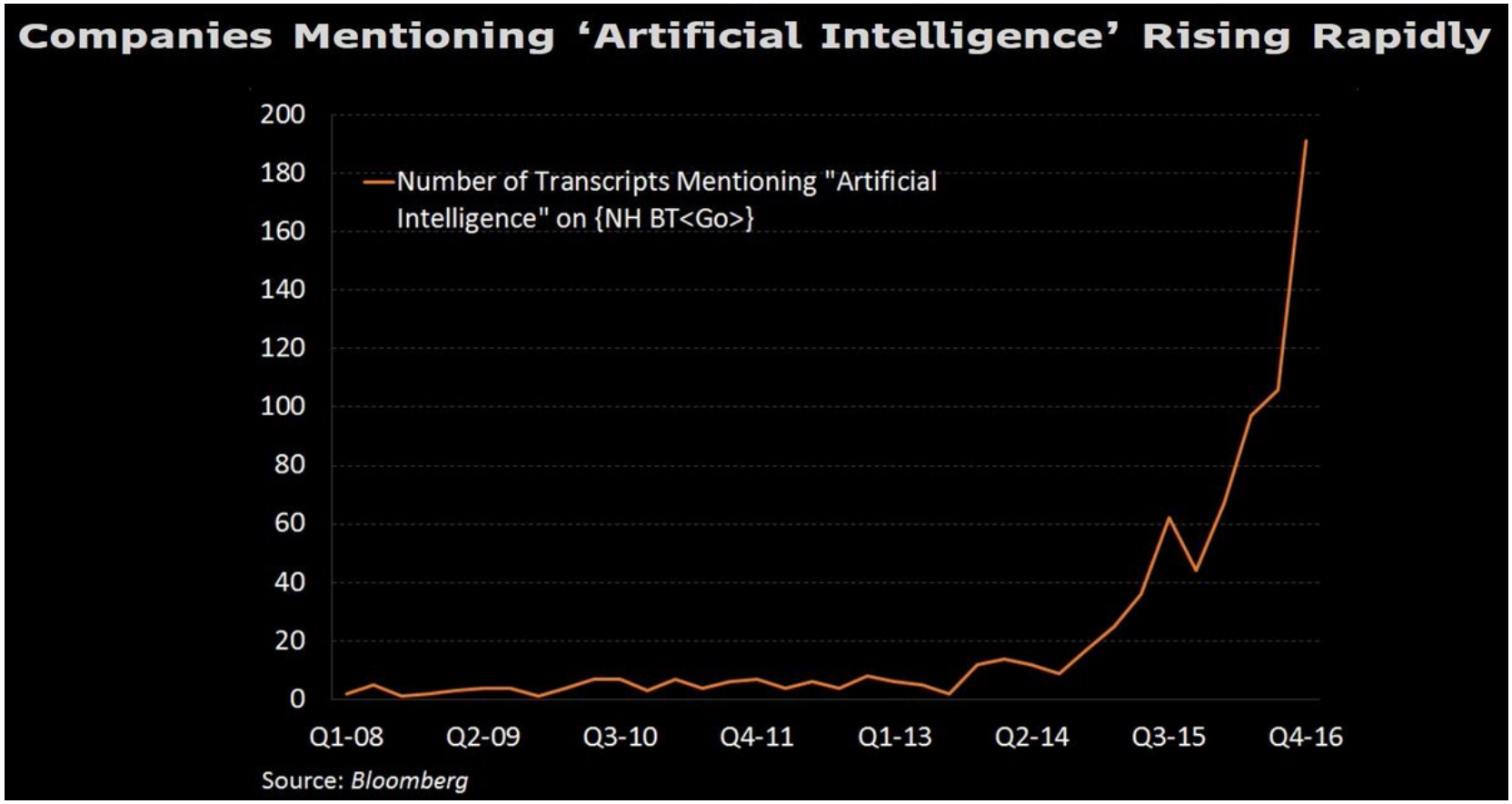
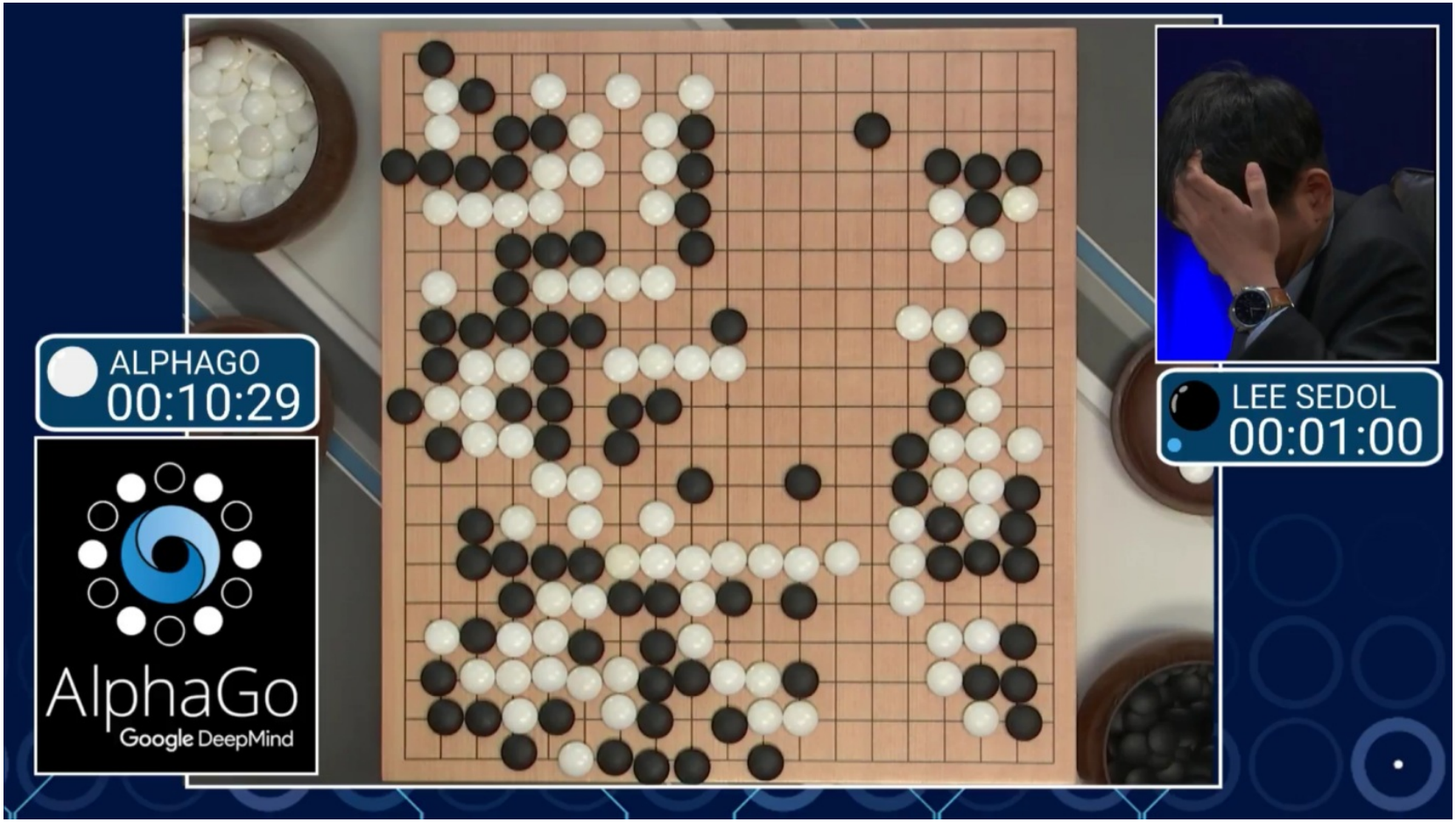
Probably Approximately Correct

A very brief tour of all of machine learning

Shane Conway
Kepos Capital



Mentioned by Paul Kedrosky (@pkedrosky), March 1, 2017.



The number of potential legal board positions in go is greater than the number of atoms in the universe.

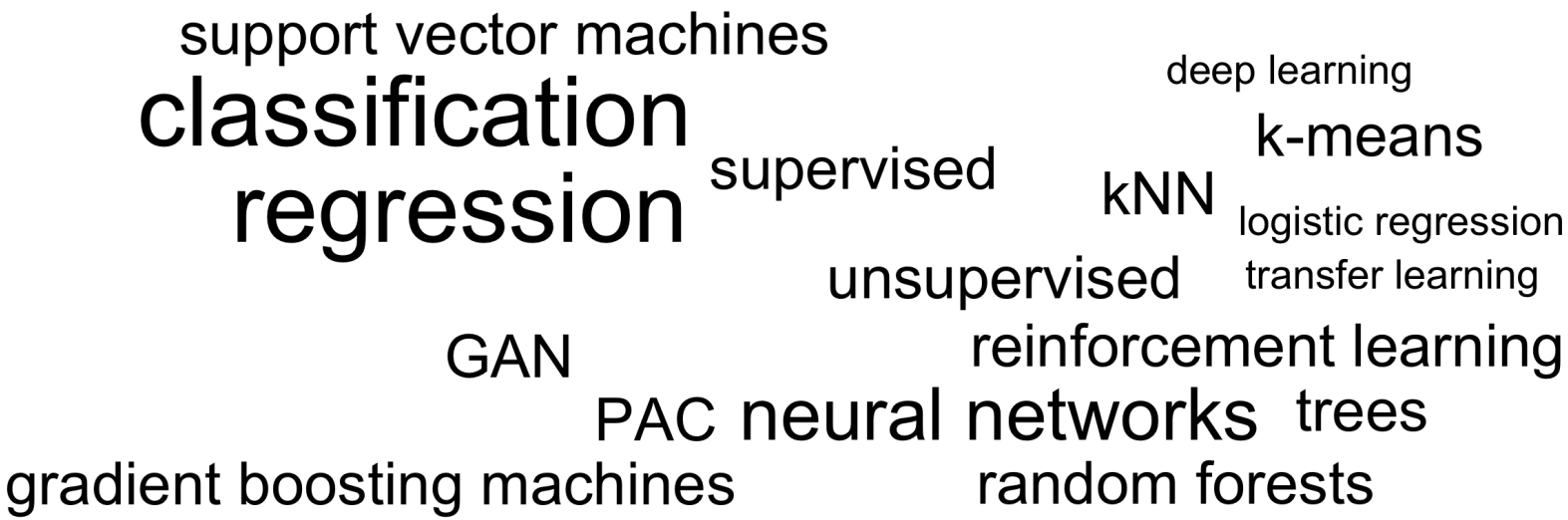
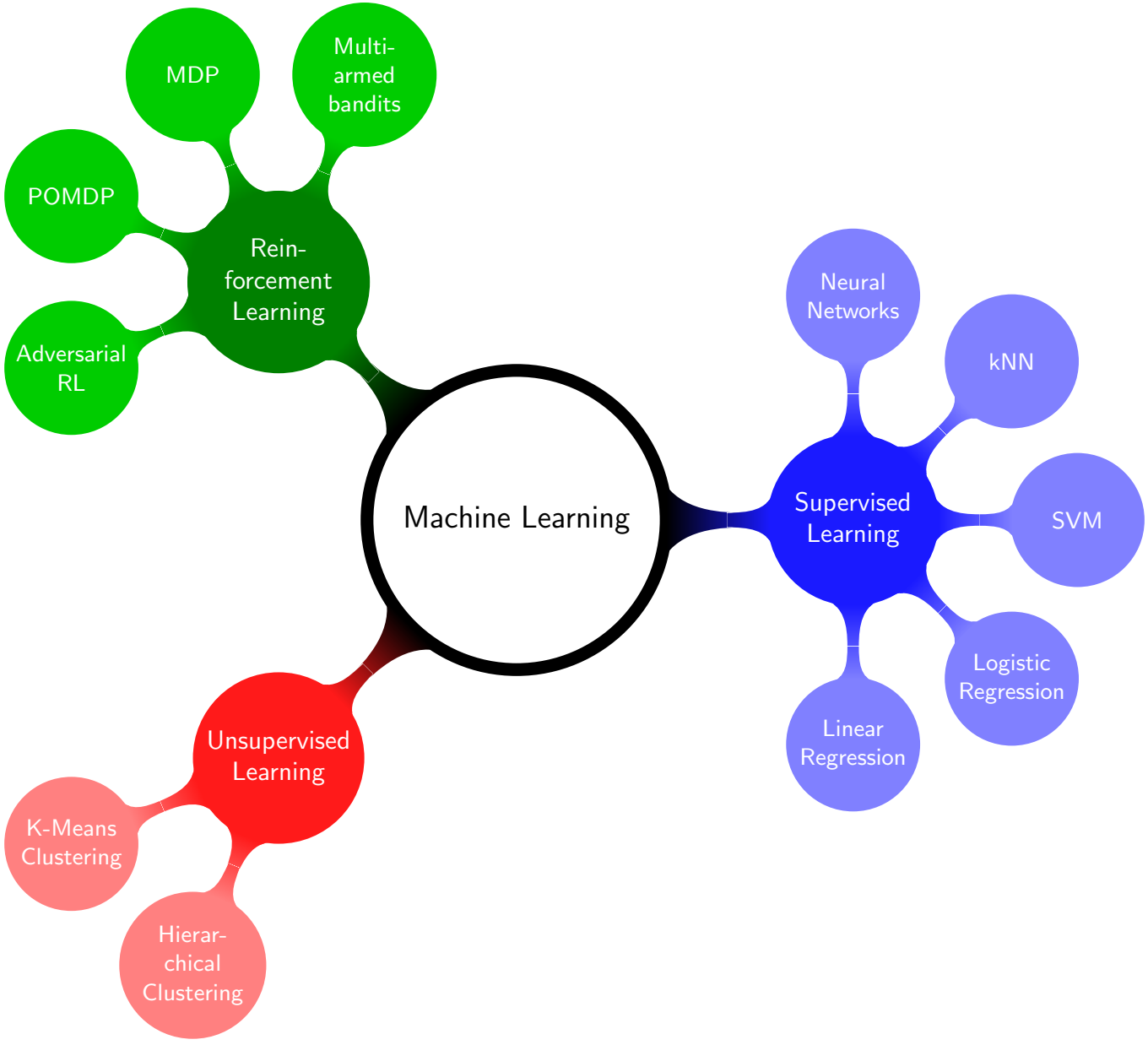
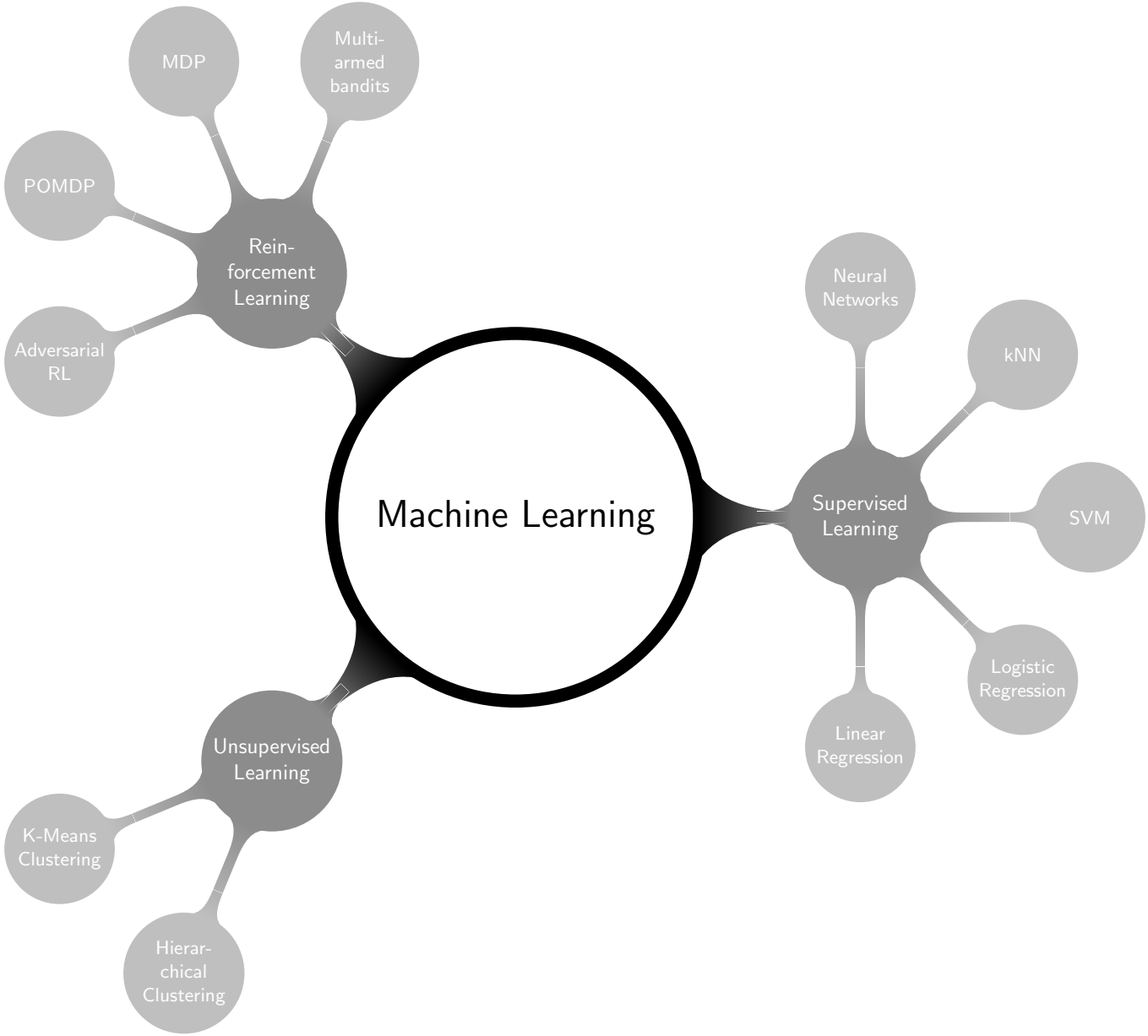


Figure : A cloud of terminology.


