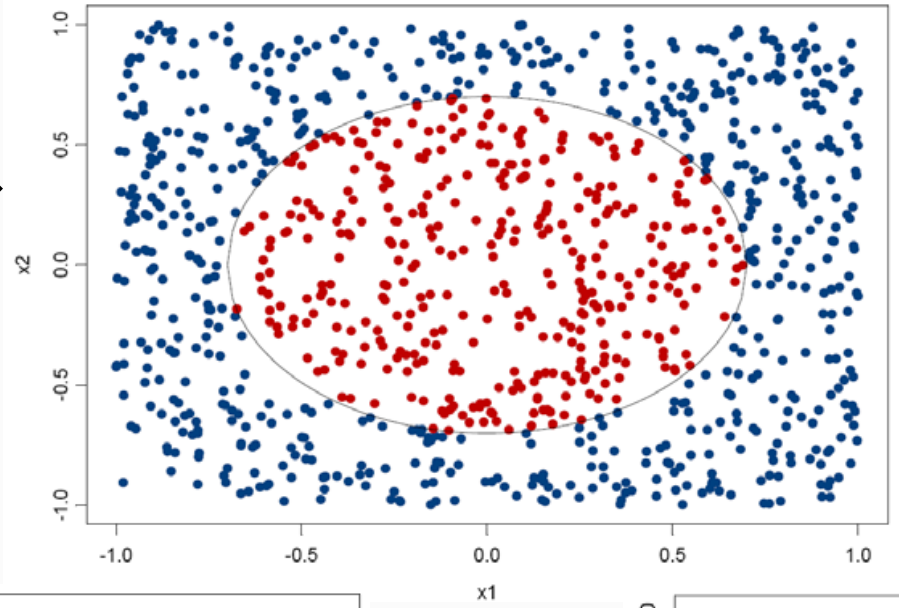
  Diagonal separation…hardest case for tree-based classifier

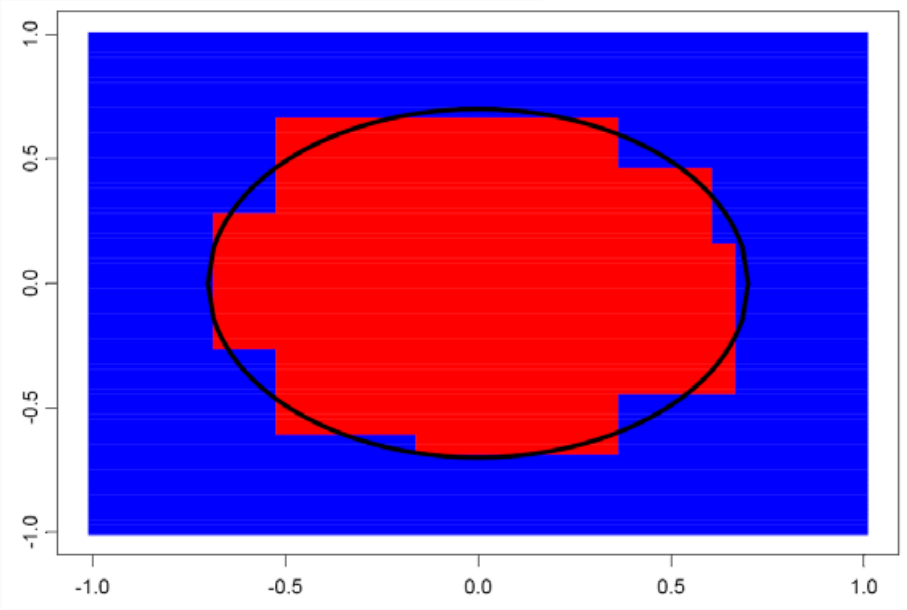


- Single tree decision boundary in orange.
- Bagged predictor decision boundary in red.

