

# Improvements

**Feedback loops and other measurements  
raising the bar**

Bhaskar Kamble - Kerstin Wagner - Pallavi Mitra

# How to improve your data product

— — —

## Basics

- Data & Flywheel
- Algorithm & Fine Tuning
- Ensembles

## More fancy stuff

- Transfer Learning
- Continual Learning
- Active Learning



## Improvements

### Data

R: K

- Get More Data.
- Invent More Data.
- Clean Your Data.
- Resample Data.
- Reframe Your Problem.
- Rescale Your Data.
- Transform Your Data.
- Project Your Data.
- Feature Selection.
- Feature Engineering.

### Algorithm

R: K

- Resampling Method.
- Evaluation Metric.
- Baseline Performance.
- Spot Check Linear Algorithms.
- Spot Check Nonlinear Algorithms.
- Steal from Literature.
- Standard Configurations.

### Fine / Tuning

R: P

- Diagnostics.
- Try Intuition.
- Steal from Literature.
- Random Search.
- Grid Search.
- Optimize.
- Alternate Implementations.
- Algorithm Extensions.
- Algorithm Customizations.
- Contact Experts.

### Flywheel

R: P

### Ensemble

R: B

- Blend Model Predictions.
- Blend Data Representations.
- Blend Data Samples.
- Correct Predictions.
- Learn to Combine.
- somehow in the presentation of Alexei

### Transfer L.

R: K

connection:  
Catastrophic forget

### Continuous L.

R: P

### Active L.

R: B

connection to transfer data:  
re-use models

# Basics

# Improve performance with data

---

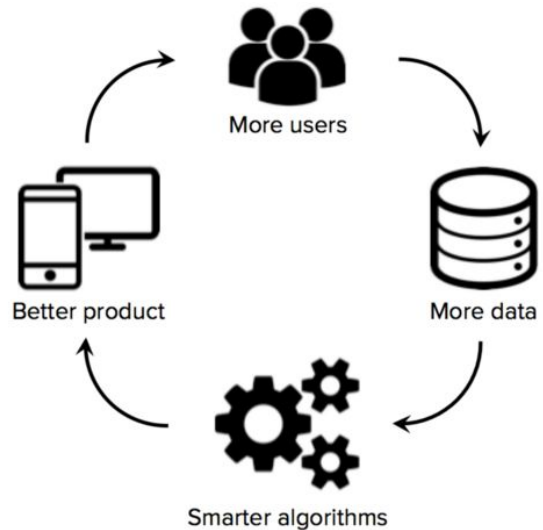
**Strategy:** Create new and different perspectives on your data in order to best expose the structure of the underlying problem to the learning algorithms.

- ❑ Get more data
- ❑ Invent your data
- ❑ Cleaning
- ❑ Resampling
- ❑ Problem reframing
- ❑ Rescaling
- ❑ Transformation
- ❑ Projection
- ❑ Feature selection
- ❑ Feature engineering

# Data Flywheel

— — —

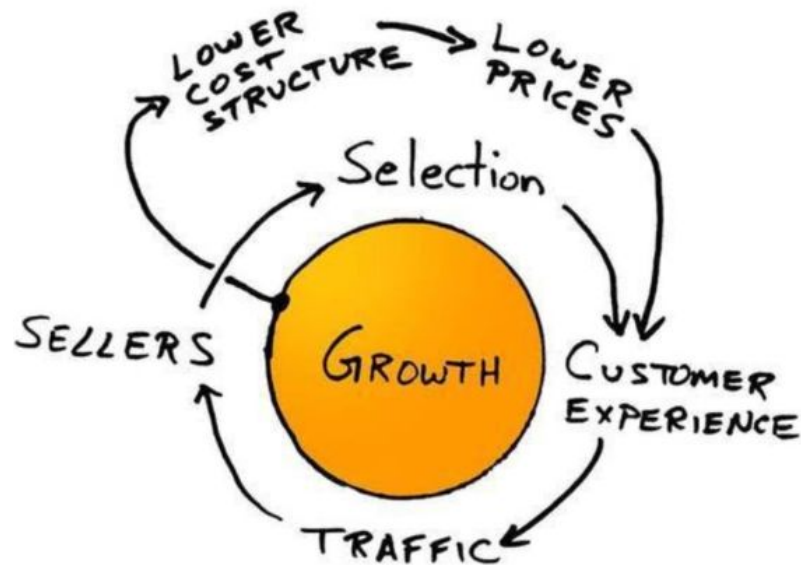
- The idea that more users get you more data which lets you build better algorithms and ultimately a better product to get more users.
- A self-reinforcing loop made up of a few key initiatives that feed and are driven by each other that builds a long-term business.
- No “one thing” powers it, and organizations that search for such a fundamentally basic solution will likely lose their way.



# Amazon Flywheel

— — —

- Customer experience is key, and all Amazon employee have this as their number one principle.
- Excellent customer experience drives traffic to Amazon.com.
- Sellers are attracted to put their products on Amazon.com.



# Improve performance with algorithms

— — —

**Strategy:** Identify the algorithms and data representations that perform above a baseline of performance and better than average. Remain skeptical of results and design experiments that make it hard to fool yourself.

- ❑ Baseline performance
- ❑ Evaluation metric
- ❑ Spot check algorithms
- ❑ Resampling
- ❑ Steal from literature
- ❑ Standard configurations

**Outcome:** You should now have a short list of well-performing algorithms and data representations.



# Improve performance with algorithm tuning

— — —

**Strategy:** Combine the predictions of multiple well-performing models.

- ❑ Diagnostics
- ❑ Try Intuition
- ❑ Steal from Literature
- ❑ Optimize
- ❑ Grid Search
- ❑ Random Search
- ❑ Alternate Implementations
- ❑ Contact Experts

**Outcome:** You should now have a short list of highly tuned algorithms on your machine learning problem, maybe even just one.

# Improve performance with ensembles

— — —

**Strategy:** Get the most out of well-performing machine learning algorithms.

- ❑ Blend Model Predictions
- ❑ Blend Data Samples
- ❑ Correct Predictions
- ❑ Learn to Combine

