Ensemble Machine Learning

What is Machine Learning Ensembles?

Intuition of Ensemble Learning

Three kids can defeat a master

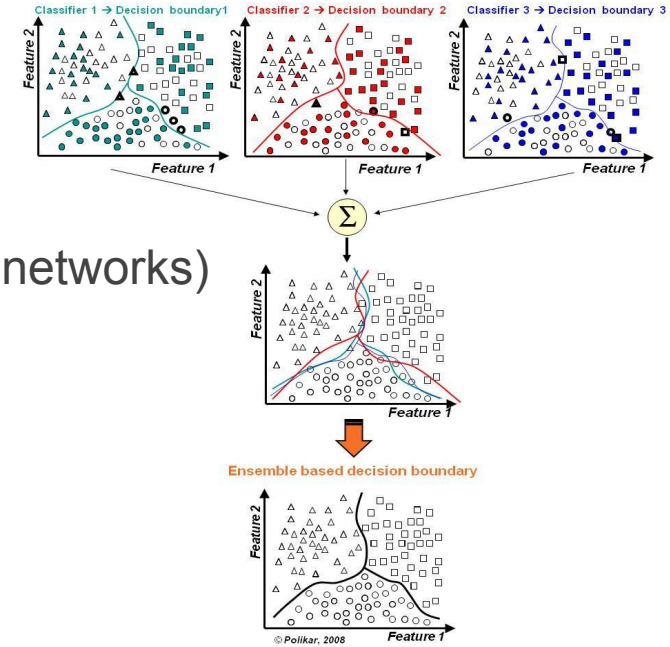
Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	**89.731**	**92.215**
2 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	88.592	90.859
2 Sep 16, 2019	ALBERT (single model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	88.107	90.902
2 Jul 26, 2019	UPM (ensemble) *Anonymous*	88.231	90.713
3 Aug 04, 2019	XLNet + SG-Net Verifier (ensemble) *Shanghai Jiao Tong University & CloudWalk* https://arxiv.org/abs/1908.05147	88.174	90.702

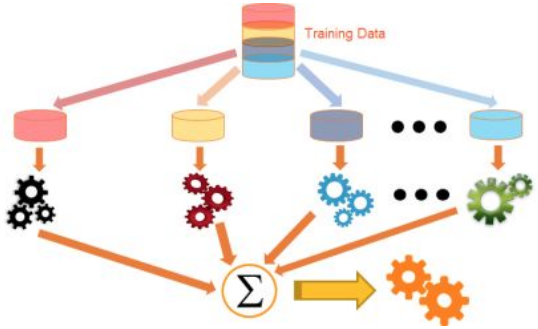
Machine Learning Ensembles

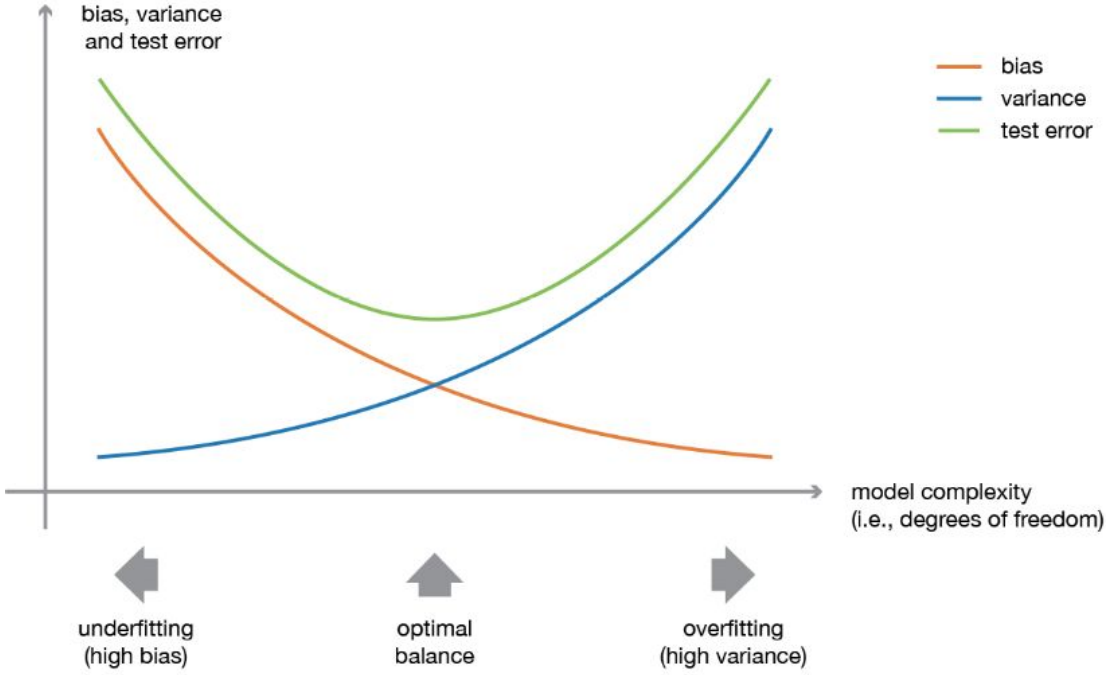
- Techniques that generate a group of base learner when combined have higher accuracy
- Strong v.s. Weak learner
- Stable (kNN) v.s. Unstable (decision trees, neural networks) machine learning algorithms



Why Ensemble?