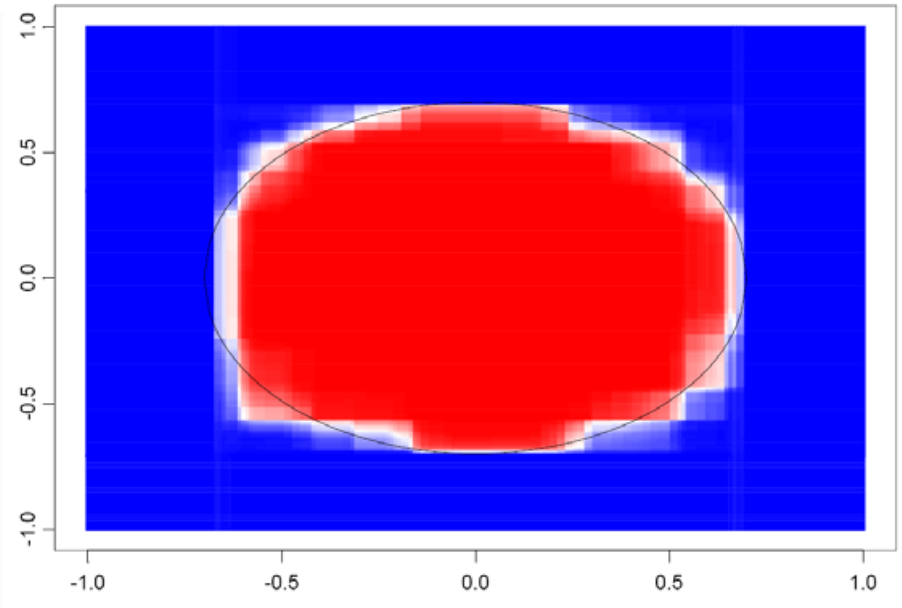
Bagging: reduces variance – Example 2

Ellipsoid separation →
Two categories,
Two predictors





Single tree decision boundary



100 bagged trees.

Random forests

```
D = training set                          F = set of tests
k = nb of trees in forest                 n = nb of tests

for i = 1 to k do:
   build data set Di by sampling with replacement from D
   learn tree Ti (Tilde) from Di:
      at each node:
         choose best split from random subset of F of size n
      allow aggregates and refinement of aggregates in tests

make predictions according to majority vote of the set of k trees.
```

Random Forest: How Many Trees to Train?

- Rule of thumb:
  - Classification problem: $\sqrt{p}$
  - Regression problem: p/3

- Optimal number is still case by case
  - Start with rule of thumb
  - Tune it to optimize performance

- In-Class Lab: Random Forest

Gradient Boosted Decision Trees

- An ensemble model
  - Ensemble of decision trees as base learners
- A powerful supervised machine learning model
- Applies to regression, classification, and ranking problems
- Won Yahoo Learning to Rank Challenge (Track 1)

AdaBoost (Adaptive Boosting)

- Boosting is a powerful technique for combining multiple "base" learners to produce a form of committee whose performance can be significantly better than that of the base learners.

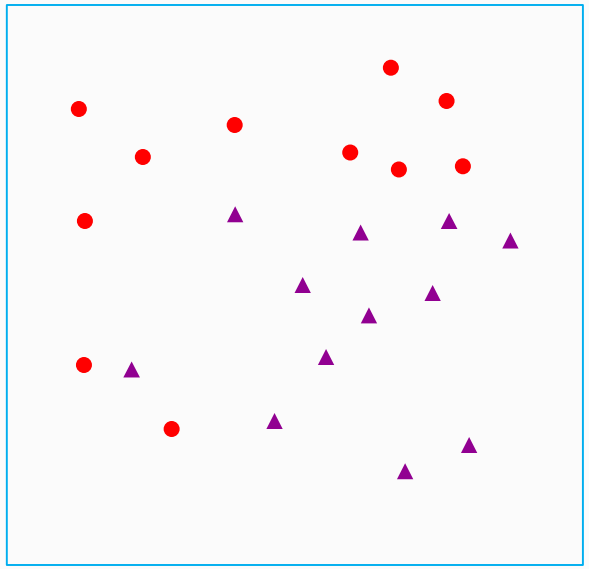
- Boosting and Bagging
  - In bagging, every base learner is trained on a random sample from the original dataset which is independent to the training set of other base learners.
  - In boosting, base learner i+1 is trained on a random sample which is dependent on the previous base learners.

