Explainability-Accuracy Tradeoff

What is Machine Learning Ensembles?

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.

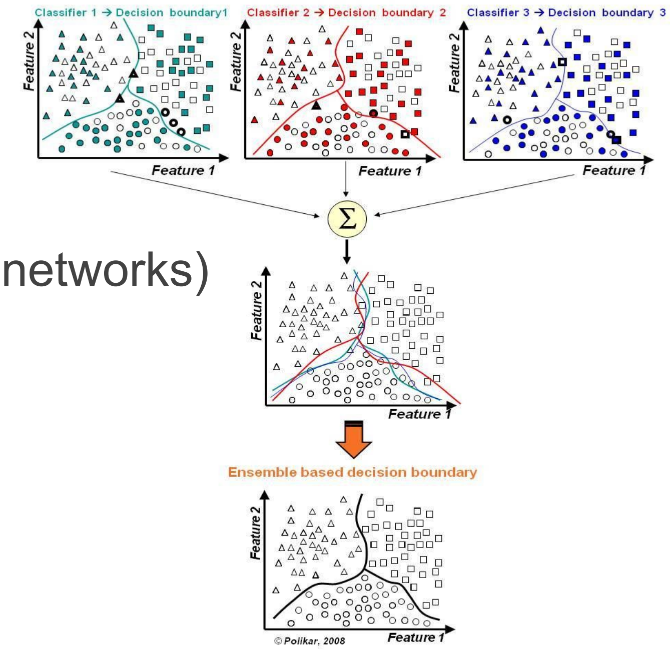
Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	**89.731**	**92.215**
2 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	88.592	90.859
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 which when combined have higher accuracy

- Strong v.s. Weak learner

- Stable (kNN) v.s. Unstable (decision trees, neural networks) machine learning algorithms.

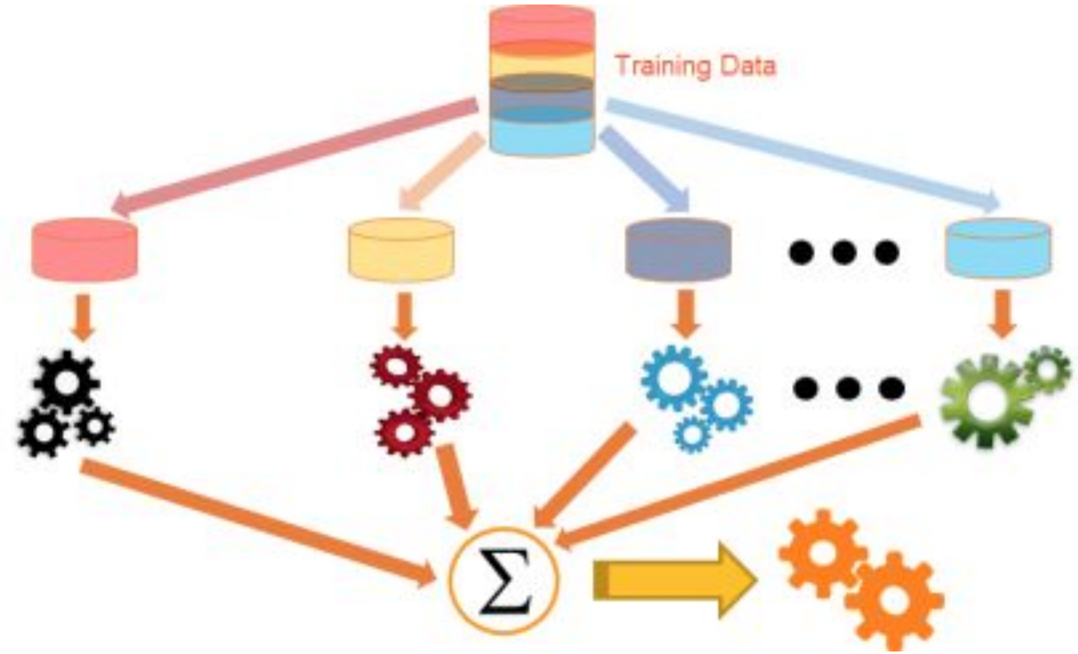


Why Ensemble?

- Reduce Bias

- Reduce Variance

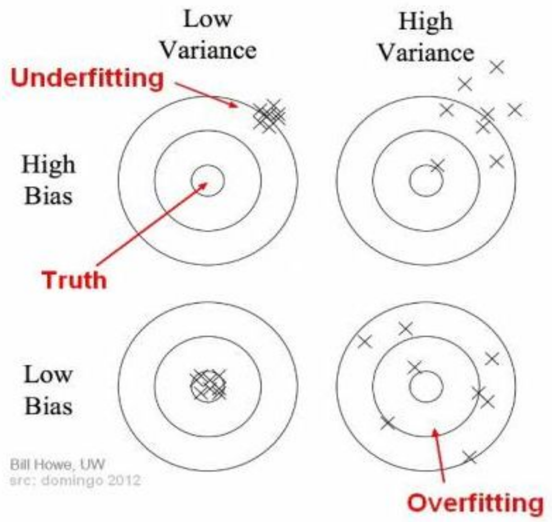
- Prediction Error:
  = Bias ^2
  + Variance
  + Irreducible Error



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.

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

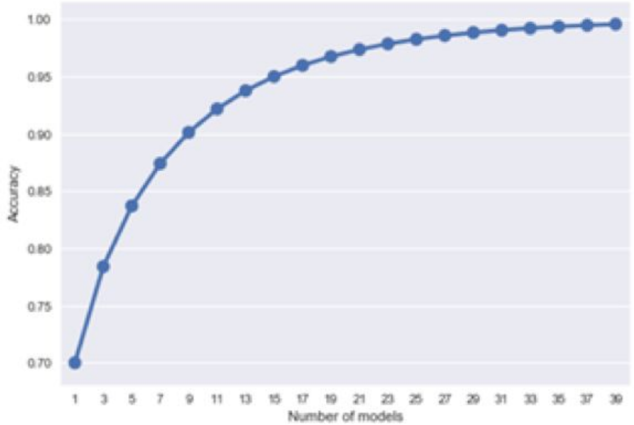
What is the probability that k choose ***i classifiers*** whose predictions are **wrong** and the rest ***k-i models***' outputs are **correct**.

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

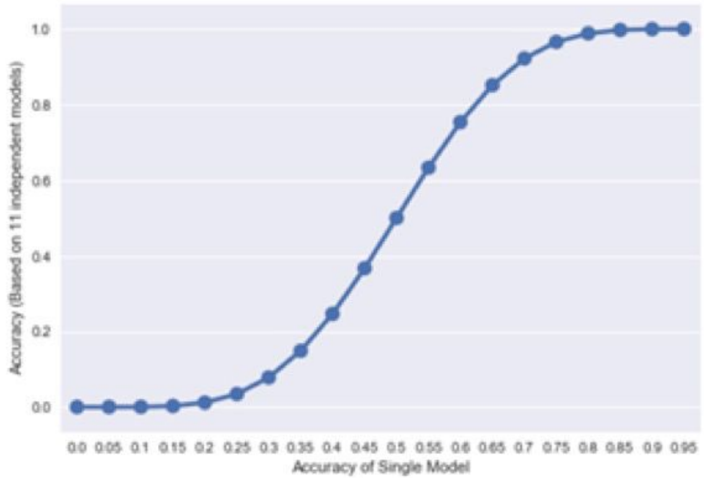
k	Ensemble Accuracy
1	0.7
3	0.784
5	0.83692
11	0.92177520904
101	0.999987057446



