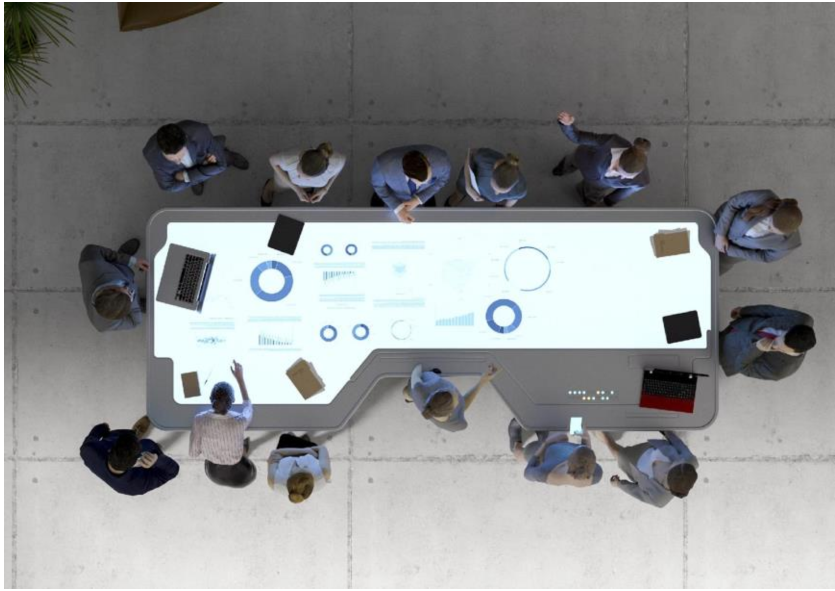
Machine Learning in Practice –

what to do if my ML models fail to achieve a desirable quality

Dr. Shou-de Lin
Chief Machine Learning Scientist, Appier
Professor, CSIE Dep, National Taiwan University
sdlin@csie.ntu.edu.tw

Talk Materials Based on Hands-On Experience in Solving Real-World ML Tasks, Including

- **Participating ACM KDD Cup for 6 years**
- **>50 Industrial collaboration**
- **Visiting Scholar in Microsoft Research from 2015~2016**
- **Serving as Chief ML Scientist in Appier since early 2020**



Team NTU’s Performance on ACM KDD Cup

KDD Cups	2008	2009	2010	2011	2012	2013
Organizer	Siemens	Orange	PSLC Datashop	Yahoo!	Tencent	Microsoft
Topic	Breast Cancer Prediction	User Behavior Prediction	Learner Performance Prediction	Recommendation	Internet advertising (track 2)	Author-paper & Author name Identification
Data Type	Medical	Telcom	Education	Music	Search Engine Log	Academic Search Data
Challenge	Imbalance Data	Heterogeneous Data	Time-dependent instances	Large Scale Temporal + Taxonomy Info	Click through rate prediction	Alias in names
# of records	0.2M	0.1M	30M	300M	155M	250K Authors, 2.5M papers
# of teams	>200	>400	>100	>1000	>170	>700
Our Record	***Champion***	***3rd place***	***Champion***	***Champion***	***Champion***	***Champion***

KDD-2012 Beijing,China August 12-16,2012
The 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
Sponsors and Supporters

Winner of the 2008
ACM SIGKDD
Knowledge Discovery and
Datamining Cup

National Taiwan University

Hung-Yi Lo, Chun Min Chang,
Tsung-Hsien Chiang, Cho-Yi Hsiao,
Anta Huang, Tsung-Ting Kuo,
Wei-Chi Lai, Ming-Han Yang,
Jung-Jung Yeh, Chun-Chao Yen,
Shou-De Lin

Sponsored by:

Siemens Medical Solutions, USA

ML Models are Evolving Fastly

- **New (and good) models come out every now and then**
  - People talked about SVM, BN before the rise of Deep learning
  - The STOA model today is likely to be infamous after 5 years

Are there some ideas about training ML models that can/shall last longer?

Whatever that worked from 2010~now will likely to still hold in 5~10 years

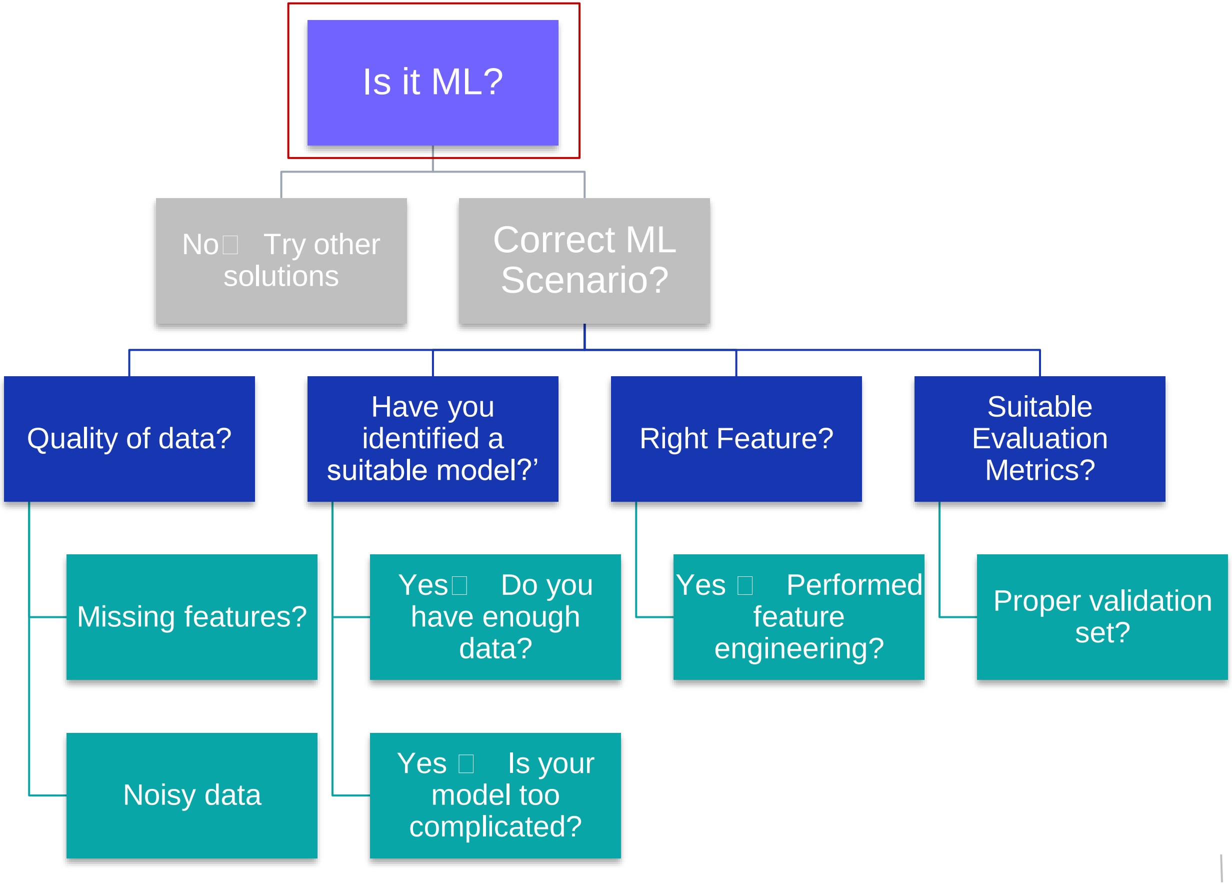
What is Machine Learning (ML)

- **ML tries to optimize a performance criterion using example data or past experience.**
- **Mathematically speaking: given some data X, we want to learn a function mapping f(X) for certain purpose**
  - f(x)= a label y classification or regression
  - f(x)= a set Y in X clustering
  - f(x)=p(x) distribution estimation
  - …
- **The ultimate goal is to obtain high quality f(x) given certain objective and evaluation metrics**

Why My Machine Learning Models Fail (meaning prediction accuracy is low)?

A series of analyses are required to understand why

The ML Diagnose Tree



Is it ML?

No□ Try other solutions

Correct ML Scenario?

Quality of data?

Have you identified a suitable model?’

Right Feature?

Suitable Evaluation Metrics?

Missing features?

Noisy data

Yes□ Do you have enough data?

Yes □ Is your model too complicated?

Yes □ Performed feature engineering?

Proper validation set?

Diagnose 1: Is it an ML task?

- Are you sure Machine Learning is the best solution for your task?

To ML or not to ML,
that is the question !!

Tasks Doubtful for ML

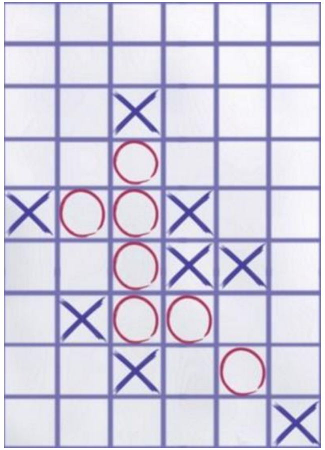
- X come from a close set with limited variation
  - simply memorize all possible X□ Y mappings
  - E.g. Word translation using dictionary
- F(x) can be easily derived by writing rules
  - E.g. compression/de-compression

Too easy!!

- X is (sort of) independent of Y
  - E.g. X□ <ID, name, wealth>, Y□ Height
- f(x) is not smooth, or f(x+△X)≠ y+ △y

Too hard!!

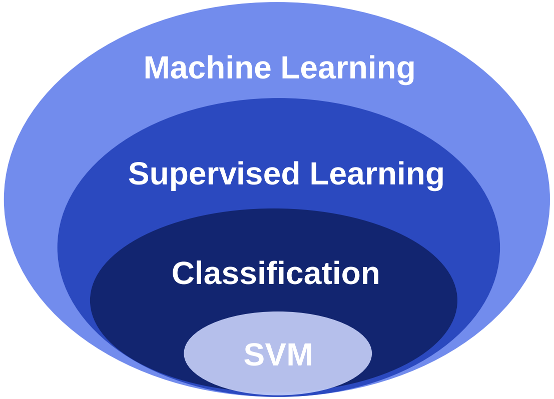
- Not enough data to be learned



OK, my task is a ML-solvable task, but I still failed

Diagnose 2: Which ML Scenario?

- Have you modeled your task into the right ML scenario?
  - ML ≠ Classification/Regression ≠ SVM, DNN, DT
- Which ML toolbox should you choose?