- AdaBoost is a widely used form of boosting algorithm

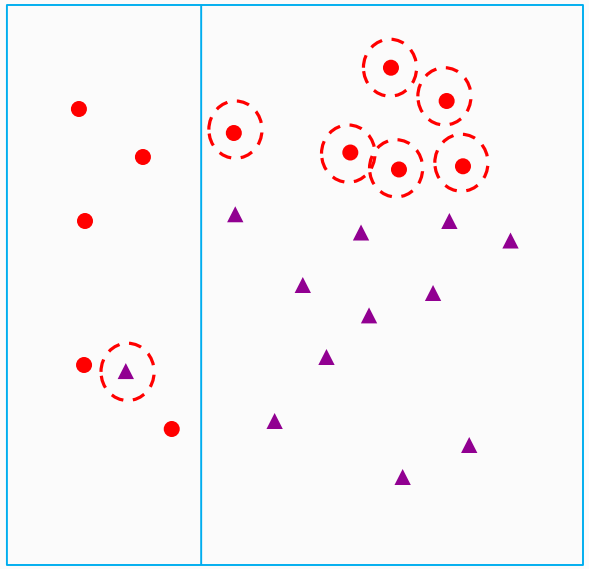
How AdaBoost Works?

- Step 0: initialize the weight for each observation to be 1/n, n=# of observations
- Step 1: for m = 1, …, M
  - Train a classifier to minimize weighted classification error. When m = 1, the weight of each observation is initialized in Step 0.
  - Increase the weights of observations that are misclassified by the current classifier
- Step 2: the final prediction is the weighted average of all M classifiers