Machine Learning
Rein-
forcement
Learning
MDP
Multi-
armed
bandits
POMDP
Adversarial
RL
Supervised
Learning
Neural
Networks
kNN
SVM
Logistic
Regression
Linear
Regression
Unsupervised
Learning
K-Means
Clustering
Hierar-
chical
Clustering


Machine Learning
Reinforcement Learning
MDP
Multi-armed bandits
POMDP
Adversarial RL
Supervised Learning
Neural Networks
kNN
SVM
Logistic Regression
Linear Regression
Unsupervised Learning
K-Means Clustering
Hierarchical Clustering

What is machine learning?

Leo Breiman (2004) "The Two Cultures":

- *statistics* $\longrightarrow$ explanation
- *machine learning* $\longrightarrow$ prediction

All generalizations are false, including this one. - Mark Twain

The truth is that machine learning (a CS field) and statistics have massive overlap, and both fields have benefitted from the interaction.

Efron and Hastie (2016) "Computer Age Statistic Inference"

- *algorithms*: "what statisticians do"
- *inference*: "why they do them"

Machine learning (ML) studies algorithms that generalize from experience.

- AI involves machines that can perform tasks that are characteristic of human intelligence. (John McCarthy 1956)
- ML is a subfield of computer science that "gives computers the ability to learn without being explicitly programmed". (Arthur Samuel, 1959)
- Study of algorithms that:
  - improve their performance P
  - at some task T
  - with experience E

Well-defined learning task: $< P, T, E >$. (Tom Mitchell)

A learning theory by any other name...

- *artificial intelligence*
- *statistics*
- *data mining* and *pattern recognition*

ML grew from different academic disciplines (statistics, computer science, neuroscience) with a tight connection to industry.

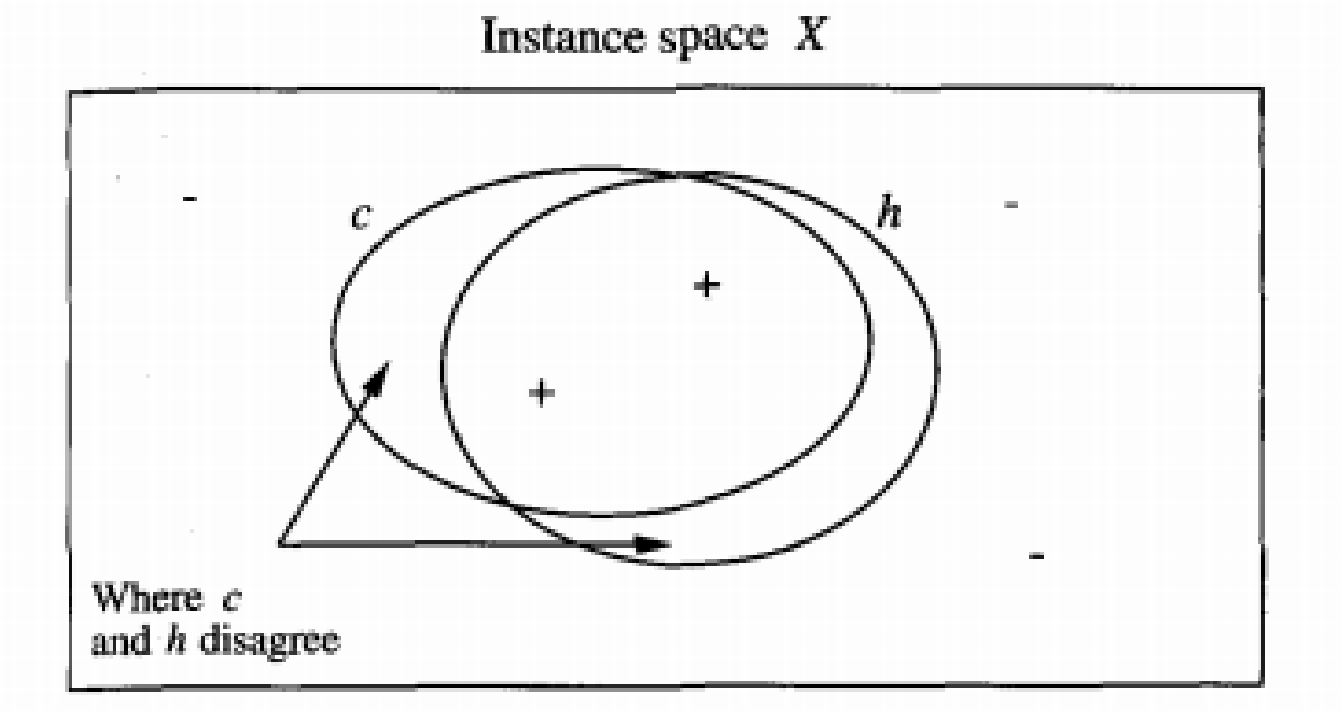
Learning Theory

Learning theory is a subfield of artificial intelligence devoted to studying the design and analysis of machine learning algorithms.

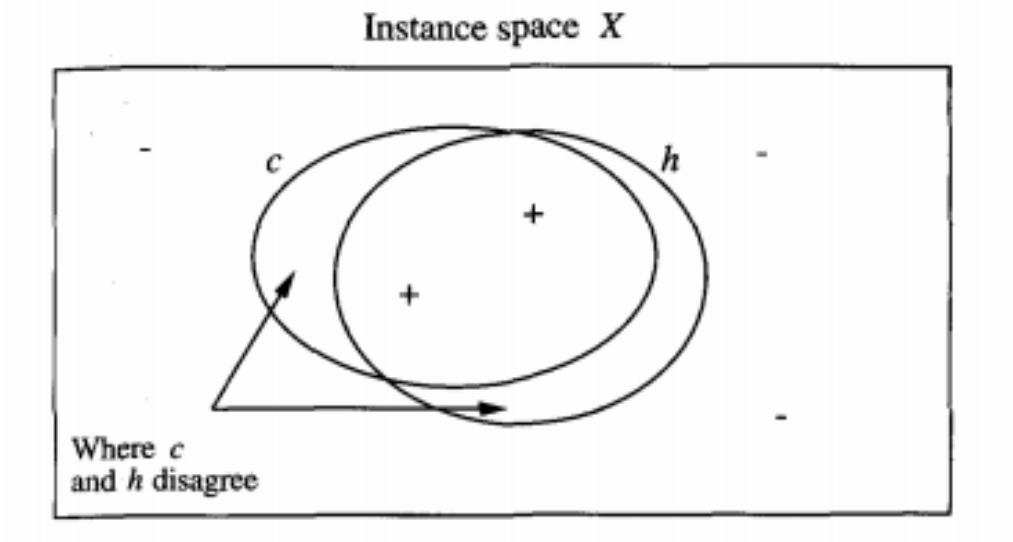
Some important ideas:

- Probably Approximately Correct (PAC) learning
- Vapnik-Chervonenkis (VC) theory
- Occam learning
- Cover's theorem

Probably Approximately Correct (Leslie Valient 1984) provides a mathematical theory to characterize selected hypotheses with high probability (Probably) to have low error (Approximately Correct).

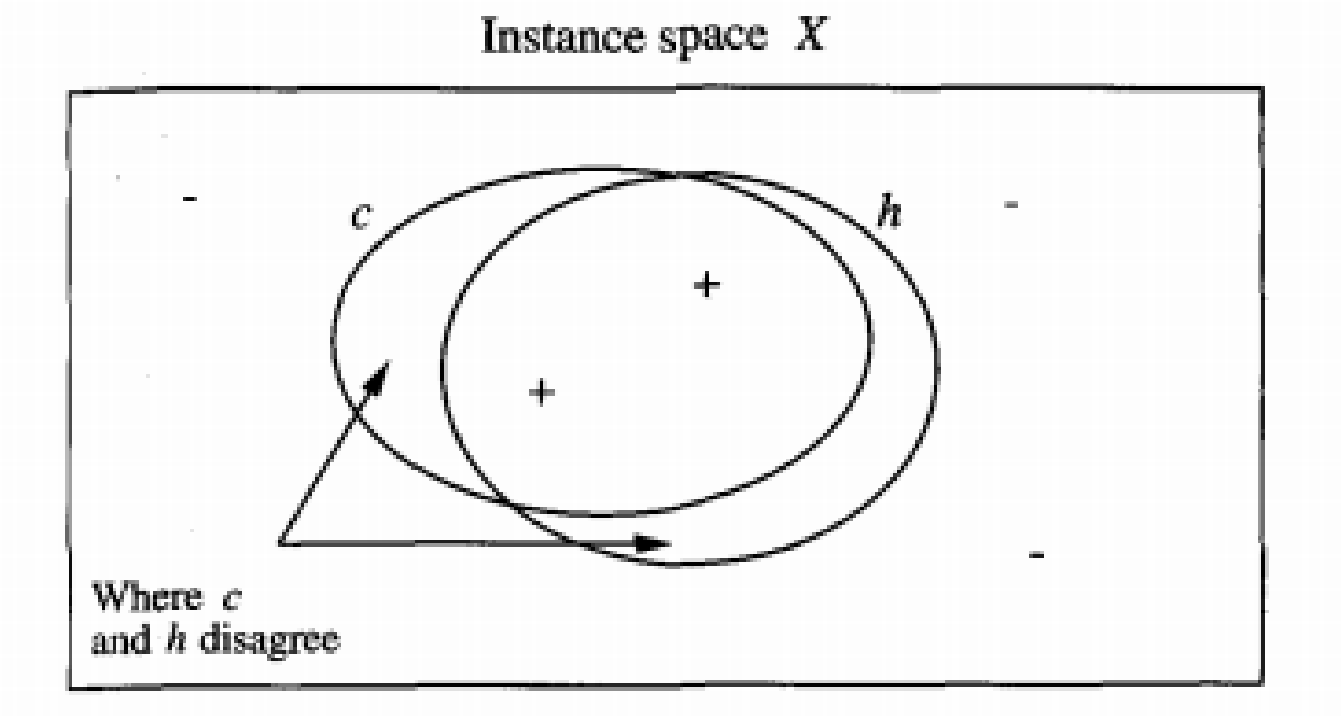


"The critical idea in PAC learning is that both statistical as well as computational phenomena are acknowledged, and both are quantified."
-Leslie Valiant



PAC learning defines the following:

- X is the instance space, and x is a specific instance
- C is the concept class in X, where c is a specific concept
- H is the class of all possible hypotheses, and h is an instance of a hypothesis
- f is the true function to be learned
- ϵ represents the level of misestimation, while δ is our confidence in the estimate



Then something is PAC learnable if there exists an algorithm A that can probably learn with sufficient accuracy in *finite time*:

$$P_D(P(h(x) \neq c(x)) < \epsilon) >= 1 - \delta$$

Running example: Fisher's "Iris flower" dataset

The *Iris flower data set* or Fisher's Iris data set is a multivariate data set introduced by Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis.



The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four *features* were measured from each sample: the length and the width of the sepals and petals, in centimetres.

Images courtesy of wikipedia.

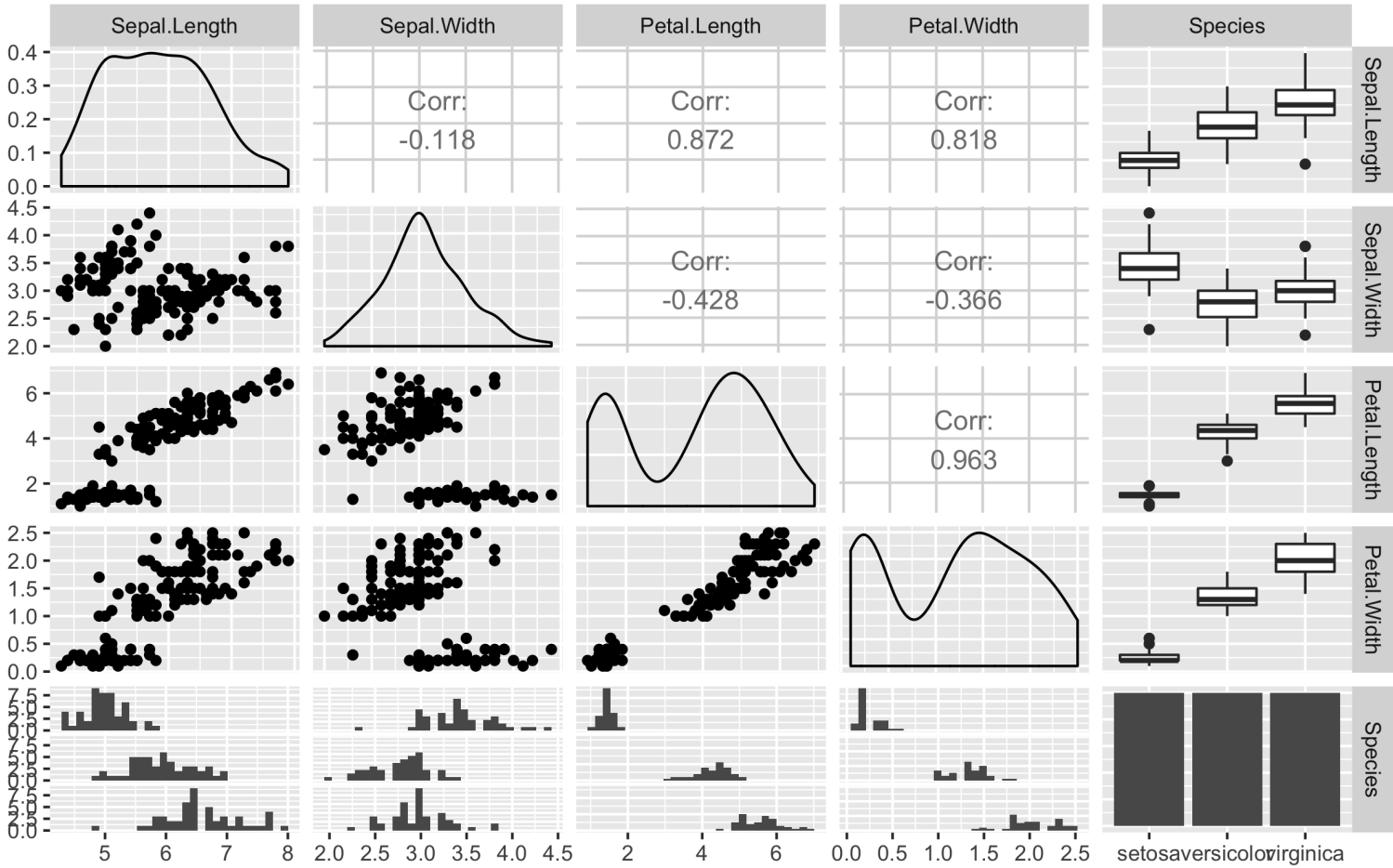
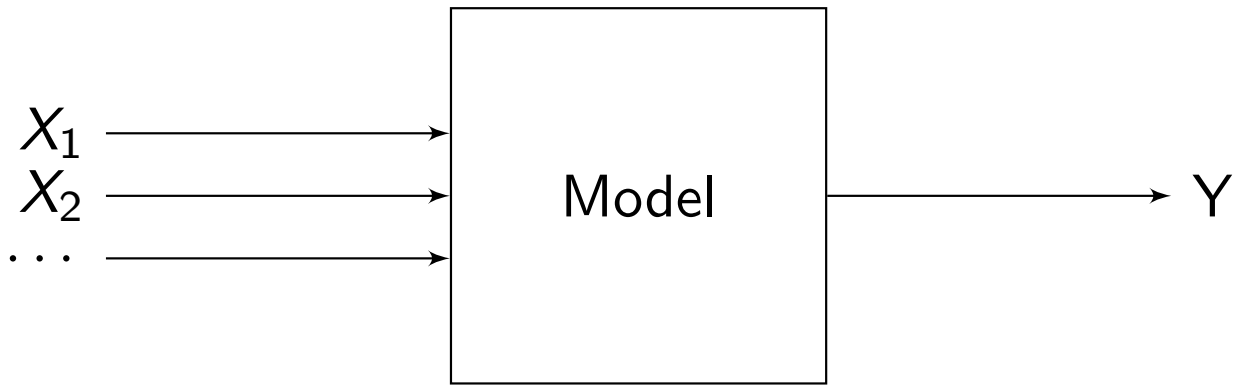