Machine Learning

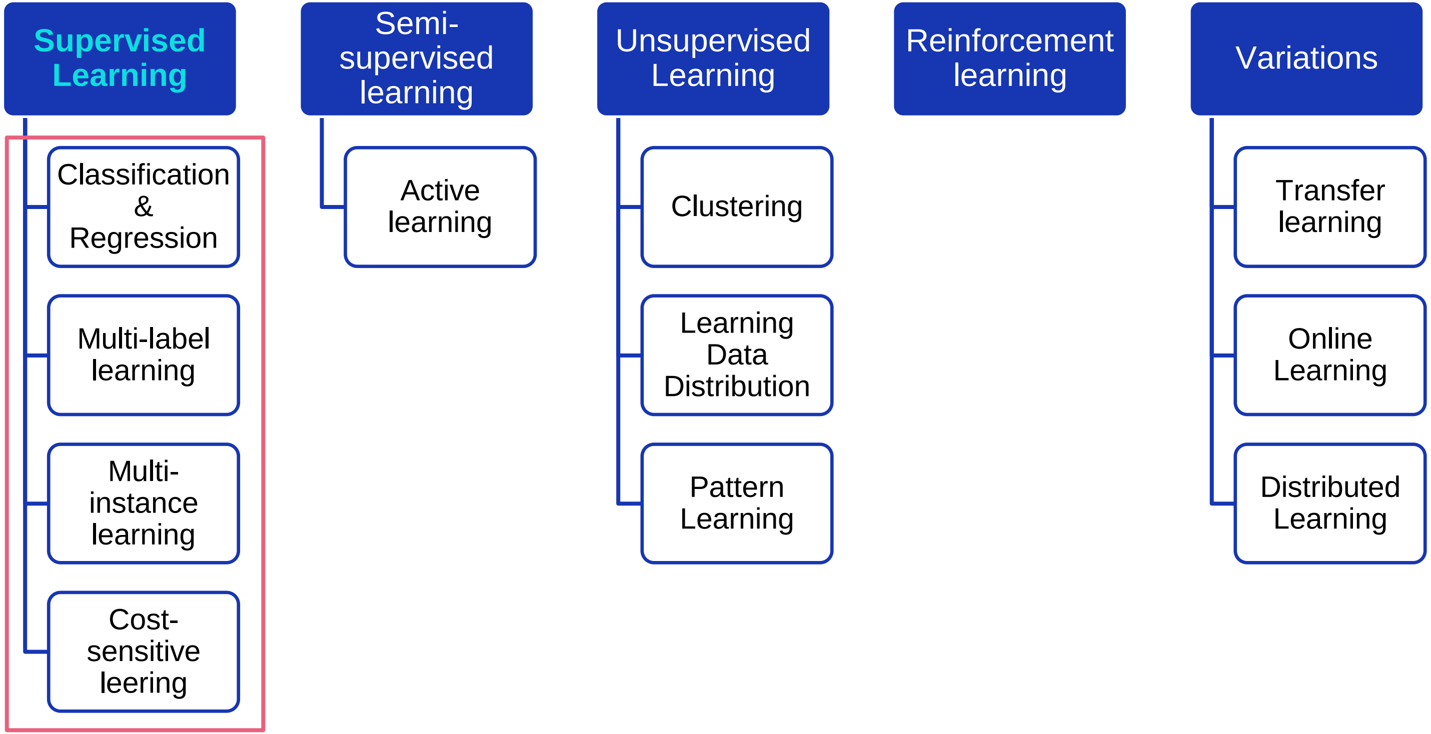
Supervised Learning

Classification

SVM

Let’s Talk About….. **Understanding Machine What Learning Can Do** in 10 Mins

A variety of ML Scenarios



Supervised Learning

- Given: a set of <input X, output Y> pairs
- Goal: given an unseen input, predict the corresponding output
- There are two kinds of outputs
  - Categorical***: classification problem***
    Binary classification vs. Multiclass classification
  - Real values***: regression problem***
- Example:

1. Binary classification:
   - input: **the sensor information**
   - output: **whether such sensor is broken**
2. Multi-class classification:
   - Input: **Lyric of a song**,
   - output: **happy/sad/surprise/angry**
3. Regression:
   - Input: **the weather/traffic/air condition**,
   - output: **PM 2.5 value**

Multi-label Learning

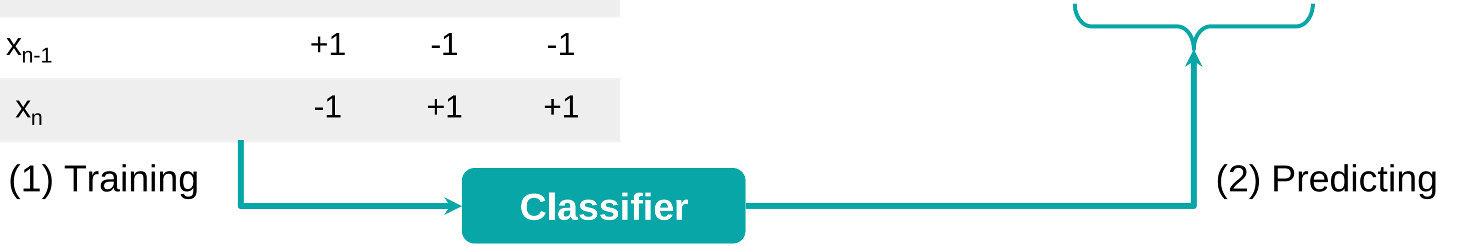
- A classification task in that an instance is associated with a set of labels, instead of a single label.

Training set

Feature Vector ($x_i \in R^d$)	ℓ_1	ℓ_2	ℓ_3
x_1	+1	-1	+1
x_2	-1	+1	-1
…	…	…	…
x_{n-1}	+1	-1	-1
x_n	-1	+1	+1

A new instance

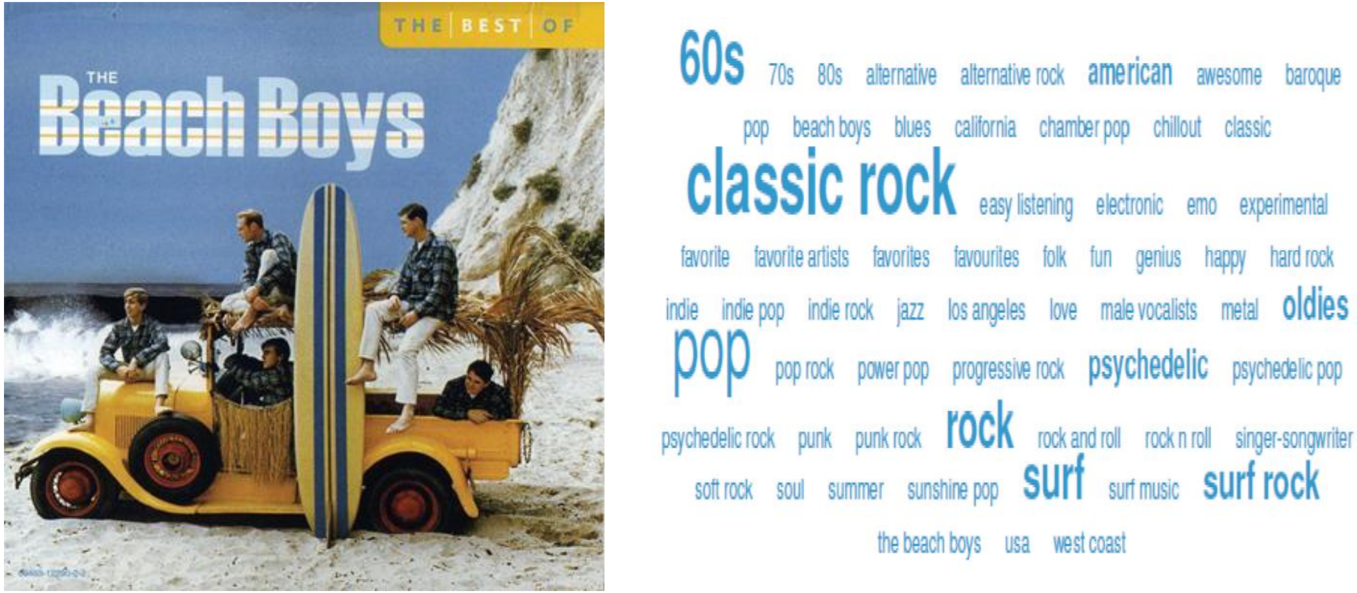
Feature Vector ($x_{new} \in R^d$)	ℓ_1	ℓ_2	ℓ_3
x_{new}	?	?	?



- Existing models: Binary Relevance, Label Powerset, ML-KNN, IBLR, …

Multimedia tagging

- Many websites allow the user community to add tags, thus enabling easier retrieval.

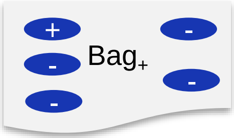

THE BEST OF
THE Beach Boys
60s 70s 80s alternative alternative rock american awesome baroque
pop beach boys blues california chamber pop chillout classic
classic rock easy listening electronic emo experimental
favorite favorite artists favorites favourites folk fun genius happy hard rock
indie indie pop indie rock jazz los angeles love male vocalists metal oldies
pop pop rock power pop progressive rock psychedelic psychedelic pop
psychedelic rock punk punk rock rock rock and roll rock n roll singer-songwriter
soft rock soul summer sunshine pop surf surf music surf rock
the beach boys usa west coast

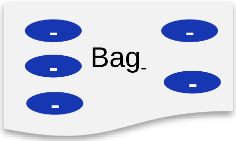
Example of a tag cloud: the beach boys, from Last.FM (Ma et al., 2010)

Multi-instance Learning

- A supervised learning task in that the training set consists of ***bags of instances***, and instead of associating labels on instances, ***labels*** are *only* assigned to ***bags***.

- In the binary case,
  - Positive bag □ at least one instance in the bag is positive
  - Negative bag □ all instances in the bag are negative





- The goal is to learn a model and predict the label of a new bag of instances.

Cost-sensitive Learning

- A supervised learning task in that the training set consists of ***bags of instances***, and instead of associating labels on instances, ***labels*** are *only* assigned to ***bags***.

- An example cost matrix L: medical diagnosis

L_{jk}	**Actual Cancer**	**Actual Normal**
Predict Cancer	0	1
Predict Normal	10000	0

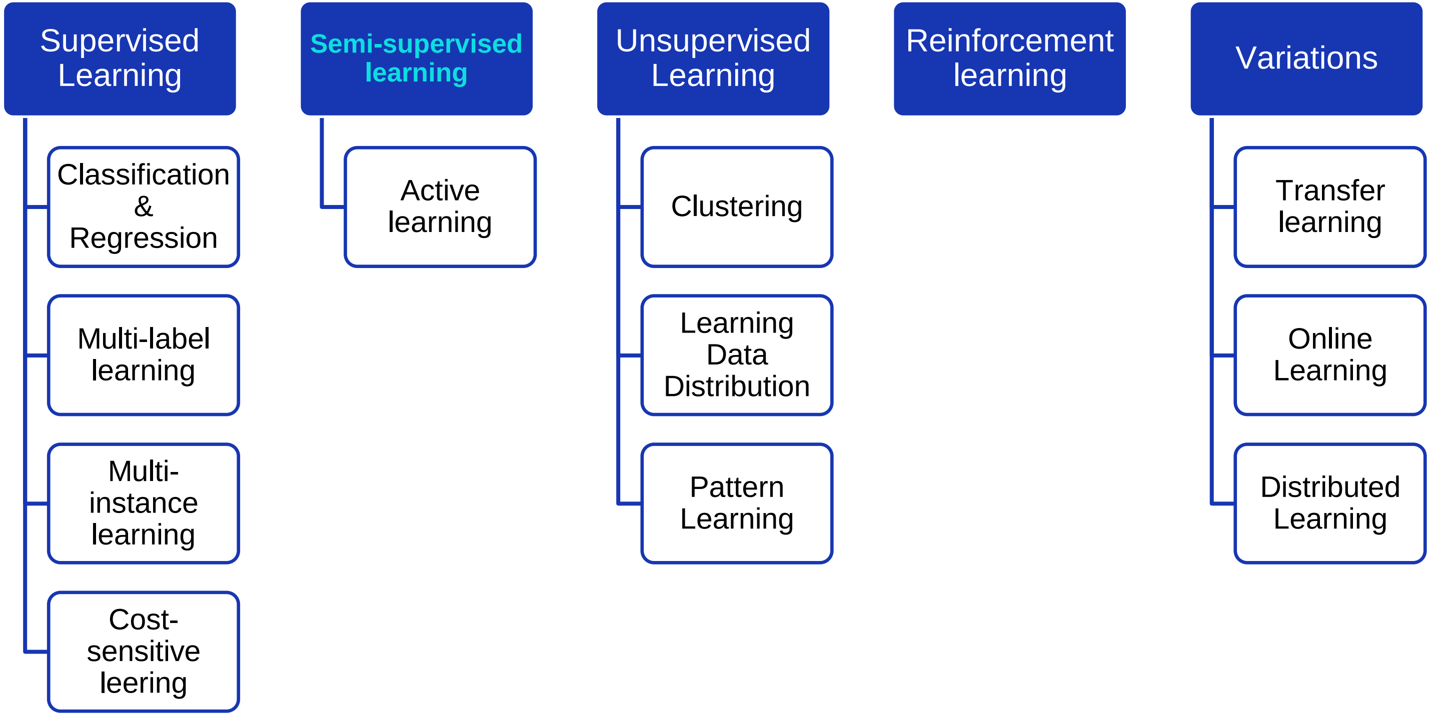
- Exemplified solution: cost-sensitive SVM, cost-sensitive sampling

Examples for Cost-sensitive Learning

- Highly non-uniform misclassification costs are very common in a variety of challenging real-world machine learning problems
  - Fraud detection
  - Medical diagnosis
  - Various problems in business decision-making**.**

Credit cards are one of the most famous targets of fraud. The cost of missing a target (fraud) is much higher than that of a false-positive.