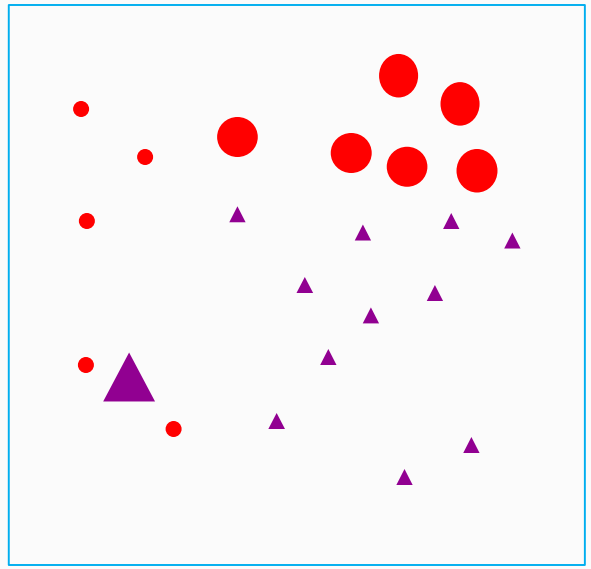
Example of AdaBoost



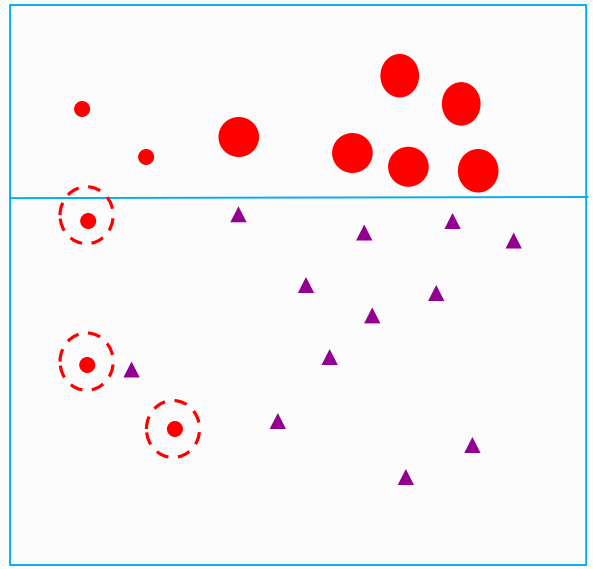
Dataset1 with equal weights



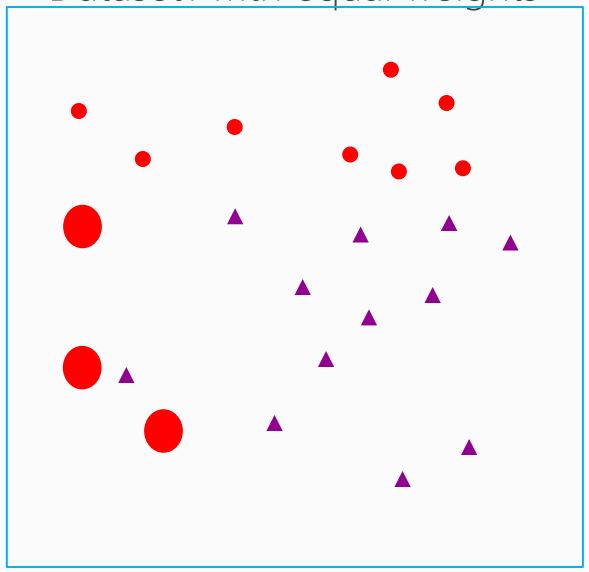
Base learner 1



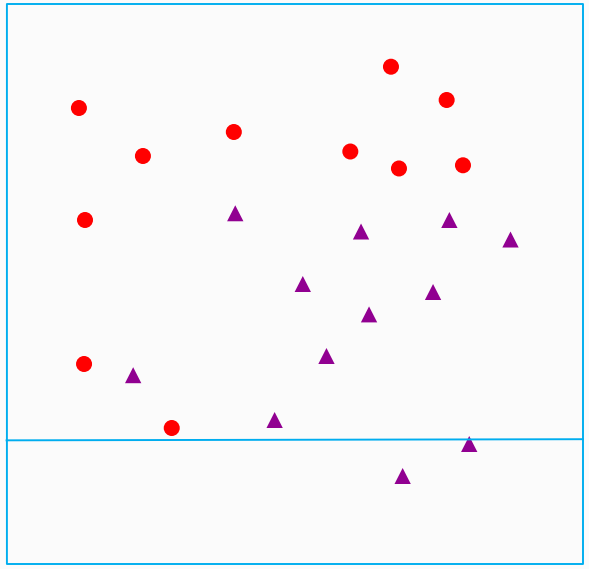
Dataset2 with adjusted weights



Base learner 2



Dataset3 with equal weights



Base learner 3

Gradient Boosted Decision Trees

- AdaBoost is using the entire training data to fit the original target variable Y for each tree, where each observation has different weight
- Gradient boosted decision trees is using the entire training data to fit the residuals of the target variable Y from previously trained models

$$F^{t-1}(x_i) + h^t(x_i) = y_i, i = 1, 2, \ldots, \mathrm{n}$$

- If we call $y_i - F^{t-1}(x_i)$ as the residual of the previous t-1 trees, $h^t(x_i)$ is a regression tree for the residuals, with the training data like
  $[(x_1, y_1 - F^{t-1}(x_1)), (x_2, y_2 - F^{t-1}(x_2)), \ldots, (x_n, y_n - F^{t-1}(x_n))]$
- Final prediction will be, where ρ is named the shrinkage rate (learning speed):

$$\boldsymbol{F^t(x_i) = \sum_{k=0}^{t} \rho h^k(x_i) = F^{t-1}(x_i) + \rho h^t(x_i)}$$

Advantages and Disadvantages of Gradient Boosted Decision Trees

- Advantages:
  - Can be more accurate than adaboost and random forest

- Disadvantages:
  - More trees can bring severe overfitting, since each additional tree is fitting on the residuals
  - Not easy to parallelize since tree t+1 is depending on the residuals from the previous trees