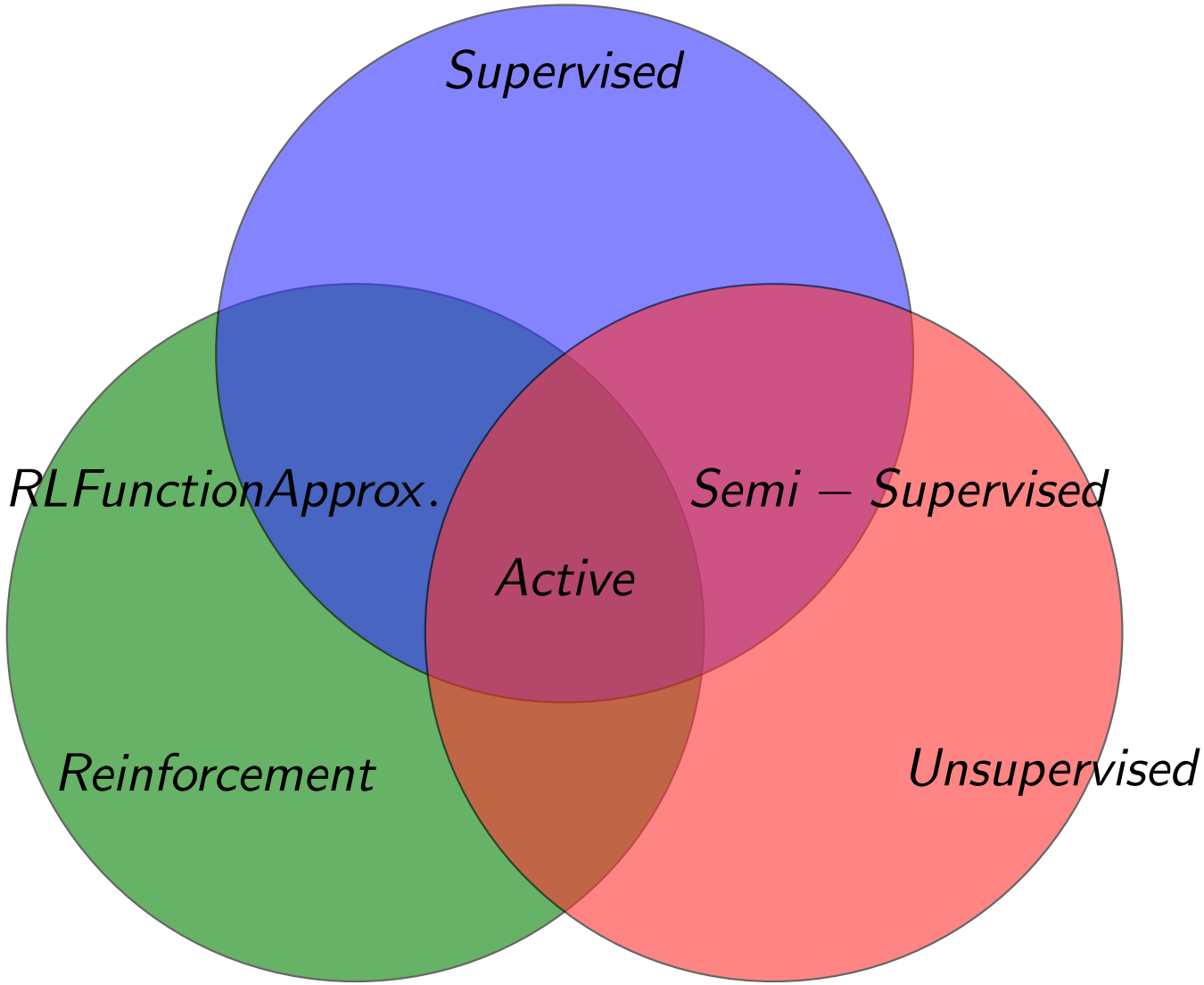
Figure : Scatterplot matrix of iris dataset. Flower dimensions are input features, while the species is usually a target variable in classification problems.

How machine learning *works*


X_1
X_2
...
Model
Y

Machine learning includes several different kinds of *models*:

- Supervised learning: learn a function by fitting labeled targets and input features.
- Unsupervised learning: learn a function by fitting to input features (without labeled targets).
- Reinforcement learning: learn a policy based on receiving rewards for taking actions in states.


Supervised
RLFunctionApprox.
Semi − Supervised
Active
Reinforcement
Unsupervised

The decision for which model is a factor of a number of issues:

- Is there a sequential aspect to my data? Yes $\longrightarrow$ Reinformcement Learning (or other time series model)
- Do I have labelled data? Yes $\longrightarrow$ Supervised Learning, else Unsupervised Learning
  - Is target variable continuous? Yes $\longrightarrow$ Regression, else Classification

There are a number of other considerations:

- Is there enough data?
- Is target "learnable"?
- Is the relationship linear or non-linear?
- Can the data fit in memory?
- Do I care more about accuracy or speed?

Machine learning in practice is more than *models*; it includes a set of tools to make predictions robust.

- Feature engineering
- Regularization
- Feature selection
- Hyperparameter tuning
- Model selection
- Ensembling

Automated Machine Learning (AutoML) is a subfield of machine learning that aims to completely remove human decision making from the machine learning pipeline.

Some examples:

- Commercial: DataRobot, Ayasdi, SparkBeyond
- Open Source: Auto-Sklearn, Tpot, Caret

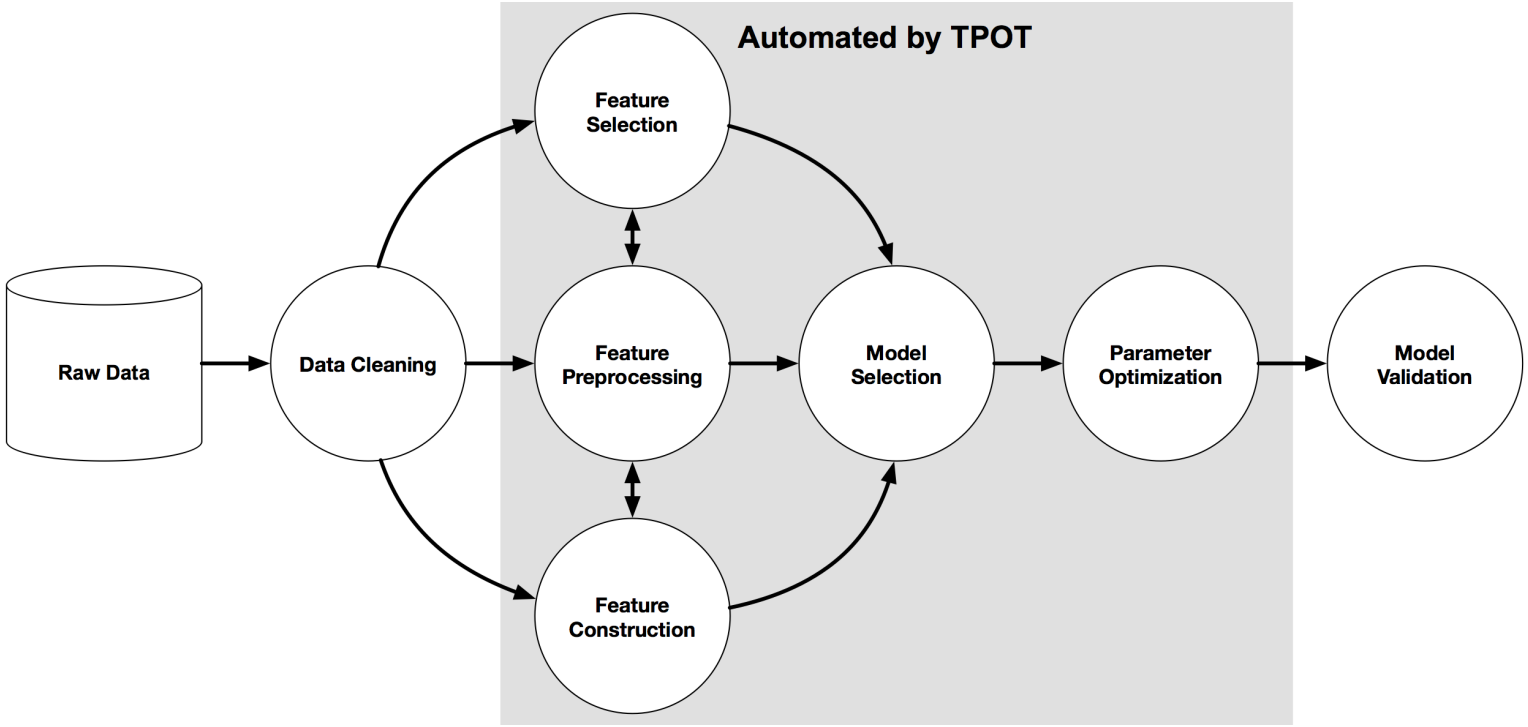


Figure : An example of a machine learning *pipeline* from tpot.

We are interested in how our models will perform on unseen data.

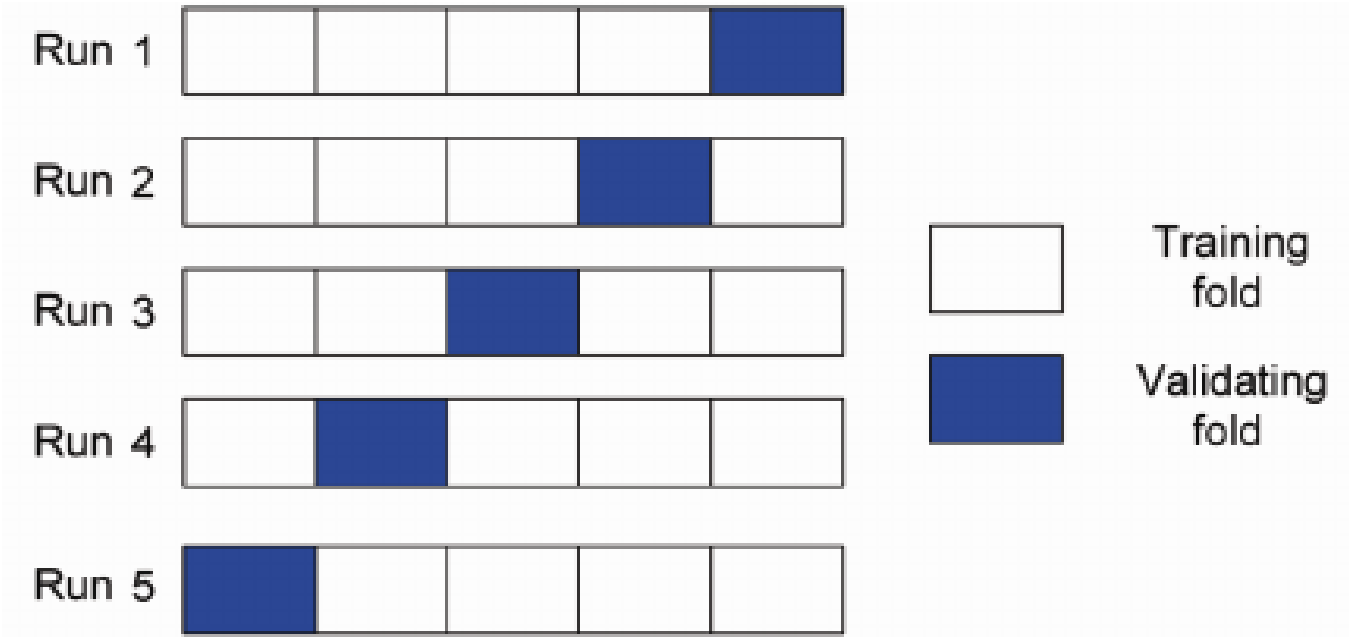


Figure : *Cross-validation* divides a dataset into folds, and runs through by comparing the *training* and *validation* performance.

A large part of machine learning is aiming to solve the *bias-variance trade-off*.

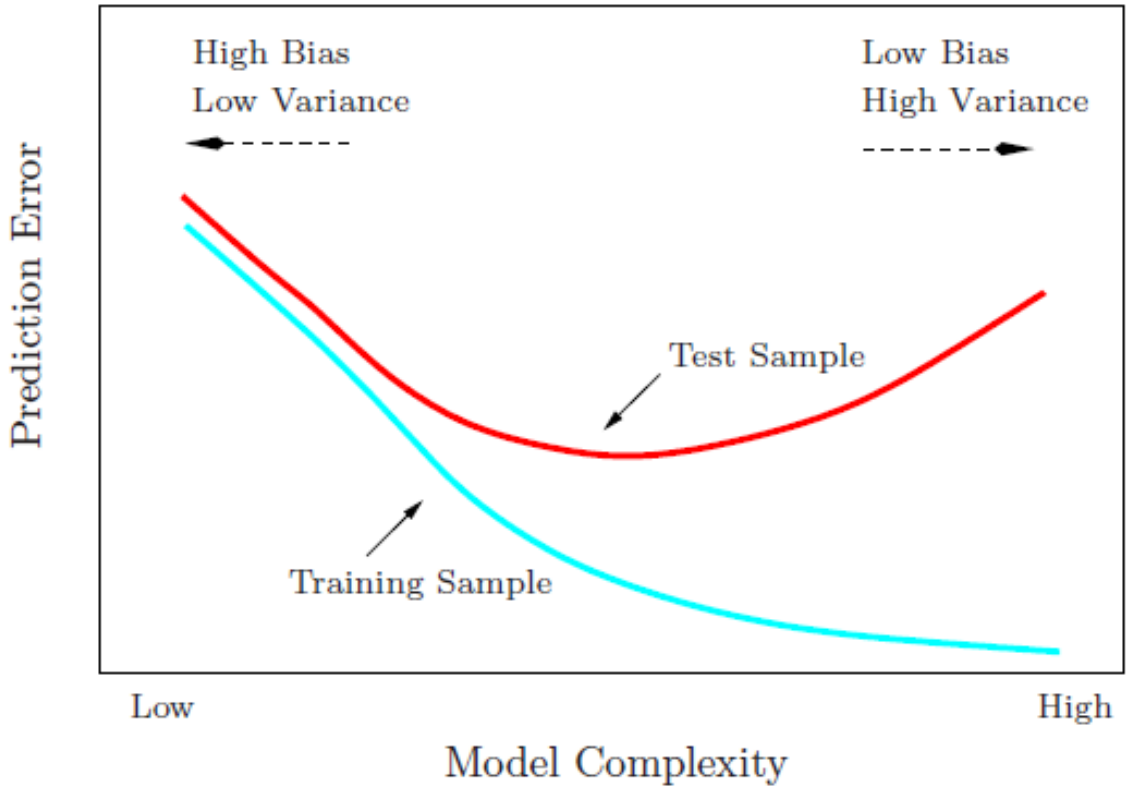


Figure : Increasing the model complexity can improve performance (lower bias), but eventually will cause over-fitting (higher variance).

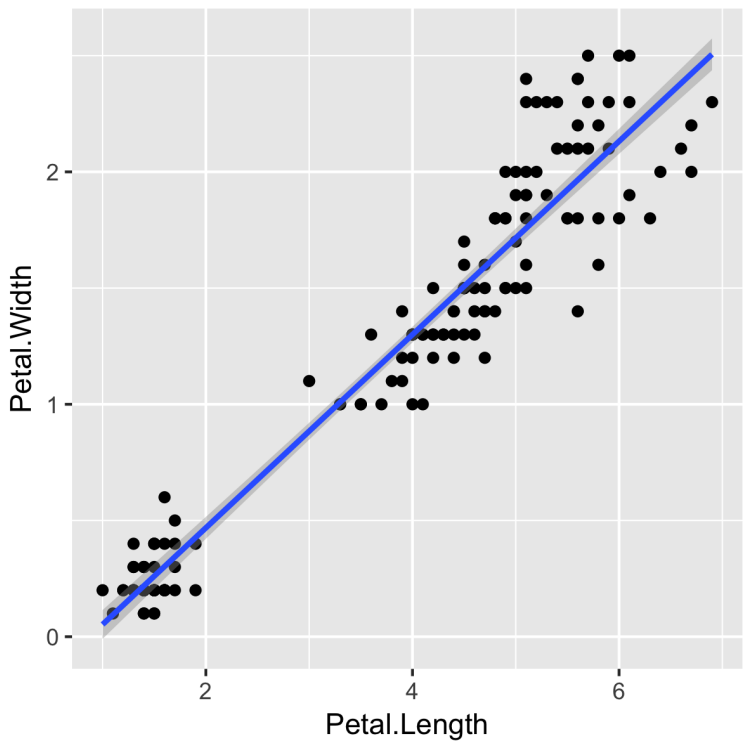
Linear *regression* predicts a continuous variable from a number of inputs:

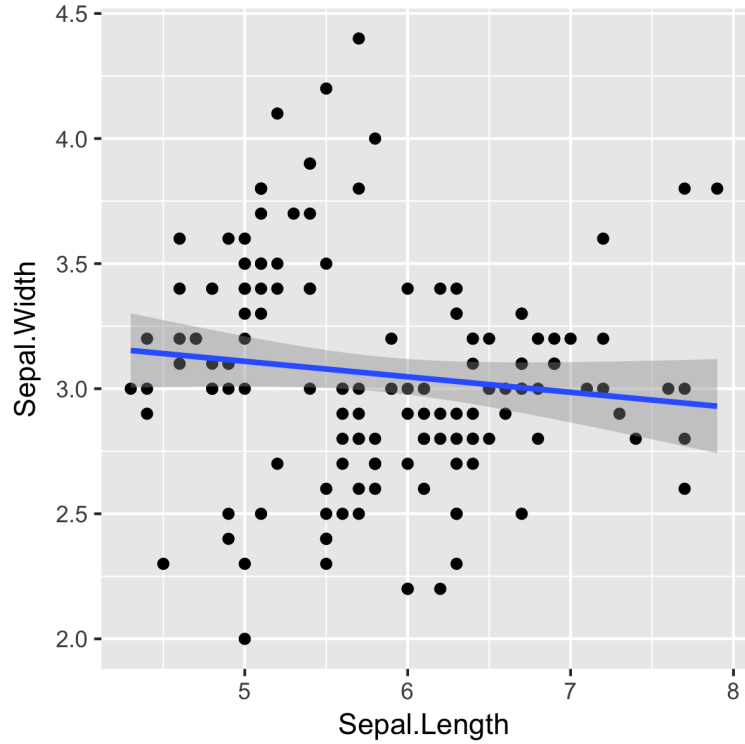
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \epsilon$$