A variety of ML Scenarios



Supervised Learning

- Classification & Regression
- Multi-label learning
- Multi-instance learning
- Cost-sensitive leering

Semi-supervised learning

- Active learning

Unsupervised Learning

- Clustering
- Learning Data Distribution
- Pattern Learning

Reinforcement learning

Variations

- Transfer learning
- Online Learning
- Distributed Learning

Semi-supervised Learning (1/2)

- We have a large amount of data, but only a small portion of them are annotated (usually due to high annotation cost)
- Very common scenario in practice

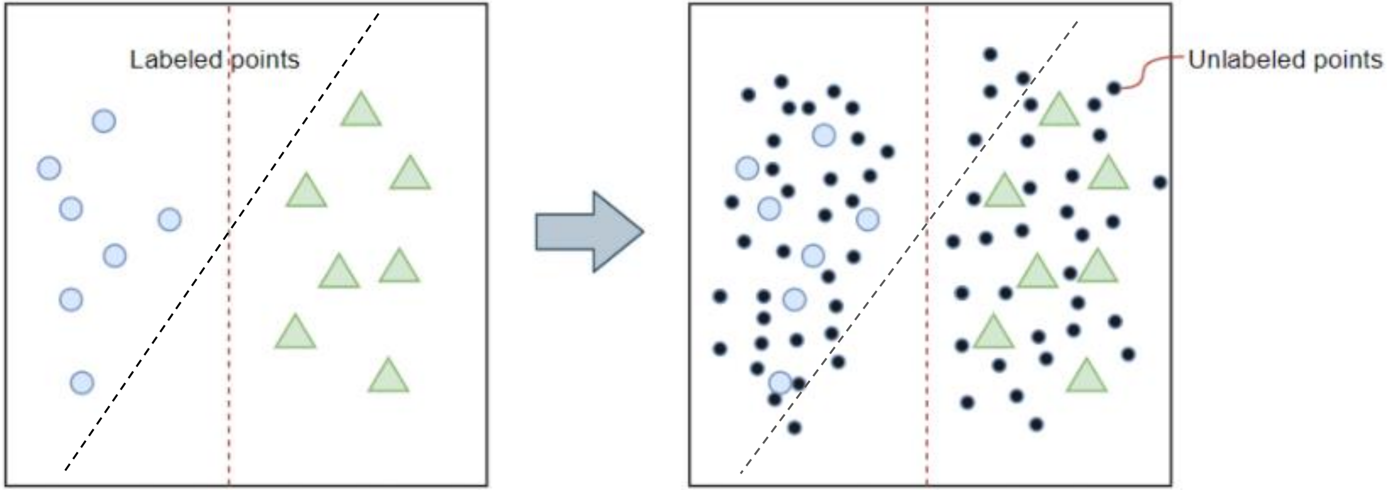
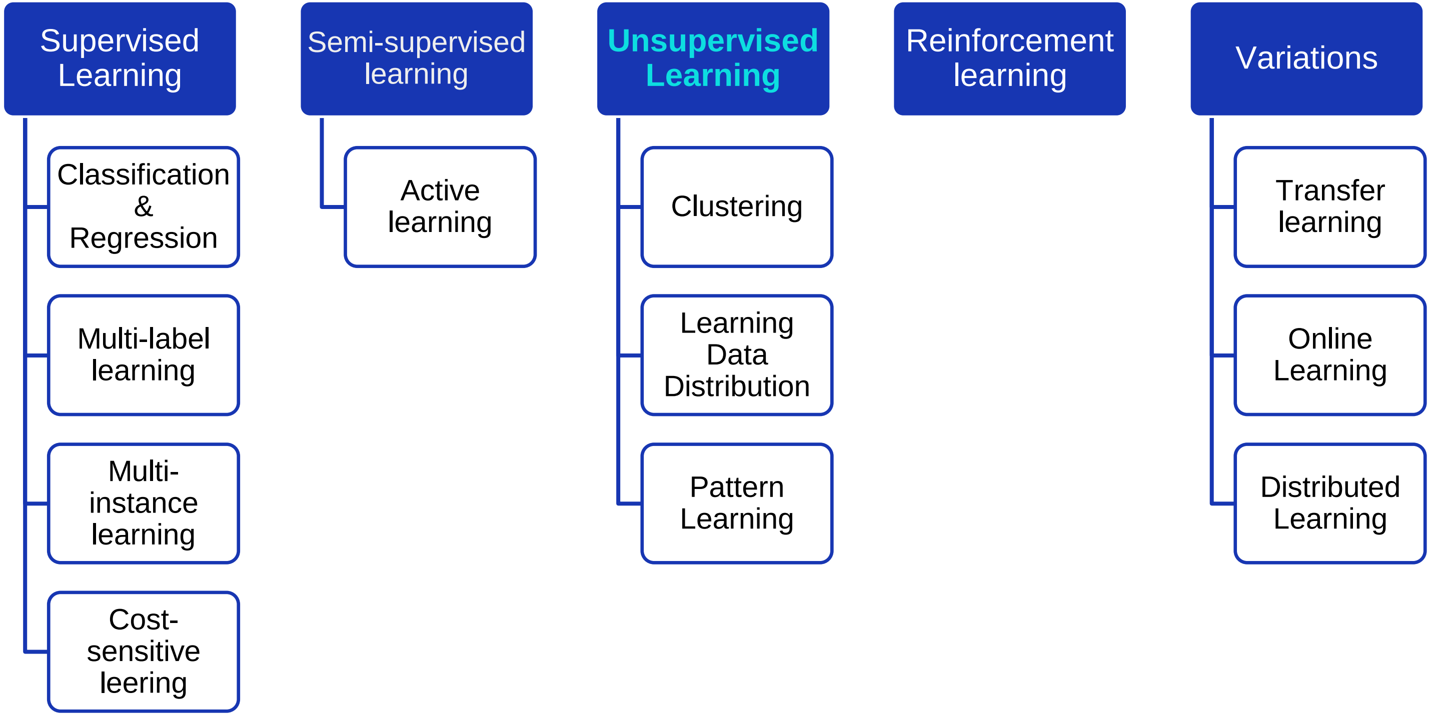


Image from
https://www.ecloudvalley.com/mlintroduction/

A variety of ML Scenarios



Supervised Learning
- Classification & Regression
- Multi-label learning
- Multi-instance learning
- Cost-sensitive leering

Semi-supervised learning
- Active learning

Unsupervised Learning
- Clustering
- Learning Data Distribution
- Pattern Learning

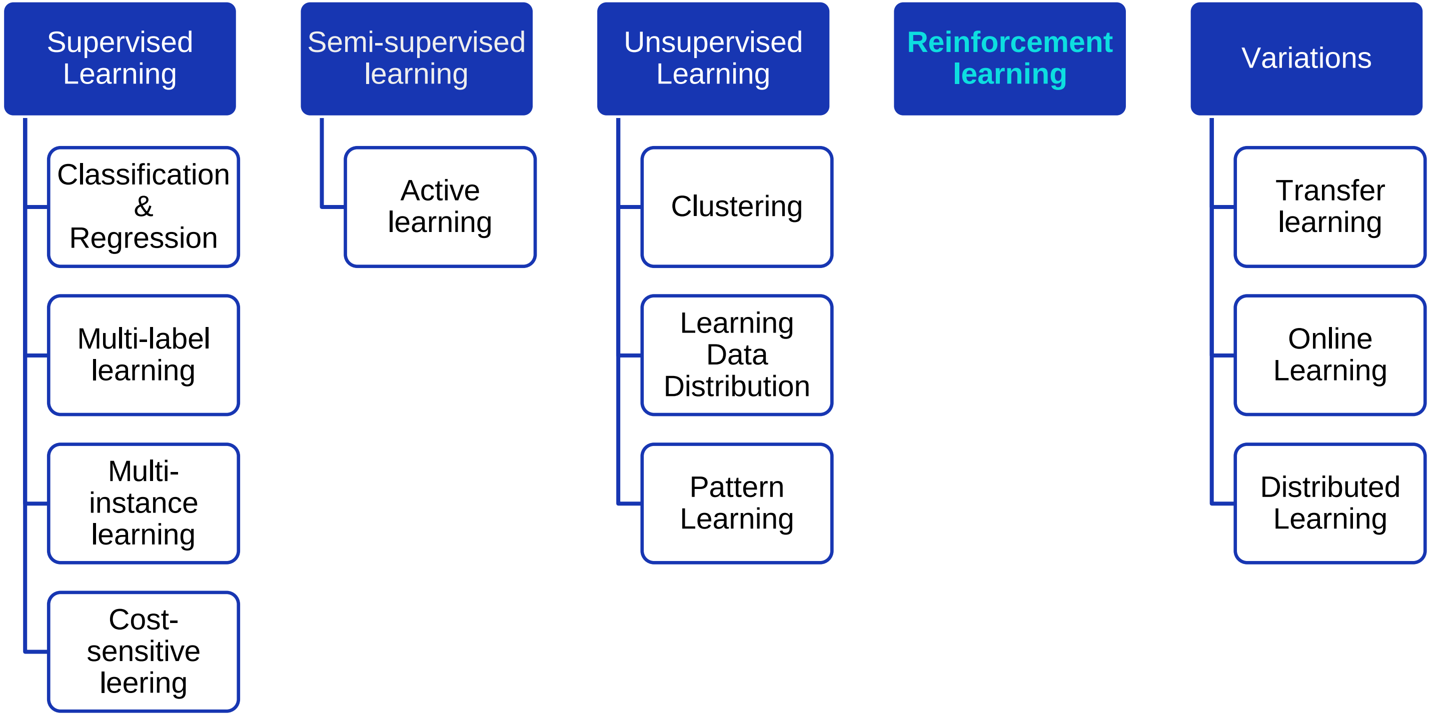
Reinforcement learning

Variations
- Transfer learning
- Online Learning
- Distributed Learning

Unsupervised Learning

- Learning without teachers (presumably harder than supervised learning)
  - Learning “what normally happens”
  - Think of how babies learn their first language (unsupervised) comparing with how people learn their 2nd language (supervised).
- Given: a bunch of input X (there is no output Y)
- Goal: depending on the tasks, e.g.
  - Estimate P(X) □ then we can find augmax P(X) □ Bayesian
  - Finding $P(X_2|X_1)$□ we can know the dependency between inputs □ Association Rule, causality model
  - Finding $Sim(X_1,X_2)$ □ then we can group similar X’s □ clustering

A variety of ML Scenarios



Supervised Learning

- Classification & Regression
- Multi-label learning
- Multi-instance learning
- Cost-sensitive leering

Semi-supervised learning

- Active learning

Unsupervised Learning

- Clustering
- Learning Data Distribution
- Pattern Learning

Reinforcement learning

Variations

- Transfer learning
- Online Learning
- Distributed Learning

Reinforcement Learning (RL)

- RL is for "consecutive decision making"
  - How an agent should make a series of decisions tc maximize the long-term rewards
- RL is associated with **a sequence** of **states X and actions Y** (i.e. think about Markov Decision Process) with certain "rewards".
- It's goal is to find an optimal policy to guide the decision.