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



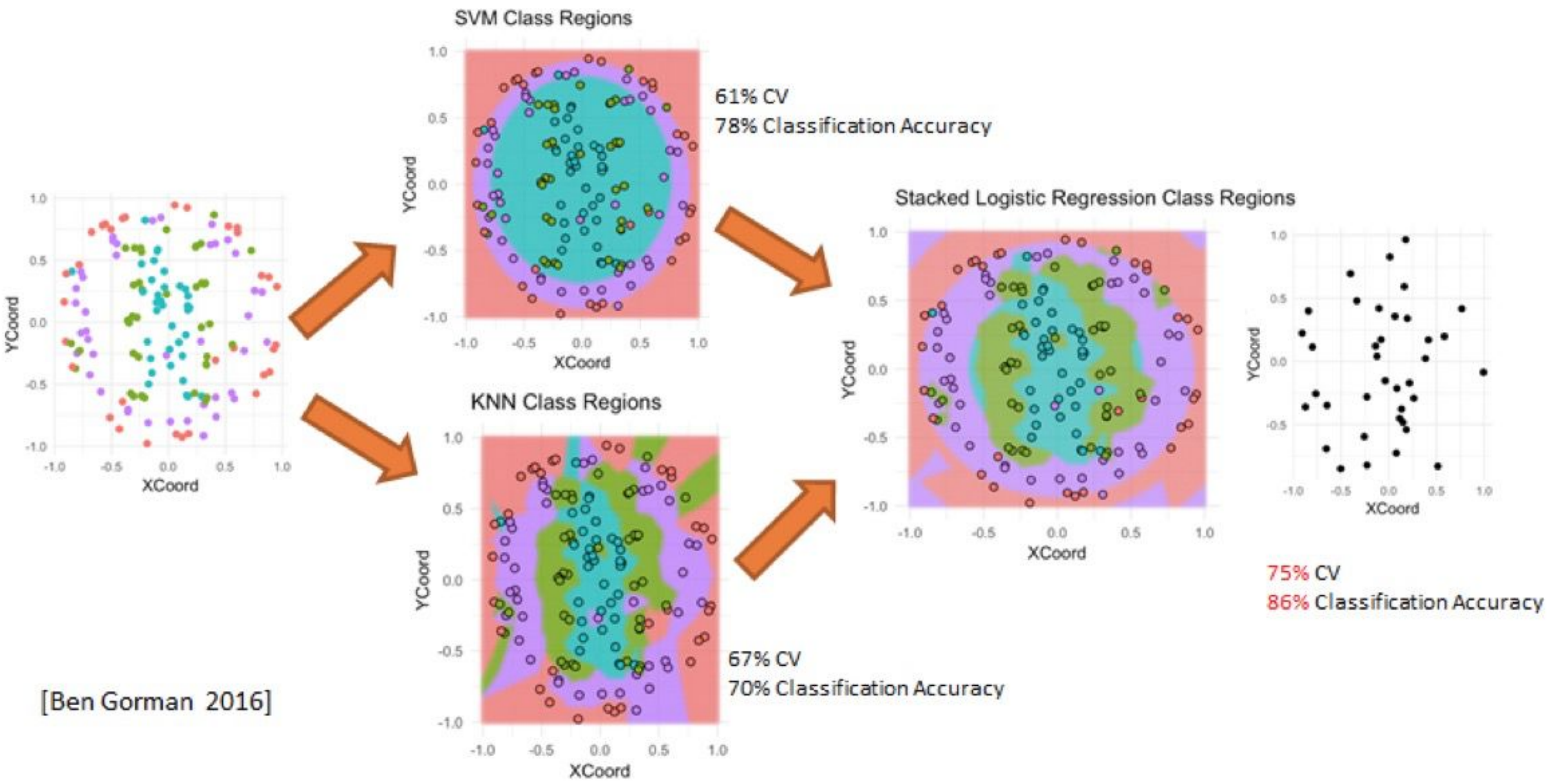
Reduce Variance

- 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:

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

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

Machine Learning Ensembles



[Ben Gorman 2016]

Common Ensemble Techniques

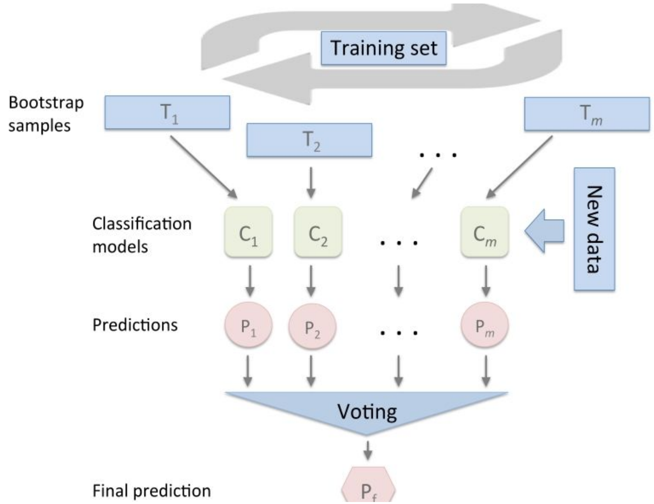
Ensemble Learning

- 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 very accurate
  - Bias toward precision from recall
  - Suitable for the cases you can not afford to make a mistake

Bagging

Bagging

- A.k.a Bootstrap aggregation
- Train m classifier from m bootstrap replica
- Combine outputs by voting
- Decreases error by decreasing the variance
- Random Forest (Randomly select features)
- ExtraTrees (Randomized top-down split)



3.2.4.3.1. sklearn.ensemble.RandomForestClassifier

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None*) [source]

Majority Voting

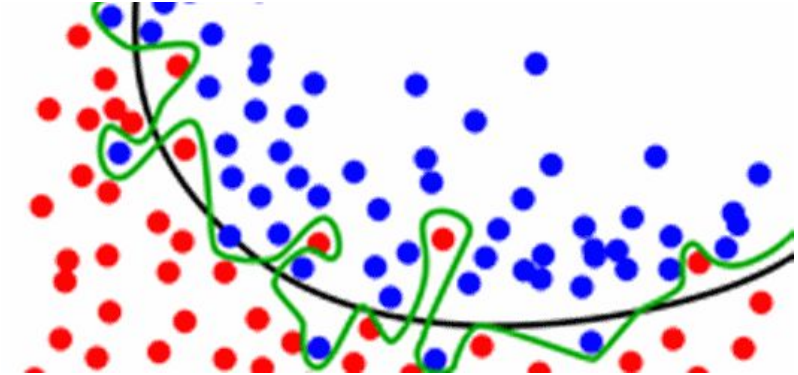
- **Equal**: the difference between the average

- **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

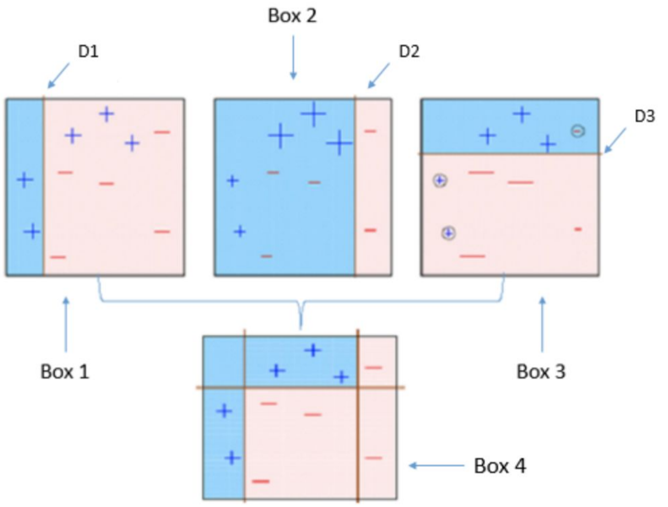
- Take the average of several models' output
- Average multiple green lines -> black line (reduce overfit)