- Reduce Bias
- Reduce Variance
- Prediction Error:
  = Bias ^2
  + Variance
  + Irreducible Error

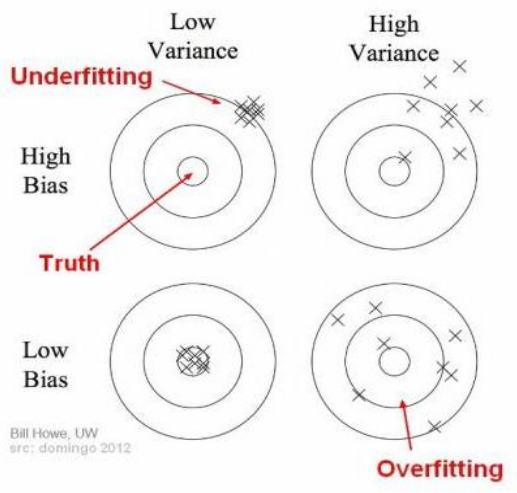




Bias-Variance

- **Bias**: the difference between the average prediction of our model and the correct value which we are trying to predict

- **Variance**: the variability of model prediction for a given data point or a value which tells us spread of our data



Reduce Bias

- Assume a test set of 10 samples and k (assume k is odd) **independent** binary classifiers, where each classifier has ***p*** accuracy.

Combining these k classifiers, using majority voting

The final Acc. will be the prob that majority of classifiers are correct.

$$\sum_{i=0}^{int(\frac{k}{2})} \binom{k}{i} p^{k-i}(1-p)^{i}$$

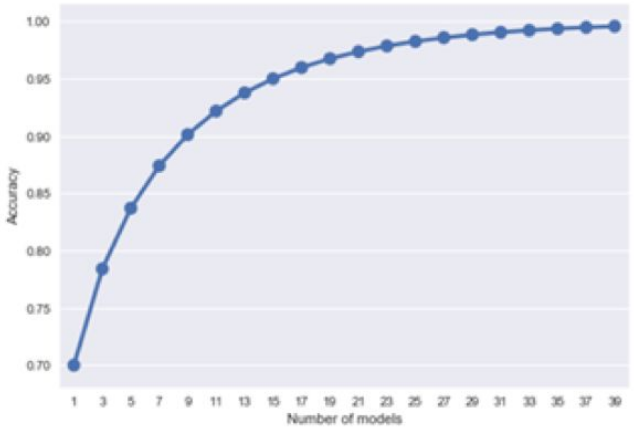
What is the probability that k choose ***i classifiers*** whose predictions are **wrong** and the rest ***k-i models***' outputs are **correct**.

Reduce Bias

$$\sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{i} p^{k-i}(1-p)^i$$

If p = 0.7, then we have

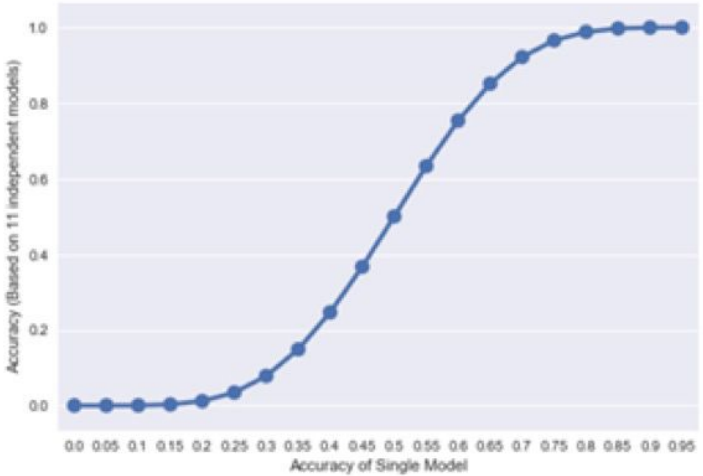
k	Ensemble Accuracy
1	0.7
3	0.784
5	0.83692
11	0.92177520904
101	0.999987057446



Reduce Bias

$$\sum_{i=0}^{\lfloor \frac{k}{2} \rfloor} \binom{k}{i} p^{k-i}(1-p)^i$$

Fix # of classifiers to be 11



Reduce Variance

- Suppose we have n **independent** models: **M1**, **M2**, …. **Mn** with the same variance σ **^2**. The ensemble **M*** constructed from these models using averaging will have the variance as follows:

$$
\begin{aligned}
Var(M^*) &= Var(\frac{1}{n}\sum_i M_i) \\
&= \frac{1}{n^2} Var(\sum_i M_i) \\
&= \frac{1}{n^2} * n * Var(M_i) \\
&= \frac{Var(M_i)}{n}
\end{aligned}
$$