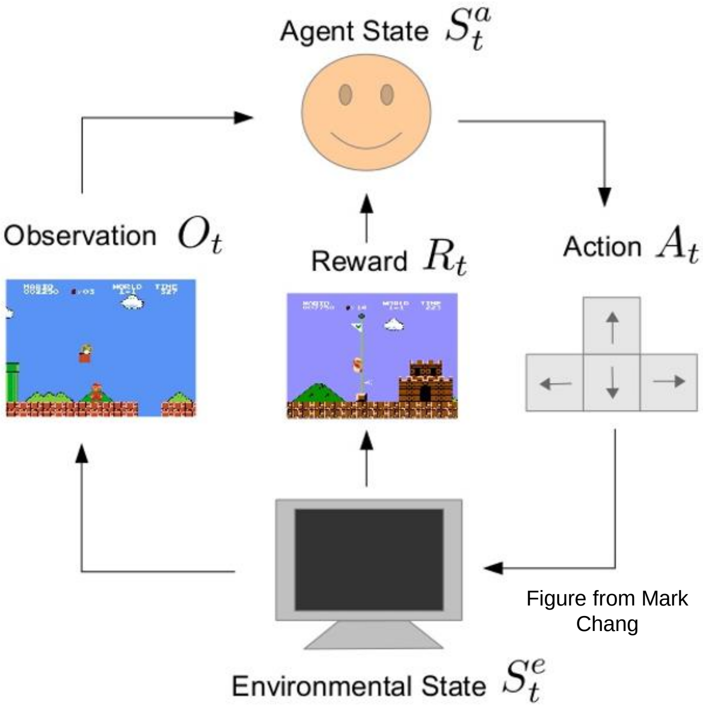


Figure from Mark Chang

Image from Nintendo Mario

AlphaGo: SL+RL

- 1st Stage: Multi-class classification
  - Data: previous moves from experts
  - Learning: f(X)=Y, Y=next move
  - Results: AI can outperform normal players, but not the best ones
- 2nd Stage: Reinforcement Learning
  - Data: generating from playing with 1st Stage AI
  - Learning: reward□ if win, action□ next move

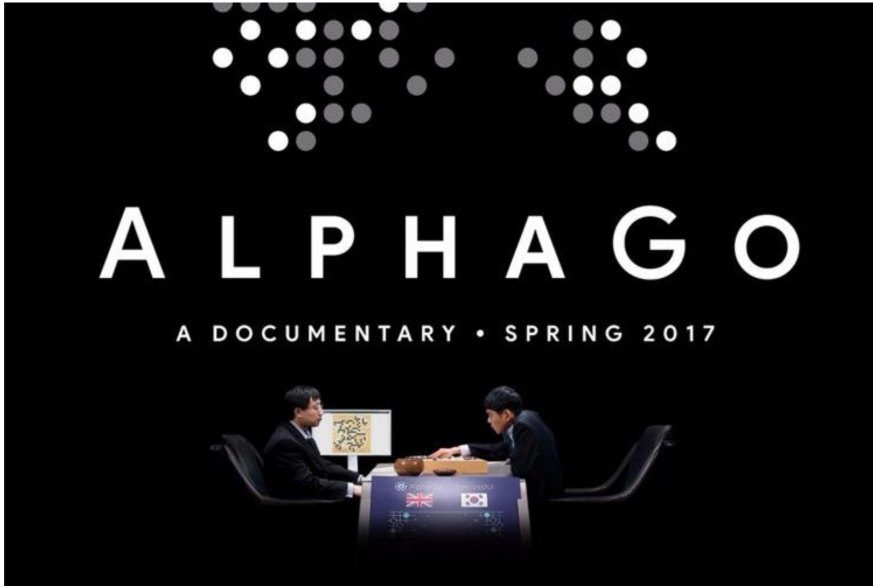
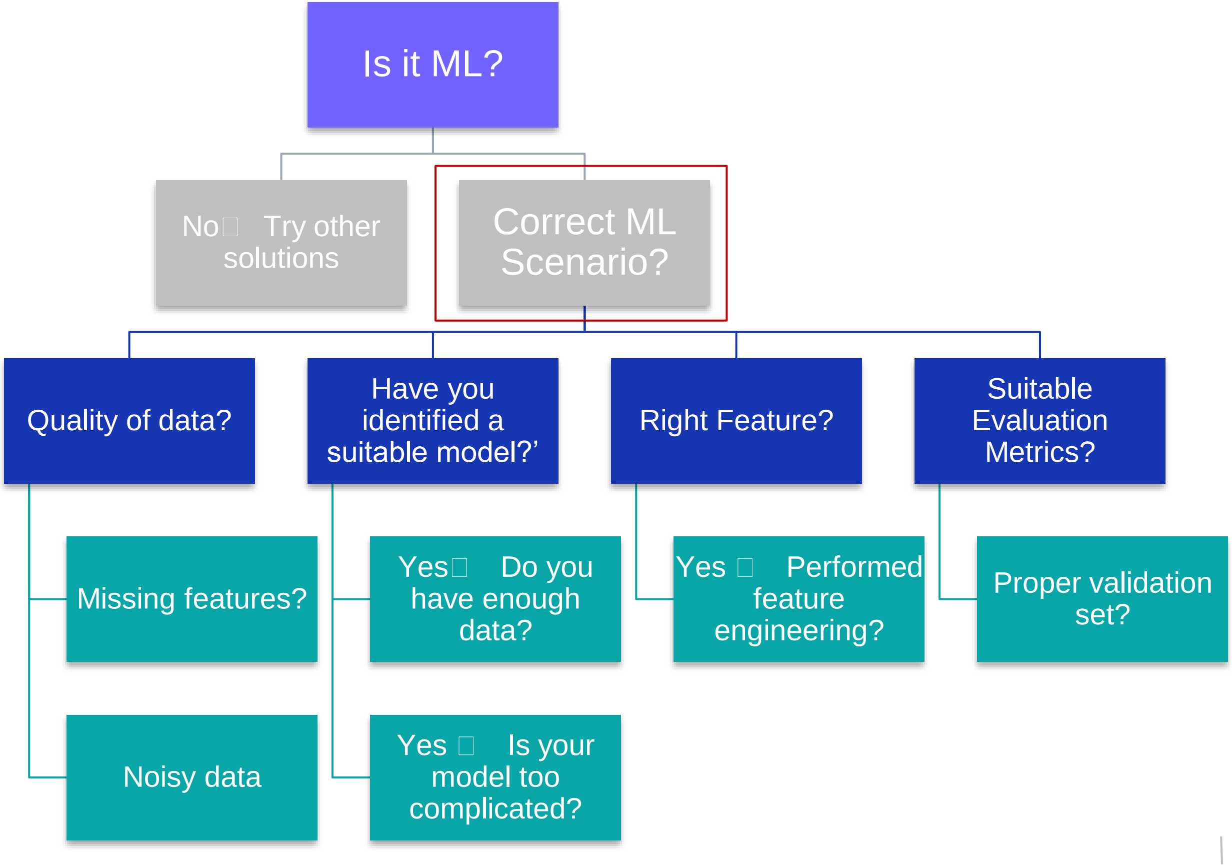


Image from
https://twitter.com/alphagomovie

The ML Diagnose Tree


Is it ML?
No□ Try other solutions
Correct ML Scenario?
Quality of data?
Have you identified a suitable model?’
Right Feature?
Suitable Evaluation Metrics?
Missing features?
Noisy data
Yes□ Do you have enough data?
Yes □ Is your model too complicated?
Yes □ Performed feature engineering?
Proper validation set?

Diagnose 2: Which ML Scenario?

- Have you modeled your task into the most suitable ML scenario?

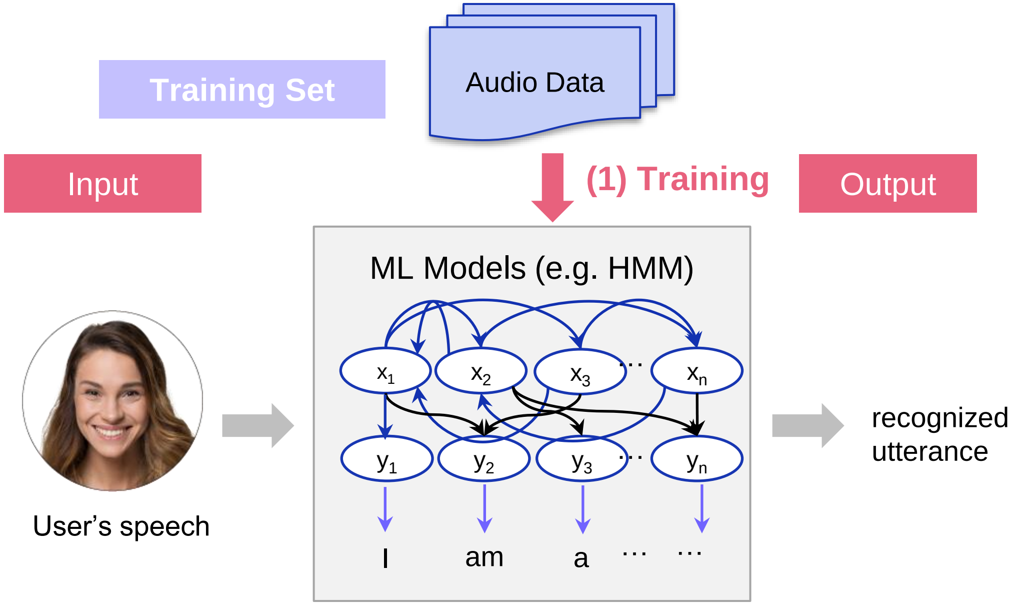
toolbox 1: multi-label learning

toolbox 2: clustering

toolbox 3: reinforcement learning

Case Study 1: Sequence Labeling Tasks (Speech recognition, OCR, Tagging)

- This task can be modeled as
  - Supervised Learning task
    - Multi-class classification problem
    - Sequential labeling problem (e.g. CRF, HMM, RNN)
  - Unsupervised Learning task (EM)



Training Set

Audio Data

Input

(1) Training

Output

ML Models (e.g. HMM)

User's speech

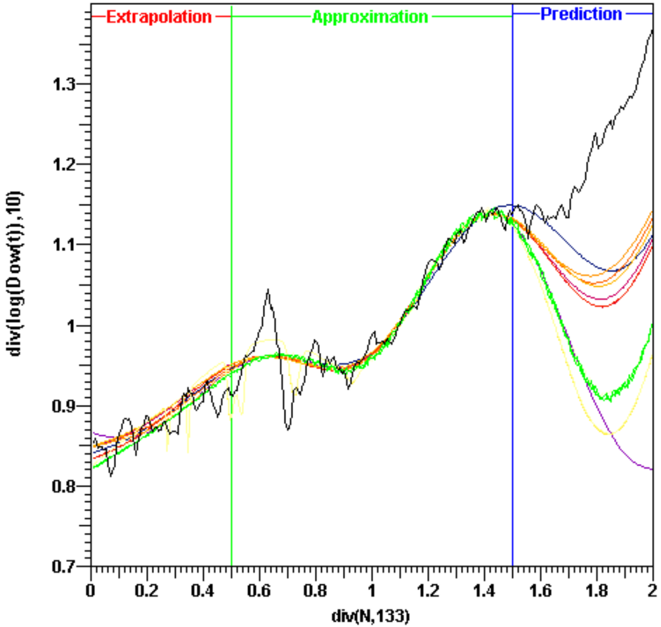
recognized utterance

Case Study 2: Click Through Rate (CTR) Prediction

- CTR: for an advertisement displayed to users, what is the ratio that the users click into it

- It looks like a regression task, but is it?
  - CTR= #click/#view □ User1: 5/10 vs. User2: 500/1000
  - If eventually we care about ‘user-level CTR’ accuracy, then User1 and User2 shall be treated equally during training ⇒ regression
  - If eventually we care about ‘click-level CTR’ accuracy, then we shall transfer 5(00)/10(00) to 5(00) positive cases and 5(00) negative cases ⇒ binary classification

Case Study 3: Temporal Value Prediction



http://alphard.ethz.ch/

- It can be modeled as
  - a regression problem (concatenate all sequences into one)
  - an online learning problem (i.e. data come incrementally to update a model)
  - a multi-task learning problem.