**Outcome:** You should have one or more ensembles of well-performing models that outperform any single model.

# More Fancy Stuff

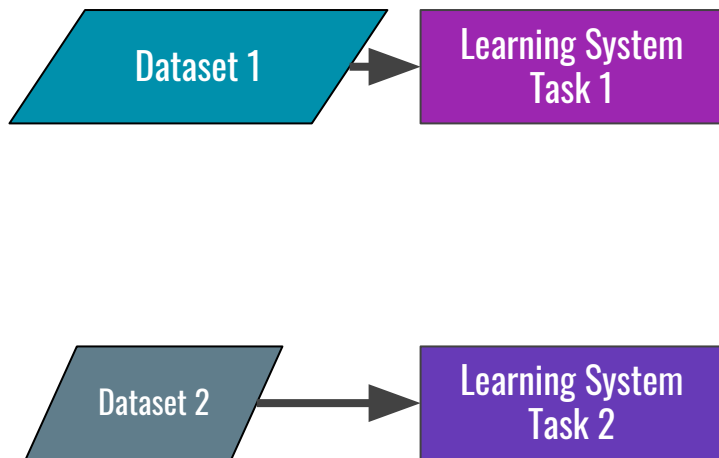
# Transfer Learning



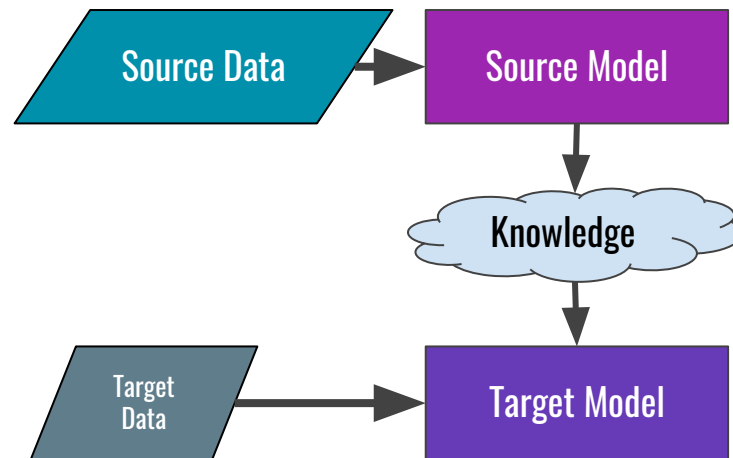
# Traditional ML vs Transfer Learning

— — —

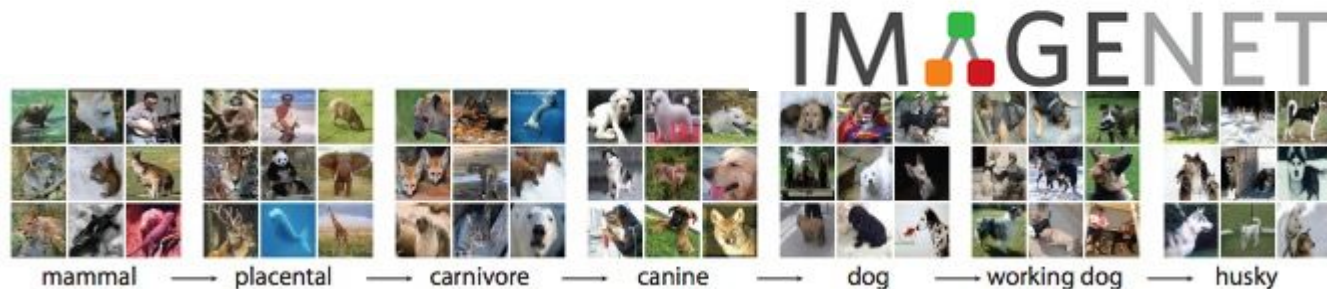
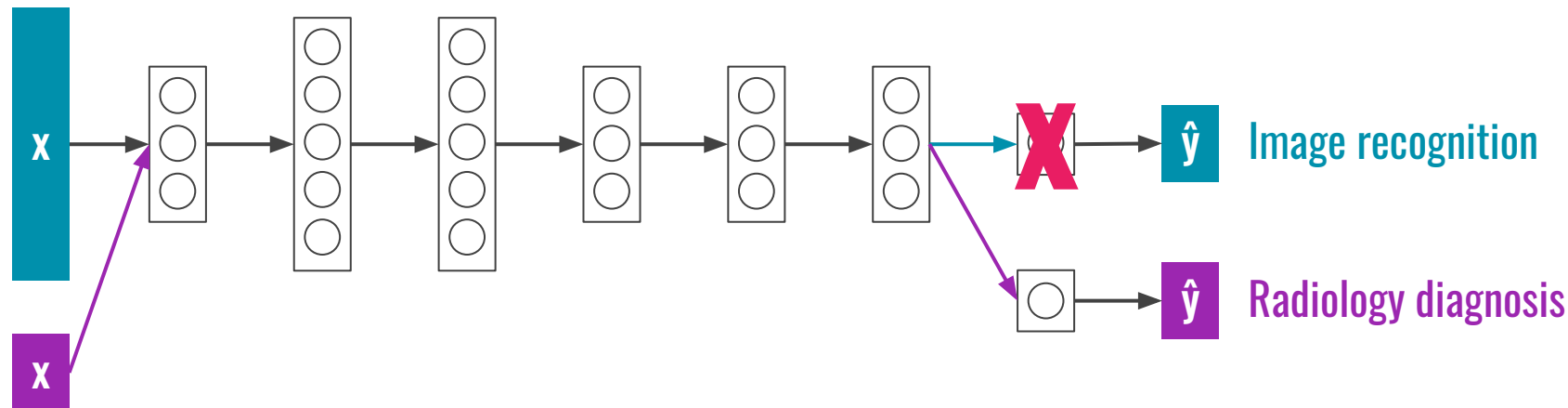
Isolated, single task learning:



Learning of new tasks relies on the previous learned tasks:

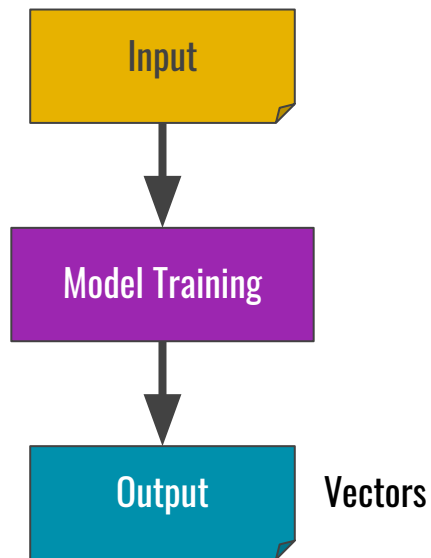


# Example: Computer Vision

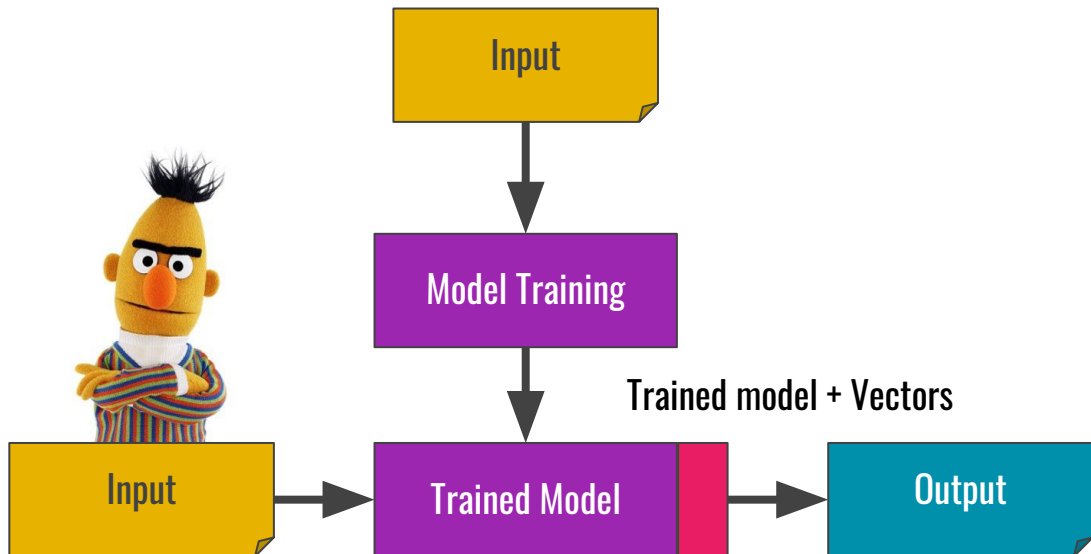


# What about NLP?


## Past Approach with Word2vec & Co

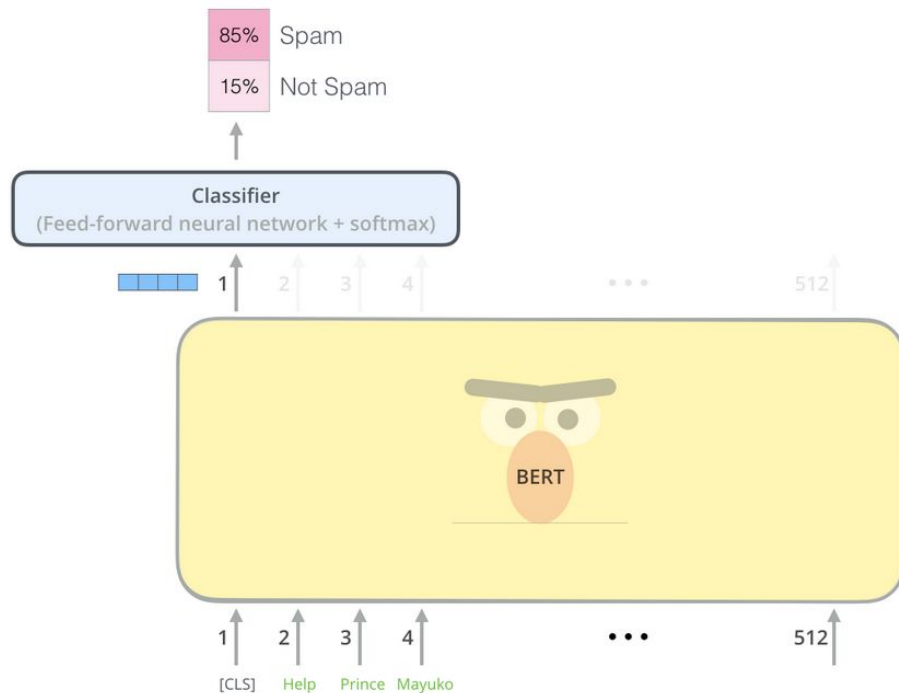
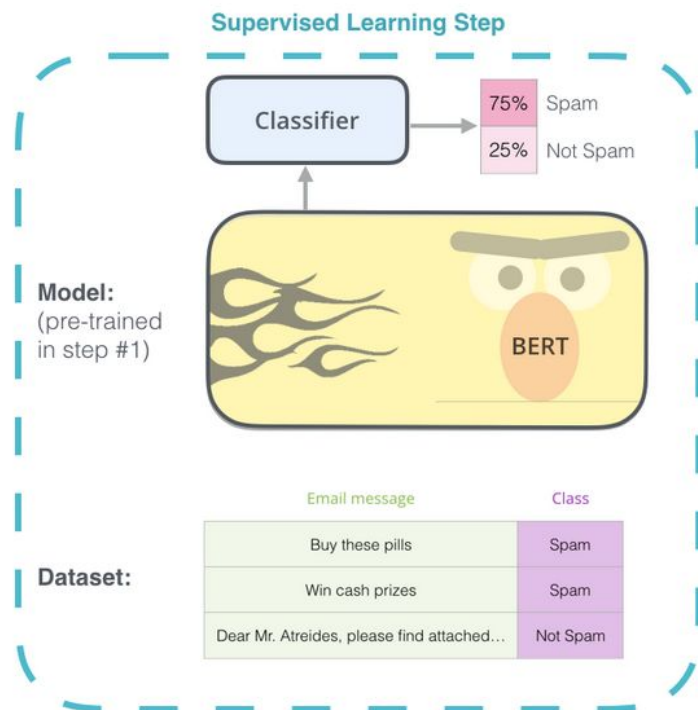