$$\sigma^2 \rightarrow \frac{\sigma^2}{n}$$

Common Ensemble Techniques

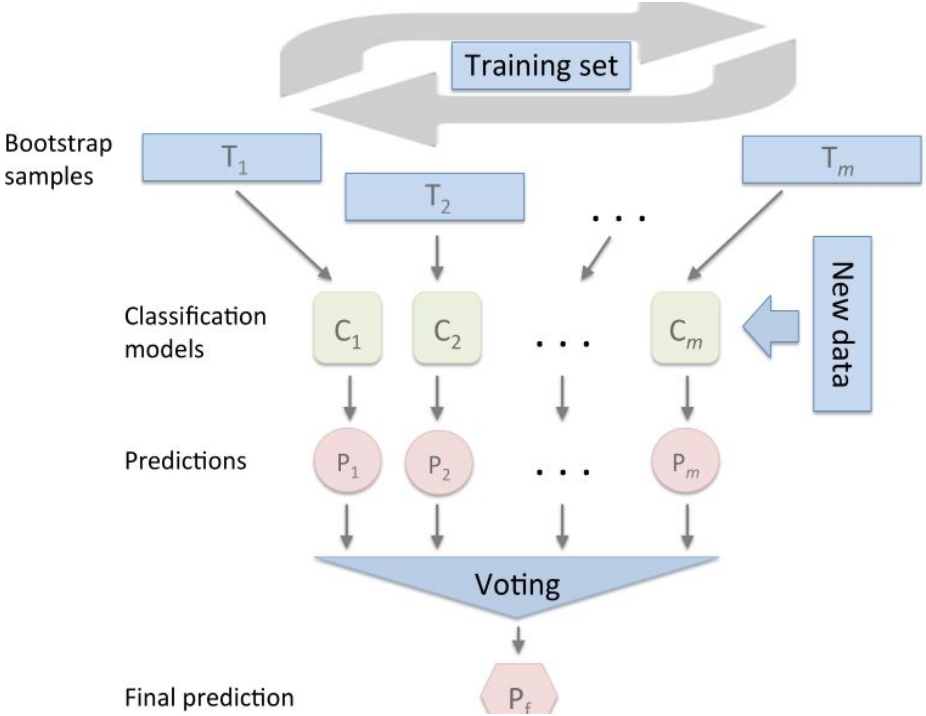
Ensemble Learning

- Bagging: reduce the variance in a model
  - Random Forest
- Boosting: reduce the bias in a model
  - Ada-Boost, XGBoost, Gradient Boosted Decision Trees
- Stacking: increase the prediction accuracy of a model
  - Mlxtend library
- Cascading: the class of models is very accurate
  - Suitable for the cases you can not afford to make a mistake

Bagging

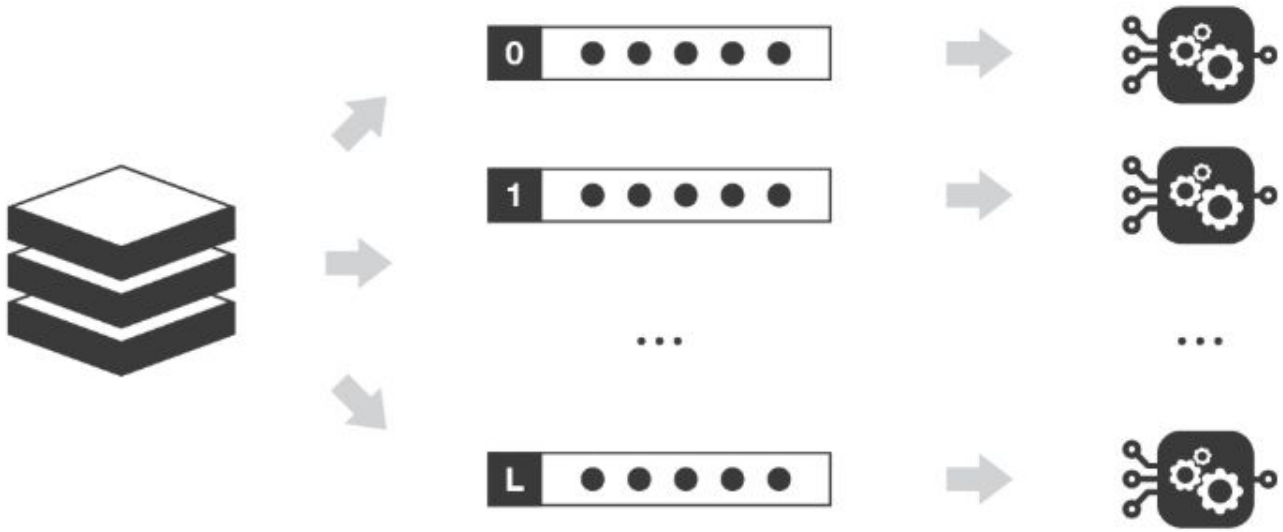
Bagging

- Bootstrap aggregation
- Train *m* classifier from *m* bootstrap replication
- Combine outputs by voting/averaging
- Decreases error by decreasing the variance
- Random Forest (Randomly select features)
- ExtraTrees (Randomized top-down split)