Boosting

Boosting

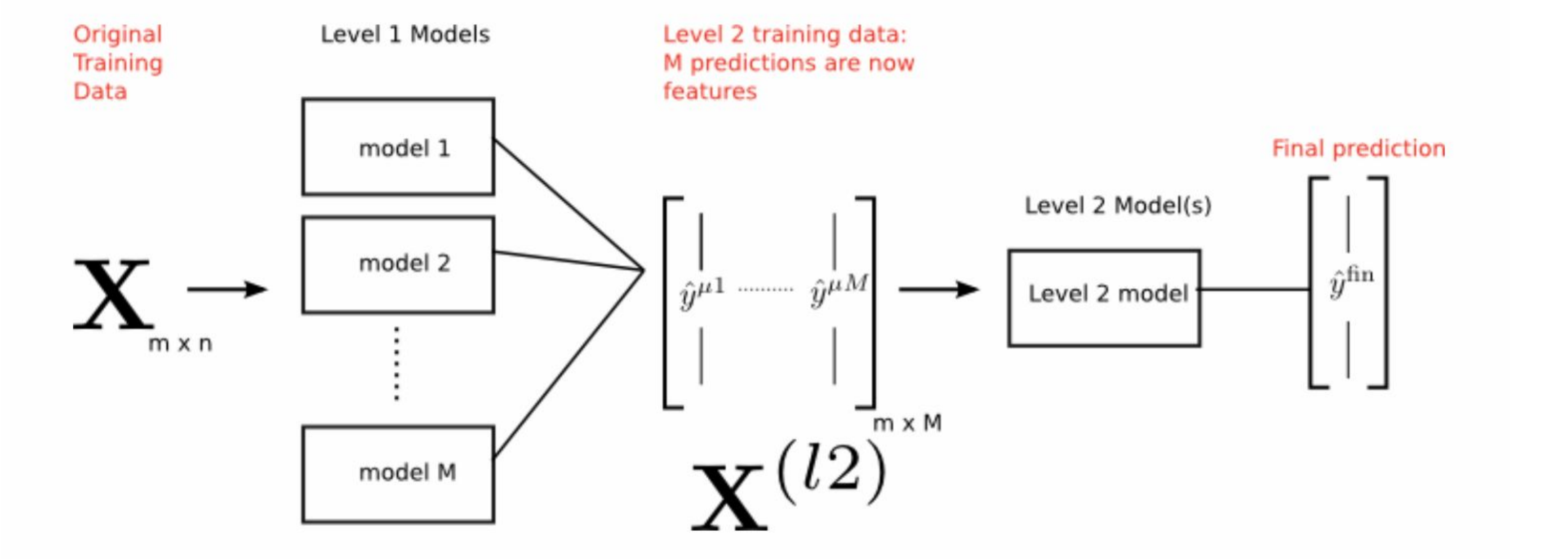
- Training samples are given weights (initially same weight)
- At each iteration, a new hypothesis is learned.
- Training samples are reweighted to focus the model on samples that the most recently learned classifier got wrong.
- Combine output by voting
- Gradient Boosting, Adaboost, XGBoost, LightGBM