The most popular method for regression is OLS, commonly solved using the matrix formulation:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

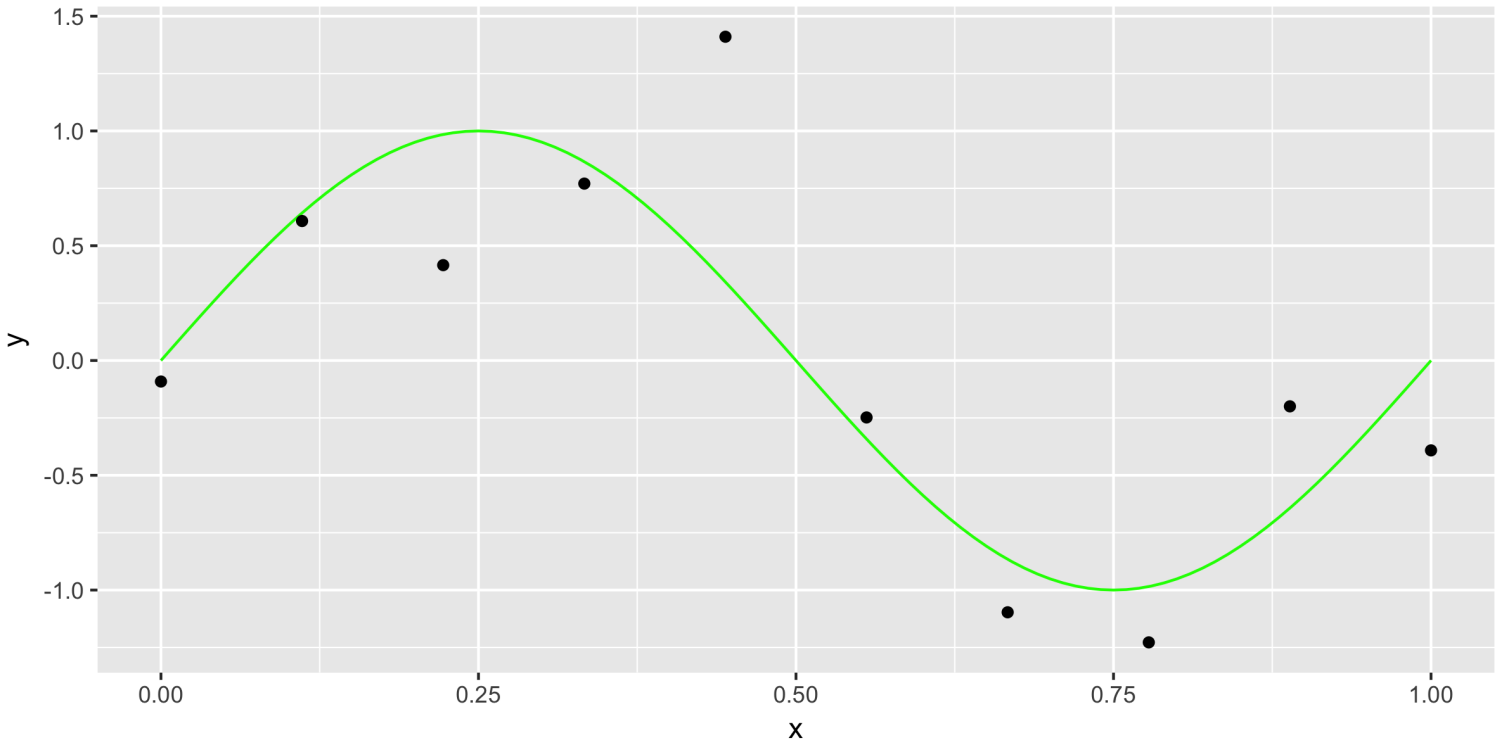
This shows that there is a strong linear relationship between petal dimensions, but not between sepal dimensions in the iris dataset.





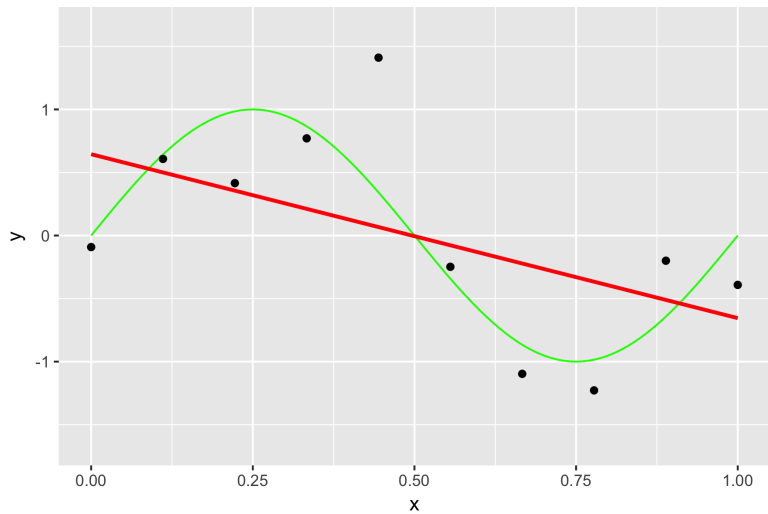
As an example, suppose that we have a simple function, that we observe with some noise ϵ:

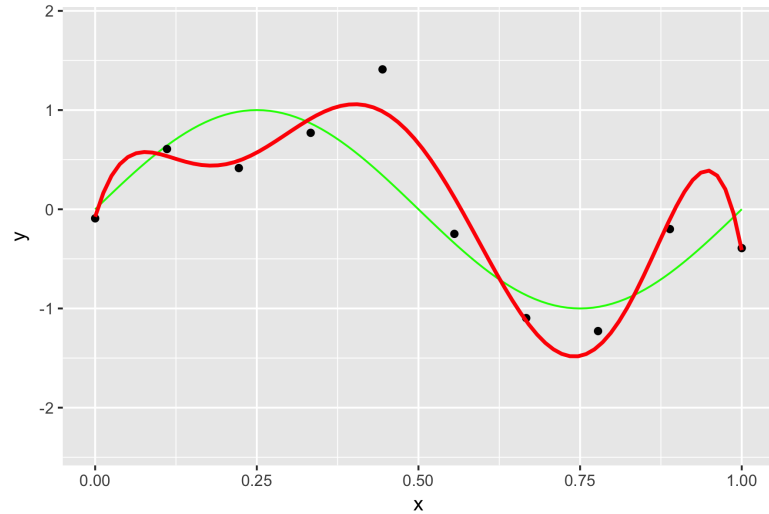
$$y = sin(2\pi x) + \epsilon$$

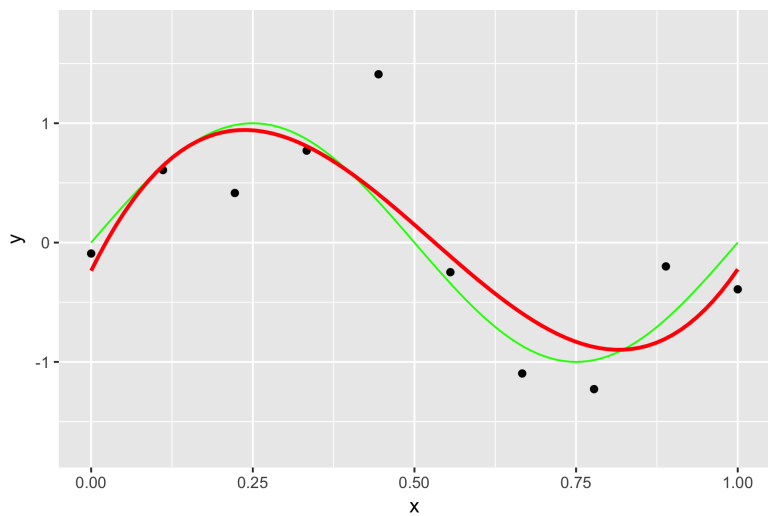


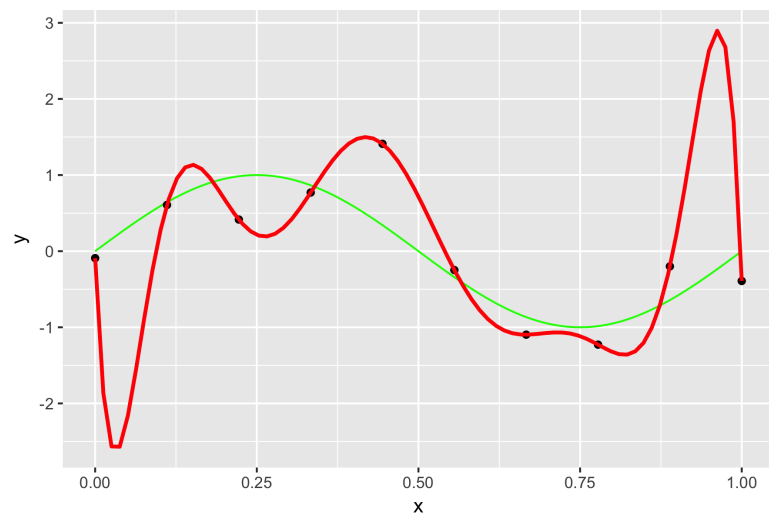
Without knowing the real function, we can estimate it by fitting polynomials:

$$\hat{y} = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \epsilon$$

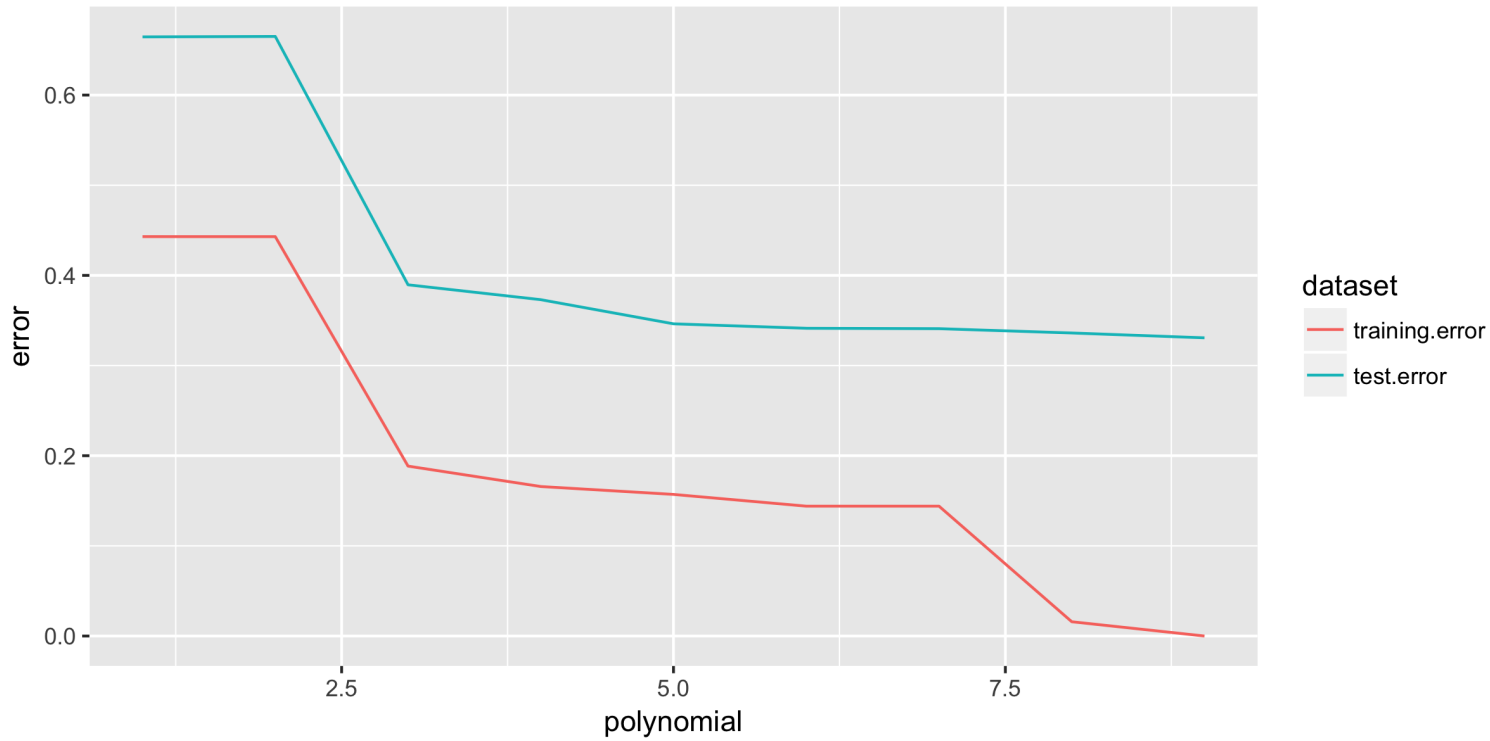




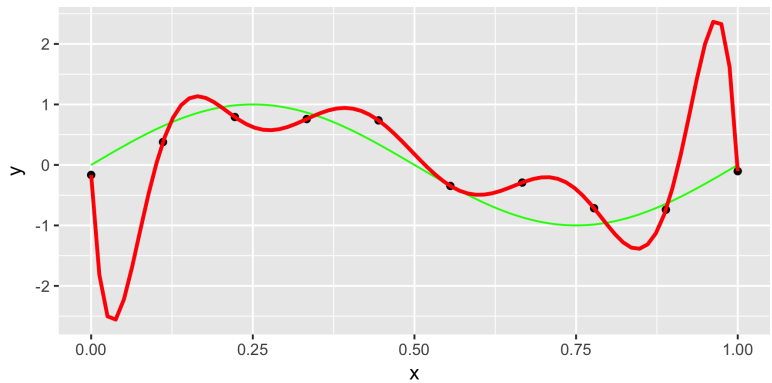


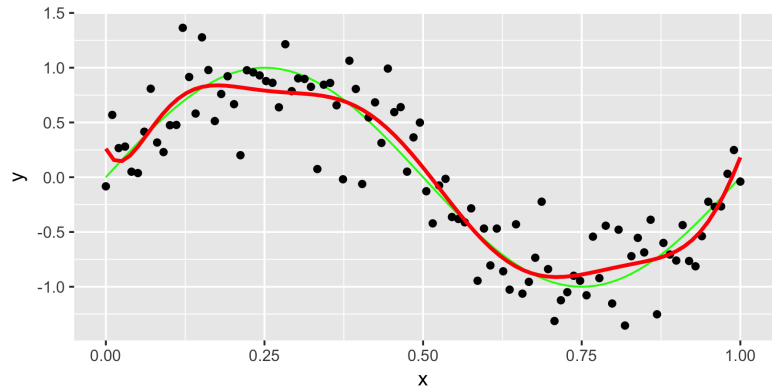


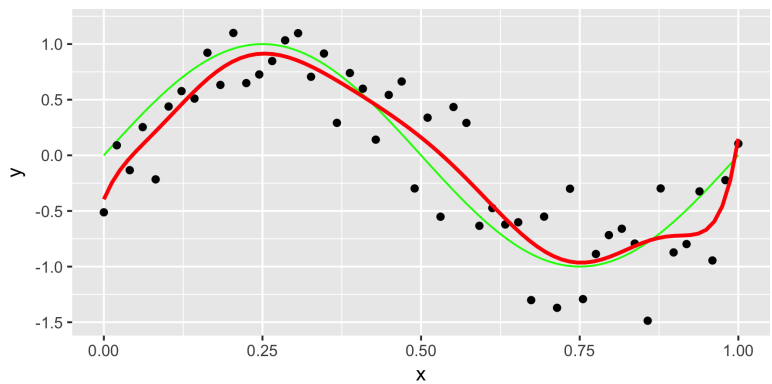
This shows the error rates for different numbers of polynomials.

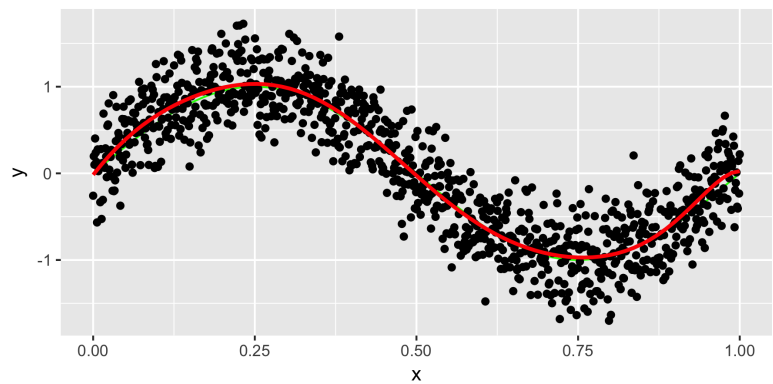


One "simple" way to solve for overfitting is by getting more data, which is one benefit of Big Data.









Another way to solve for overfitting is to use *regularization*, which is a method that penalizes features, and thus can be used for variable or model selection.

Lasso regression uses a l_1 norm:

$$RSS(\beta) = \sum_{i=0}^{n} (y_i - x_i^T \beta)^2 + \lambda |\beta|$$

Ridge regression uses an l_2 norm:

$$RSS(\beta) = \sum_{i=0}^{n} (y_i - x_i^T \beta)^2 + \lambda |\beta|^2$$

These function shrink the coefficients toward zero. Ridge never reaches zero.