Bootstrapping

Base Learner



Source: Towards Data Science

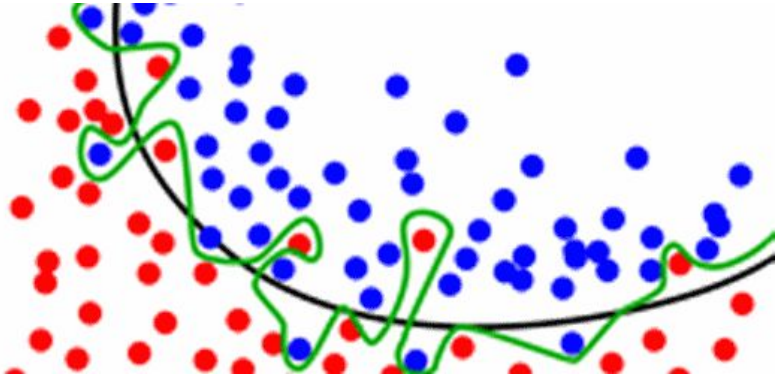
Majority Voting

- **Equal**: the difference between the average
- **Weighted**: best model get more weight in a vote

MODEL	PUBLIC ACCURACY SCORE
GradientBoostingMachine	0.65057
RandomForest Gini	0.75107
RandomForest Entropy	0.75222
ExtraTrees Entropy	0.75524
ExtraTrees Gini (Best)	**0.75571**
Voting Ensemble (Democracy)	0.75337
Voting Ensemble (3*Best vs. Rest)	**0.75667**

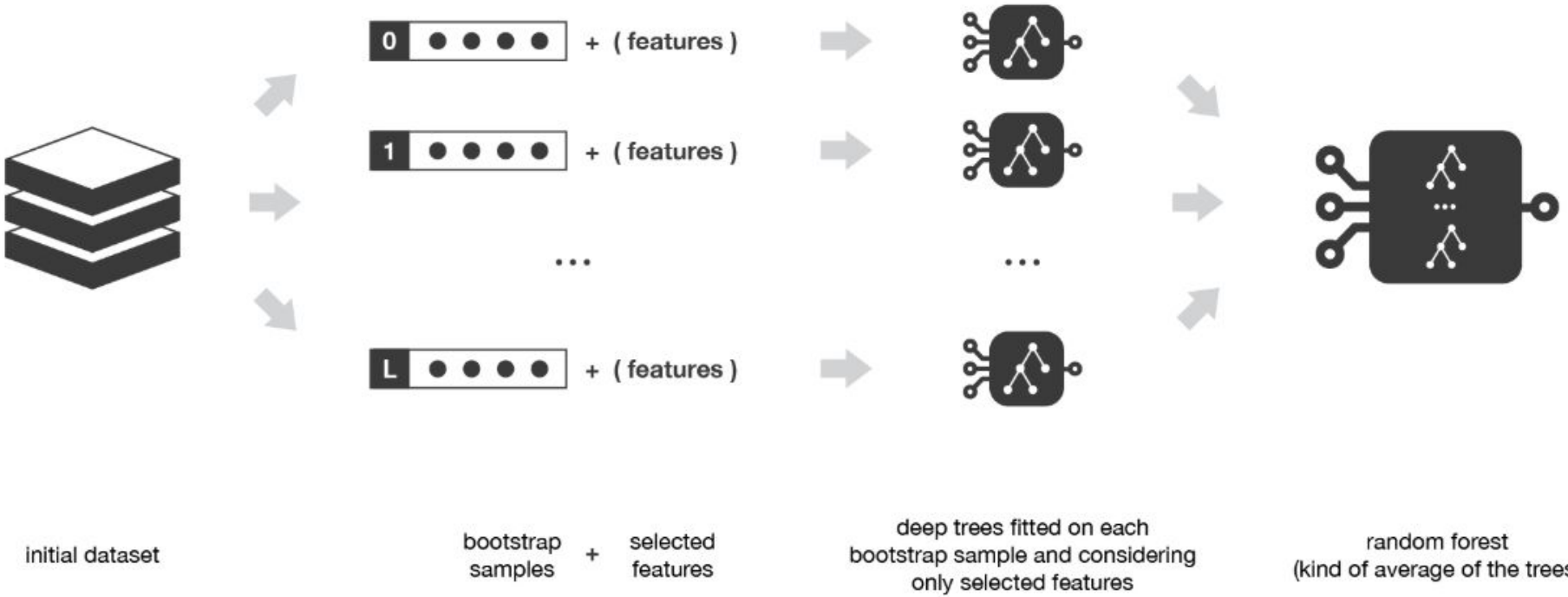
Average

- Take the average of several models' output
- Average multiple green lines -> black line (reduce overfit)