Multi-task Learning for Temporal Prediction

Figure 2: Proximal Constraints on Features

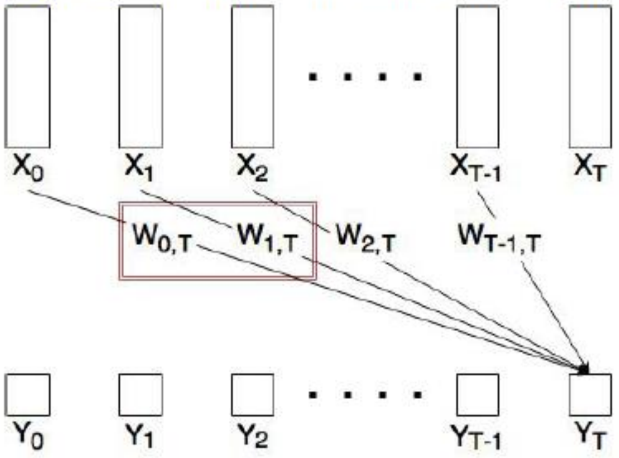
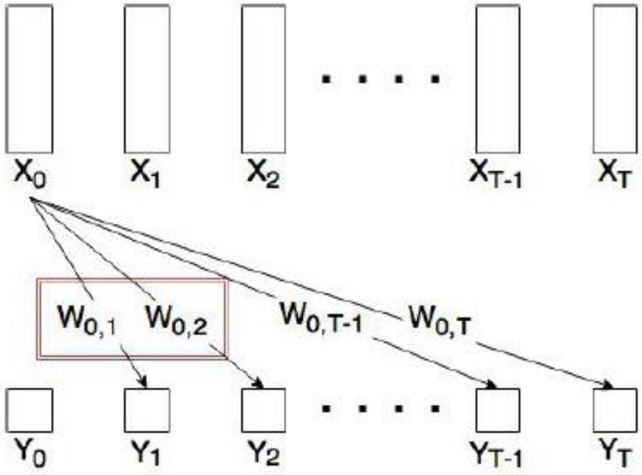
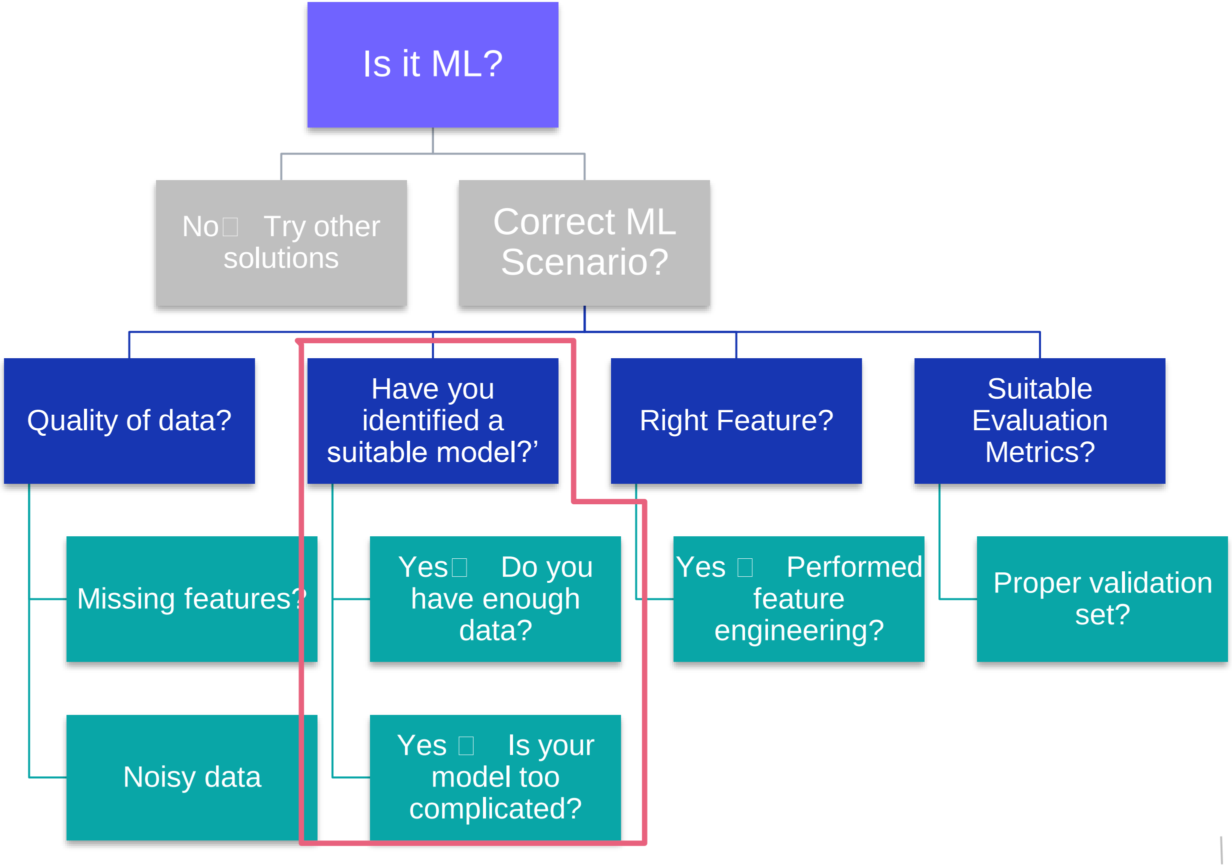


Figure 1: Proximal Constraints on Target Scores



OK, I have identified an ideal ML scenario, but my model still doesn’t work

The ML Diagnose Tree



Is it ML?

No□ Try other solutions

Correct ML Scenario?

Quality of data?

Missing features?

Noisy data

Have you identified a suitable model?’

Yes□ Do you have enough data?

Yes □ Is your model too complicated?

Right Feature?

Yes □ Performed feature engineering?

Suitable Evaluation Metrics?

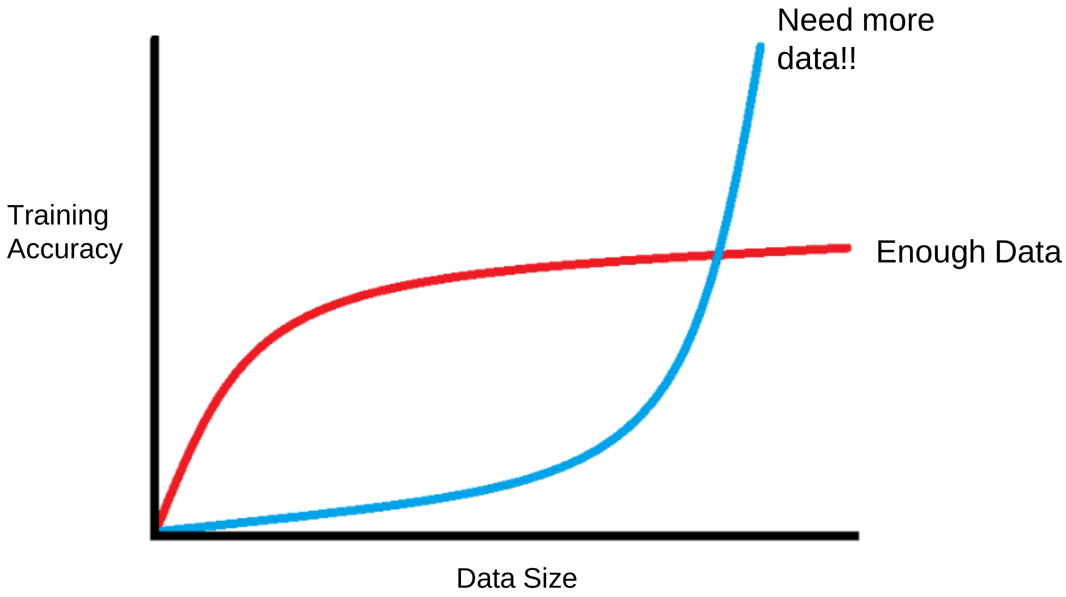
Proper validation set?

Diagnose 3: Did you choose a proper model?

- A proper model considers
  - **The size of data**
    - Small data □ linear (or simpler) model
    - Large data □ linear model or non-linear model
  - **The sparsity of data**
    - Sparse data □ more tricks to perform better and faster
    - Dense data □ requires light algorithm that consumes less memory
  - **The balance condition of data**
    - Imbalanced data □ special treatment for minority class
  - **The quality of data (whether there are noise, missing values, etc)**
    - Some loss function (e.g. 0/1 loss or L2) are more robust to noise than others (e.g. hinge loss or exponential loss)

Diagnose 4: Do you have enough data to train the given model?

- Draw the **learning curve** to understand whether your data is sufficient



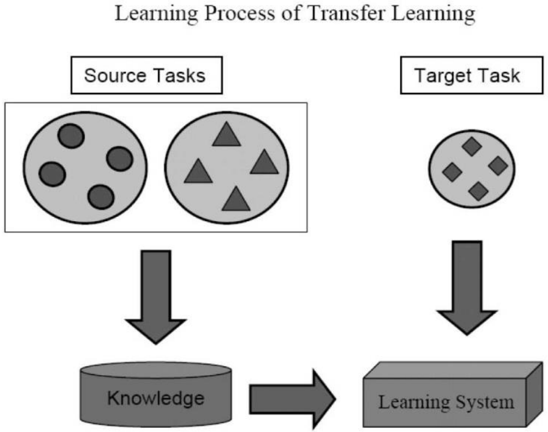
What shall I do if I have a lot of Data, but they are not labelled ?

- This is by far the most common question I have been asked.
- My honest answer: try to get them labelled because you anyway need ground truth for evaluation.
- In several cases, labelling are too costly
  - Semi-supervised solution
  - Transfer learning (using labelled data in other domains)

Transfer Learning (or domain adaptation)

- Improving a learning task via incorporating knowledge from learning tasks in other domains with different feature space and data distribution.

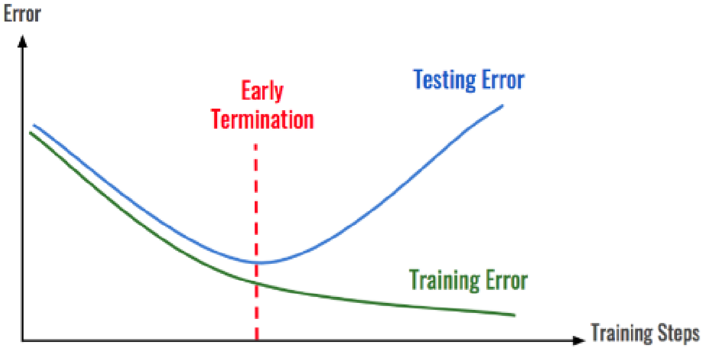
- Example 1: the knowledge for recognizing an airplane may be helpful for recognizing a bird.
- Example 2: I need to build a recommender system for company R1, but I have only obtained rating data from company R2



[Pan & Yang 2010]