## Present Approach with BERT & Co



# How to use BERT

1. Pre-training on large amounts of text 
2. **Supervised training** on a specific task with a labeled dataset





# Transfer Learning Applications: Adapting to new domains

---



Different areas



Different text types



Different accents/dialects

# Continual Learning



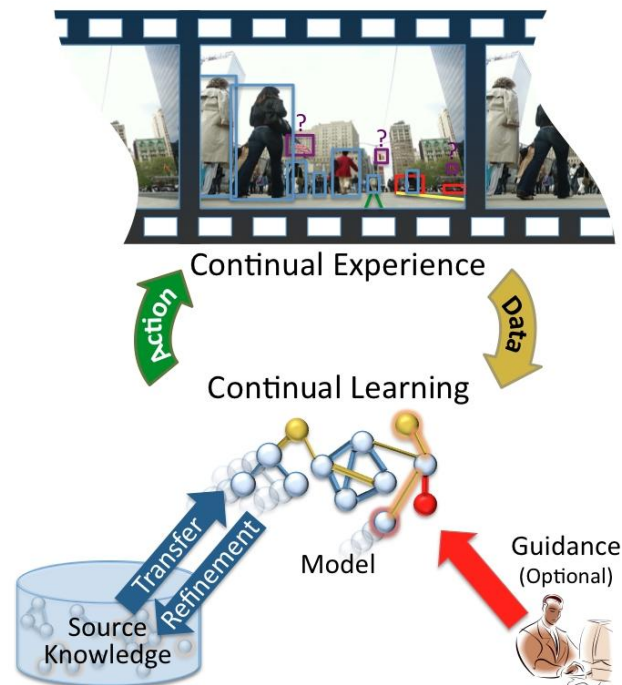
# Catastrophic forgetting

— — —

- When a neural network is used to learn a sequence of tasks, the learning of the later tasks may degrade the performance of the models learned for the earlier tasks.
- **An example of catastrophic forgetting is transfer learning using a deep neural network.**

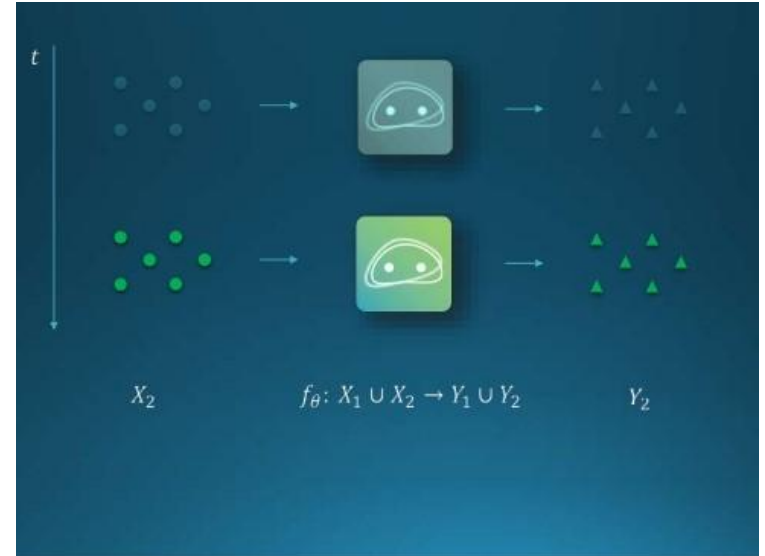
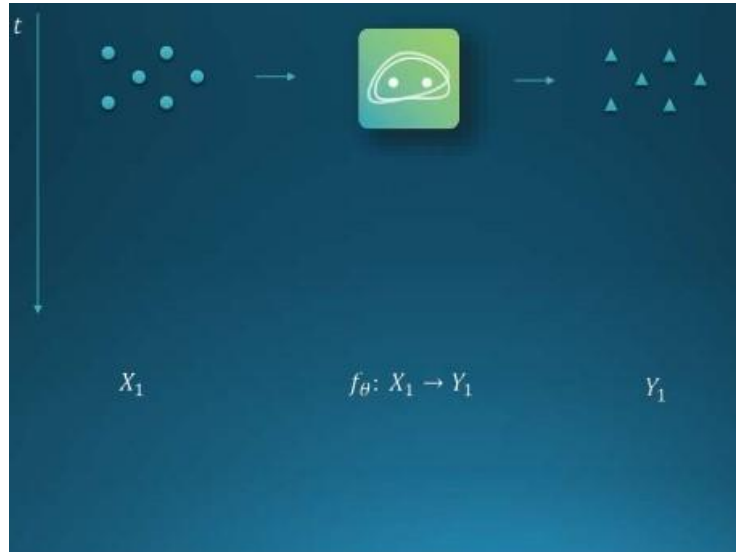
# Continual Learning

- Domain of machine learning which tries to mimic the human cognitive system by learning continually from a stream of data without forgetting previous knowledge
- Deals with an higher and realistic time-scale where data and tasks becomes available only during time, we have no access to previous perception data and it's imperative to build on top of previously learned knowledge.



# Illustration

— — —



# Properties

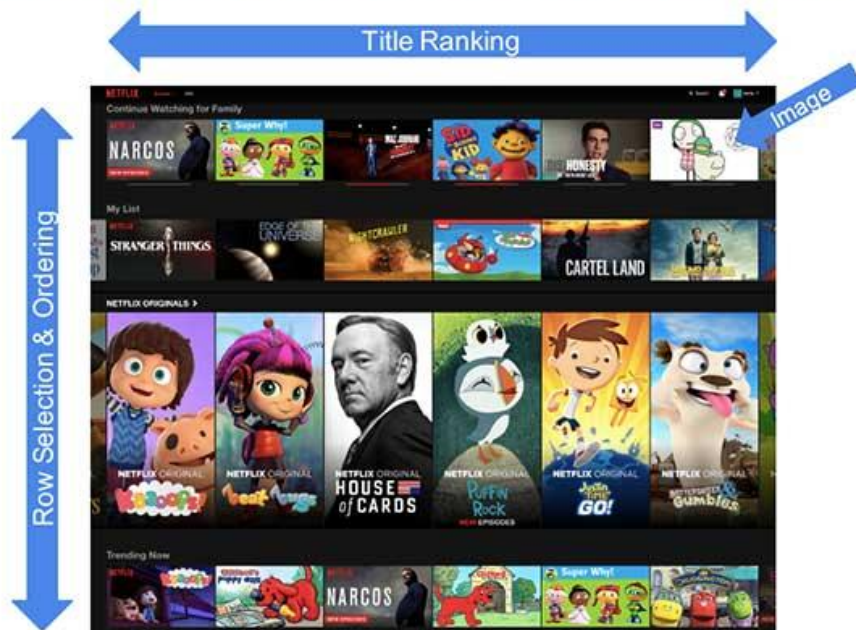
— — —

- Retain previously acquired knowledge to avoid catastrophic forgetting
- Efficiently utilise previous knowledge (transfer learning)
- Learn new knowledge