Random Forests

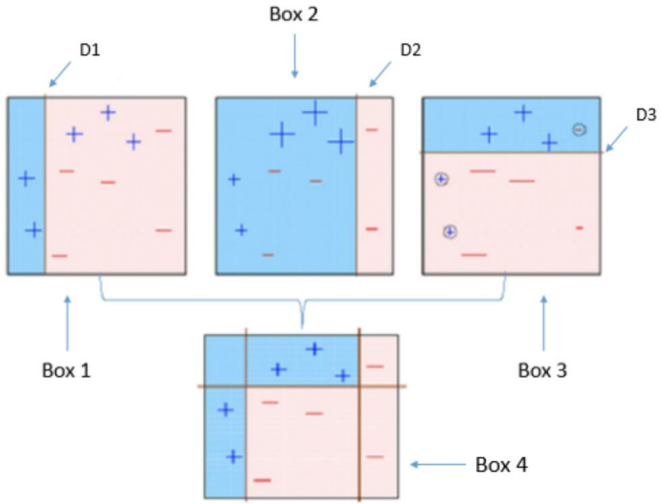
Source: Towards Data Science



Boosting

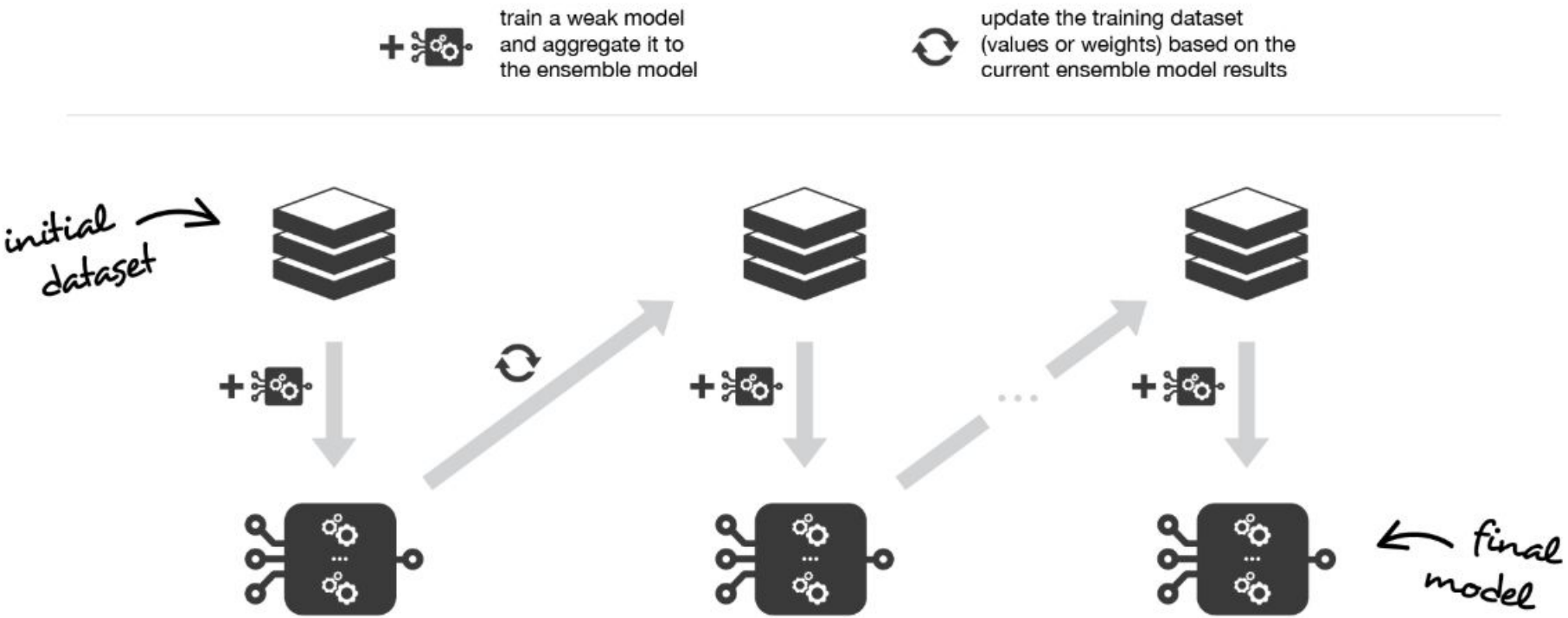
Boosting

- Training samples are given weights (initially same weight)
- At each iteration, a new hypothesis is learned.
- Training samples are reweighted to focus the model on samples that the most recently learned classifier got wrong.
- Combine output by voting
- Gradient Boosting, Adaboost, XGBoost, LightGBM