- In-class Lab: Gradient Boosted Decision Trees

Netflix Prize

Began October 2006

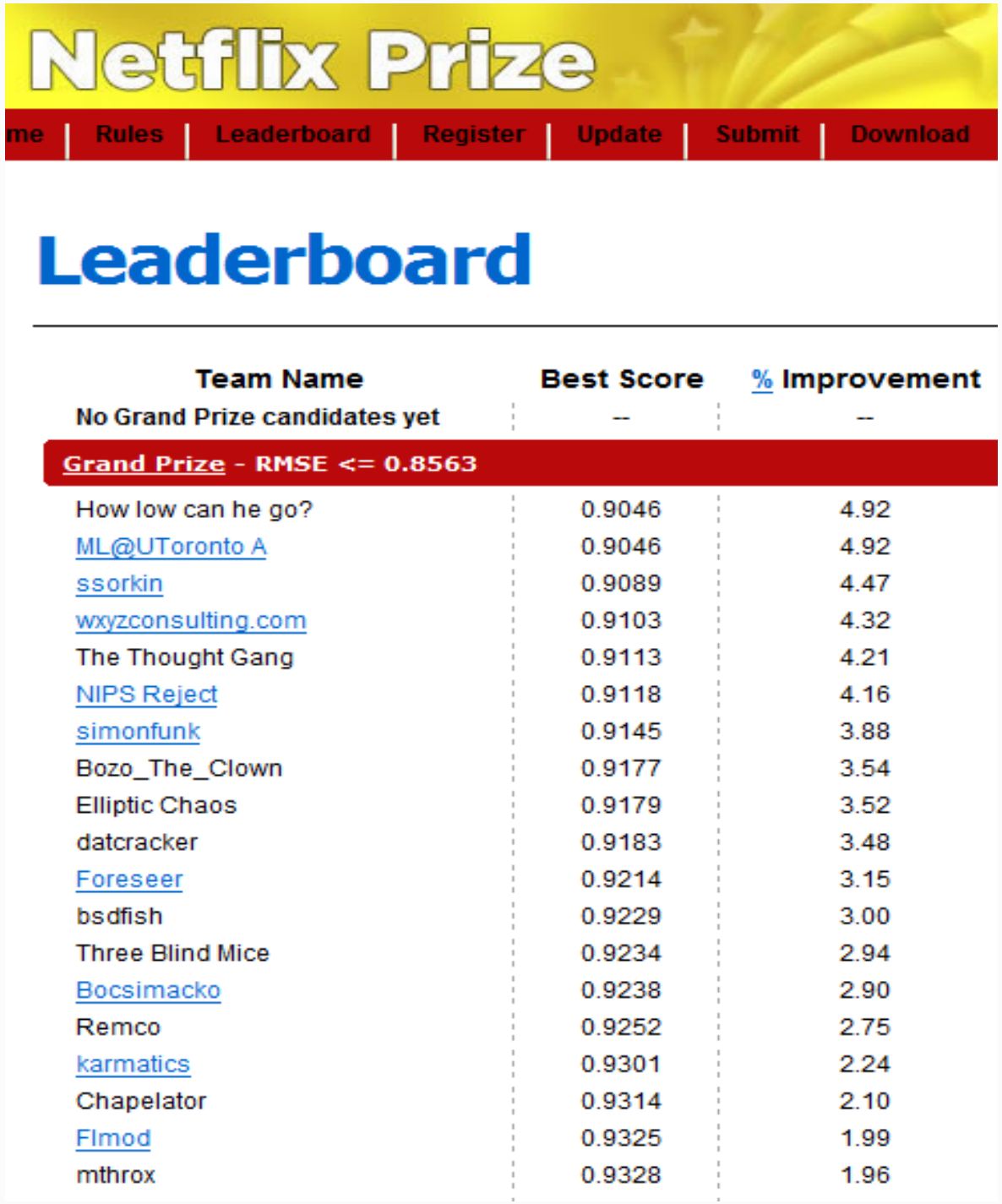
Supervised learning task

- Training data is a set of users and ratings (1,2,3,4,5 stars) those users have given to movies.
- Construct a classifier that given a user and an unrated movie, correctly classifies that movie as either 1, 2, 3, 4, or 5 stars

$1 million prize for a 10% improvement over Netflix's current movie recommender/classifier

(MSE = 0.9514)