# An Example

## Everything is a Recommendation



Recommendations are driven by machine learning algorithms

**Over 80%** of what members watch comes from our recommendations

# Never-Ending Language Learner

— — —

- Semantic machine learning system developed by a research team at Carnegie Mellon University, and supported by grants from DARPA, Google, NSF, and CNPq with portions of the system running on a supercomputing cluster provided by Yahoo!
- The goal of NELL and other semantic learning systems, such as IBM's Watson system, is to be able to develop means of answering questions posed by users in natural language with no human intervention in the process.
- NELL was programmed by its developers to be able to identify a basic set of fundamental semantic relationships between a few hundred predefined categories of data, such as cities, companies, emotions and sports teams



# Never-Ending Language Learner (Contd.)

— — —

- Never-ending learning agent to be a system that, like humans, learns many types of knowledge, from years of diverse and primarily self-supervised experience, using previously learned knowledge to improve subsequent learning, with sufficient self-reflection to avoid plateaus in performance as it learns.
- It consists of a collection of learning tasks, and constraints that couple their solutions.
- $\mathcal{L} = (L, C)$  where  $L = (\langle T, P, E \rangle)$  and  $C = \{\varphi_k, V_{ki}\}$

# Properties of NELL agent

— — —

- learns many different types of inter-related knowledge; that is,  $L$  contains many learning tasks, coupled by many cross-task constraints,
- from years of diverse, primarily self-supervised experience; that is, the experiences  $\{E_i\}$  on which learning is based are realistically diverse, and largely provided by the system itself,
- in a staged, curricular fashion where previously learned knowledge supports learning subsequent knowledge; that is, the different learning tasks  $\{L_i\}$  need not be solved simultaneously—solving one helps solve the next, and
- where self-reflection and the ability to formulate new representations, new learning tasks, and new coupling constraints enables the learner to avoid becoming stuck in performance plateaus; that is, where the learner may itself add new learning tasks and new coupling constraints that help it address the given learning problem  $\mathcal{L}$ .

# Input-Output Specification

— — —

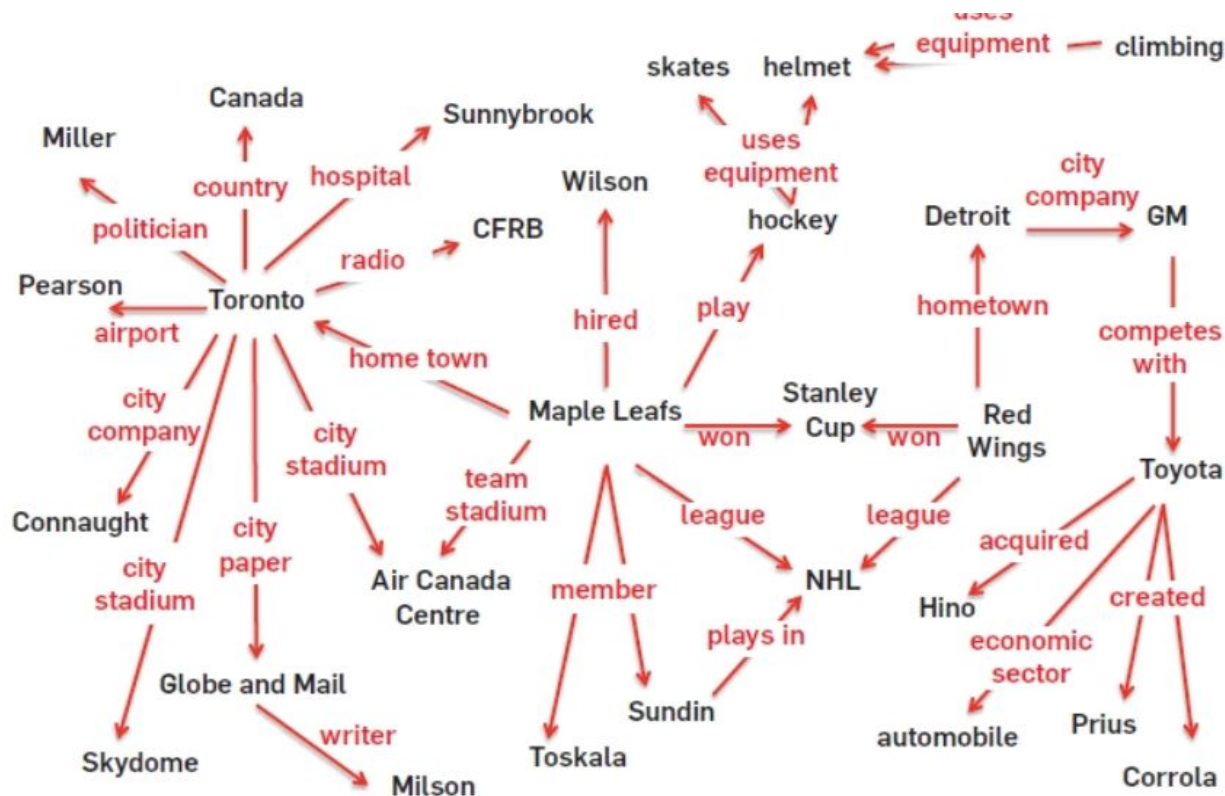
## Input

- an initial ontology defining categories
- approximately a dozen labeled training examples for each category and relation
- the web
- occasional interaction with humans

## Output

- Do: Run 24 hours/day, forever
- each day:
  - a. read (extract) more beliefs from the web, and remove old incorrect beliefs, to populate a growing knowledge base containing a confidence and provenance for each belief
  - b. learn to read better than the previous day