Boosting

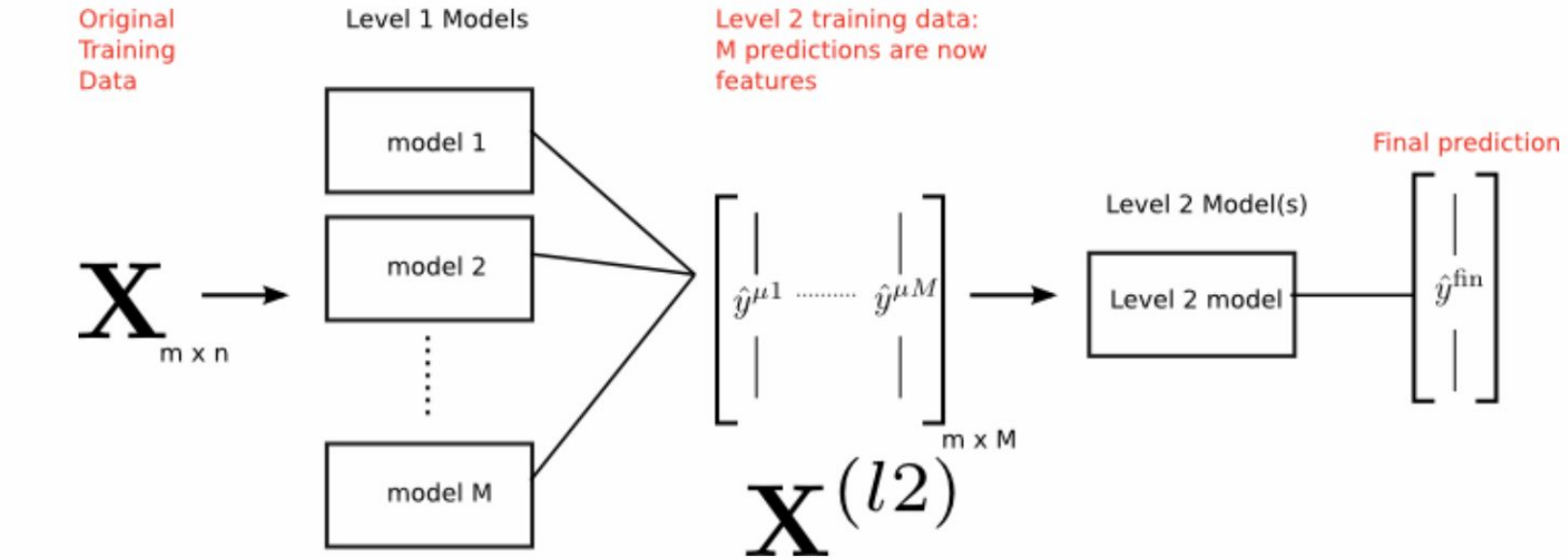
Source: Towards Data Science



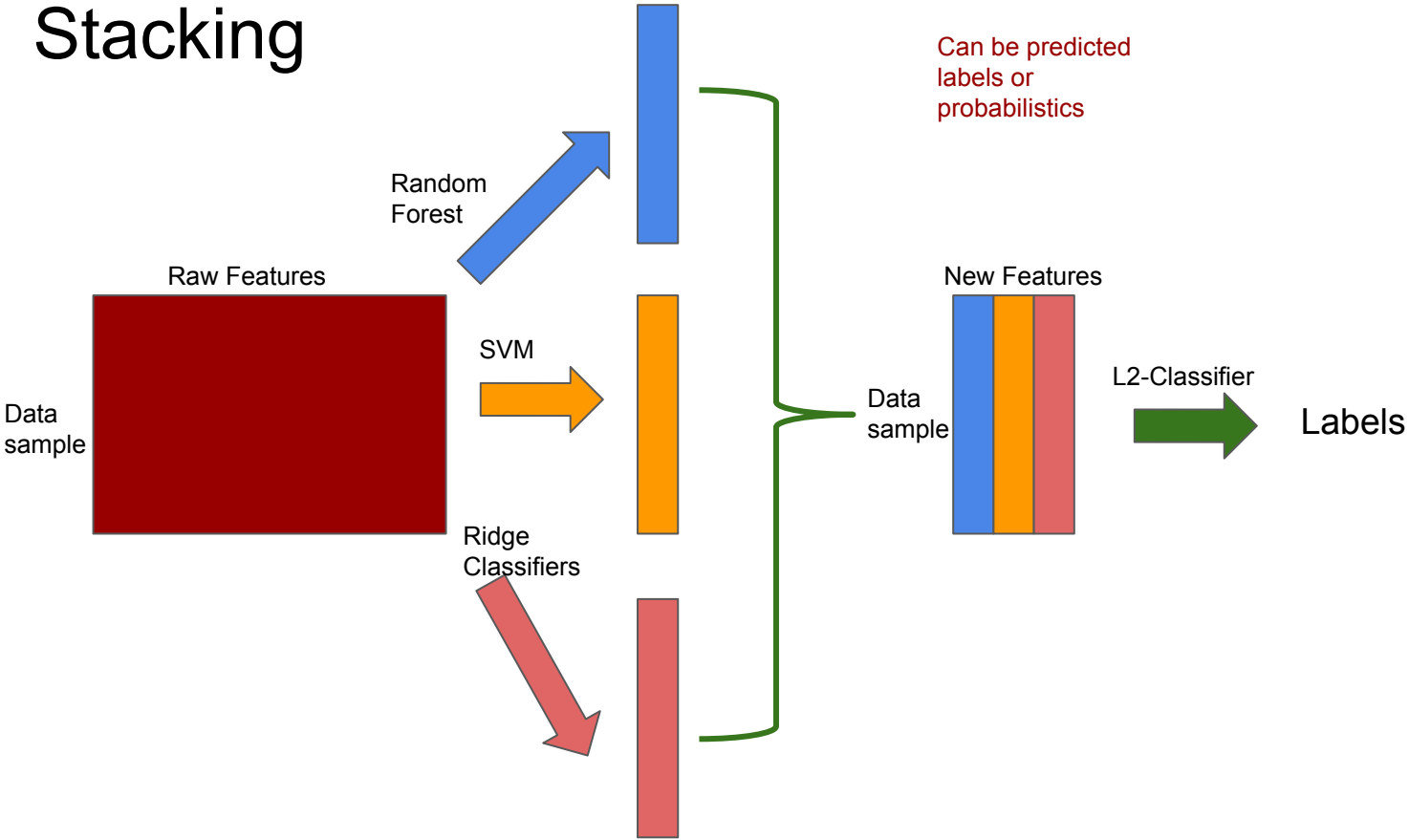
Stacking

Stacking

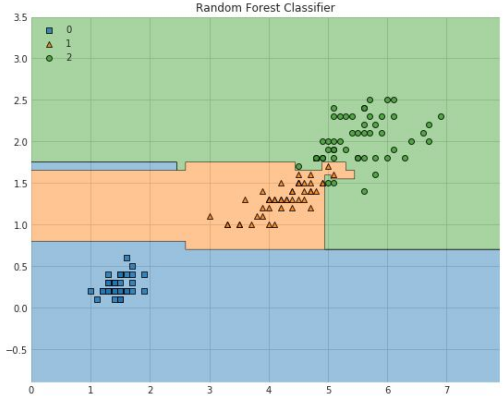
- Core idea: use a pool of base classifiers, then using another classifier (stacker) to combine their prediction for the final decision

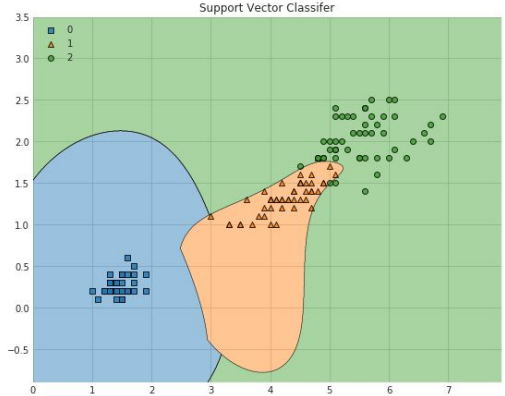


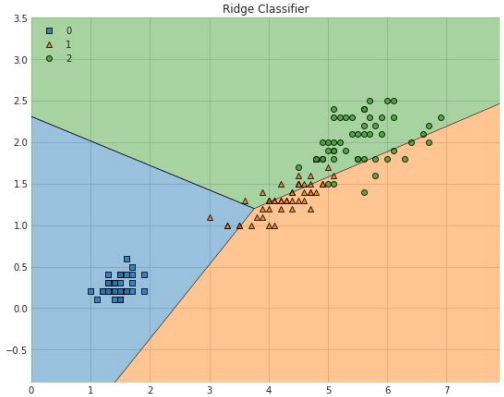
Stacking