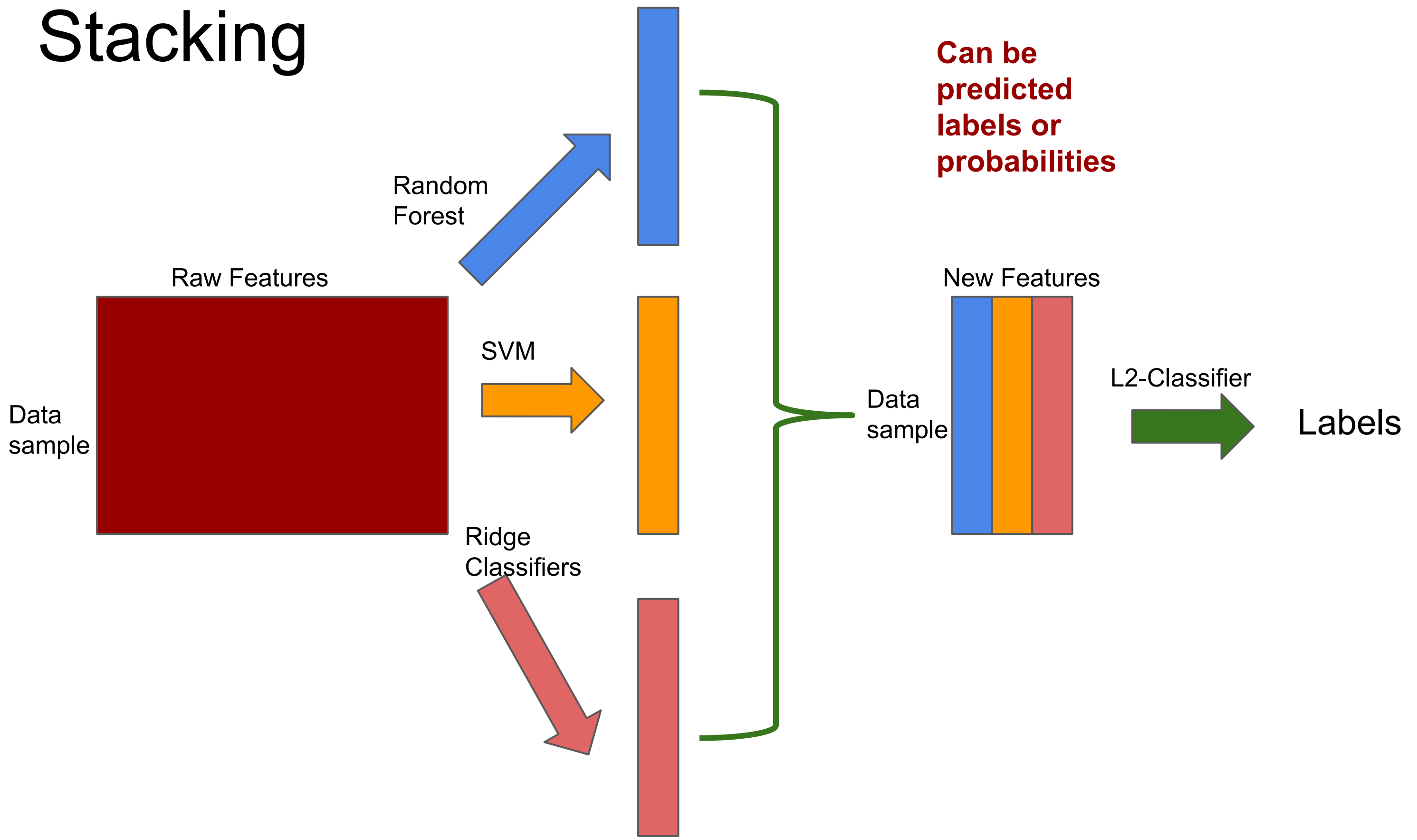
Stacking

Stacking

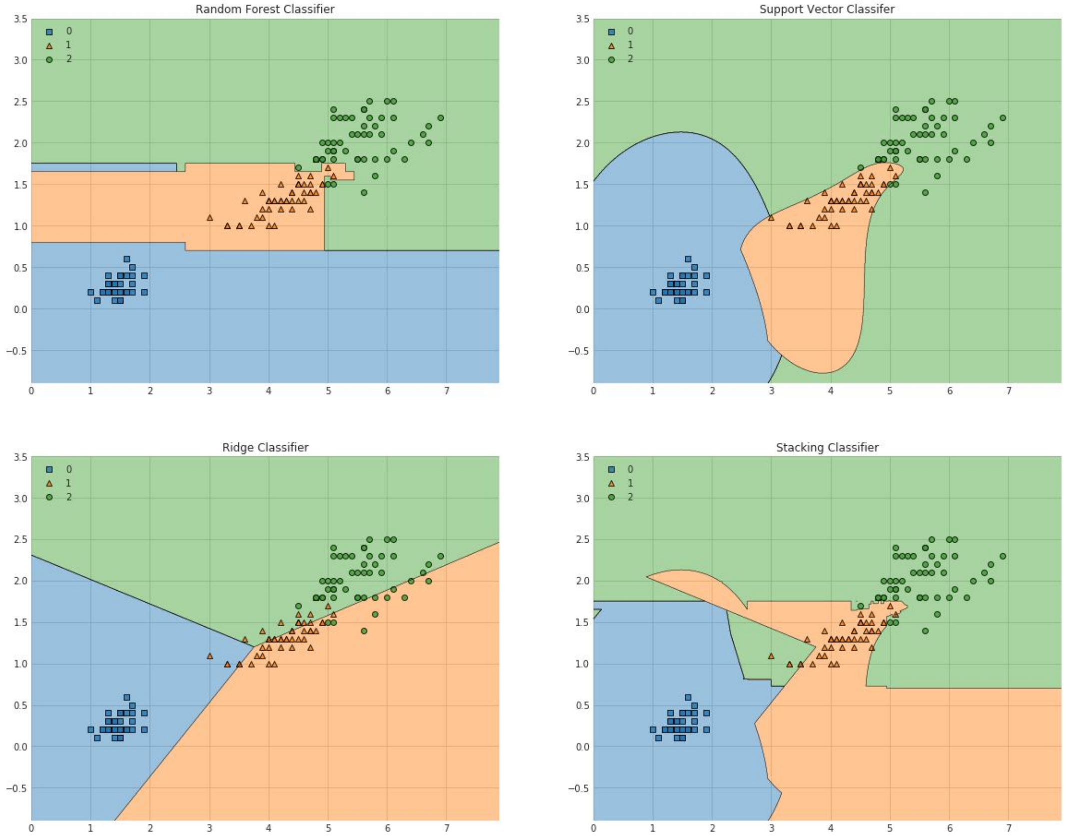
- Core idea: use a pool of base classifiers, then using another classifier (stacker) to combine their prediction for the final decision



Stacking


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

Decision Regions: Demo Case



Cascading

Cascading

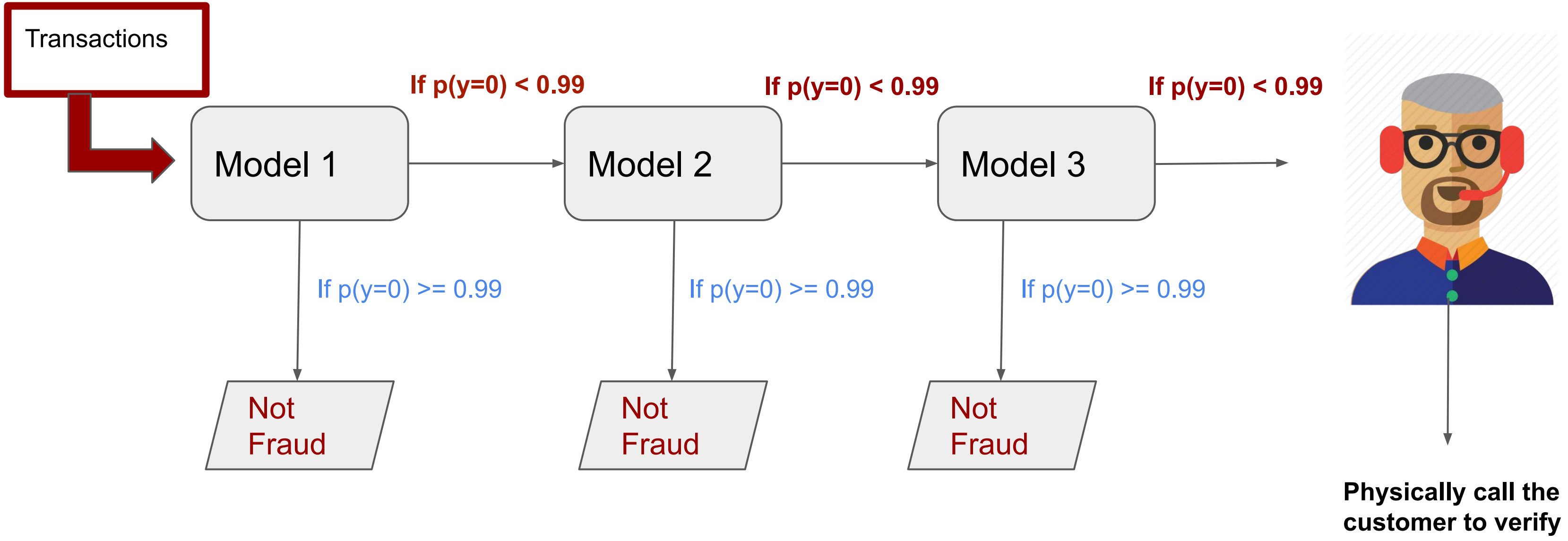
- Literally, cascading means "a process whereby something, typically information or knowledge, is successively passed on"

- 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*



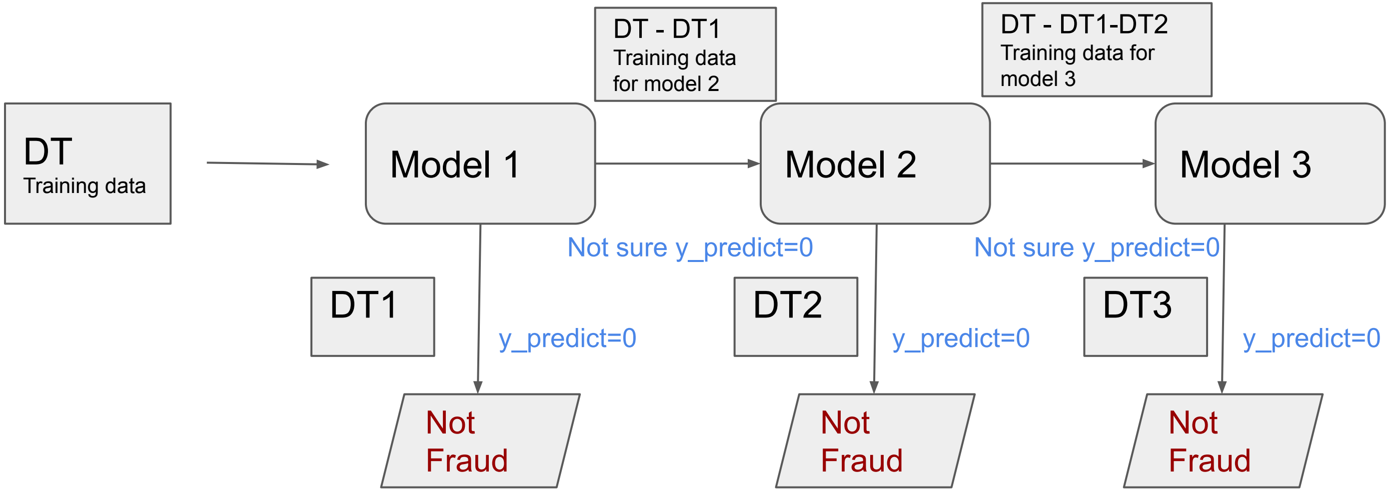
Training

- 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
y_predict=0
DT2
y_predict=0
DT3
y_predict=0
Not
Fraud
Not
Fraud
Not
Fraud

From Competition to Industry

Netfilx Competition

Netflix Prize COMPLETED

Home | Rules | Leaderboard | Update | Download

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 infra. to maintain and update these models.
- Greater chance of data leakage between models or stages in the whole training.