# NELL knowledge fragment

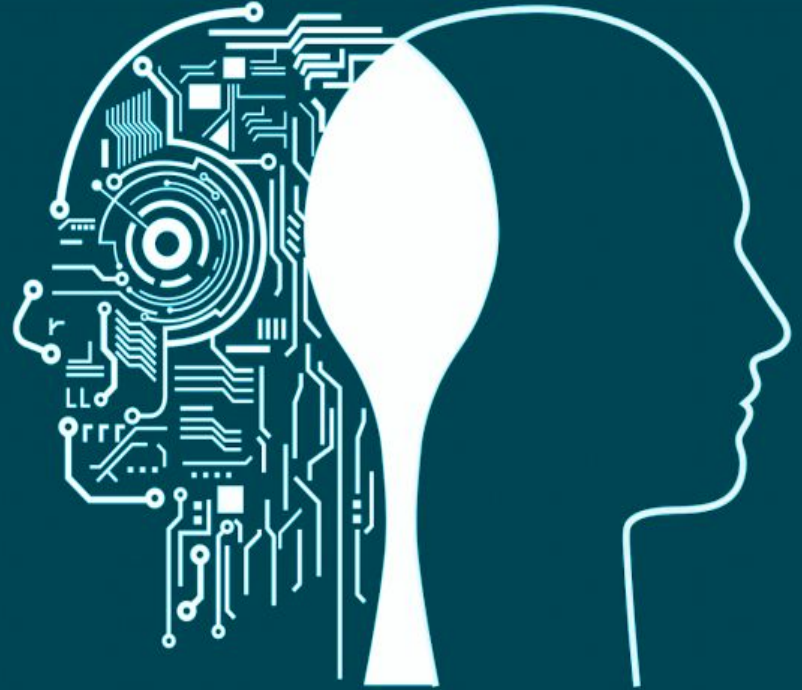


# Problems in CL

— — —

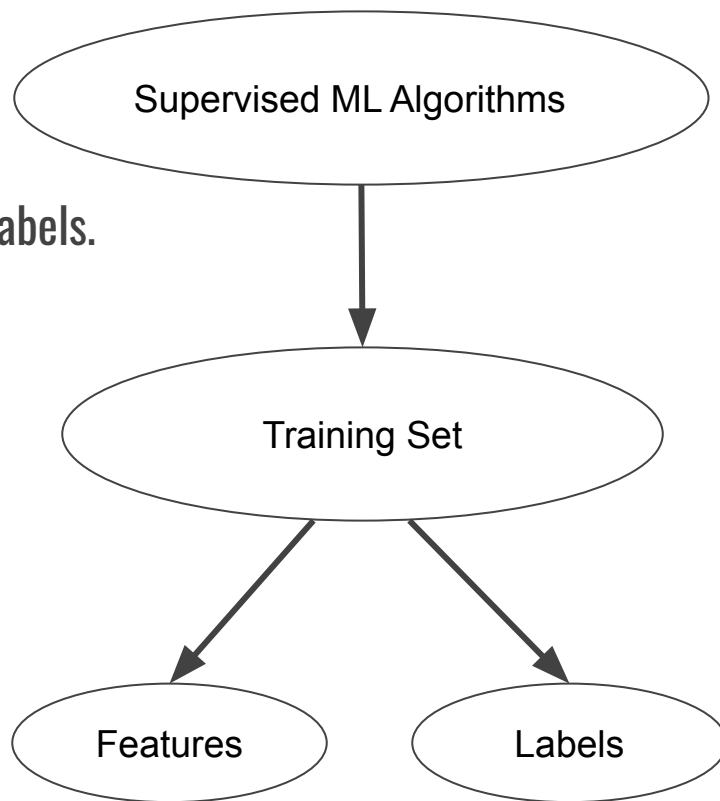
1. Not completely clear how evaluate CL techniques, and a more formalized framework may be needed.
2. Not clear how to behave after the saturation of the capability of the model, neither how to selectively forget.

# Active Learning



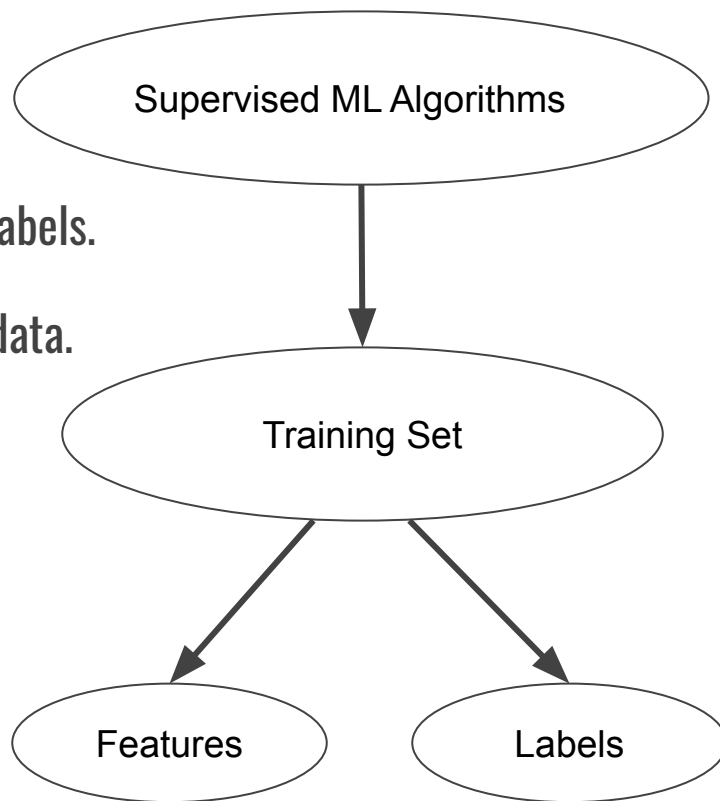
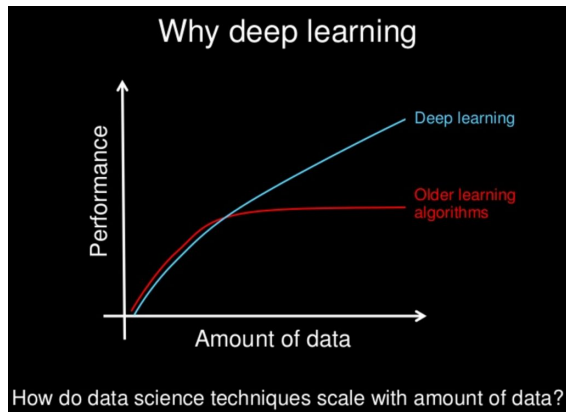
# Motivation

1. Training set for supervised ML algorithms: features + labels.



# Motivation

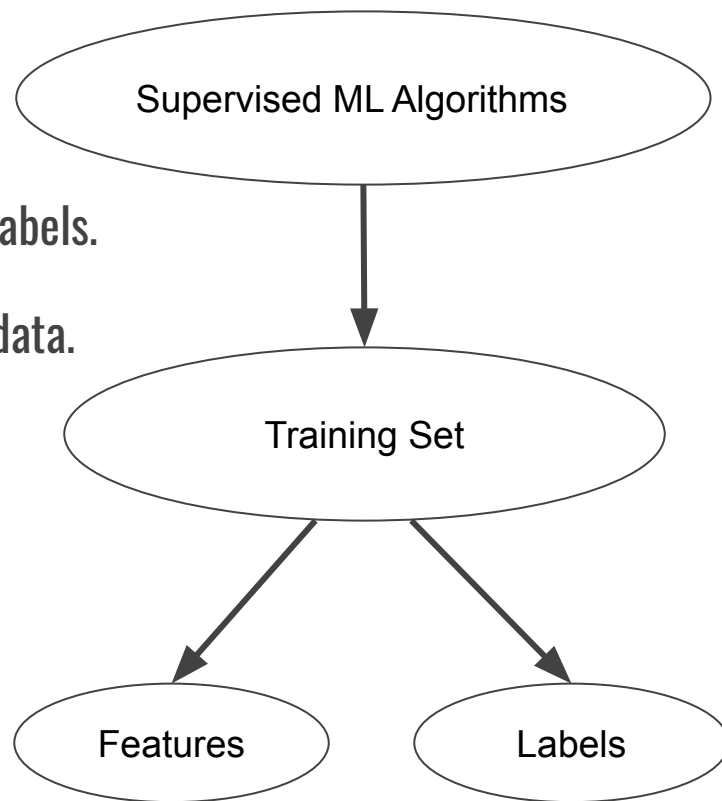
- 1. Training set for supervised ML algorithms: features + labels.
- 2. Accuracy of ML algorithms increases with increasing data.





# Motivation

- 1. Training set for supervised ML algorithms: features + labels.
- 2. Accuracy of ML algorithms increases with increasing data.
- 3. Data (Features) increasing at a tremendous rate.
- 4. Easy to get data through IoT etc.



# Motivation

1. Training set for supervised ML algorithms: features + labels.

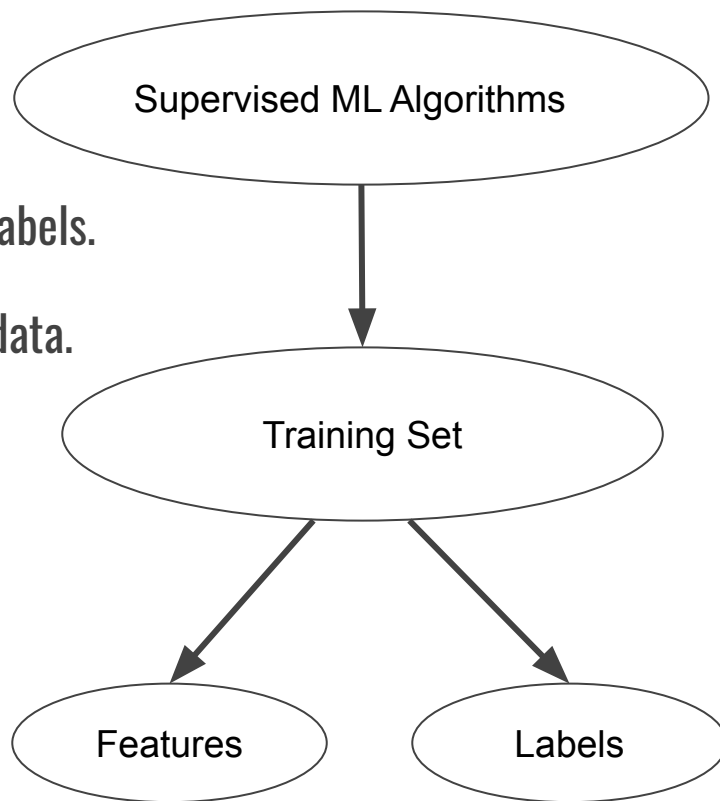
2. Accuracy of ML algorithms increases with increasing data.

3. Data (Features) increasing at a tremendous rate.

4. Easy to get data through IoT etc.



That's the easy part!