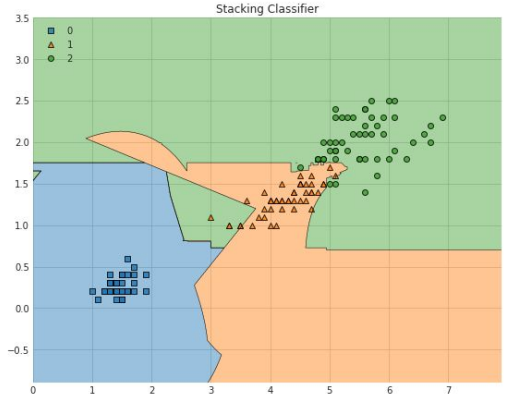

Can be predicted labels or probabilistics
Random Forest
Raw Features
SVM
Data sample
Ridge Classifiers
New Features
Data sample
L2-Classifier
Labels

Decision Regions: Demo Case









Cascading

Cascading

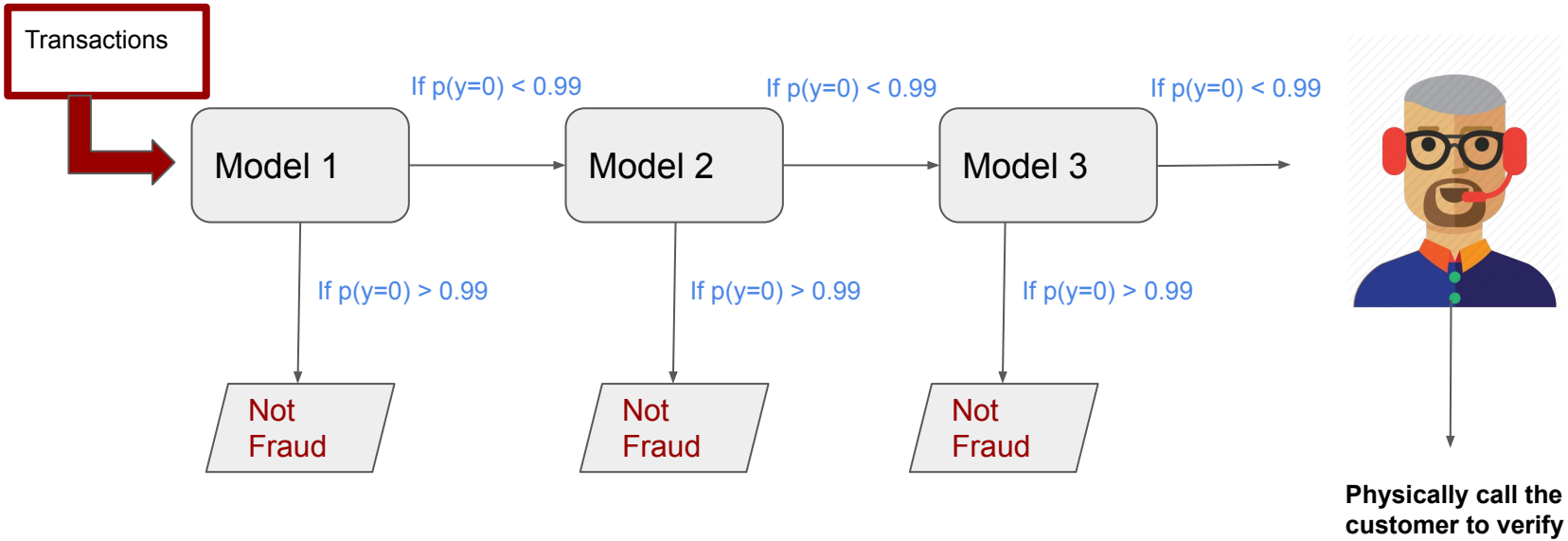
- Literally, cascading means “a process whereby something, typically information or knowledge, is successively passed on”