In a nutshell

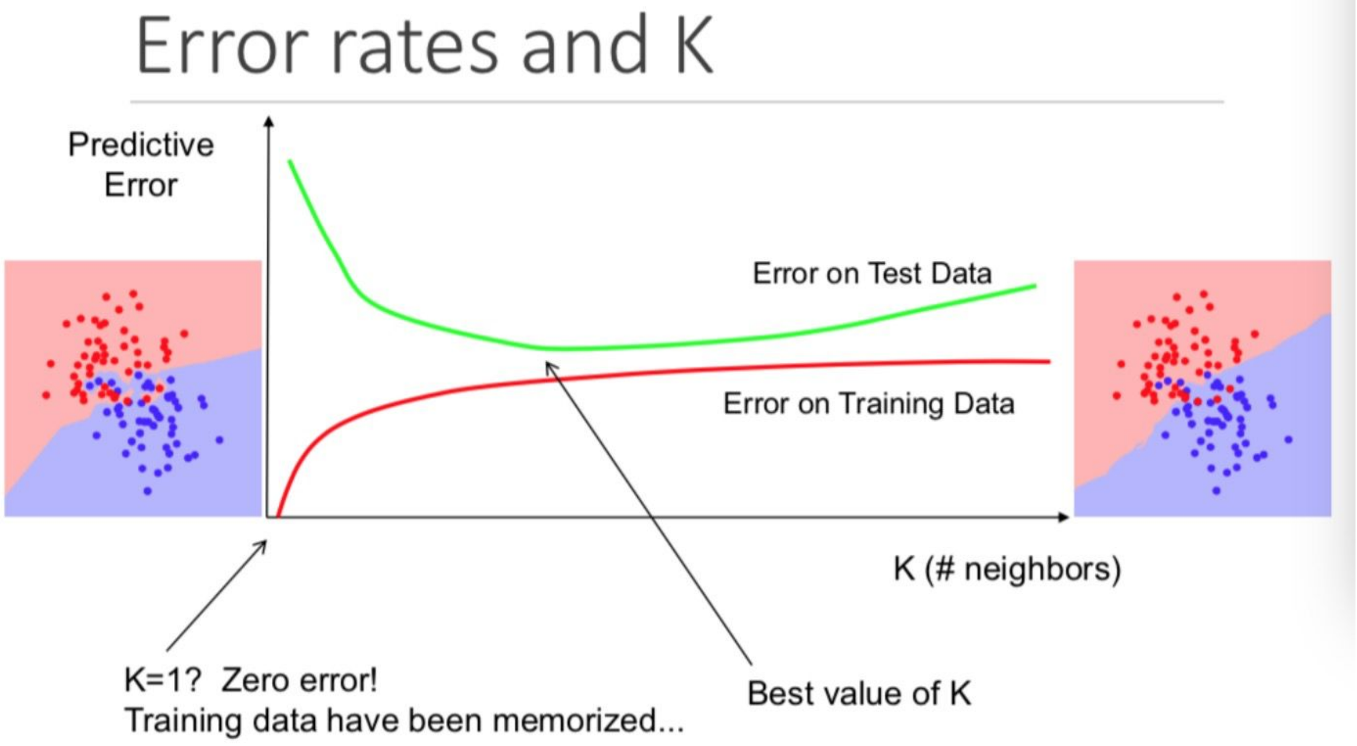
- **No Free Lunch Theorem**: There is no one algorithm that is always the most accurate.
- Our efforts should focus on obtaining base models which make different kinds of errors, rather than obtaining highly accurate base models
- What we need to do is to build weak learners that are at least more accurate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

Proposal Submission

- Due on Feb 14 @23:59pm
- File should be named as proposalXX.pdf where XX is your group number.
- 3-5 pages
- It is totally acceptable that there is some inconsistency between the final demo and your initial proposal.
- Key messages in proposal should contain:
    - What kind of problem you want to solve?
    - How can the proposed problem be solved by ML techniques?
    - Why is it interesting or what is its business impact?

Open Questions

KNN’s Complexity



Error rate and K. Credit: http://sameersingh.org/courses/gml/fa17/sched.html

From Notebook in Week2

Why is this improper cross-validation?

- Normally, we split the data into training and testing sets **before** creating the document-term matrix. But since `cross_val_score` does the splitting for you, we passed it the feature matrix (`X_dtm`) rather than the raw text (`X`).
- However, that does not appropriately simulate the real world, in which your out-of-sample data will contain **features that were not seen** during model training.

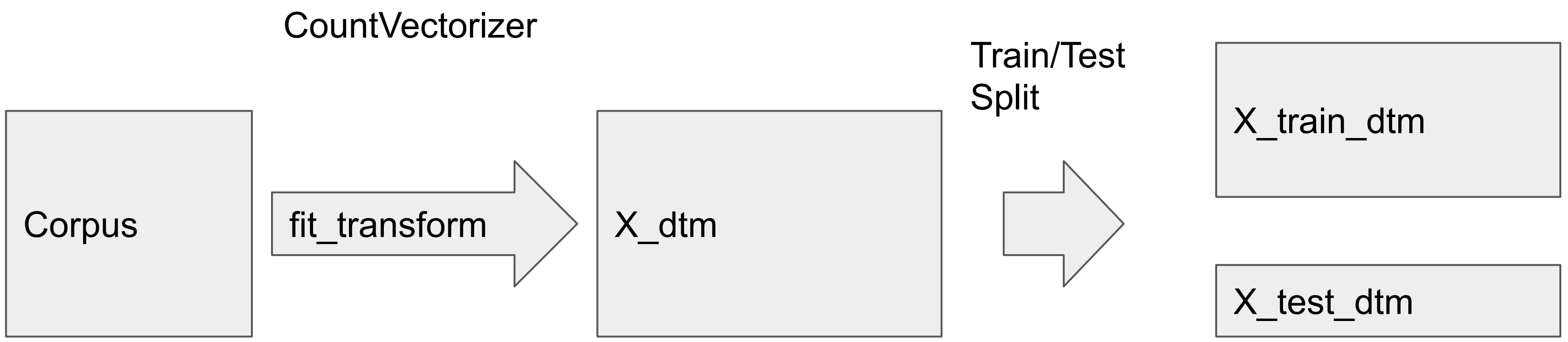
What's the solution?

- We need a way to pass `X` (not `X_dtm`) to `cross_val_score` , and have the feature creation (via `CountVectorizer`) occur **within each fold** of cross-validation.
- We will do this by using a `Pipeline` .

Data Leakage

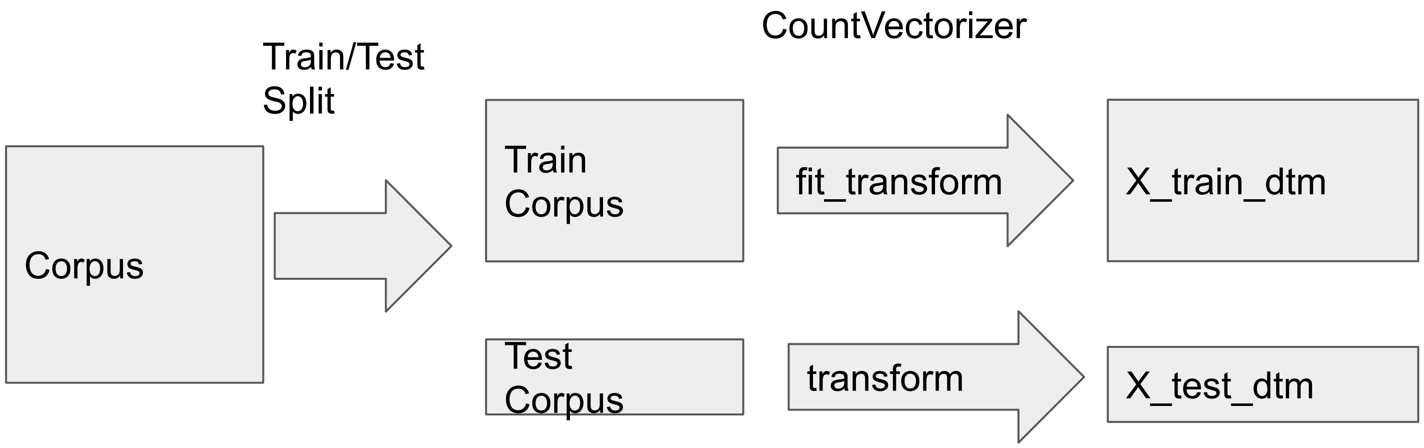
- **When the data you are using to train a machine learning algorithm happens to have the information you are trying to predict.**
- **If any other feature whose value would not actually be available in practice at the time you's want to use the model to make a prediction, is a feature that can introduce leakage to your model.**

Data leakage can cause you to create overly optimistic if not completely invalid predictive models.