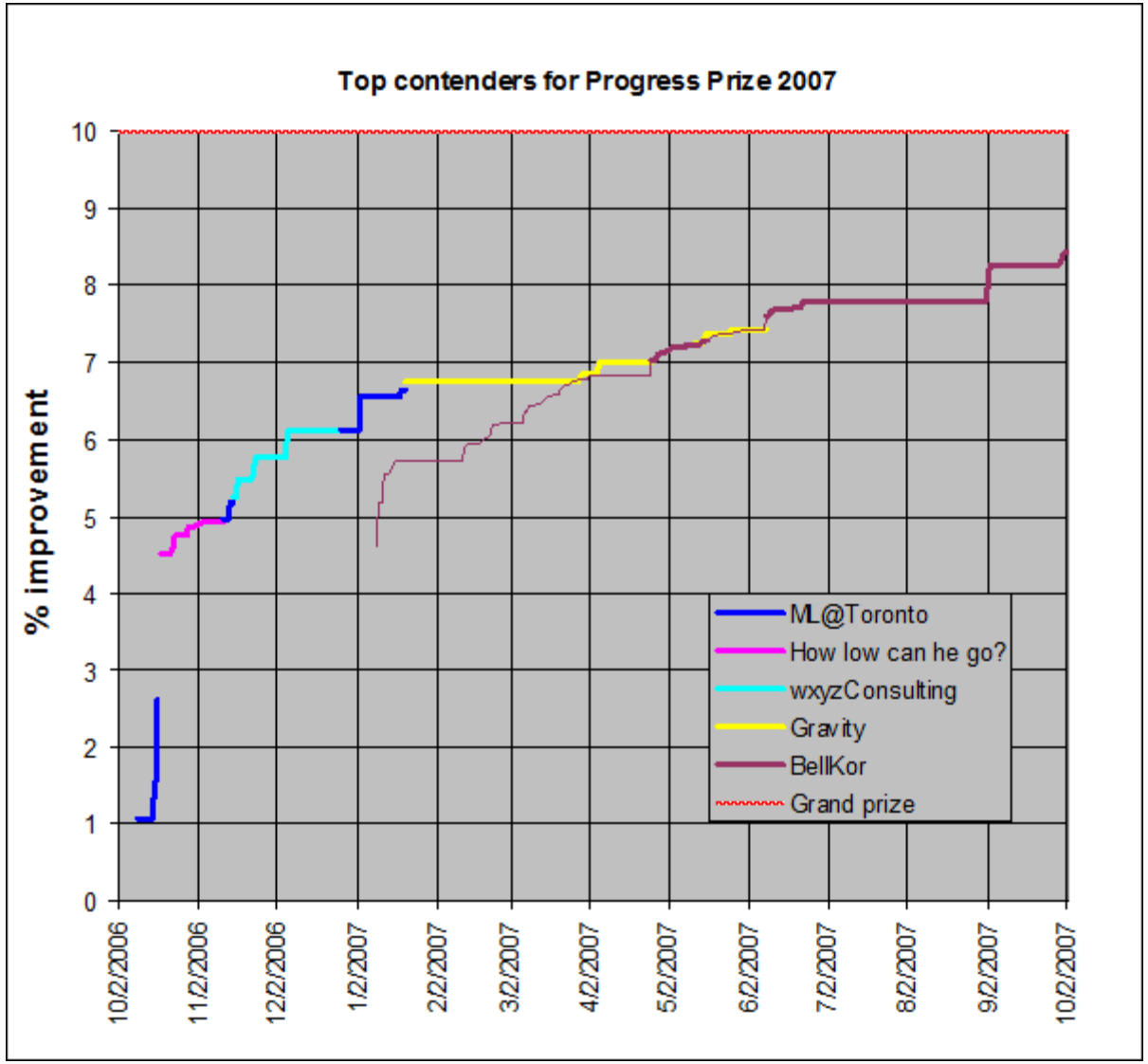
http://www.wired.com/business/2009/09/how-the-netflix-prize-was-won/, a light read (highly suggested)



Just three weeks after it began, at least 40 teams had bested the Netflix classifier.

Top teams showed about 5% improvement.

However, improvement slowed…



from http://www.research.att.com/~volinsky/netflix/

--	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

The top team posted a 8.5% improvement.

Ensemble methods are the best performers…

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

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. The best method in my ensemble is regularized SVD with biases, post processed with kernel ridge regression"

http://rainbow.mimuw.edu.pl/~ap/ap_kdd.pdf

--	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

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."

http://www.cs.toronto.edu/~rsalakhu/papers/rbmcf.pdf

--	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

Gravity

Table 5: Best results of single approaches and their combinations

Method/Combination	RMSE
MF	0.9190
NB	0.9313
CL	0.9606
NB + CL	0.9275
MF + CL	0.9137
MF + NB	0.9089
MF + NB + CL	0.9089

home.mit.bme.hu/~gtakacs/download/**gravity**.pdf

--	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

When Gravity and Dinosaurs Unite

"Our common team blends the result of team Gravity and team Dinosaur Planet."