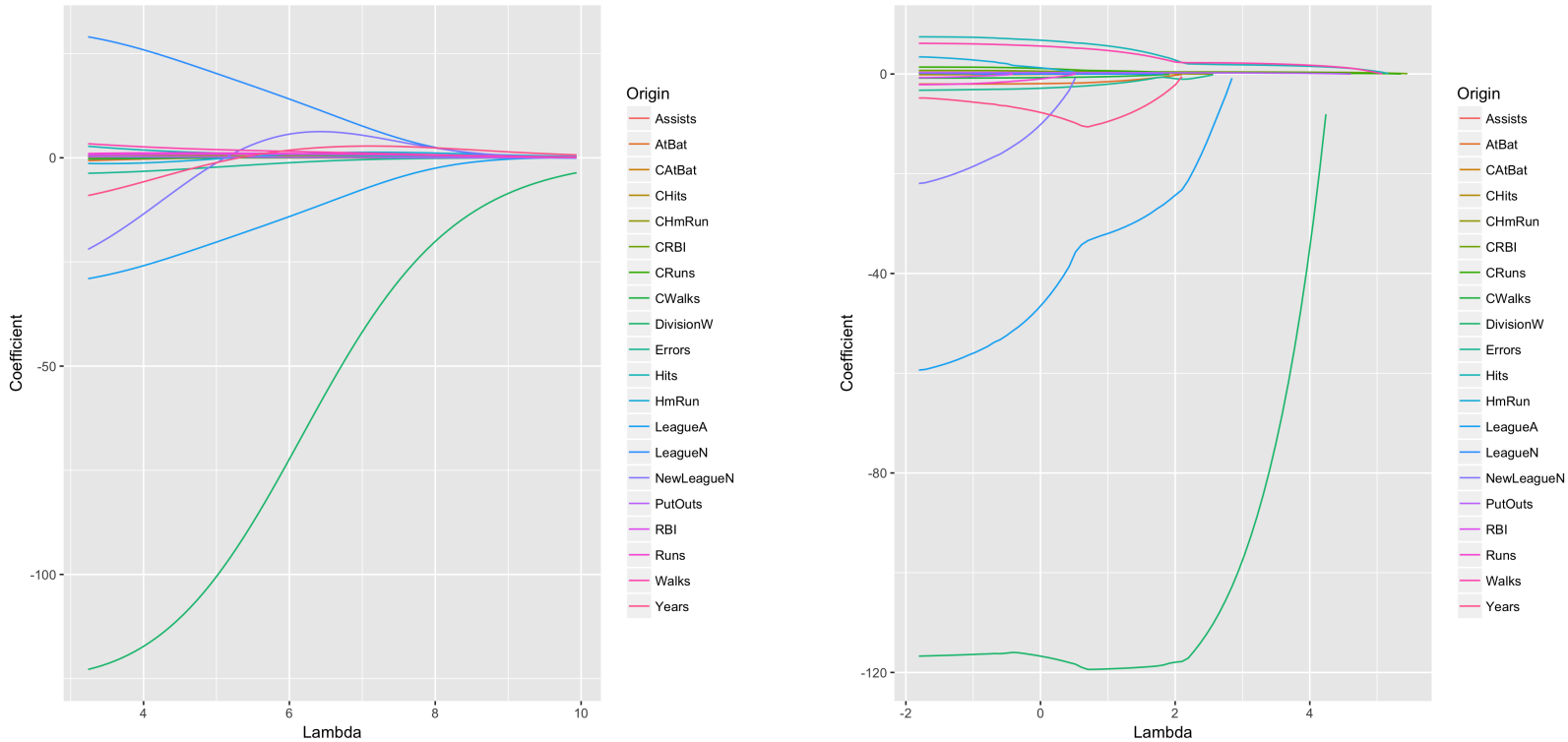
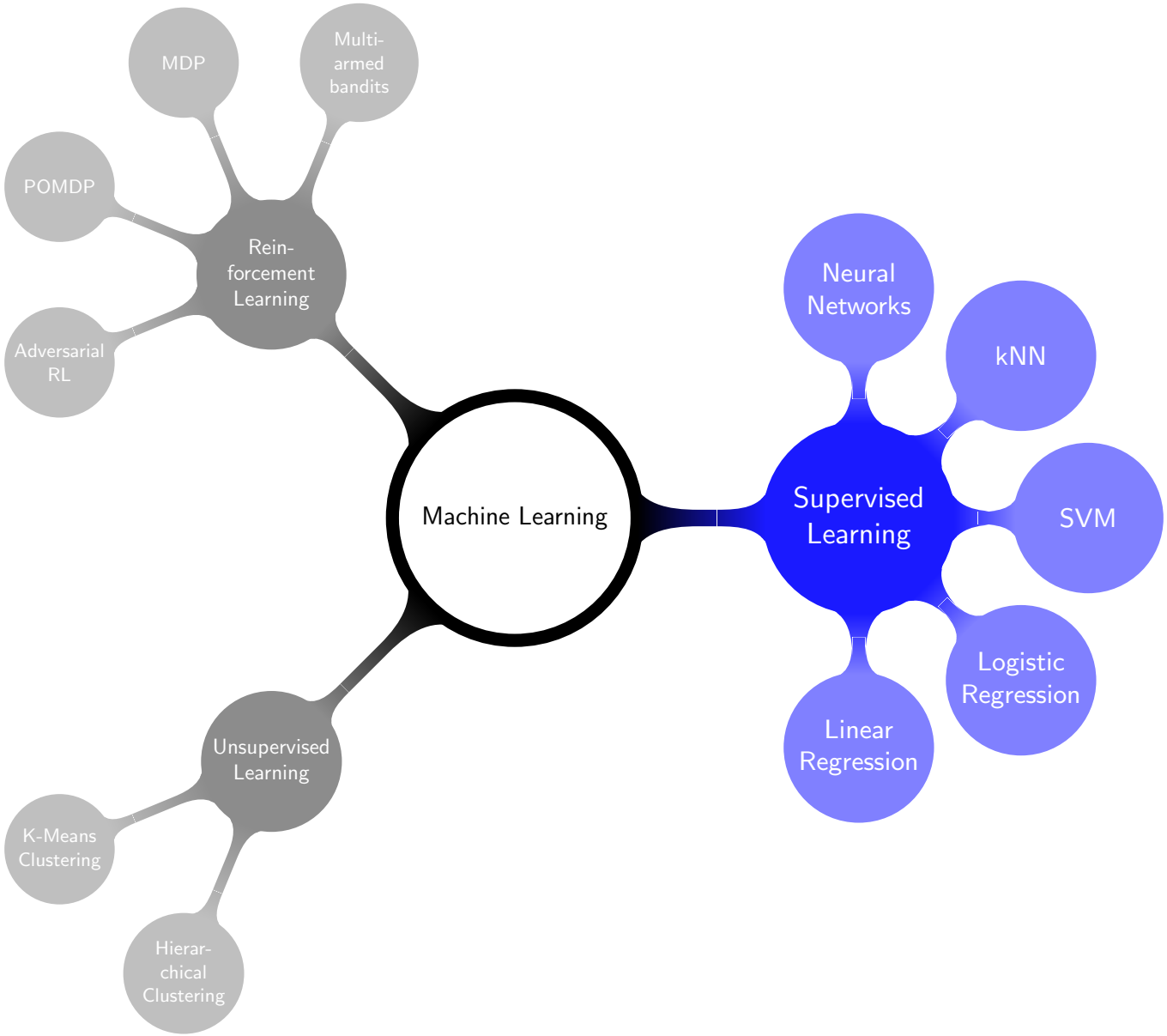


Figure : Path for coefficients as lambda is increased in (a) ridge and (b) lasso. Data predicting salaries for MLB players from 1986/87 from Games/Witten/Hastie/Tibshirani (2013).

We can generalize most machine learning problems down to a loss function:

$$J(w) = \sum_i L(m_i(w)) + \lambda R(w)$$


Machine Learning
Supervised Learning
Neural Networks
kNN
SVM
Logistic Regression
Linear Regression
Rein-forcement Learning
MDP
Multi-armed bandits
POMDP
Adversarial RL
Unsupervised Learning
K-Means Clustering
Hierar-chical Clustering

Supervised Learning

Supervised learning involves learning a model given labeled examples and input features.

$$y = f(X)$$

Examples of models:

- Linear regression
- Logistic regression
- Support vector machines
- Neural network

Classification predicts a discrete variable from a number of inputs.

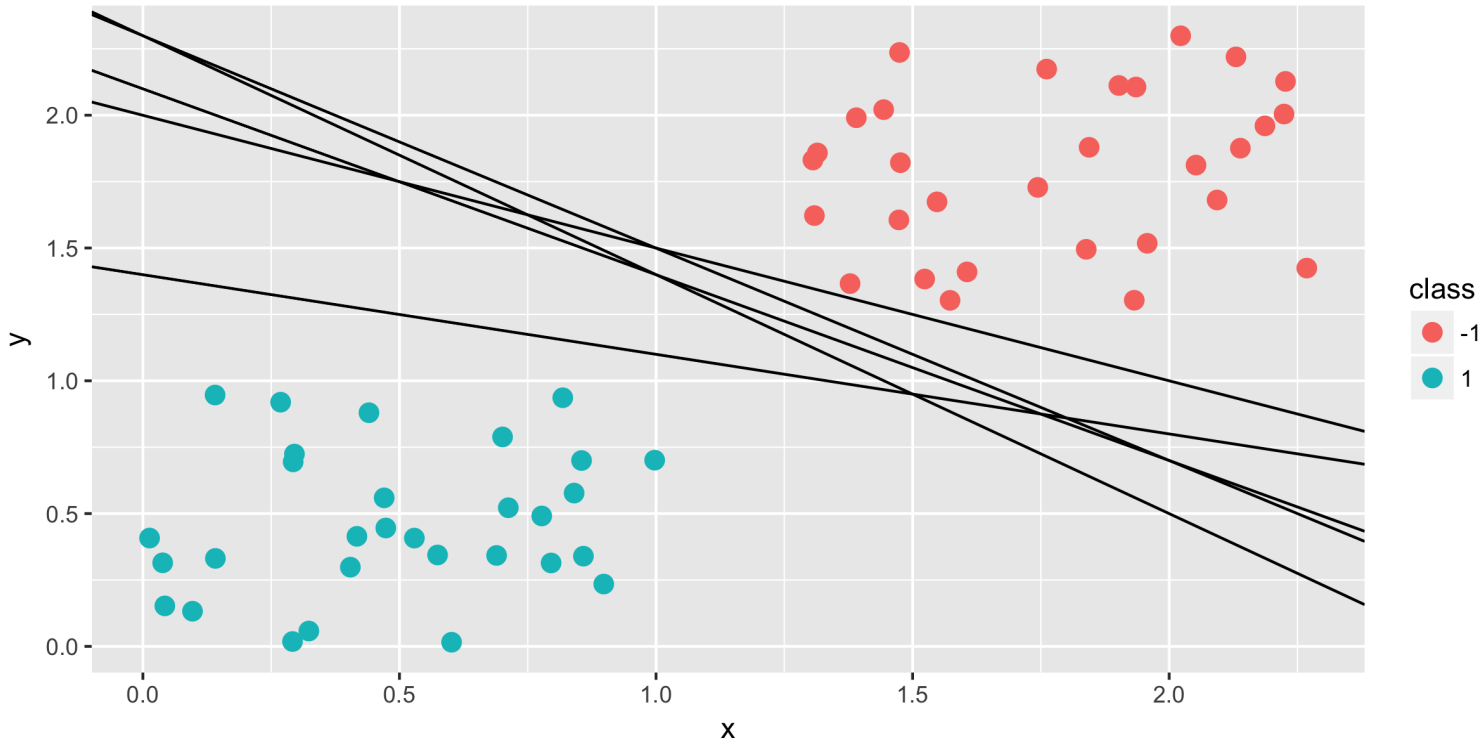
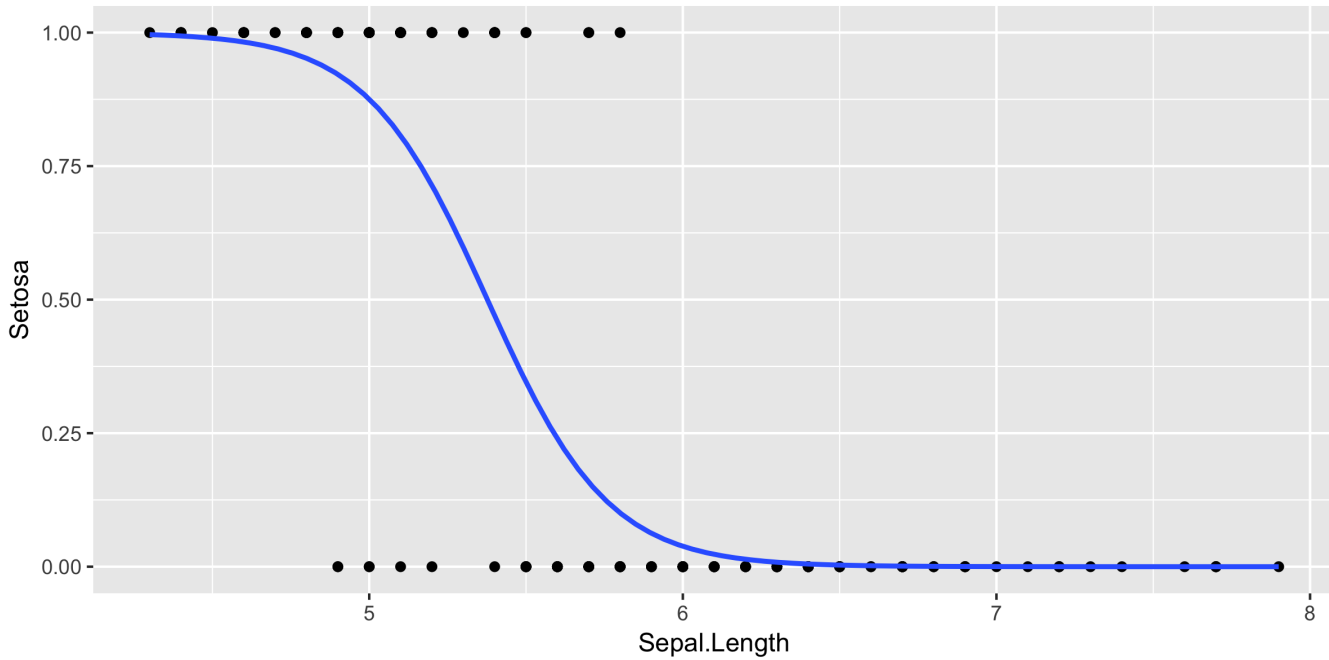


Figure : Taking two classes of data, we can separate them using many different lines. Classifiers will try to minimize a particular *loss function*.

One common method for classification is *logistic regression*, which computes a probability of being in one class or another by using a sigmoid function:

$$p(x) = \frac{1}{1 + e^{-x}}$$



K-Nearest Neighbors (KNN) makes a comparison to the k nearest neighbors values.

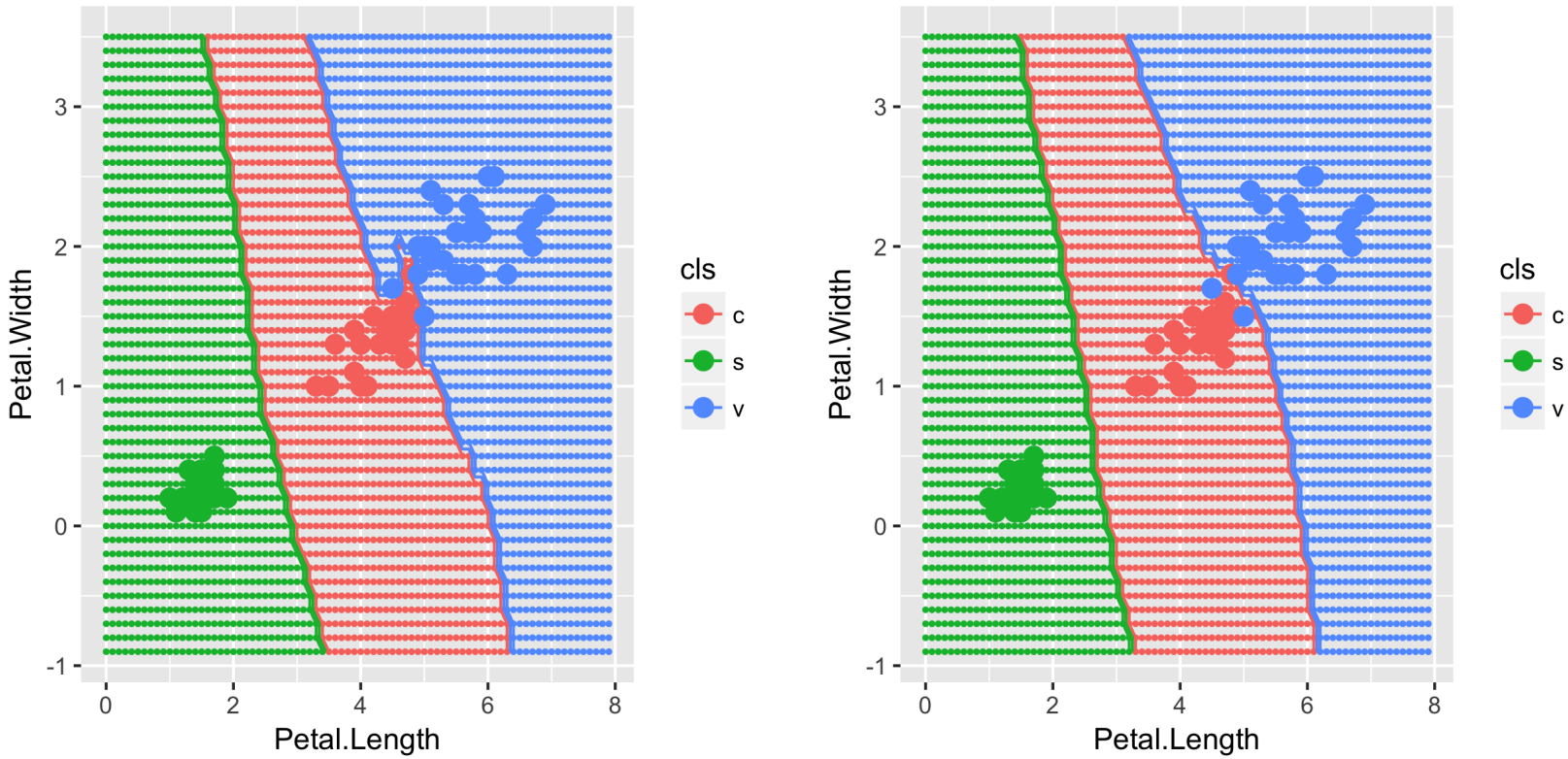


Figure : K nearest neighbors to classify the Iris dataset with (a) k = 1 and (b) k = 3.