- In ML context, we build a sequence of models. The informations are the model outputs.

- It is suitable for the scenarios that requires a very high accuracy.
  - For example, credit card fraud detection

One of Human-Centered AI Systems

- Fraud detection: binary classification
  - The accuracy of fraud case should be very high. It means that we should not miss any fraud transactions that may cause losses
  - Label 0: *Normal*; Label 1: *Fraud*



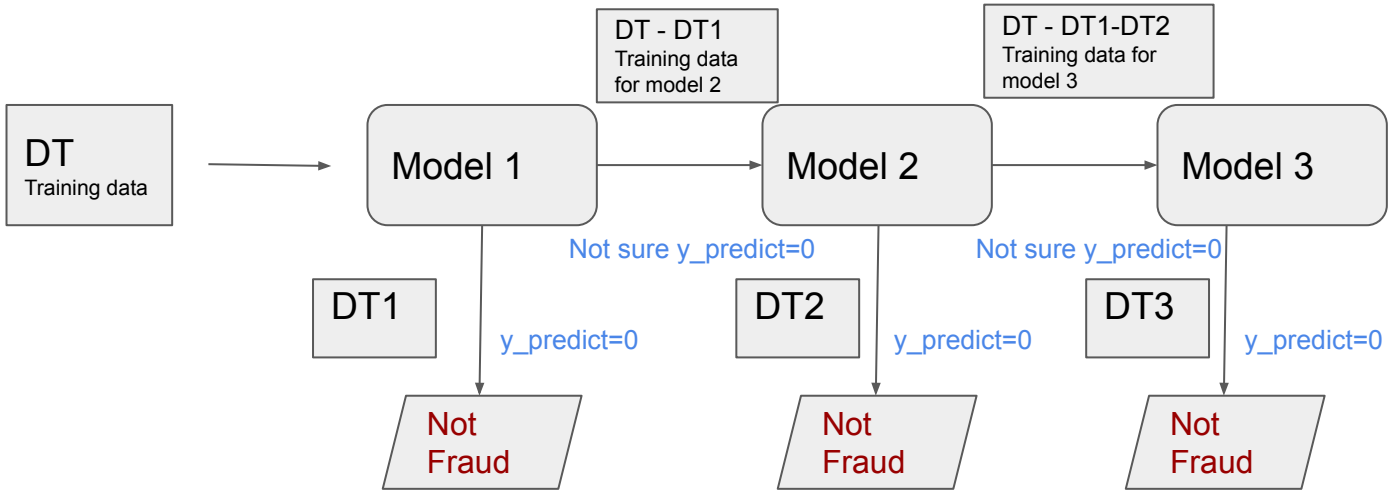
Training

- Training data denoted as DT. It contains data samples with labels 0 and 1

- Train model 1 on the whole DT. Then, we apply the model 1 on the whole DT. DT1 dataset will be the collections of all points with predicted labels of 0.
- Train model 2 on the dataset difference DT - DT1. Then, apply the model 2 on the whole DT-DT1. DT2 dataset will be the collections of all points with predicted labels of 0.
- Repeat the process for model 3, …..

The key: the subsequent model will only train over the datasets that the previous models are not confident.

Training


DT
Training data
Model 1
DT - DT1
Training data
for model 2
Model 2
DT - DT1-DT2
Training data for
model 3
Model 3
Not sure y_predict=0
Not sure y_predict=0
DT1
DT2
DT3
y_predict=0
y_predict=0
y_predict=0
Not
Fraud
Not
Fraud
Not
Fraud

From Competition to Industry

Netfilx Competition

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 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

1 The winning solution is a final combination of **107** algorithms;

2 Are not fully implemented.

Some possible pitfalls

- Exponentially increasing training times and computational requirements
- Increase demand on infrastructure to maintain and update these models.
- Greater chance of data leakage between models or stages in the whole training.

In a nutshell

- **No Free Lunch Theorem**: There is no one algorithm that is always the most accurate.
- Our efforts should focus on obtaining base models which make different kinds of errors, rather than obtaining highly accurate base models
- What we need to do is to build weak learners that are at least more accurate than random guessing
- Keep trying (experimenting, tuning, etc.) !