# Motivation

— — —

The hard part



Getting the labels

1. Labeling data done manually by an “oracle” (Amazon Mechanical Turk for example).
2. Especially relevant in medical cases, text documents, images.
3. Data labeling can be expensive.
  - X-Ray reports, blood tests for cancer and other illnesses (doctor: expensive oracle)
  - Text documents for hate speech or other content (lawyer: expensive oracle)
  - Labeling images for object recognition.
  - Can be a lot of work: predicting if an oil well exists, requires actual digging.

# Motivation

— — —

On the other hand:

1. More does not mean better
  - additional data may not be necessarily useful or better.
  - data often duplicated, eg. text and images on the net.
  - Too much of typical/redundant data (on which the model can predict well) slows up the learning process.
2. Which data is most relevant or most useful? Would be nice to know in advance, that way we can ask the oracle to label only the relevant data.

# Motivation

— — —

Three ways to approach the problem:

1. Prioritize which labeled data points you want, focus on getting them only. This happens before training.
2. Manual Override: Manually label only a small portion of the available data, use this model to predict over the entire data set, and then use manual override to correct the misclassifications.
3. Active Learning: choose whether to use a label during the training itself.

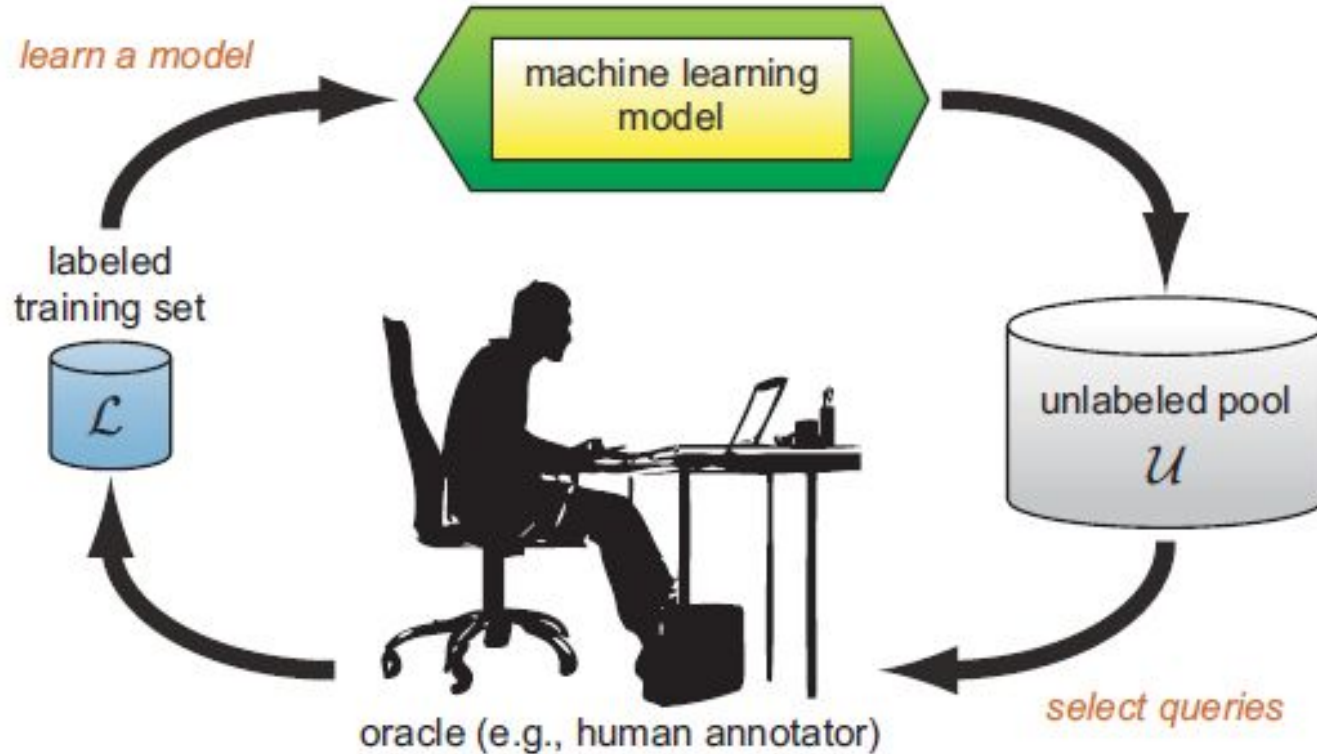
Active Learning: Most promising

# How it works (contd.)

— — —

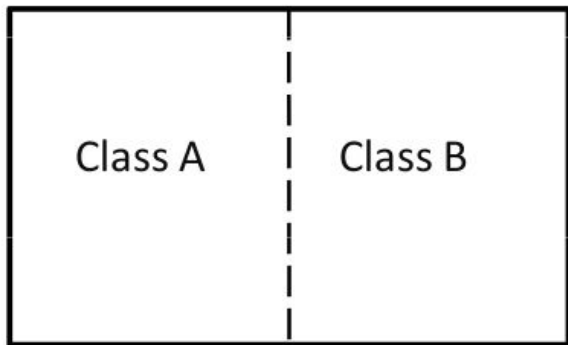
1. AL proceeds in rounds.
2. Each round has a current model learned using the labeled data seen so far.
3. The current model is used to assess informativeness of unlabeled examples using one of the query selection strategies (tbd later).
4. The most informative examples are selected and sent to oracle for labeling.
5. These are now included in the training data and the model is retrained.
6. The process goes to the next round and repeated until no budget left for getting labels or desired accuracy achieved.