Data leakage happens:

1. **when countvectorizer was building vocabulary, it had adopted the information from testing data. Here, it is vocabulary from testing data.**
2. **In practice, you only have training corpus to build vocabulary when you want to get BoW features or Document-term matrix**



Vocabulary is only built from the training corpus.

Explainable AI

Treatment Recommendation

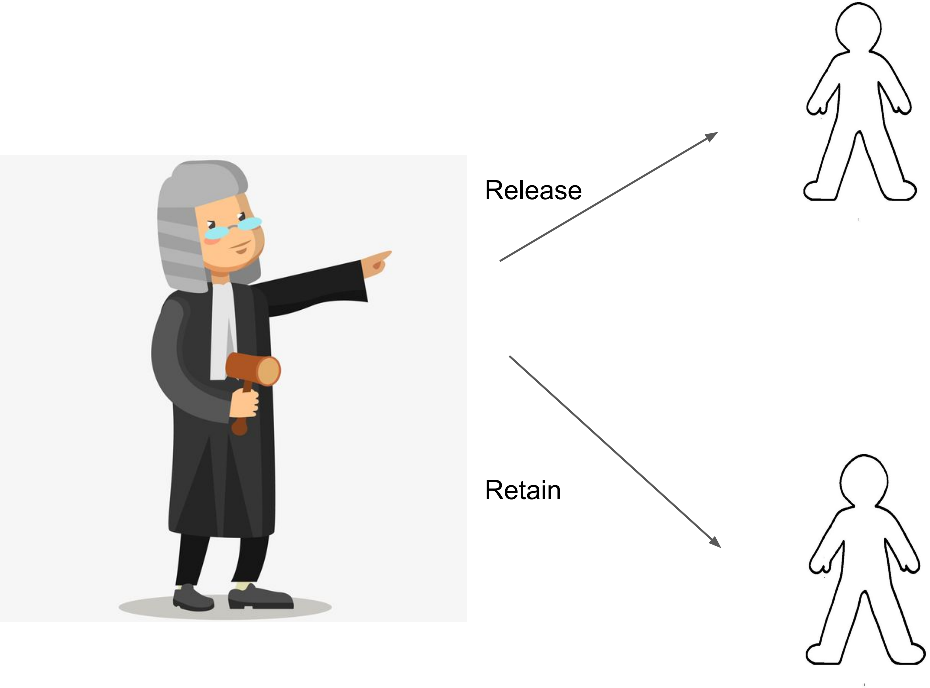
Which treatment should be given?
Options: quick relief drugs (mild),
controller drugs (strong)

Demographics: **age, gender, ..**
Medical History: **Has asthma?**

Symptoms: **Severe Cough, Sleepy**

Test Results: **Peak flow: Positive**

Bail Decision



High-Stakes Decisions

- The above examples all belong to high-stakes decisions. The decisions have a **huge impact on human well-being**.

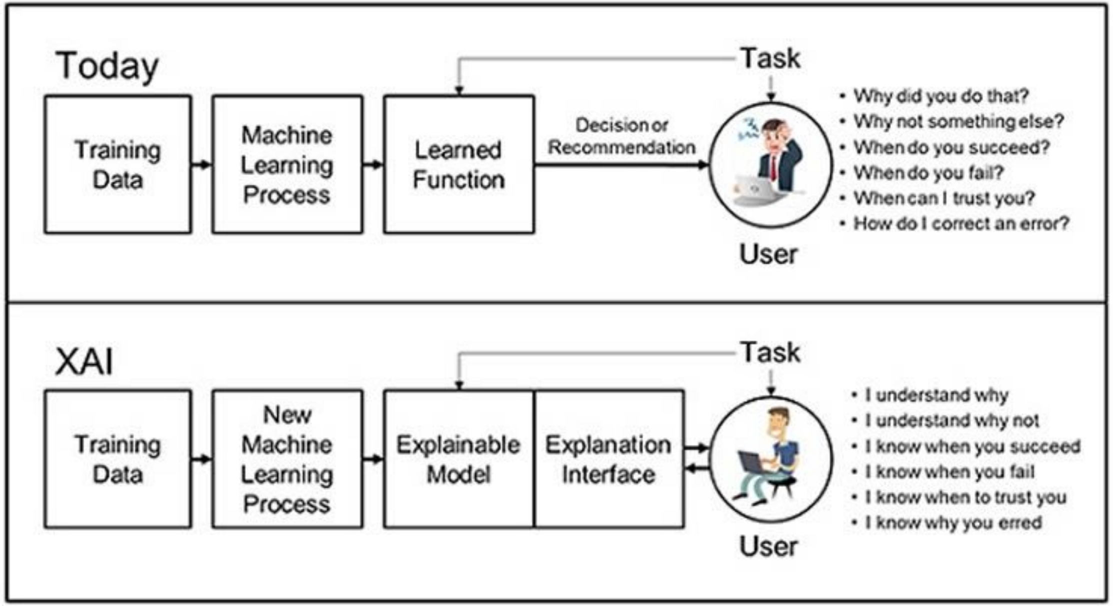
- What are those non high-stakes decisions?
  - Recommendations in E-commerces websites
  - When should I get up tomorrow?
  - …….

Black-Box Model

- If ML system is deployed in high-stakes decisions environment:
  - **Is accuracy important**?
  - Can we trust the machine learning model?

XAI

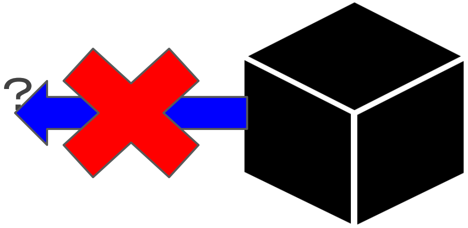
- **XAI**: ML models are explainable that enable end users to understand, appropriately trust, and effectively manage the emerging generation for AI systems.



DARPA's report

Why Model Insights Valuable

- When ML algorithms give us their predictions:
  - Do we understand our data?
  - Do we understand the model and the returned answers ?
  - It all comes to **model interpretability/insights**



- In banking, insurance and other heavily regulated industries, model interpretability is a serious legal mandate.

- In lots of critical areas such healthcare, government, bioinformatics, etc, rationale for models' decision is necessary for trust.

What is Interpretability

- Ability to explain or present in understandable terms to our humans

- However, no clear answers in psychology to:
    - What constitutes an explanation?
    - What makes some explanations better than the others?
    - When are explanation sought?

Properties of Interpretable Models

- Transparency
  - How exactly does the model work?
  - Details about its inner workings, parameters etc.
  - It has two dimensions: **Simulatability** and **Decomposability**

Transparency: Simulatability

- Can a person contemplate the entire model at once?
  - Need a very simple model

- A human should be able to take input data and model parameters and calculate prediction

- Simulatability: size of the model + computation required to perform inference
  - Decision trees: size of the model may grow faster than time to perform inference

Transparency: Decomposability

- Understanding each input, parameter, calculation
  - Decision trees, linear regression
- Inputs must be interpretable
  - Models with highly engineered or anonymous features are not decomposable

Linear Models First

- Prediction is the linear combinations of the features values, weighted by the model coefficients.