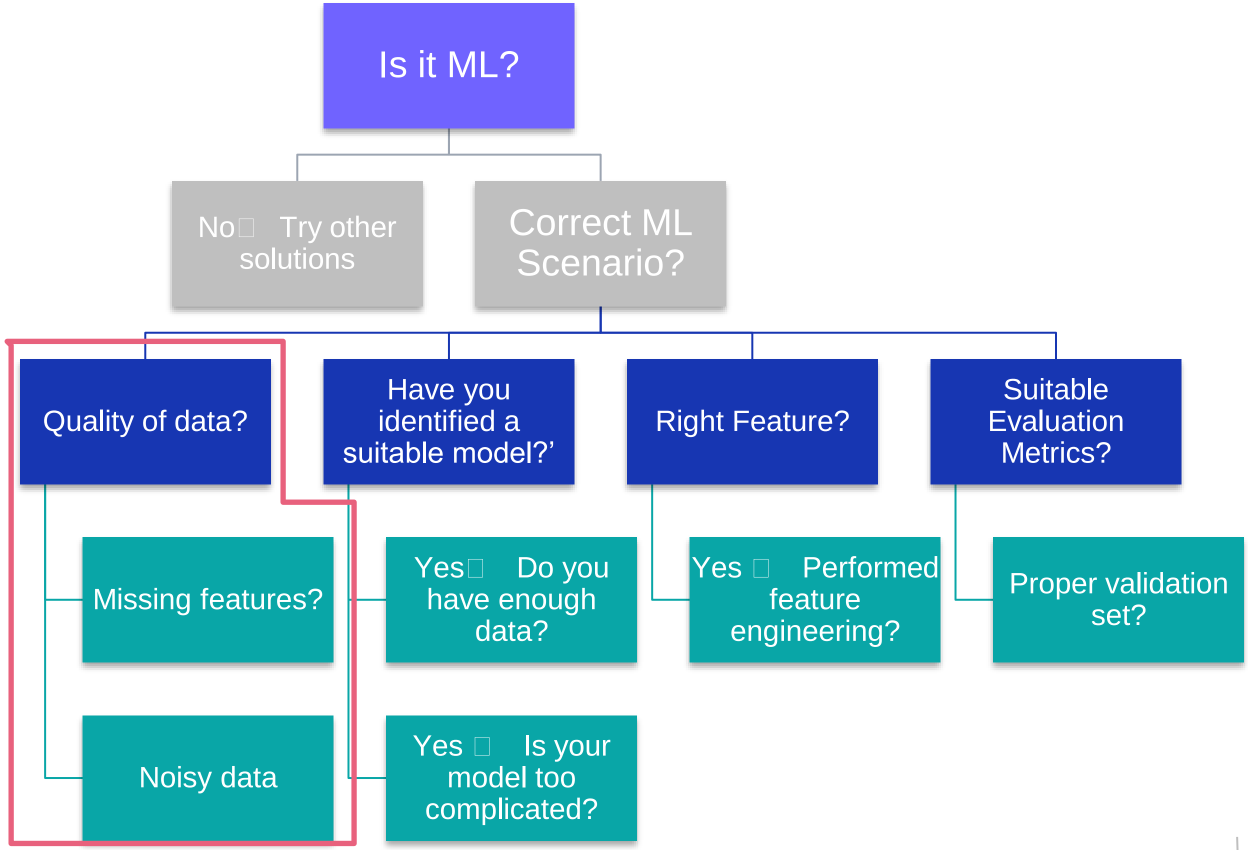
Diagnose 5: Is the model too complicated □ overfitting

- How to avoid overfitting?
  - **Occam's Razor: simpler model first**
    - Always starting from simple models as the benchmarks
    - Always record training/validation/testing accuracy
  - **Regularization: a way to constraint the complexity of the model**
  - **Early stopping in training**
  - **Train with more *high-quality* data**
  - **Remove features (i.e. feature selection)**



by Ananda Mohon Ghosh

The ML Diagnose Tree



Is it ML?

No□ Try other solutions

Correct ML Scenario?

Quality of data?

Missing features?

Noisy data

Have you identified a suitable model?’

Yes□ Do you have enough data?

Yes □ Is your model too complicated?

Right Feature?

Yes □ Performed feature engineering?

Suitable Evaluation Metrics?

Proper validation set?

Diagnose 6: Are the data very noisy?

- A typical ML process:
  - **Data labeling** □ Model Training (parameter tuning and model selection) + validation □ Final Model Shipping
  - Human are usually involved in the earlier stage for **data labelling**
- ML process for noisy data:
  - **Data cleaning** □ **data labelling** □ model training □ **label refinement** □ Final model shipping
  - Human can be involved in 3 stages

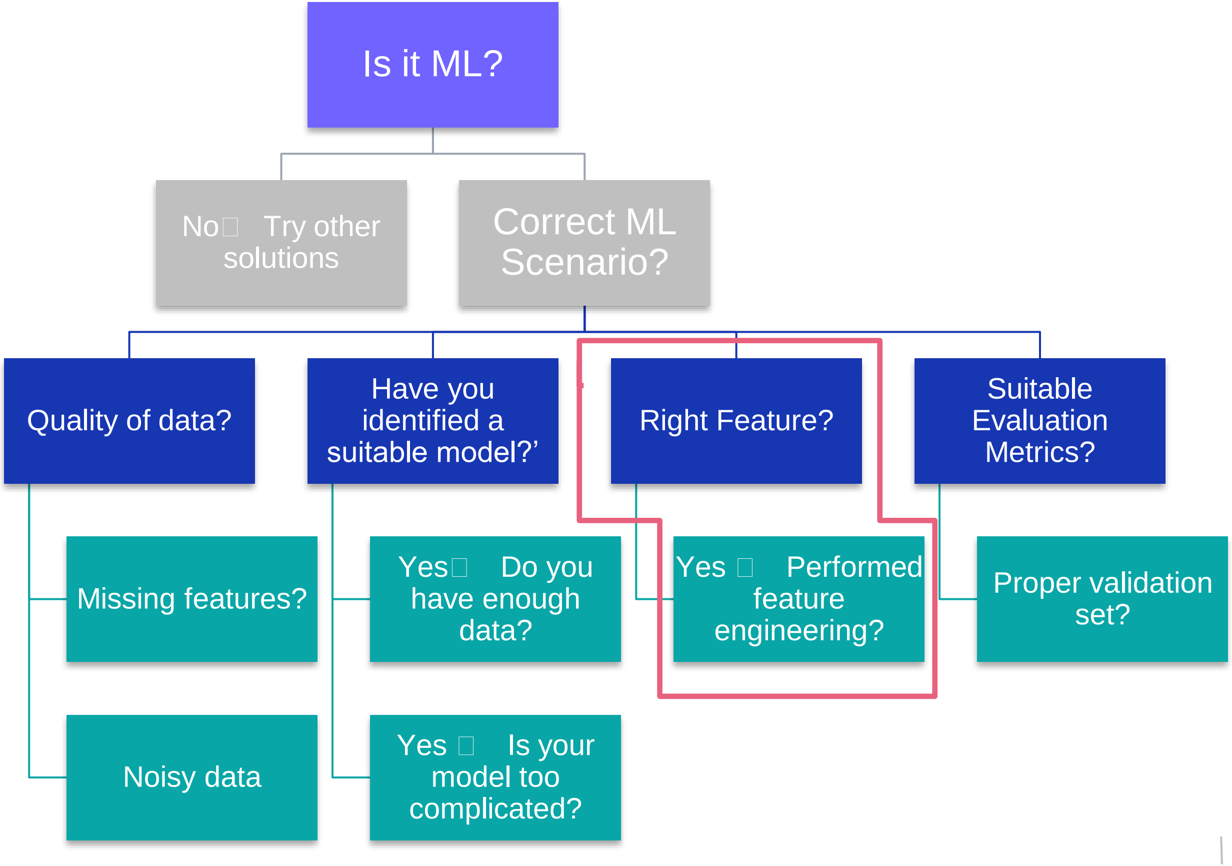
Diagnose 7: Do the data contain many missing features?

- Simply filling zero or mean for the missing features might not be optimal
- Solution 1: fill in the missing values and then perform learning
    - Be aware of different missing scenario (MCAR, MNAR, MAR)
    - Popular solutions: MICE, GAIN, MisGAIN
- Solution 2: Perform imputation and learning at the same time □ HexaGAN, GRAPE

Missing Data Imputation using Generative Adversarial Nets (ICML18)
MisGAN: Learning from Incomplete Data with Generative Adversarial Networks.(ICLR19)
Handling Missing Data with Graph Representation Learning (NeurIPS 2020)

I have checked the data and avoided overfitting, but my model still performs poorly

The ML Diagnose Tree



Is it ML?

No□ Try other solutions

Correct ML Scenario?

Quality of data?

Missing features?

Noisy data

Have you identified a suitable model?’

Yes□ Do you have enough data?

Yes □ Is your model too complicated?

Right Feature?

Yes □ Performed feature engineering?

Suitable Evaluation Metrics?

Proper validation set?

Maybe the key features haven’t been identified

- Use domain knowledge and human judgement to determine which features to obtain for training.
- The rule of thumb: If you don’t know, then you shall try because Human judgement can be misleading
  - Condition: given the data is sufficient
- If the amount of training data is limited, it is better to trust human judgement

Feature Engineering

- Feature engineering turns out to be one of the best (if not the best) strategy to improve the performance.
- The goal is to explicitly reveal important information to the model
  - domain knowledge might or might not be useful
- Original features □ different encoding of the features □ combined features

Dealing with Different Types of Features

- Categorical: need to encode to numerical ones
- Numerical: scaling (e.g. normalization to N(0,1), log (1+x), linear scaling, etc.)

Encoding Categorical Features

- Nominal features
  - Encoding without label information
    - One hot encoding (expanding to binary code for each feature values)
    - Frequency encoding (each feature value is replaced by its appearing frequency)
  - Encoding using label information (might cause overfitting)
    - Mean encoding (ratio of being positive for each feature value)
    - Probability Ratio Encoding (using P(1)/P(0))
- Ordinal features
  - Ordinal encoding (e.g. easy□ 1, medium □ 2, hard □ 3)

Existing Libraries: Python’s category_encoding library, Scikit-learn preprocessing, Pandas' get_dummies, etc

Embeddings Encoding for Categorical Features

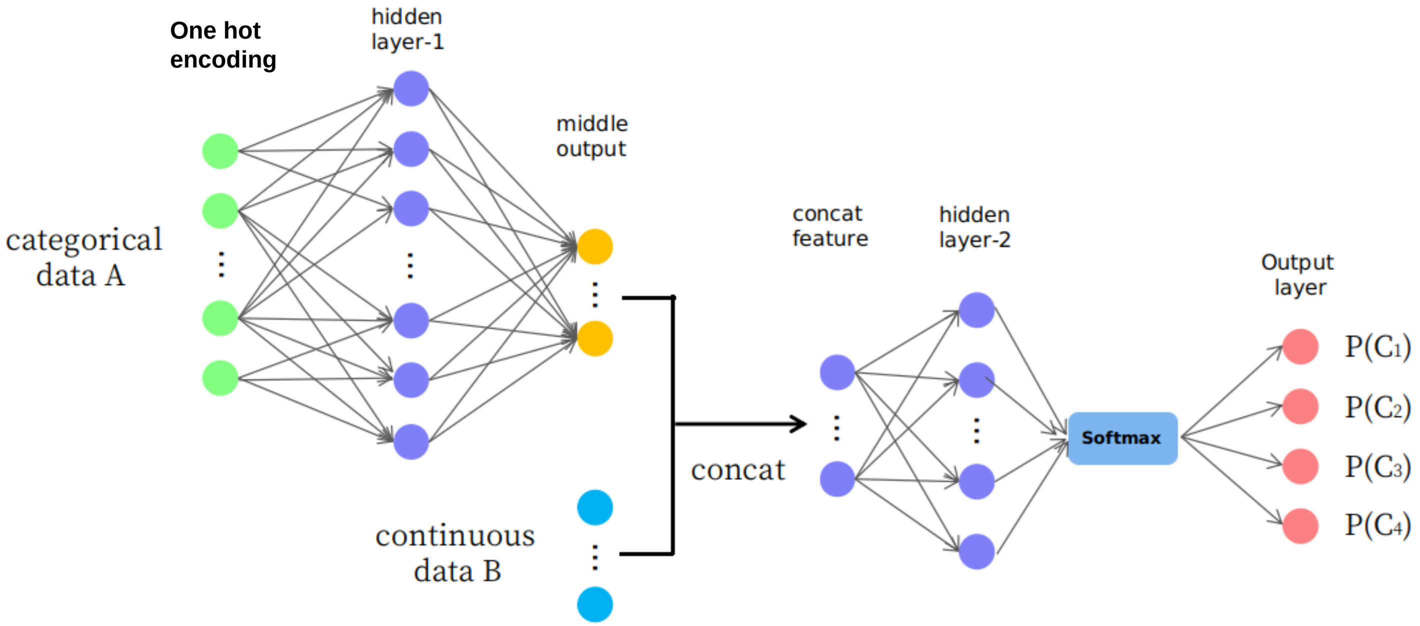
- Categorical features with large set of values can be tricky (one hot encoding□ large sparse matrix)
- Maybe we can encode each feature value using a dense (instead of sparse) vector

....