# How it works (contd.)

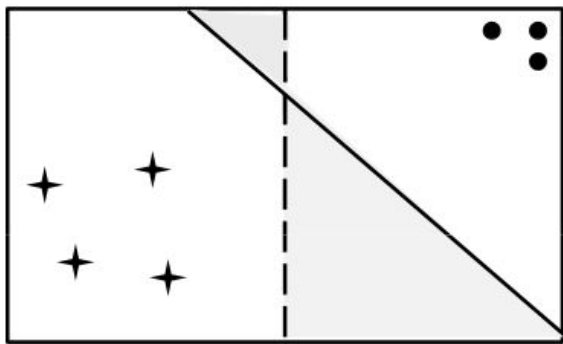


# An Example

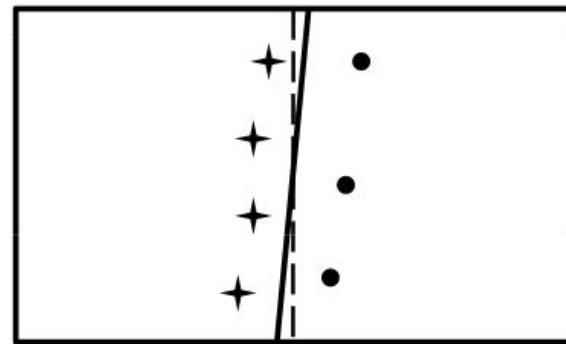
— — —



(a) Class Separation



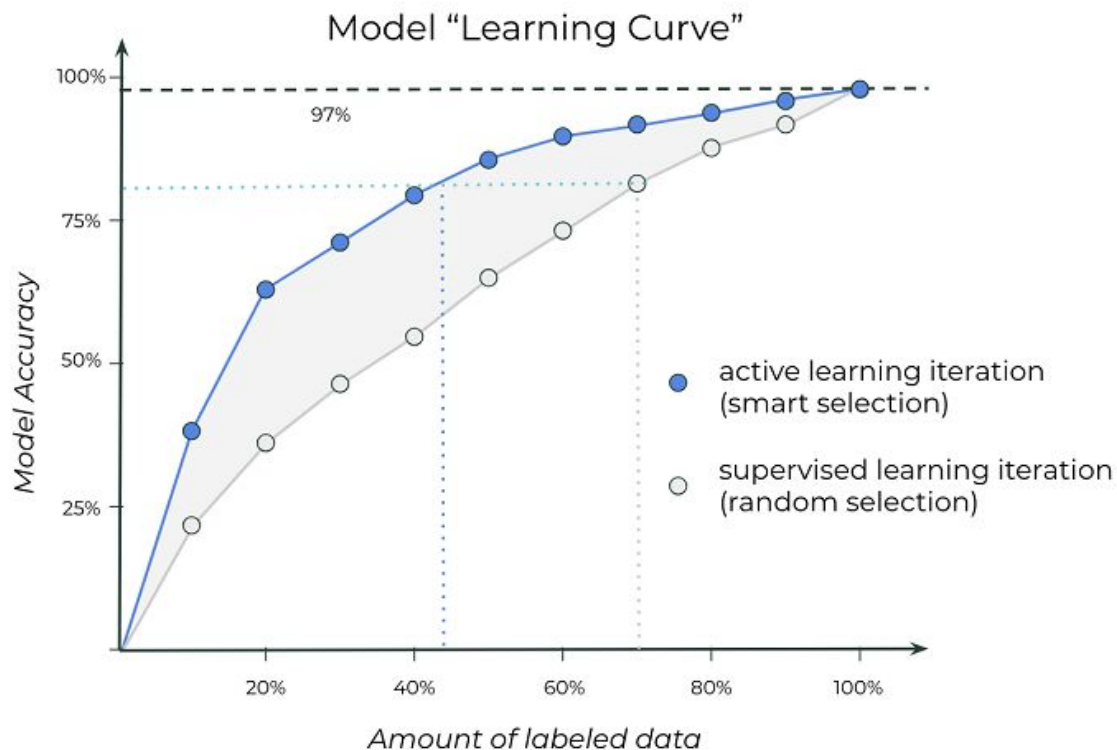
(b) Random Sample with SVM Classifier



(c) Active Sample with SVM Classifier



# Accuracy in Active Learning



# Active Learning Scenarios

— — —

1. Pool Based Scenario
2. Stream Based Selective Sampling Scenario
3. Membership Query Synthesis Scenario

# Pool Based Scenarios

— — —

1. Large pool of unlabeled data.
2. The active learner is usually initially trained on a fully labeled fraction of the data.
3. Attempts to evaluate the entire dataset before selecting the best query, or a set of best queries.
4. Disadvantage: memory-greediness.

# Stream Based Selective Sampling Scenario

— — —

1. Training examples typically keep coming continuously.
2. Machine decides immediately whether it needs a label for that data instance or not.
3. Example: Email server, with emails coming in continuously and you need to decide for each one whether its spam or not
4. Key assumption: obtaining an unlabeled instance is free (or in-expensive).
5. Disadvantage: no guarantee that the data scientist will stay within his/her budget.

# Membership Query Synthesis Scenario

— — —

1. Learner is allowed to construct its own examples for labelling. (Implies the generation of synthetic data.)
2. Easy when generating a data example is easy, difficult otherwise.

# Query Selection Strategies

— — —

1. Uncertainty Sampling
2. Query by Committee
3. Expected Model Change
4. Expected error reduction
5. Variance reduction
6. Density weighted methods

Source: [https://www.youtube.com/watch?v=8Jwp4\\_WbRio](https://www.youtube.com/watch?v=8Jwp4_WbRio)

# Uncertainty Sampling

— — —

Select examples for which the model is least confident / uncertain. Ways of quantifying uncertainty:

Options for probabilistic models:

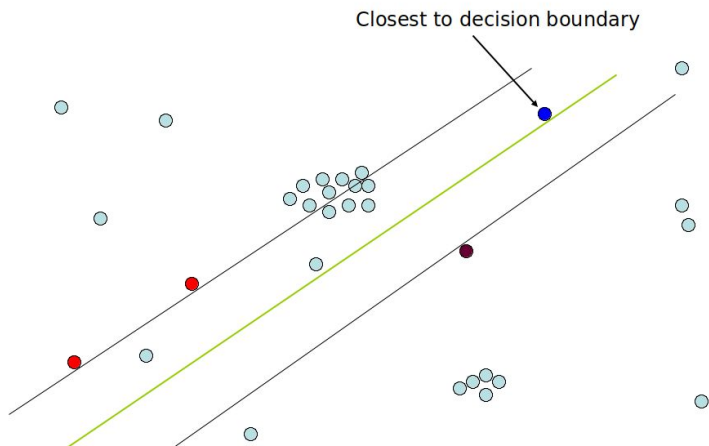
1.  $\operatorname{argmax}_x (1 - \max_k P_\theta(y_k|x))$  (High value indicates model is not confident for this example.)
2.  $\operatorname{argmin}_x (P_\theta(y_1^*|x) - (P_\theta(y_2^*|x)))$  ( $y_1^*$  and  $y_2^*$  top two most probable labels for  $x$  under model  $\theta$ . Indicates that the model predicts both  $y_1$  and  $y_2$ .)
3. Label Entropy. Choose  $x$  such that  $\operatorname{argmax}_x -\sum_i P_\theta(y_i|x) \log P_\theta(y_i|x)$  (also called margin sampling - Scheffer et al. 2001).

# Uncertainty Sampling

Select examples for which the model is least confident / uncertain. Ways of quantifying uncertainty:

1. For SVM, distance from the hyperplane serves as the measure of uncertainty.

Uncertainty Sampling (Source [1])





# Query By Committee

— — —

## Committee of classifiers

1. All models trained on the same set of labeled data.
2. Predictions are voted on the unlabeled pool.
3. Examples with maximum disagreement are chosen for labeling.

# Query By Committee (contd.)

— — —

## 4. Measure for disagreement:

▷ Vote Entropy (Dagan and Engelson, 1995):

$$x_{VE}^* = \operatorname{argmax}_x - \sum_i \frac{V(y_i)}{C} \log \frac{V(y_i)}{C}$$

$C$  = number of models in committee.

5. Each model retrained after including the new example and the process continues.

# Other Strategies

— — —

Limitation of Uncertainty Sampling or QBC:

1. Focus on outliers.
2. Outliers are thought to be informative.
3. Outliers are useless at best, misleading at worst.

Alternative idea:

Instead of using the confidence of a model on an example, *see how a labeled example affects the model itself.*

# Other Strategies (contd.)

— — —

1. Expected Model Change
2. Expected Error Reduction
3. Variance Reduction
4. Density Weighted Methods

# Expected Model Change

— — —

Select the instance that would impart the greatest change to the current model if we knew its label.

Common criterium: Expected Gradient Length (EGL):

What is that instance which, if included in the training set, most alters the rate at which the cost function changes, i.e. the gradient of the cost function?

(In the absence of this instance, the gradient is zero)

# Expected Model Change

— — —

$\nabla l_{\theta}(L)$  = Gradient of cost function ( $l$ ) for the present training set  $L$ .

$\nabla l_{\theta}(L \cup \langle x, y \rangle)$  = Gradient of cost function for the present training set  $L$  *and* the labeled instance  $\langle x, y \rangle$ . This not known (since  $\langle x, y \rangle$  is unlabeled), hence expected value used:

$$x_{EGL}^* = \operatorname{argmax}_x \sum_i P_{\theta}(y_i|x) \left\| \nabla l_{\theta}(\mathcal{L} \cup \langle x, y_i \rangle) \right\|$$

Approximation:  $\nabla l_{\theta}(L \cup \langle x, y \rangle) \approx \nabla l_{\theta}(L) + \nabla l_{\theta}(\langle x, y \rangle) = \nabla l_{\theta}(\langle x, y \rangle)$

(Since the first term in the middle is zero.)

# Expected Error Reduction

— — —

One possibility

$$x_{0/1}^* = \operatorname{argmin}_x \sum_i P_\theta(y_i|x) \left( \sum_{u=1}^U 1 - P_{\theta+\langle x, y_i \rangle}(\hat{y}|x^{(u)}) \right)$$

$U$  is the unlabeled set.  $\theta$  is the model trained on  $L$ .  $\theta^+$  is the model trained on  $L \cup \langle x, y_i \rangle$ . Since we do not know  $y_i$ , we take expectation over that to find the expected error rate.

# Variance Reduction

— — —

The mean square error consists of noise, bias and variance.

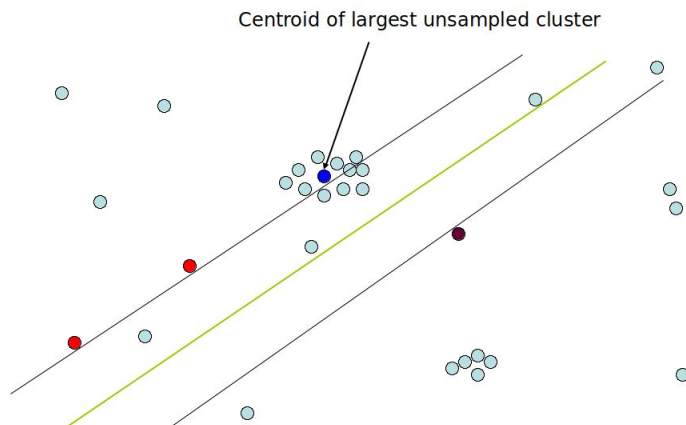
Reduce the total error by reducing the variance by suitably choosing training examples.