Support vector machines (SVM) find a linear decision surface ("hyperplane") that can separate classes and that has the largest distance between support vectors.

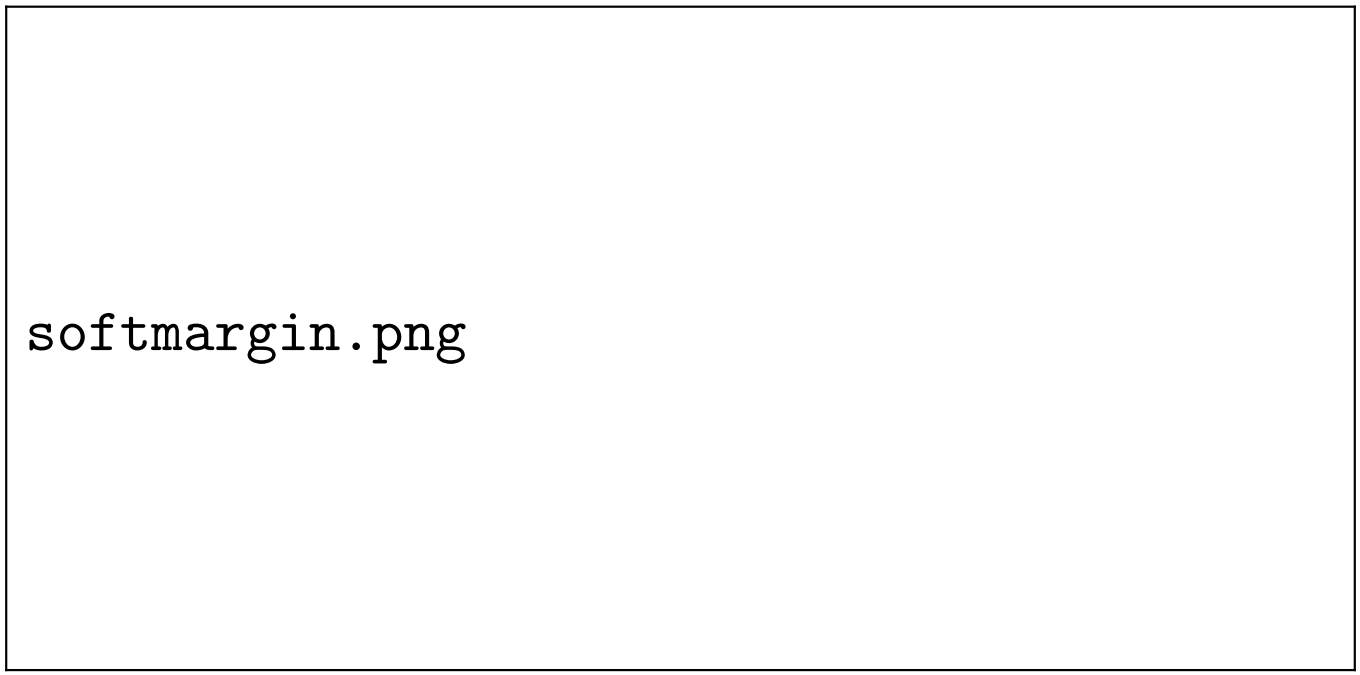


Figure : Find the value that minimizes the distance from the hyperplane, including any additional "slack" if data isn't separable.

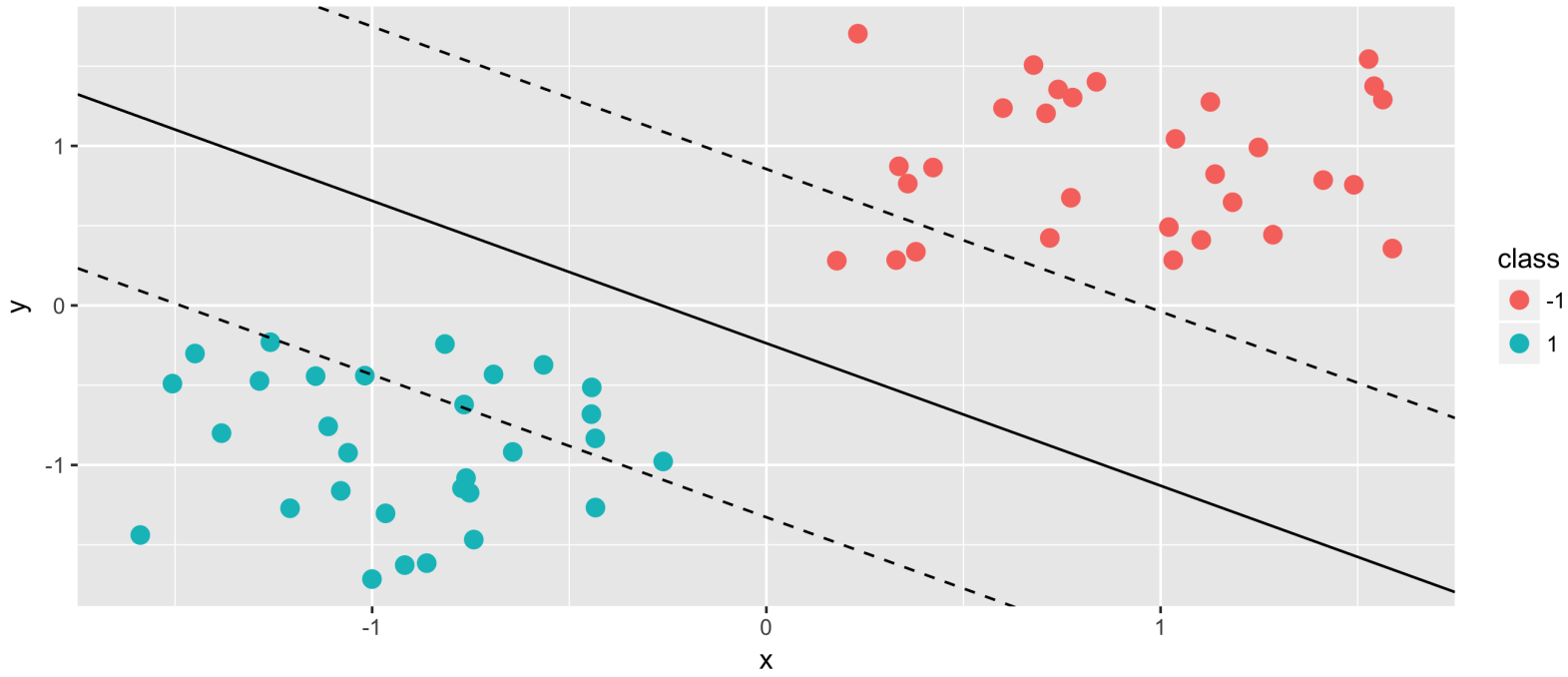


Figure : Support vector machine with decision boundaries.

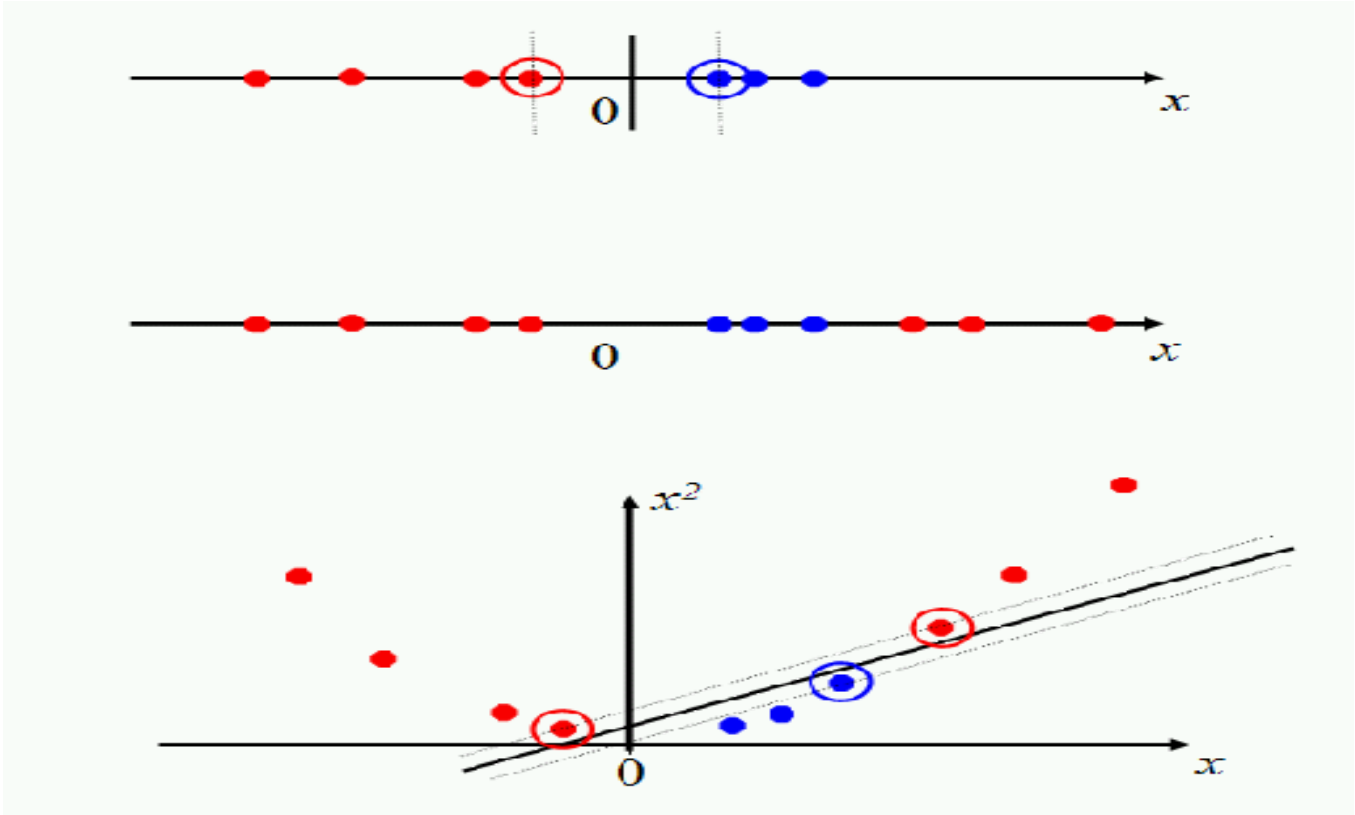


Figure : Kernels transform data into higher dimensions to allow linear decision boundaries.

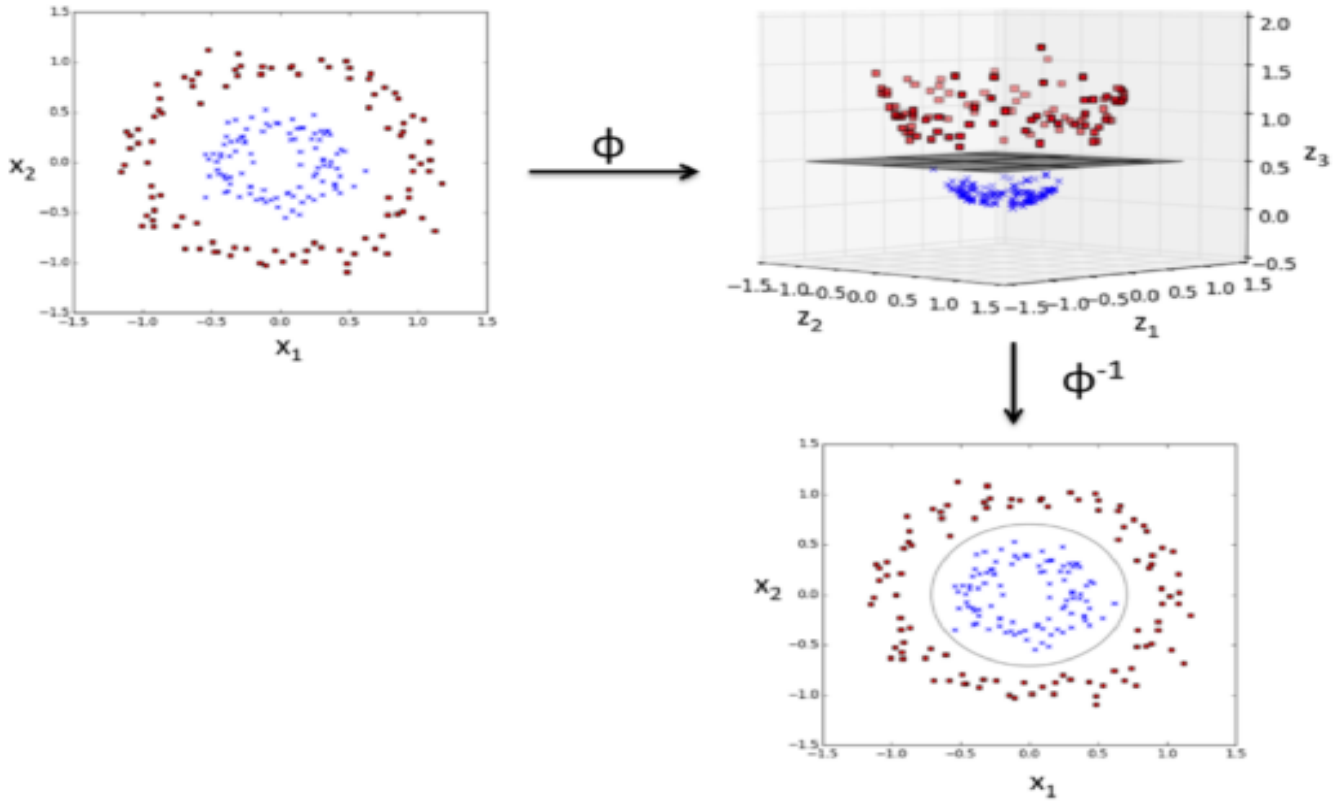


Figure : Kernels transform data into higher dimensions to allow linear decision boundaries, example in 3-dimensions.

Artificial neural networks (ANN) are learning models that were directly inspired by the structure of biological neural networks.

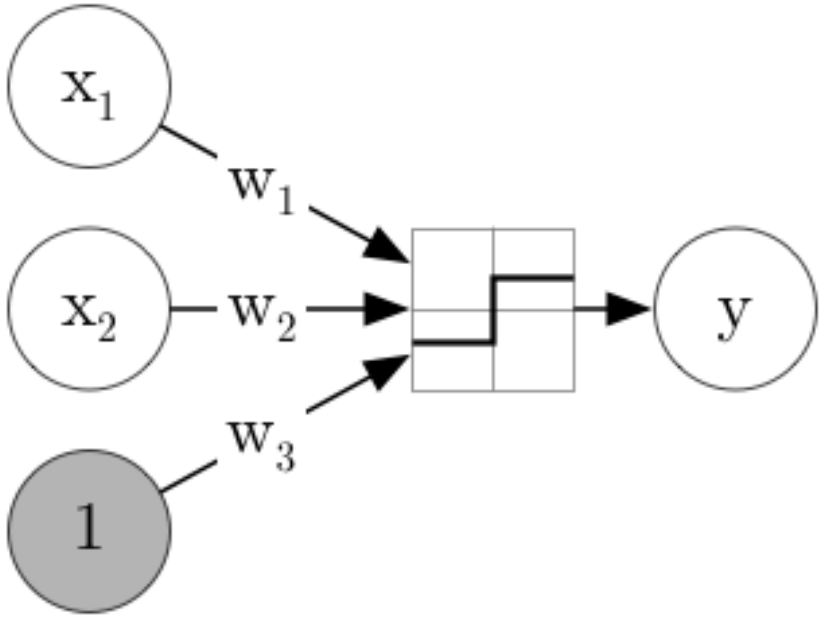


Figure : A perceptron takes inputs, applies weights, and determines the output based on an activation function (such as a sigmoid).