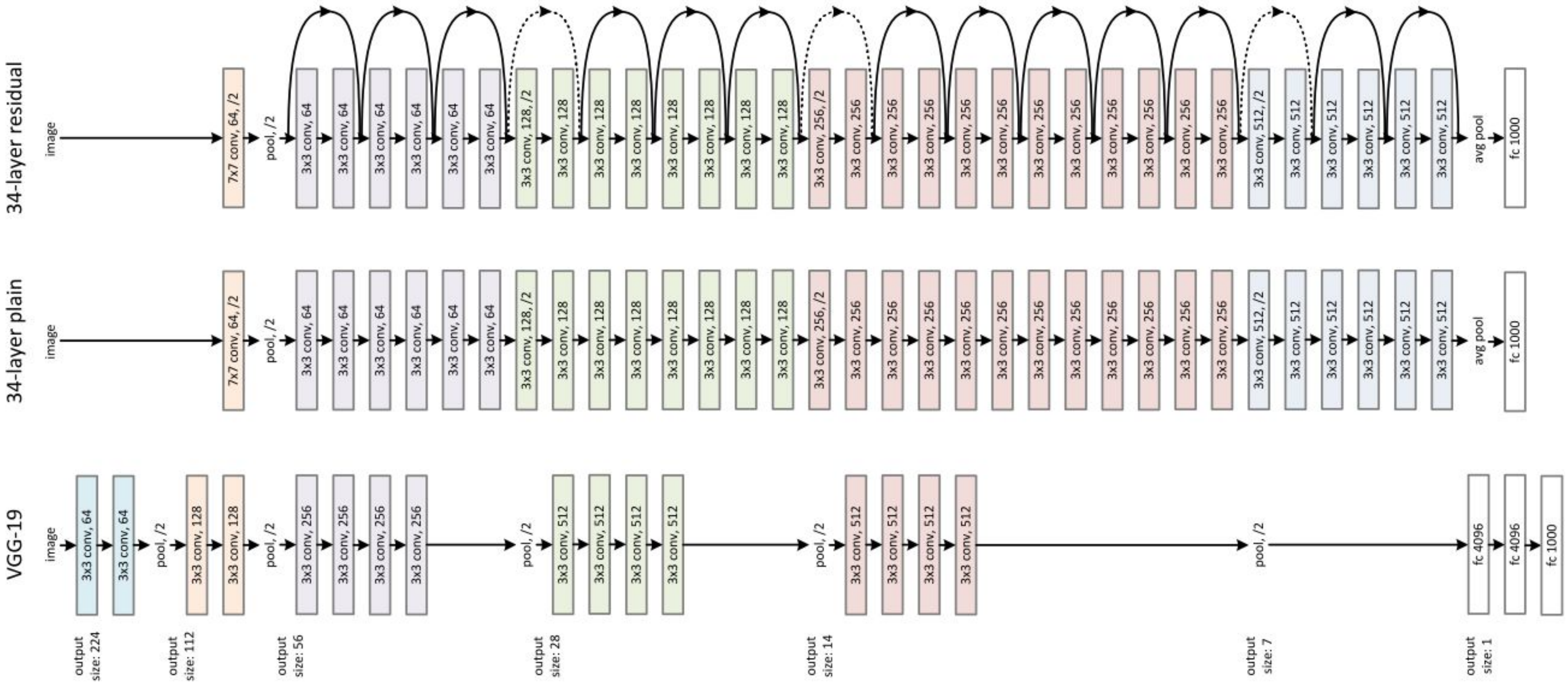
BT5153 A's chance = 0.2 + 0.1* GPA - 0.005 * Hours on Tiktok

One student's feature
GPA: 5
Hours on Tiktok: 60 hours

Feature Contributions
GPA: 0.5
Hours on Tiktok: -0.3

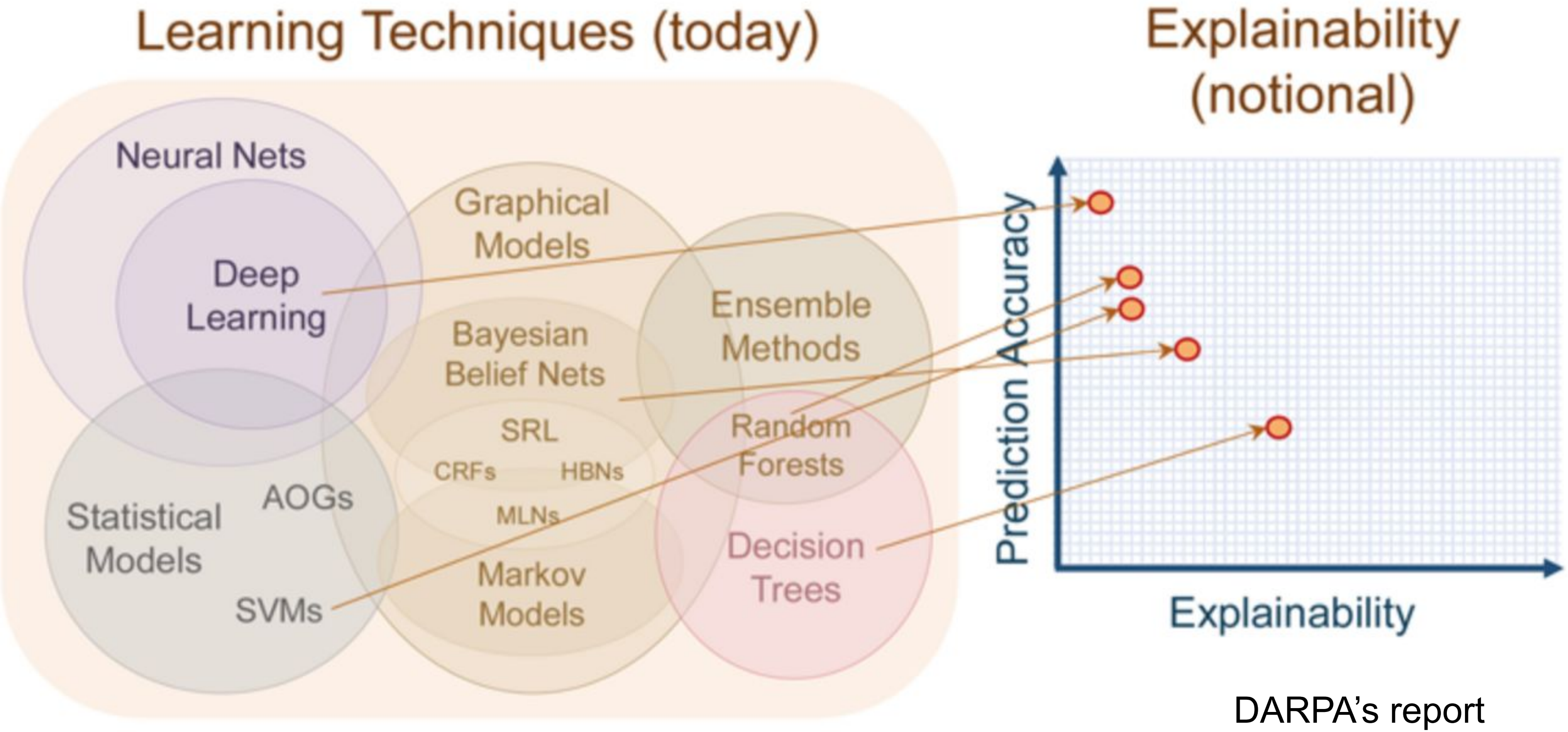
- Capability of linear models is limited.

Complex Models



For imagenet, they use 152 layers, which firstly achieved lower error rate compared to Humans in image recognition tasks.