V_k^T					
User1	1	0	0	.....	0
User2	0	0	1	.....	0

Neural Network Embeddings

- End-to-end training framework



By Junjie Chen

Numerical Scaling

1. Normalization (x-min/max-min)
   - when the distribution is far from Gaussion
   - Benefits NN-based model because it makes training faster and prevents local optima
2. Standardization (x-**μ**/s):
   - when the data follows Gaussian
   - More robust to outliers
3. Logarithmic such as log(x+1): large data values that can lead to very small weights

Feature Engineering: combining features

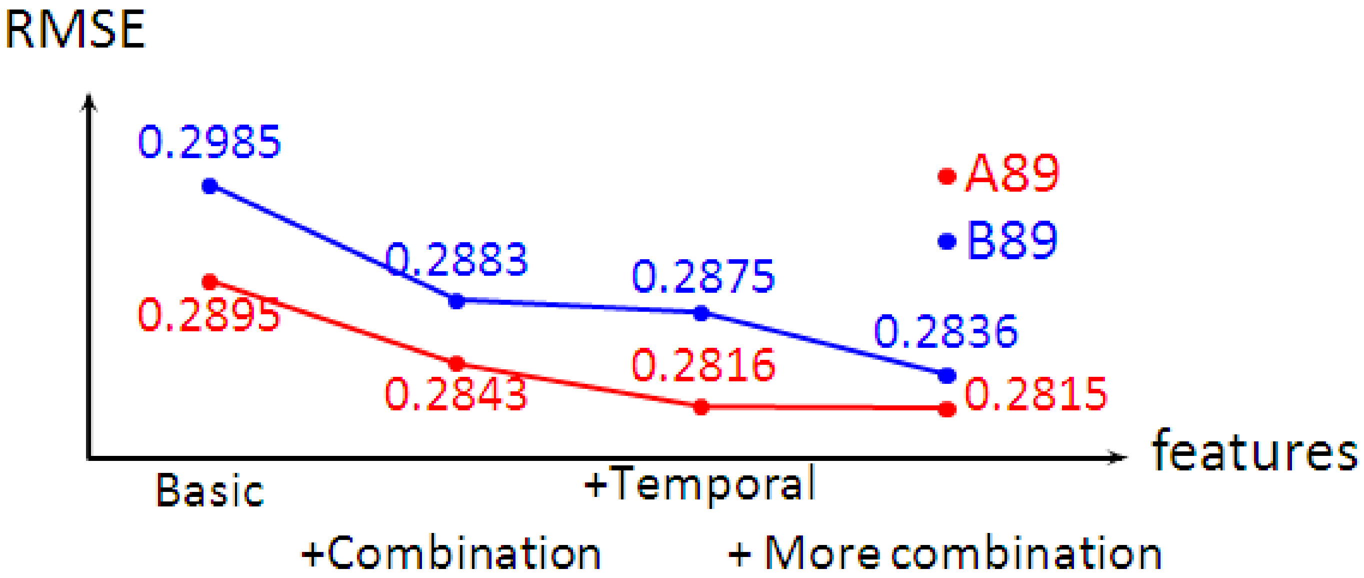
- Reason 1: we want to explicitly tell the model these combinations are useful
    - A way to inject human knowledge into models (e.g. how to use hierarchical information)
    - e.g. feature multiplication/division
- Reason 2: it allows a linear classifier to exploit non-linear dependency of features
    - Polynomial mapping (e.g. bigram/trigram features)
- Feature combination usually leads to large set of expansion on feature size
    - using linear model to evaluate its performance first

Features from Near-by Instances are sometimes helpful

- One can combine features from similar instances to build a richer model
- What are considered as similar instances?
  - kNN in terms of features
  - Instances close in time
  - Instances close in space

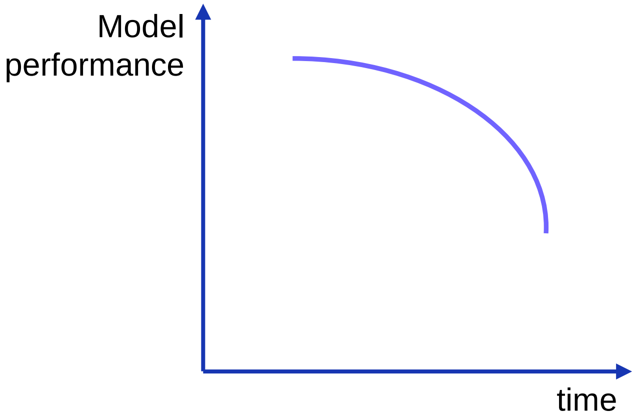
Results in KDD Cup 2010

The error rate goes down whenever a new set of features come into play !!



Check for Concept Drift

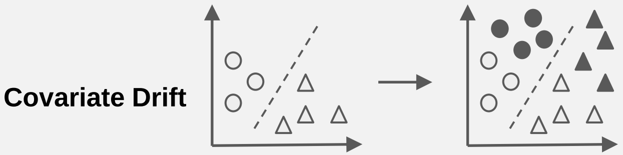
- What is concept drift?
- Over time, the context of the data or the relationship between features and labels has changed □ usually lead to performance degradation



A Deeper Look into What Causes Degradation: Different Types of Drift

- $P(X, y) = P(X) \times P(y|X)$ □ distribution of features and lables [1,2]

- <u>Covariate Drift (e.g. user distributions vary across time)</u>
  - Distribution of feature space changes while decision boundary remains.



$$P_{t0}(X) \neq P_{t1}(X), P_{t0}(y|X) = P_{t1}(y|X)$$

Figure adapted from [1]

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

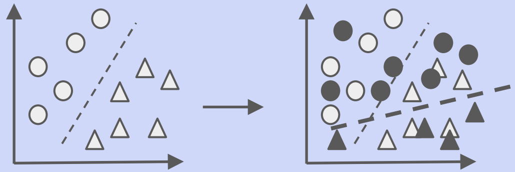
[2] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46, 4, Article 44 (April 2014), 37 pages. DOI:https://doi.org/10.1145/2523813

A Deeper Look into What Causes Degradation: Different Types of Drift

- $P(X, y) = P(X) \times P(y|X)$ □ distribution of features and lables [1,2]

- Actual Drift (e.g Users interests change over time)
    - Distribution of feature space remains while decision boundary changes.

Actual Drift



$$P_{t0}(X) = P_{t1}(X), P_{t0}(y|X) \neq P_{t1}(y|X)$$

Figure adapted from [1]

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

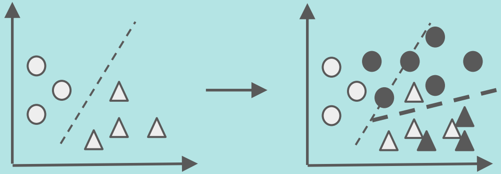
[2] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46, 4, Article 44 (April 2014), 37 pages. DOI:https://doi.org/10.1145/2523813

A Deeper Look into What Causes Degradation: Different Types of Drift

- $P(X,y) = P(X) \times P(y|X)$ □ distribution of features and lables [1,2]

- <u>Concept Drift (user distribution changes, and so are users’ interests)</u>
  - Both feature space distribution and decision boundary change.

Concept Drift



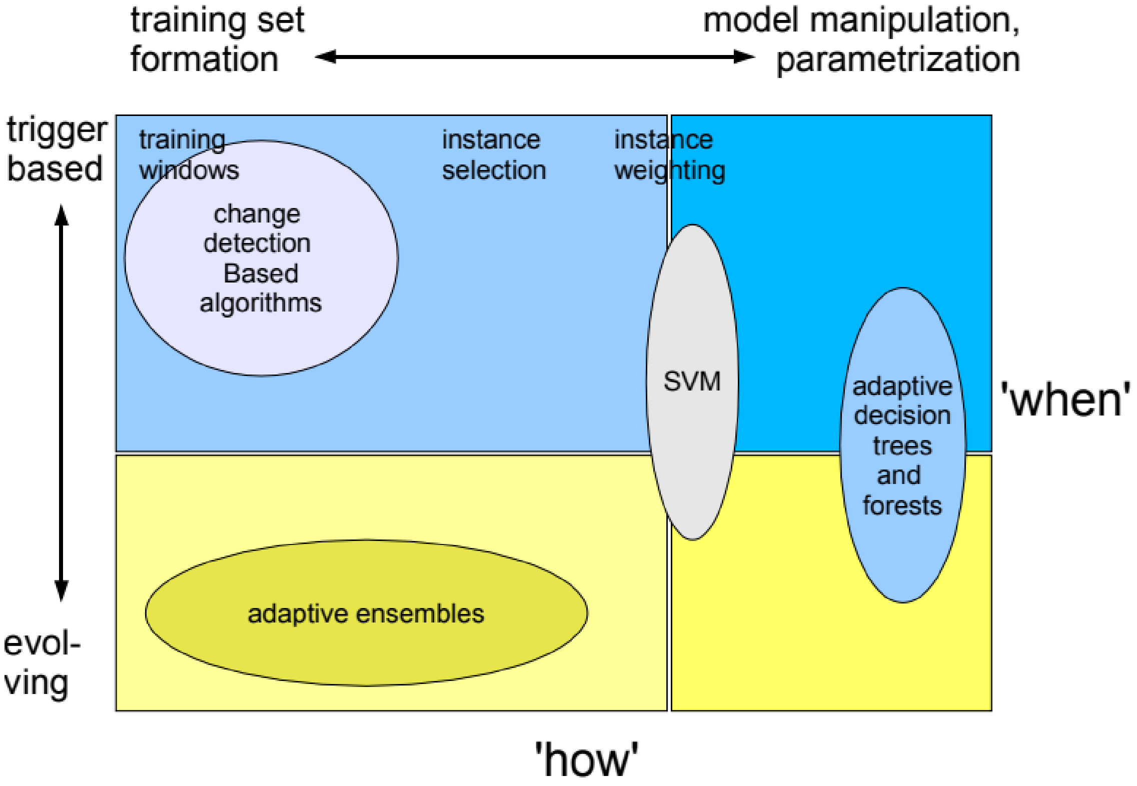
$$P_{t0}(X) \neq P_{t1}(X), P_{t0}(y|X) \neq P_{t1}(y|X)$$

Figure adapted from [1]

[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

[2] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. ACM Comput. Surv. 46, 4, Article 44 (April 2014), 37 pages. DOI:https://doi.org/10.1145/2523813

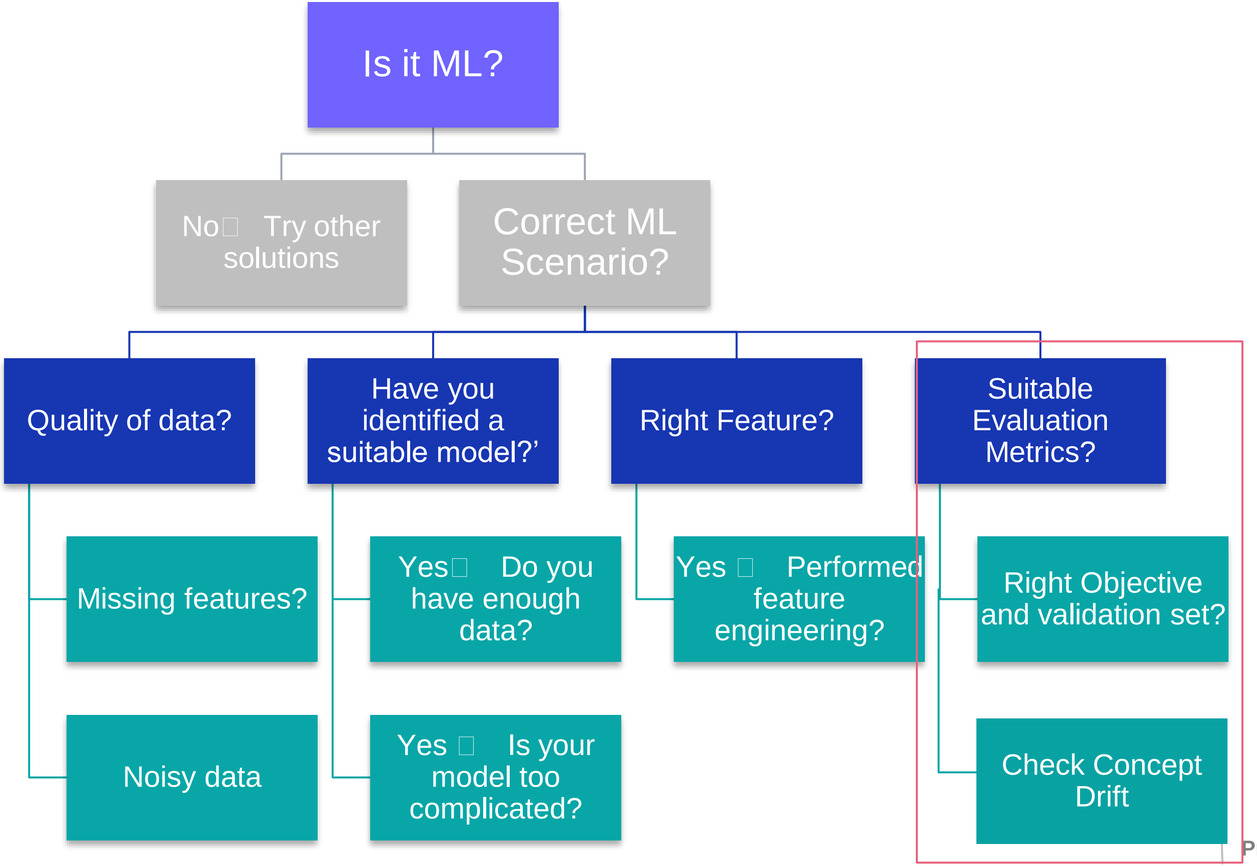
Solution overview



[1] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama and G. Zhang, "Learning under Concept Drift: A Review," in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346-2363, 1 Dec. 2019, doi: 10.1109/TKDE.2018.2876857.