Image source: @jaschaephraim

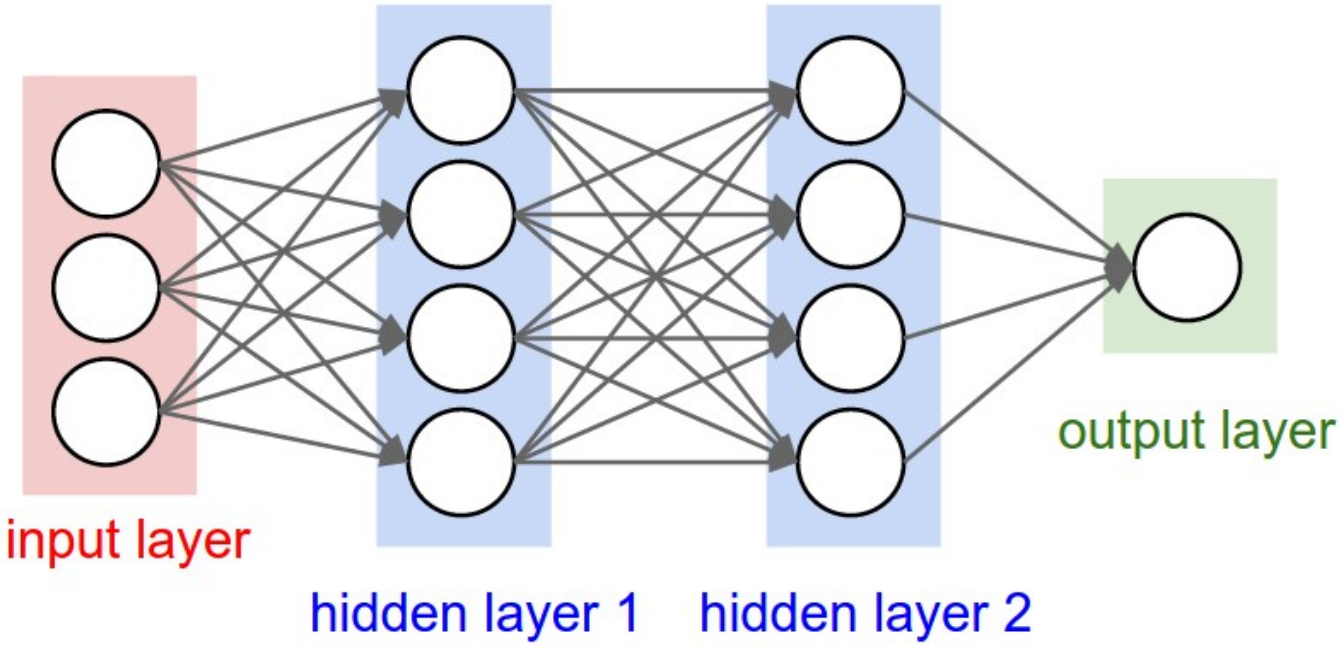


Figure : Multiple layers can be connected together.

Multi-layer neural networks can fit complex functions:

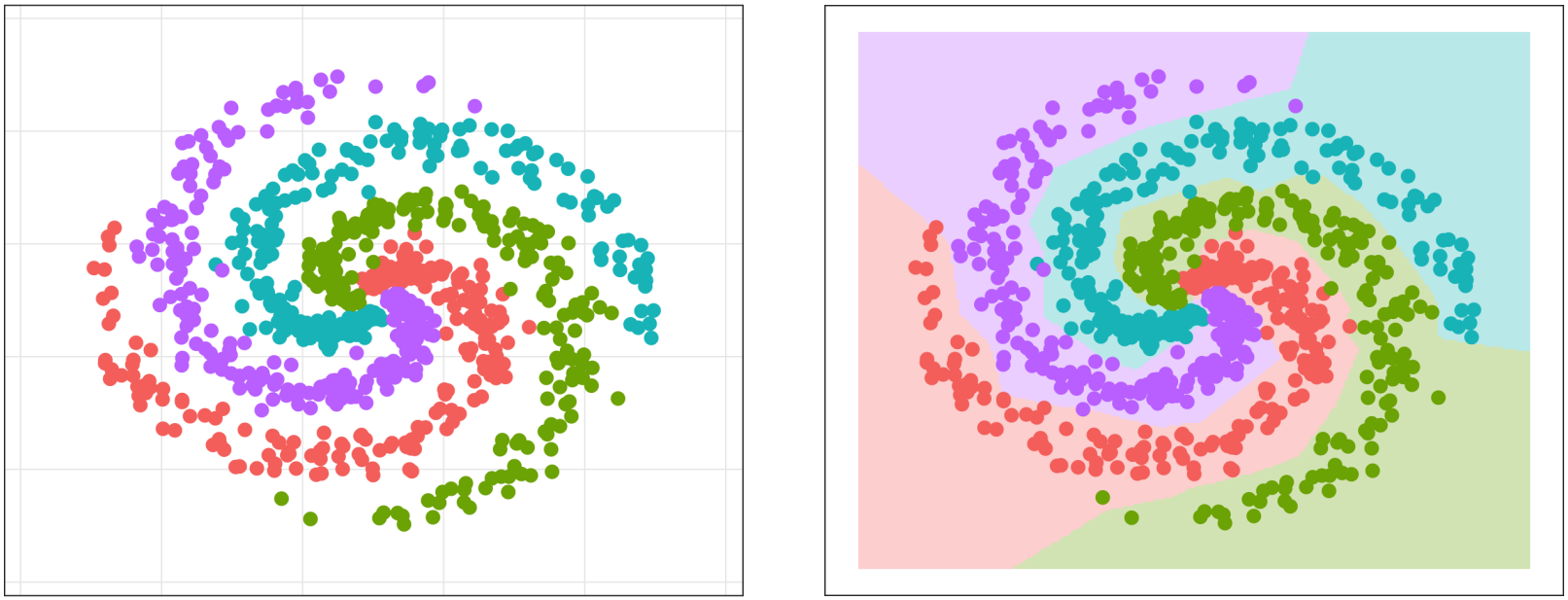
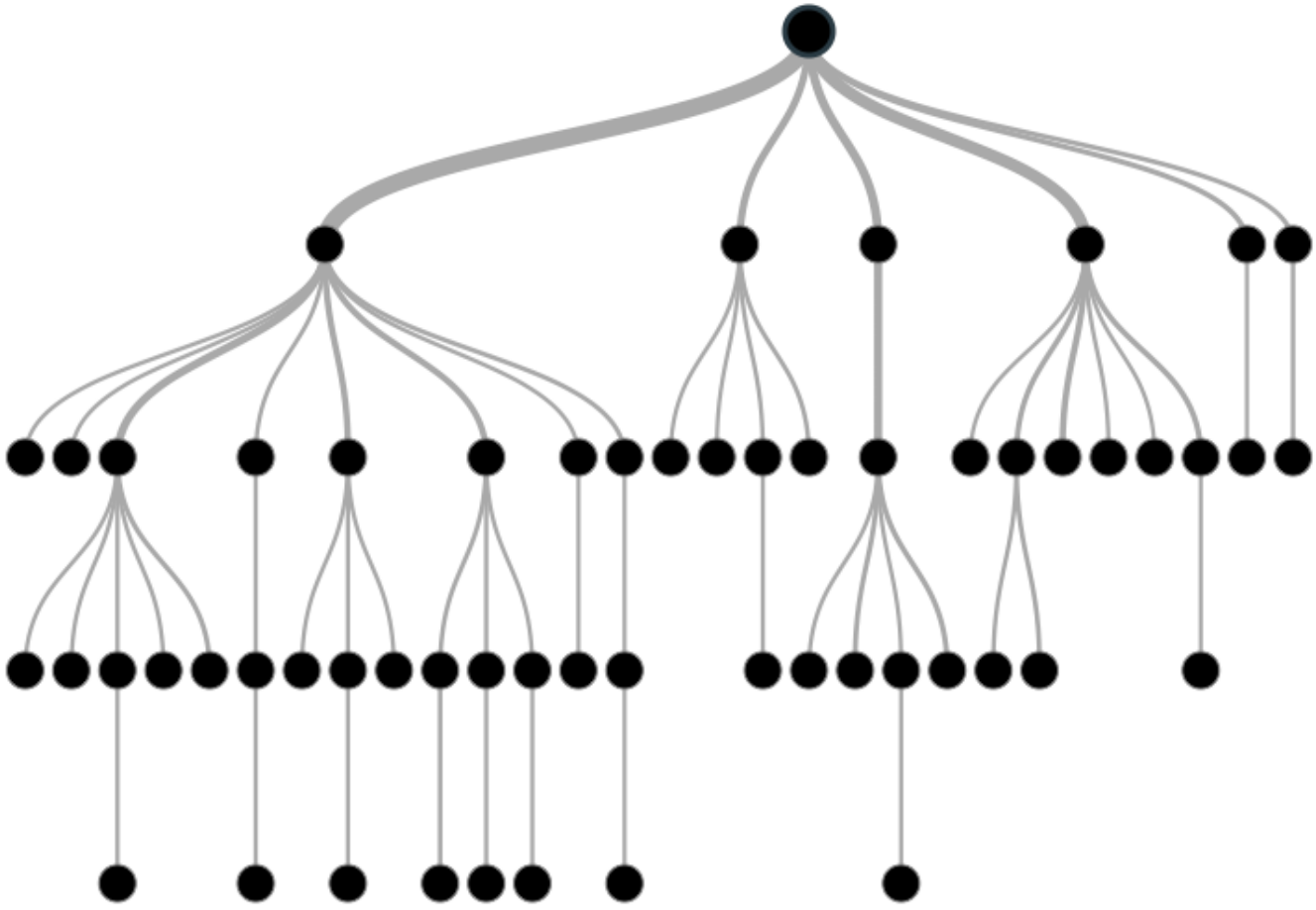


Figure : Spiral dataset with 4 different classes. The shaded region represents the neural network's predictions.

Trees "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining". (Hastie et al., ESL)



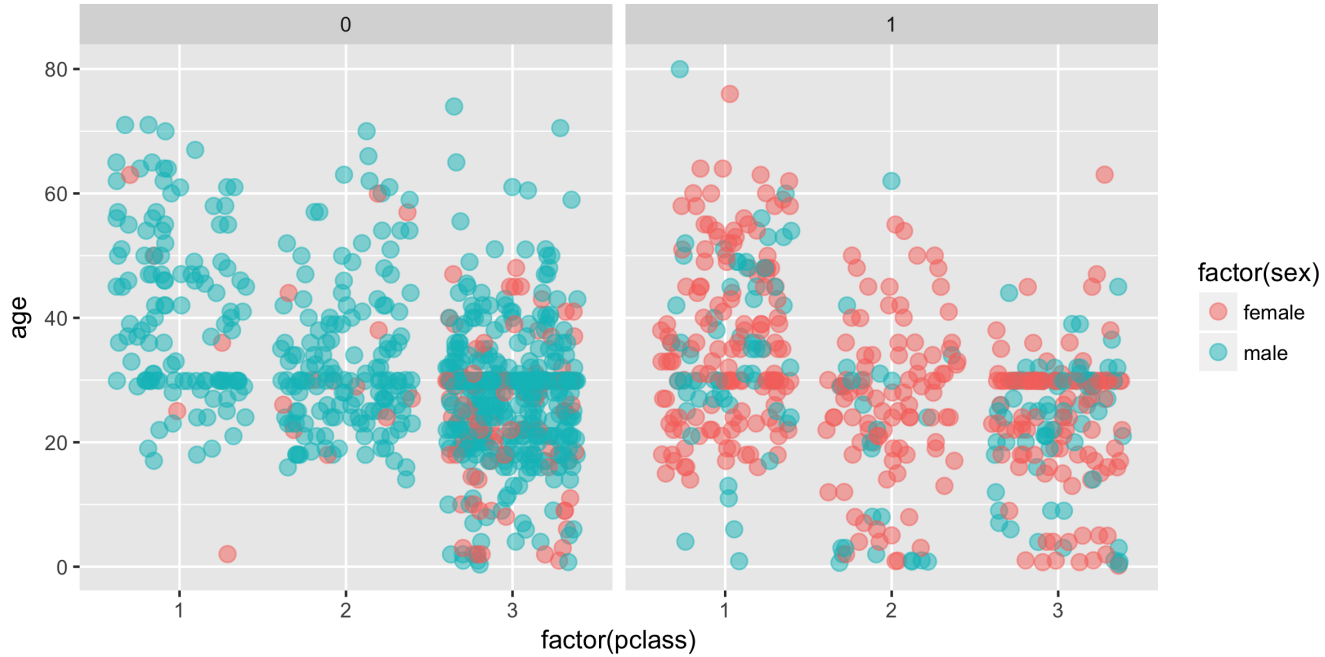


Figure : Titanic data from Kaggle competition shows a pattern predicting who survived: more likely to be female, wealthy, and/or young. Can we model this?

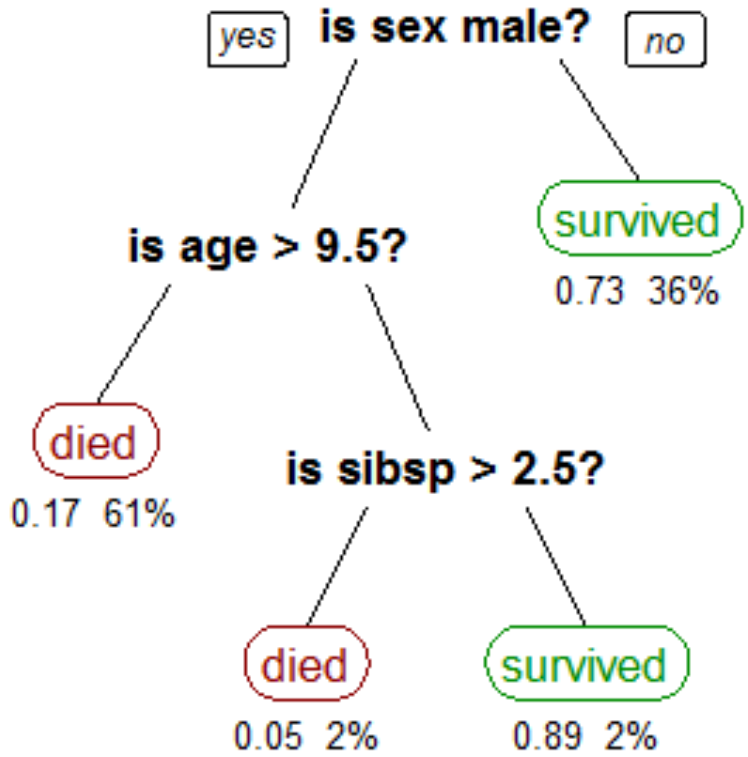
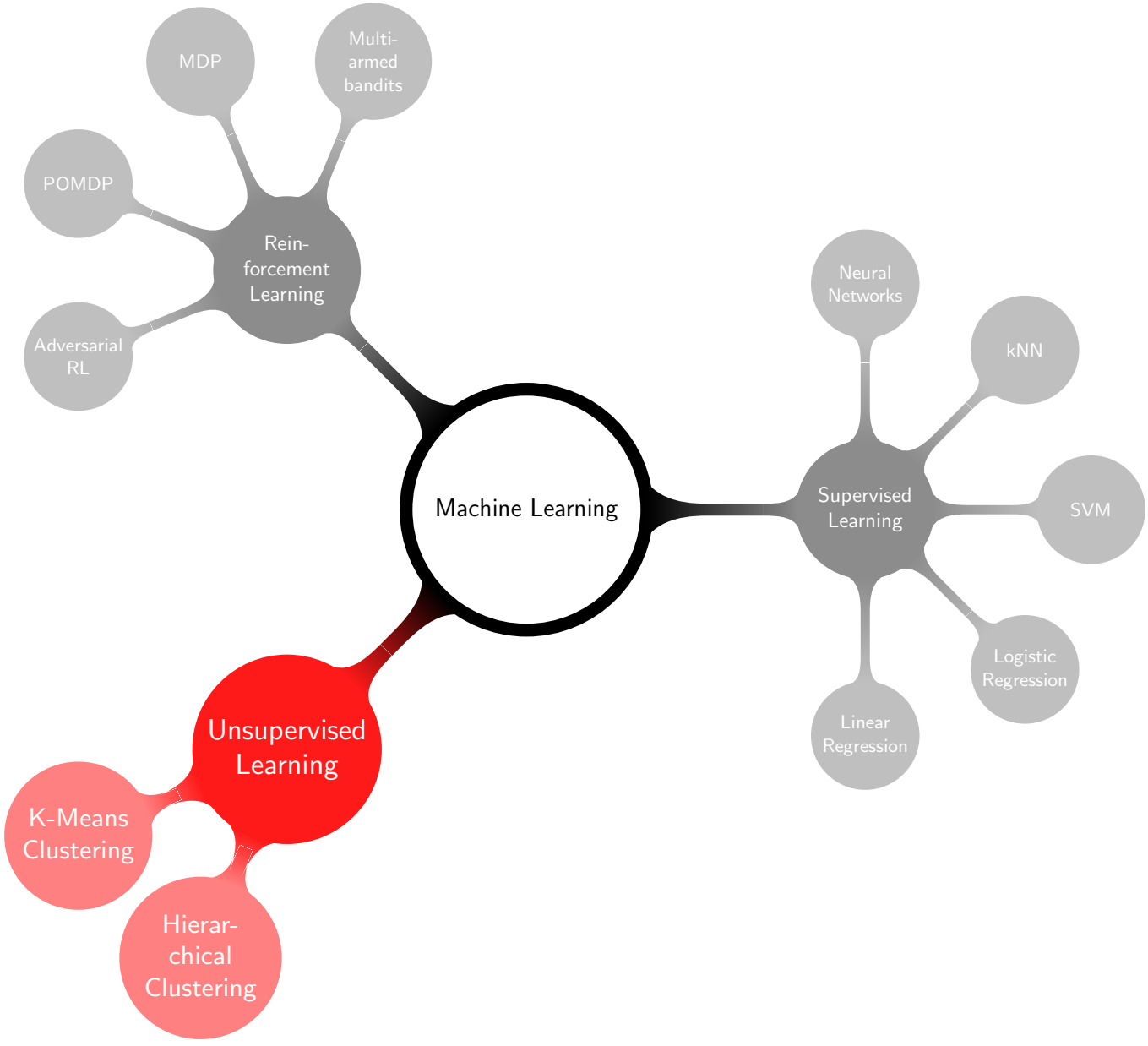


Figure : A decision tree that predicts who survived the Titanic, constructed using CART.