# Density Weighted Methods

Weight the informativeness of an example by its average similarity to the entire unlabeled pool of examples. This way an outlier won't get substantial weight.

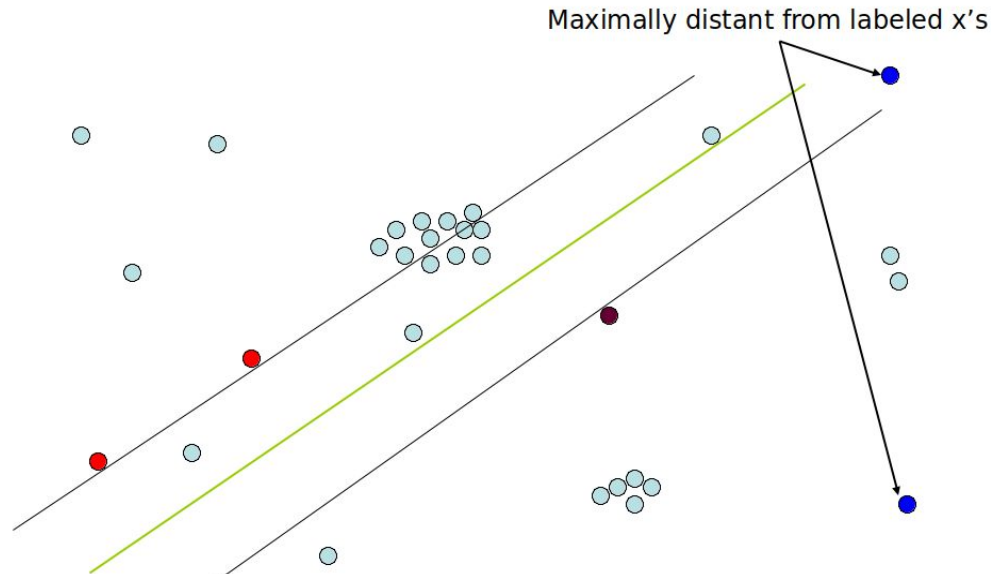
Density-Based Sampling (Source [1])



# Diversity Based Sampling

— — —

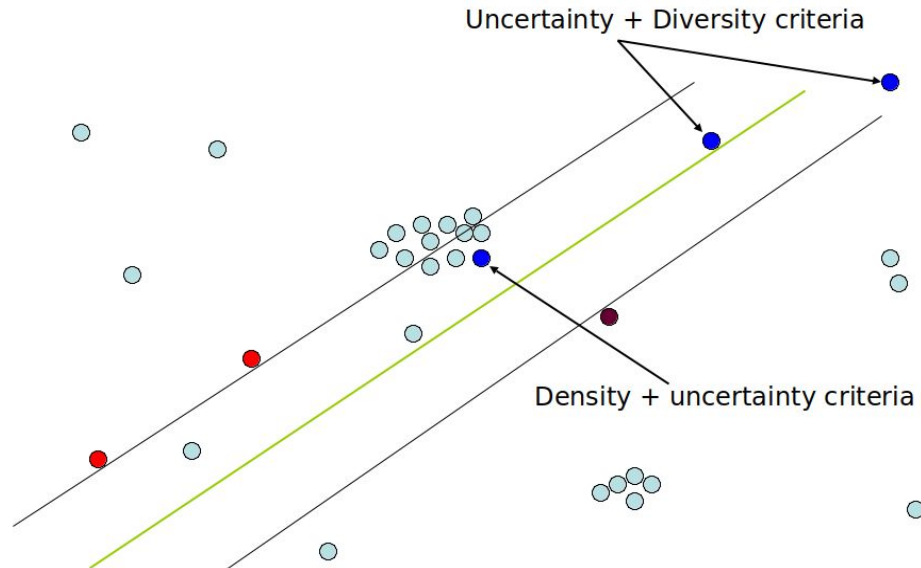
Maximal Diversity Sampling (Source [1])



# Ensemble Based Possibilities

— — —

Ensemble-Based Possibilities (Source [1])



# Further Application -Generalizing Labels

— — —

Number of labels limited by human ability (e.g. cat or dog in image classification).

Train on labeled data set with limited number of labels.

Predict on unlabeled data, that now contains also horses and birds.

Predictions with low confidence: probably horses and birds - give for labeling.

# Summary

— — —

AL is a label efficient learning strategy in which data are labeled based on their informativeness.

Several variants possible (as discussed here)

Being used in industry (IBM, Microsoft, Siemens, Google, etc.)

Further questions:

- The actively labeled dataset does not reflect the true training/test data distribution (i.e. Sampling is biased).
- Can an actively labeled dataset be used to train a new different model?

# Thank you!

Further questions?

# Sources: Basics

— — —

- <https://machinelearningmastery.com/machine-learning-performance-improvement-cheat-sheet/>
- <https://machinelearningmastery.com/spot-check-classification-machine-learning-algorithms-python-scikit-learn/>
- <https://machinelearningmastery.com/improve-deep-learning-performance/>
- <http://faculty.marshall.usc.edu/gareth-james/ISL/index.html> (James et al)
- <https://pages.awscloud.com/data-flywheel.html>
- <https://www.cbinsights.com/research/team-blog/data-network-effects/>
- <https://medium.com/swlh/the-amazing-flywheel-effect-80a0a21a5ea7>
- <https://becominghuman.ai/top-machine-learning-algorithms-you-should-know-to-become-a-data-scientist-17b16bc85077>

# Sources: Transfer Learning

— — —

- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- <https://www.kdnuggets.com/2019/11/transfer-learning-coding.html>
- <https://www.youtube.com/watch?v=yofjFQddwHE>
- <https://blog.slavv.com/a-gentle-intro-to-transfer-learning-2c0b674375a0>
- <https://runder.io/nlp-imagenet/>
- <https://runder.io/state-of-transfer-learning-in-nlp/>
- <http://jalammar.github.io/illustrated-bert/>
- <https://yashuseth.blog/2019/06/12/bert-explained-faqs-understand-bert-working/>



# Sources: Continual Learning

— — —

- The Netflix Recommender System: Algorithms, Business Value, and Innovation -<https://dl.acm.org/citation.cfm?id=2843948>
- <https://www.cs.uic.edu/~liub/lifelong-learning/continual-learning.pdf>
- <https://www.slideshare.net/VincenzoLomonaco/continual-learning-with-deep-architectures-workshop-computer-visioners-conference-2018>
- <https://cacm.acm.org/magazines/2018/5/227193-never-ending-learning/fulltext>
- [http://talukdar.net/papers/NELL\\_aaai15.pdf](http://talukdar.net/papers/NELL_aaai15.pdf)
- <https://medium.com/continual-ai/why-continuous-learning-is-the-key-towards-machine-intelligence-1851cb57c308>
-

# Sources: Active Learning

— — —

- <https://www.youtube.com/watch?v=V33Ut36eUsY> (Jennifer Prendki)
- <https://www.kdnuggets.com/2018/10/introduction-active-learning.html> (Jennifer Prendki )
- Burr Settles
- C. Aggarwal, X. Kong, Q. Gu, J. Han, P. Yu. Active Learning: A Survey, Data Classification: Algorithms and Applications, CRC Press 2014.  
(<http://www.charuaggarwal.net/active-survey.pdf> )
- [https://www.youtube.com/watch?v=8Jwp4\\_WbRio](https://www.youtube.com/watch?v=8Jwp4_WbRio) (Jordan Boyd-Graber Digging into Data: Active Learning)
- <https://towardsdatascience.com/active-learning-tutorial-57c3398e34d>
- <https://towardsdatascience.com/how-active-learning-can-help-you-train-your-models-with-less-data-389da8a5f7ea>
- <https://www.datacamp.com/community/tutorials/active-learning>
- <https://medium.com/towards-artificial-intelligence/how-to-use-active-learning-to-iteratively-improve-your-machine-learning-models-1c6164bdab99>
- <https://www.slideserve.com/holly/active-learning-02-750>