Might have guessed from the name…

--	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

BellKor / KorBell

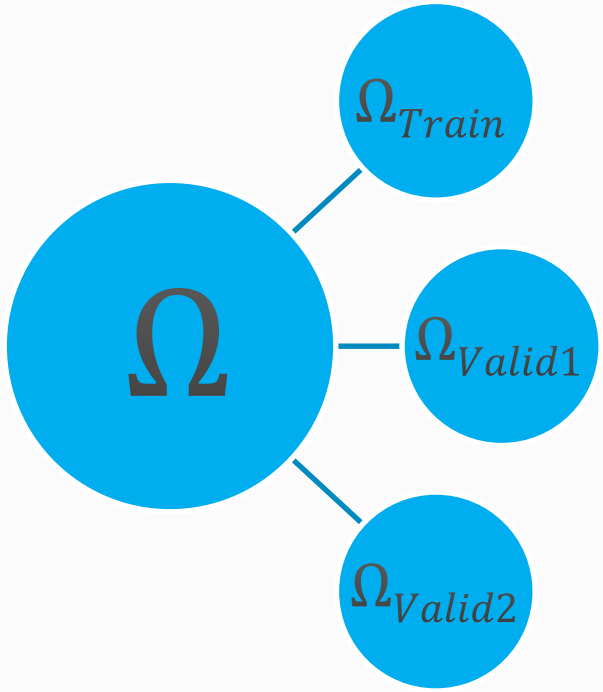
And, yes, the top team which is from AT&T…

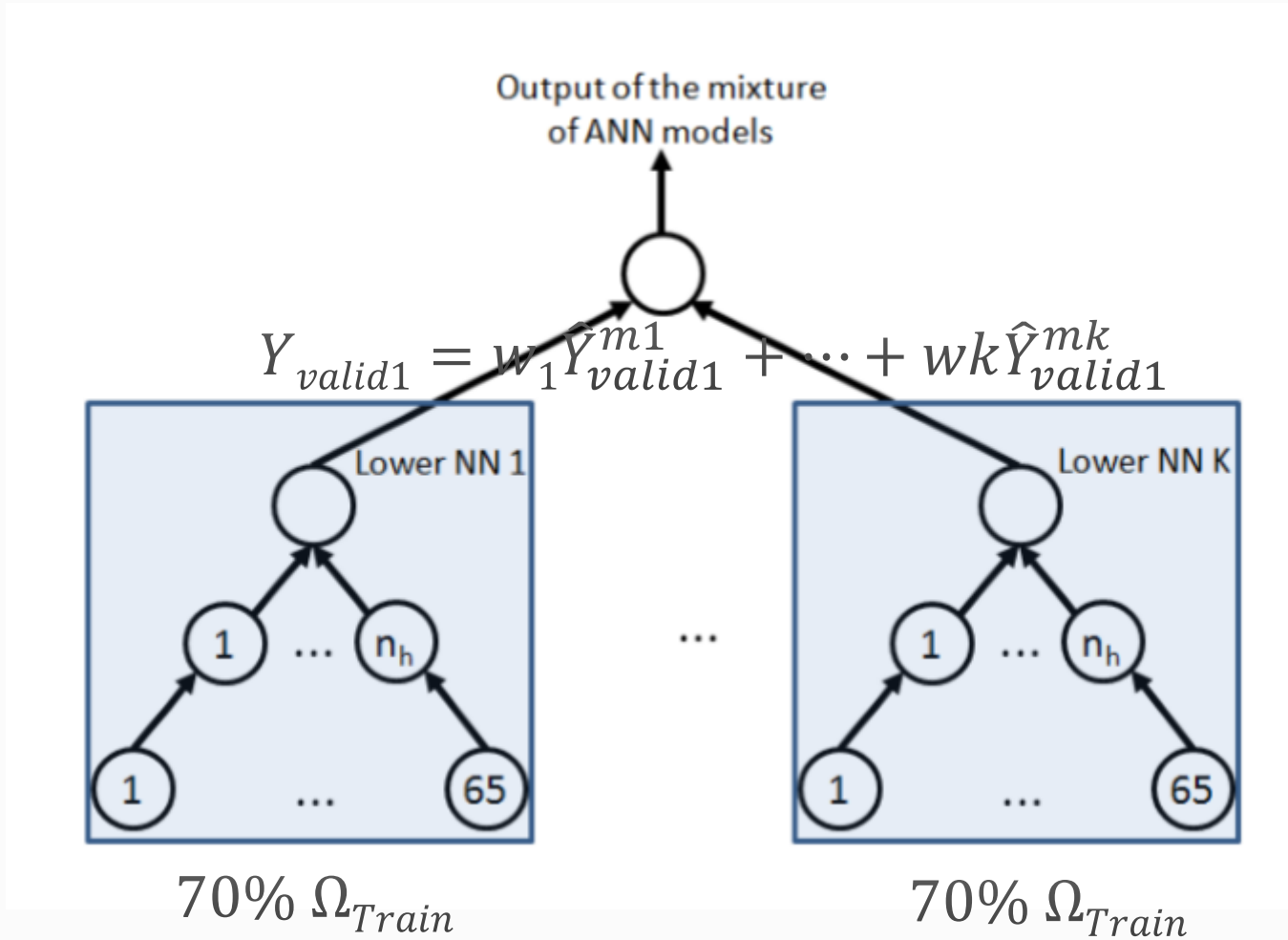
"Our final solution (RMSE=0.8712) consists of blending 107 individual results. "