Trade-off

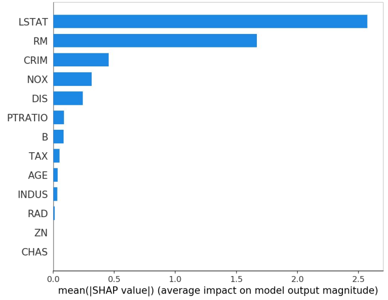


DARPA's report

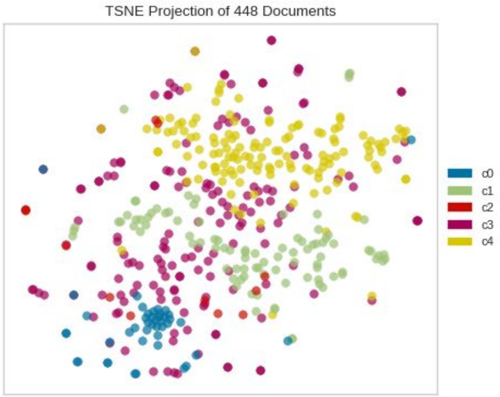
Taxonomy of Interpretability

- Intrinsic
  - Interpretability achieved through constraints imposed on the complexity of the ML model
  - Applied on tree-based, linear model
  - Constraints: Sparsity, monotonicity, causality or physical constraints

- Post hoc:
  - Explanation methods that are applied after model training
  - Open-source packages: LIME, SHAP, etc



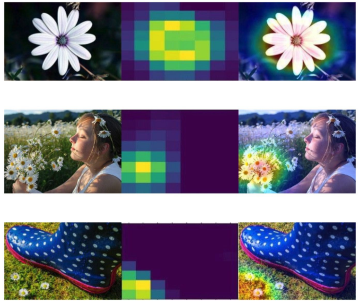
Post-hoc: Visualization



Source: yellowbricks

- Visualize high-dimensional data with t-SNE
  - 2D visualization in which nearby data points appear close
  - It works well on neural networks' hiddens outputs

- Perturb input data to enhance activations of certain nodes in neural nets:
  - Helps understand which nodes correspond to what aspects of the image
  - Eg., certain nodes might correspond to
    Concept: *flowers*



Images labeled as flowers

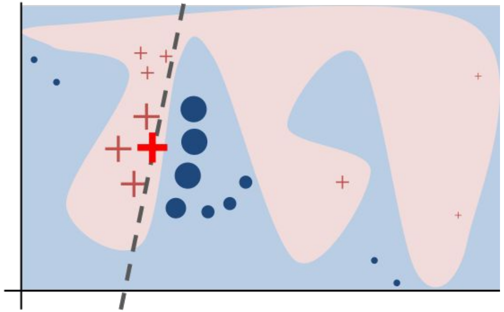
Source: https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b

Post-hoc: Example Explanations

- Reasoning with examples
- Eg., Patient A has a tumor because he is similar to these k other data points with tumors
- K neighbors can be computed by using some distance metric on learned representations.
    - Such as word2vec

Post-hoc: Local Explanations

- Hard to explain a complex model in its entirety
  - How about explaining smaller regions?



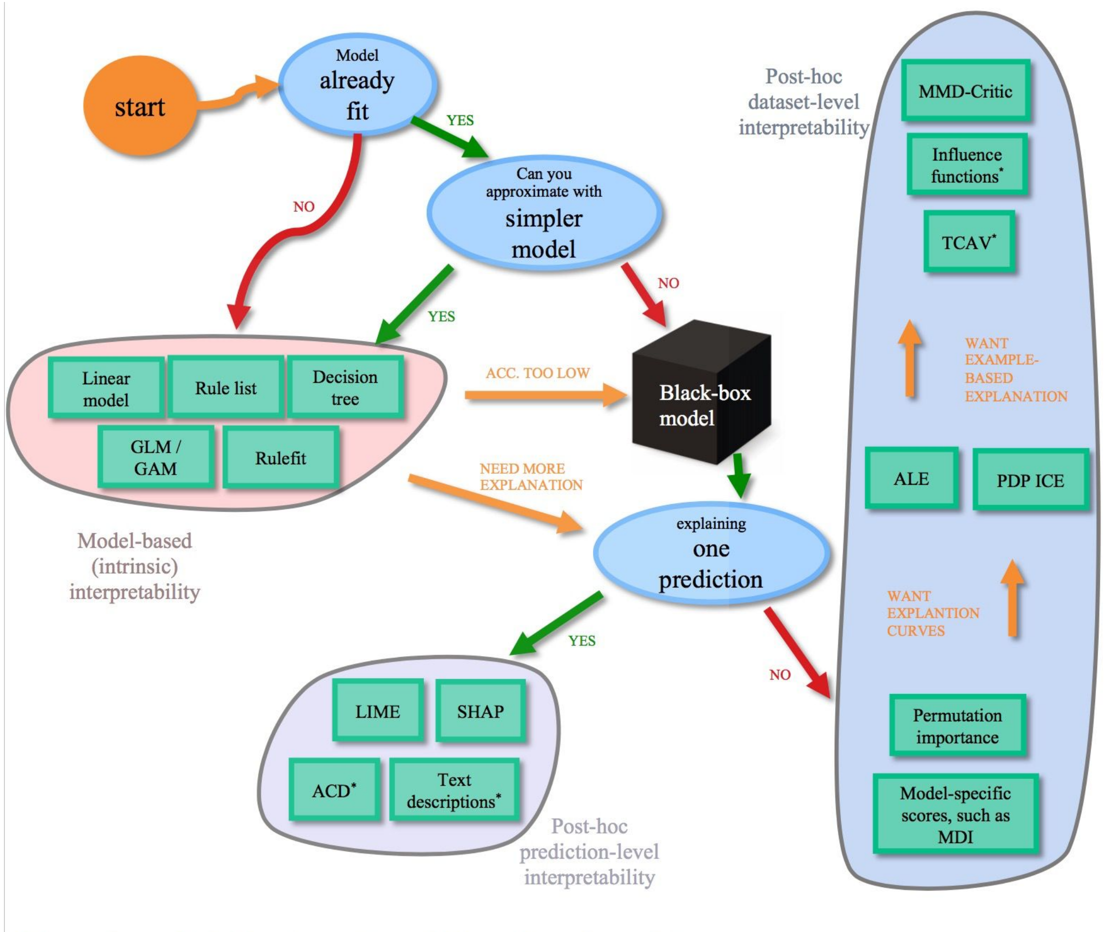
LIME (Ribeiro et. al)

  - Explains decisions of any model in a local region around a particular point
  - Learns sparse linear model

Post-hoc interpretations can mislead

- Do not blindly embrace post-hoc explanations!

- Post-hoc explanations can seems plausible but be misleading
    - They do not claim to open up the black-box;
    - They only provide plausible explanations for its behavior



* Denotes that a method only works on certain models (e.g. only neural networks)

interpretability cheat-sheet

View on github

Based on this interpretability review and the sklearn cheat-sheet.
More in this book + these slides.

Summaries and links to code

RuleFit – automatically add features extracted from a small tree to a linear model

LIME – linearly approximate a model at a point

SHAP – find relative contributions of features to a prediction

ACD – hierarchical feature importances for a DNN prediction

Text – DNN generates text to explain a DNN's prediction (sometimes not faithful)

Permutation importance – permute a feature and see how it affects the model

ALE – perturb feature value of nearby points and see how outputs change

PDP ICE – vary feature value of all points and see how outputs change

TCAV – see if representations of certain points learned by DNNs are linearly separable

Influence functions – find points which highly influence a learned model

MMD-CRITIC – find a few points which summarize classes

Source: https://github.com/csinva/csinva.github.io/blob/master/_notes/cheat_sheets/interp.pdf