Sadly, my ML model doesn’t work well in real-world scenario

The ML Diagnose Tree



Is it ML?

No□ Try other solutions

Correct ML Scenario?

Quality of data?

Missing features?

Noisy data

Have you identified a suitable model?'

Yes□ Do you have enough data?

Yes □ Is your model too complicated?

Right Feature?

Yes □ Performed feature engineering?

Suitable Evaluation Metrics?

Right Objective and validation set?

Check Concept Drift

Diagnose 8: Does my ML model objective align well with the ultimate application target?

In digital advertisement:

click → view → in_cart → purchase

In automated driving:

color/shape → object → scenario

In AI dialog system:

word-level understanding → sentence-level understanding → intent understanding

It's a paradox: The labels our customers really care about are usually hard to obtain for training !!

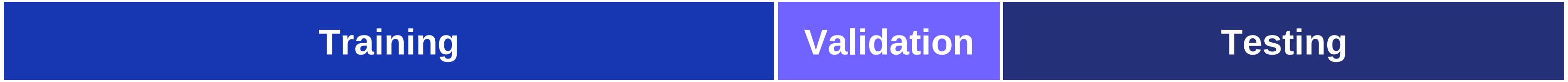
- *Focusing on the Longterm: It's Good for Users and Business. KDD '15*
- *Data-Driven Metric Development for Online Controlled Experiments: Seven Lessons Learned. KDD '16*
- *Measuring Metrics. CIKM '16*
- *Top Challenges from the first Practical Online Controlled Experiments Summit. KDD '19*

Diagnose 9: Is your model evaluated correctly?

- **Accuracy** is NOT always the best way to evaluate a machine learning model

- Case Study: An 99.999% accurate system in Detecting Malicious Personnel
    - Randomly picked person □ not likely a terrorist.
    - Thus, a model that always guess ‘non-terrorist’ will achieve very high accuracy
        - But it is useless !!
    - “Area under ROC Curve” (or AUC) is generally used to evaluate such system.

Diagnose 10: Have the Right Evaluation Data Been Used?

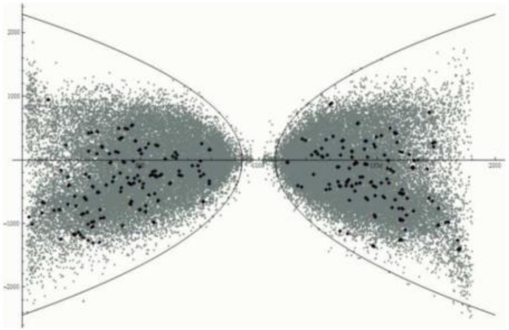


- We normally divide labeled data (or ground truth data) into three parts: **training**, **validation (or heldout),** and **testing**
- Performance on **training** data is **obviously biased** since the model was constructed to fit this data.
  - Accuracy must be evaluated on an independent (usually disjoint) test set.
  - Cannot peak the test set labels!!
- Use validation set to adjust hyper-parameters

How the validation set is chosen can affect the performance of the model!!

Case Study: Be aware of Leakage in validation

- **Training set: a set of positive and negative instances for cancer patient detection**
    - Each positive patient contain a set of negative instances (i.e. an ROI in the X-ray) and at least one positive instances.
    - ALL instances in a negative patient are negative
    - It's a multi-instance classification problem.
- **Random division for CV:**
    - training: 90%, testing: 72% □
    - significantly overfitting
- **Patient-based CV:**
    - training: 80%, testing: 77%



Survey paper: Leakage in data mining: formulation, detection, and avoidance (KDD 2011)

Case Study: Sampling a representative validation set

- **How to sample a validation dataset?**
  - Random sample □ ok but not good enough
  - Sample several different sets and test on a variety algorithms.
    - choose one that obtain similar **ranking across algorithms** with the testing (assuming aggregated performance for testing is available)

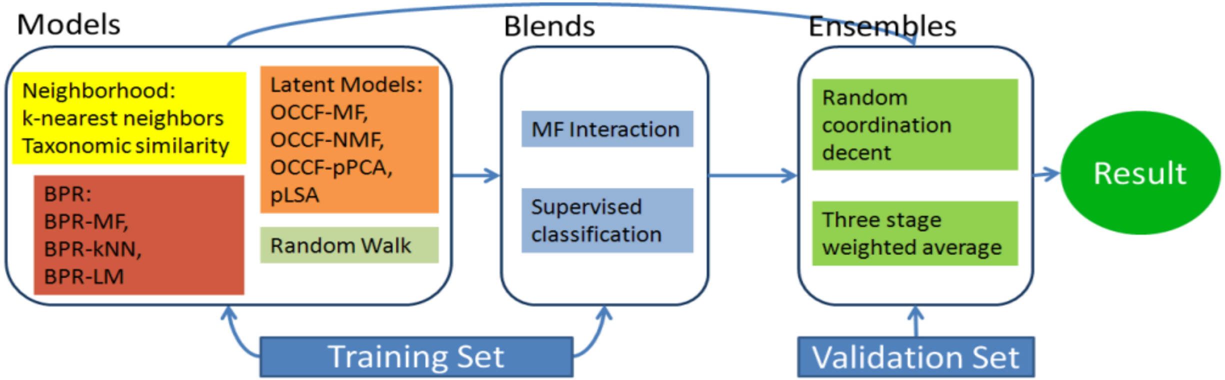
I’ll be fired if my ML model cannot do better!!

What to do if I absolutely need to boost the accuracy?

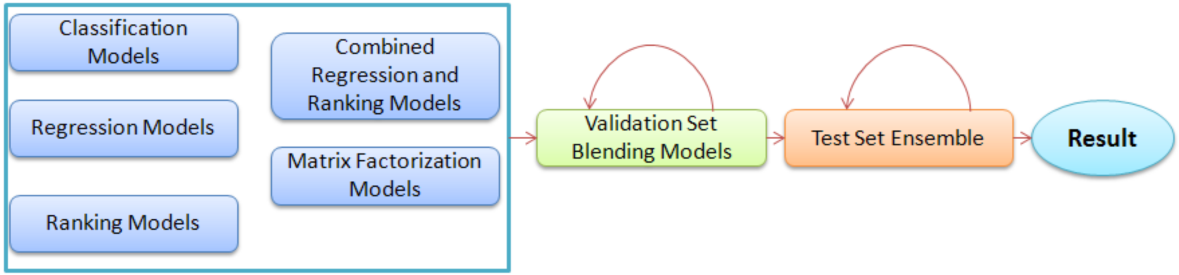
- **Blending and Ensemble:**
  - Quote from a winning team in ML competition: “everytime we add one more model into our ensemble, we have a big jump on the scoreboard”.
- **Blending: combine the results from some models**
  - Usually the number of models are not a lot
  - Non-linear methods such as kernel-SVM or neural network can be exploited
- **Ensemble: combine the results from blending models and individual models**
  - Usually takes a large amount of models
  - Simple linear or voting methods are exploited
  - Be careful, can cause overfitting.

Case Study: KDD Cup 2011 and 2012

2011



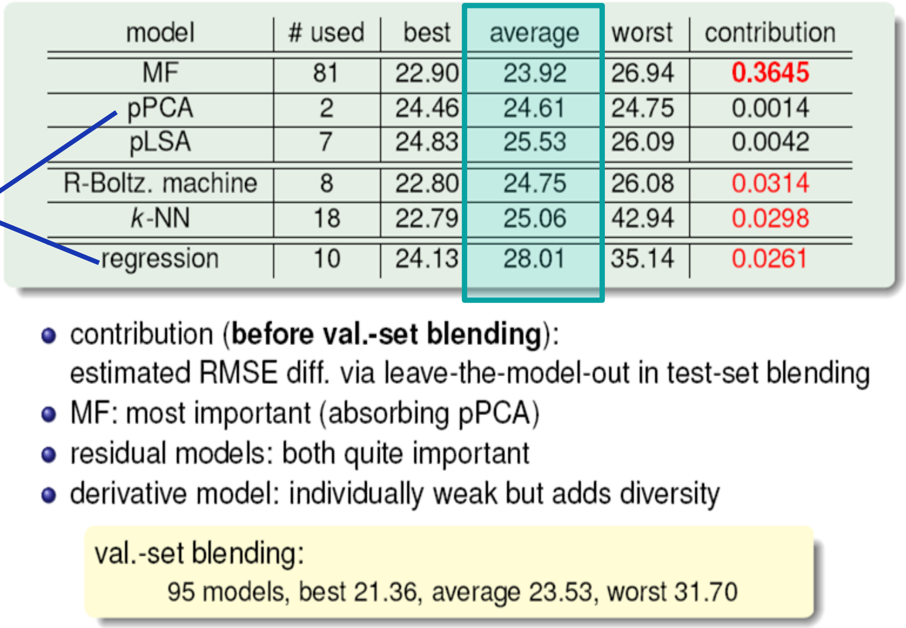
2012


Classification Models
Regression Models
Ranking Models
Combined Regression and Ranking Models
Matrix Factorization Models
Validation Set Blending Models
Test Set Ensemble
Result

Ensemble brings a different mindset to assess the quality of a model

- Does a worse model really has no value?
- A worse model is useful as long as it brings diversity
- A superior model might not be useful if it does not bring diversity.

Glance of Single Model RMSE



model	# used	best	average	worst	contribution
MF	81	22.90	23.92	26.94	**0.3645**
pPCA	2	24.46	24.61	24.75	0.0014
pLSA	7	24.83	25.53	26.09	0.0042
R-Boltz. machine	8	22.80	24.75	26.08	0.0314
k-NN	18	22.79	25.06	42.94	0.0298
regression	10	24.13	28.01	35.14	0.0261

- contribution (**before val.-set blending**): estimated RMSE diff. via leave-the-model-out in test-set blending
- MF: most important (absorbing pPCA)
- residual models: both quite important
- derivative model: individually weak but adds diversity

val.-set blending:
95 models, best 21.36, average 23.53, worst 31.70

Final Remark

- **Building ML models is very attractive because there is a clear metric to evaluate the performance**
  - **need to understand the definition of 'success' in advance**
- **While performance is very important, we need to further consider (1) cost and (2) maintenance while building an ML model**
  - **cost (human efforts + computation): spending 10 hours to tune a 100-layer DNN+attention model with 90% accuracy vs. spending 2 hours to apply a tree-based model to achieve 87% accuracy**
  - **maintenance: is the model too complicated to maintain? is debugging easy? is it too sensitive to the data/concept drift?**

Thank you and enjoy the life as an ML practitioner !!