--	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

KDD Cup 2011

- Predict whether a user is going to rate a music track highly or not





Performance Gain Decreases as Ensembled Models Increase

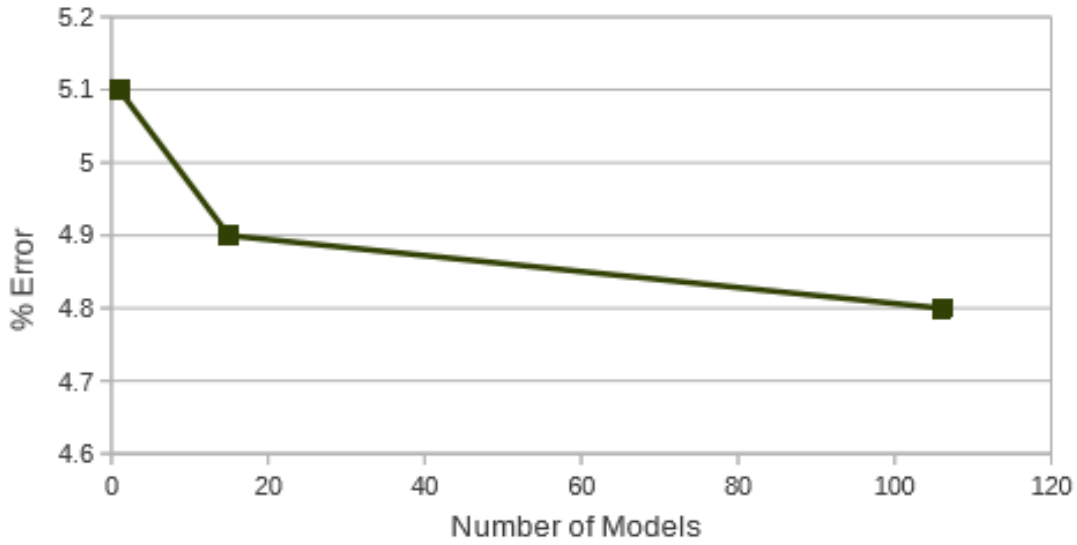


Figure 5: Error on Combinations of Neural Nets.

Cons of Ensemble of Models

- Loss of interpretability quickly
- Difficult to operationalize
- Risk of overfitting
  - Heritage Health Prize: prediction number of hospital stay days of next year
  - Ensemble of around 50 models
  - Ranked #1 on public leaderboard, but ranked #20 on private leaderboard

Risk Factors of Overfitting in Ensemble Models

- Highly correlated models:
  - Different models trained on same or similar feature set
  - Similar models trained on similar feature sets

- Weights of models are fine tuned too much in order to optimize the performance on validation data
  - Overfitting on validation data
  - Does not generalize well in production