Machine Learning
Rein-
forcement
Learning
MDP
Multi-
armed
bandits
POMDP
Adversarial
RL
Supervised
Learning
Neural
Networks
kNN
SVM
Logistic
Regression
Linear
Regression
Unsupervised
Learning
K-Means
Clustering
Hierar-
chical
Clustering

Unsupervised Learning

Unsupervised learning involves learning the labels for an unlabled dataset.

Examples of models:

- K-means clustering
- Hierarchical clustering

K-means clustering aims to partition data into k clusters, so that each observation belongs to the cluster with the nearest mean (which is found using the "centroid" of all the observations in each cluster).

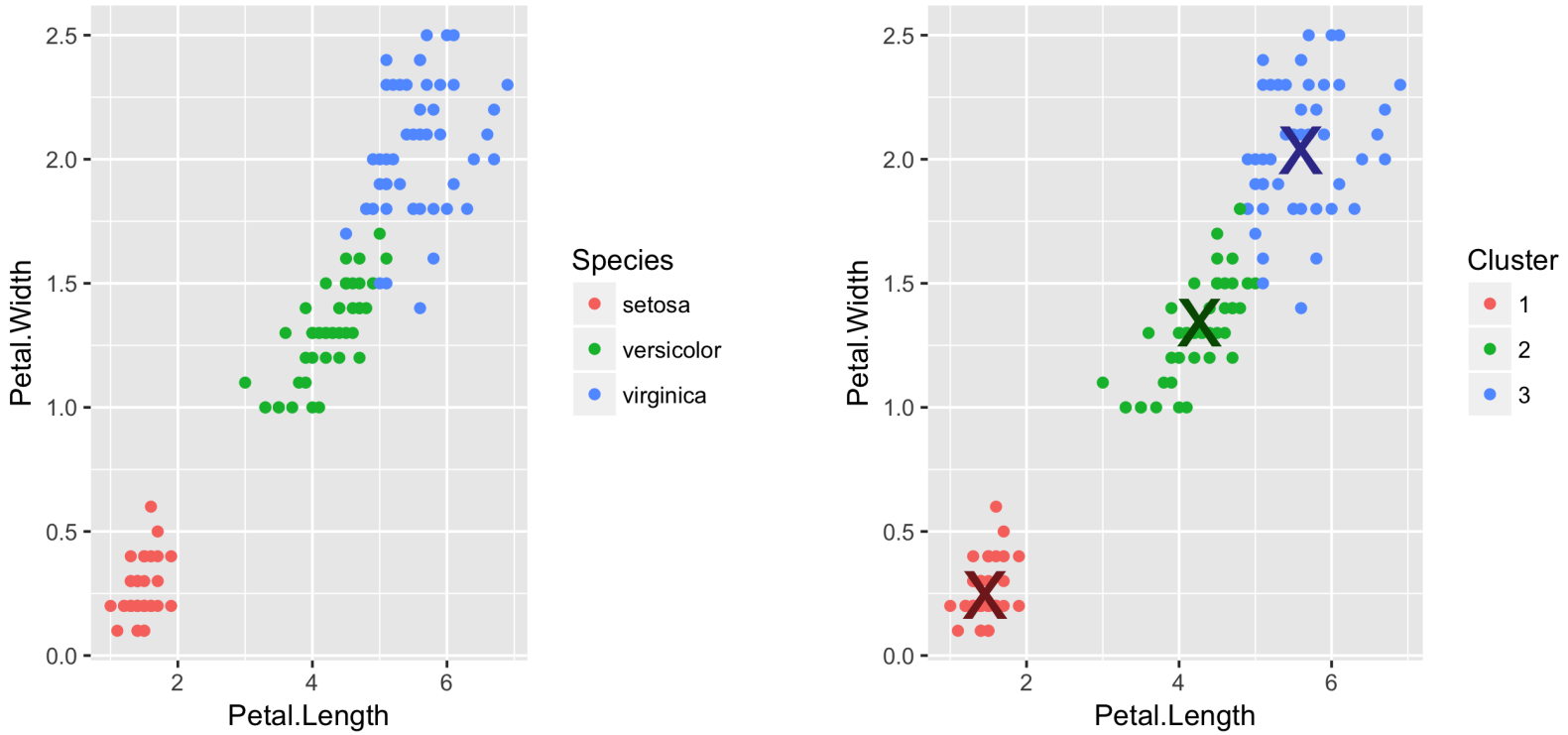
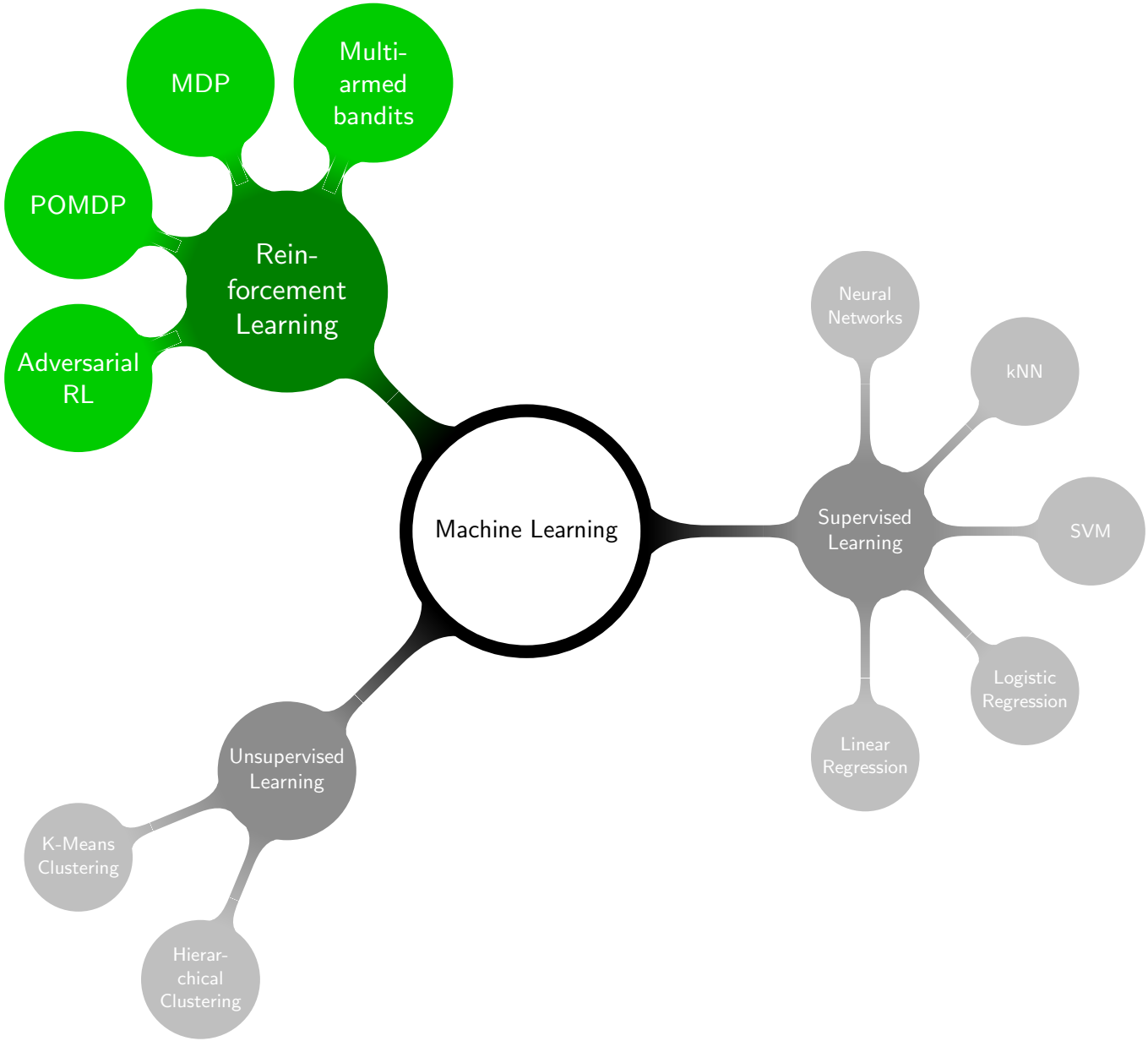


Figure : Iris dataset with (a) original labels and (b) 3-mean clusters.


Machine Learning
Rein-
forcement
Learning
MDP
Multi-
armed
bandits
POMDP
Adversarial
RL
Supervised
Learning
Neural
Networks
kNN
SVM
Logistic
Regression
Linear
Regression
Unsupervised
Learning
K-Means
Clustering
Hierar-
chical
Clustering

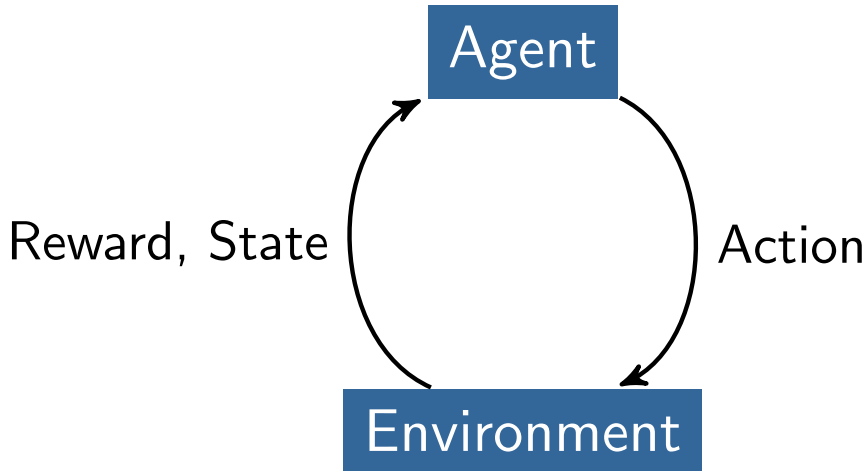
Reinforcement Learning

Reinforcement learning is an example of *online learning* which tries to learn a *policy* (set of actions) based on receiving rewards to maximize a long-run expected value.

Examples of models:

- Multi-armed bandit
- Markov decision process (MDP)
- Partially-observable markov decision process (POMDP)
- Multi-agent reinforcement learning

In a single agent version, we consider two major components: the *agent* and the *environment*.



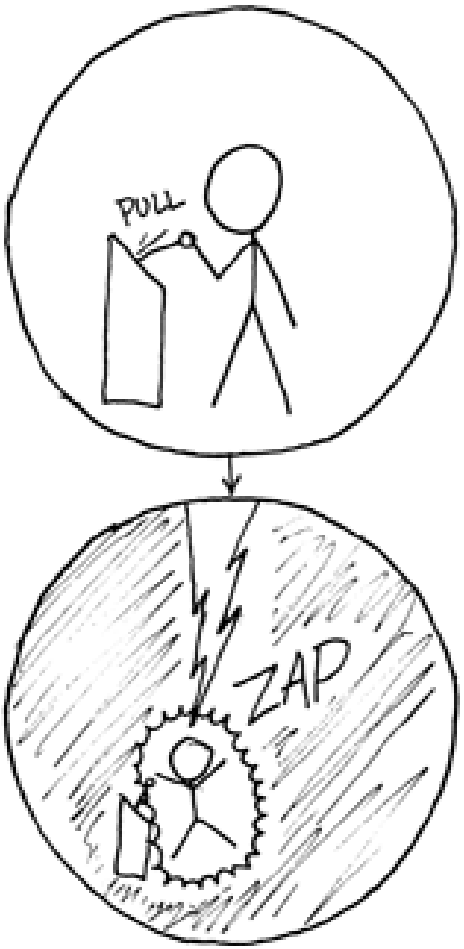
The agent takes actions, and receives updates in the form of state/reward pairs.

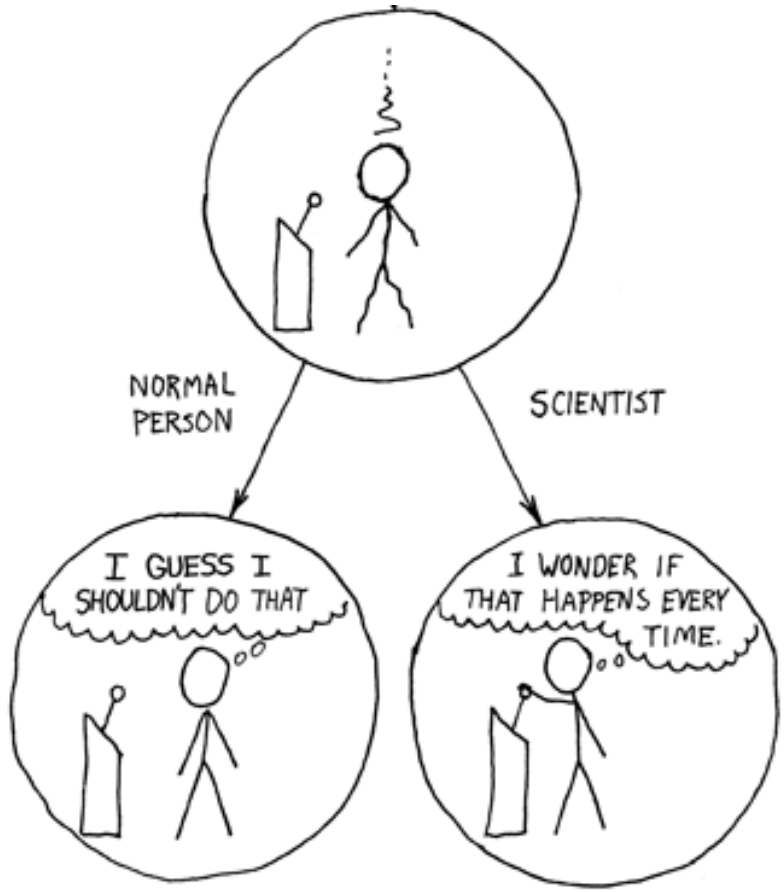
A simple introduction to the reinforcement learning problem is the case when there is only one state, also called a *multi-armed bandit*. This was named after the slot machines (one-armed bandits).

Definition

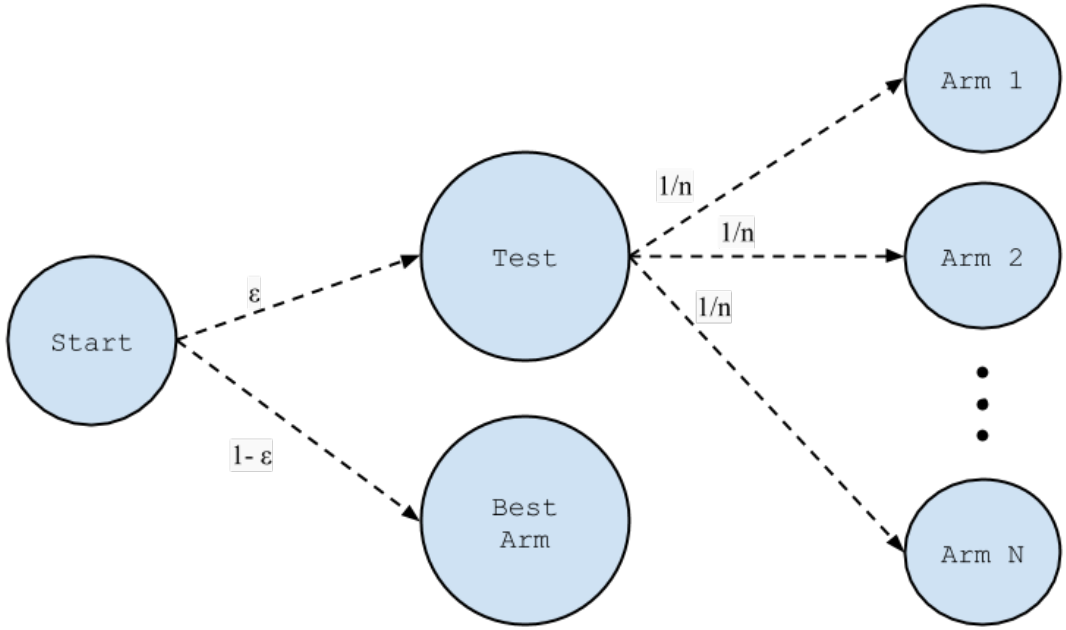
- Set of actions $A = 1, ..., n$
- Each action gives you a random reward with distribution $P(r_t | a_t = i)$
- The value (or utility) is $V = \sum_t r_t$

Online learning are a form of sequential optimization, and need to solve the *exploration vs. exploitation trade-off*.



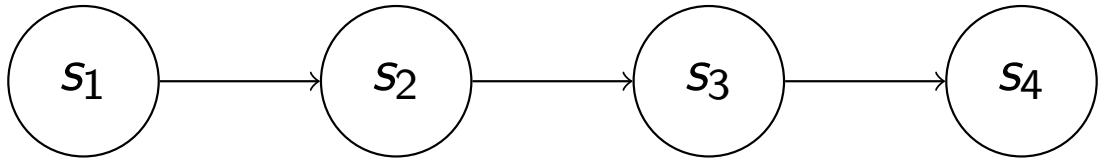


The ϵ-Greedy algorithm is one of the simplest and yet most popular approaches to solving the exploration/exploitation dilemma.



Picture courtesy of "Python Multi-armed Bandits" by Eric Chiang, yhat

Markov Processes are elementary in time series analysis.

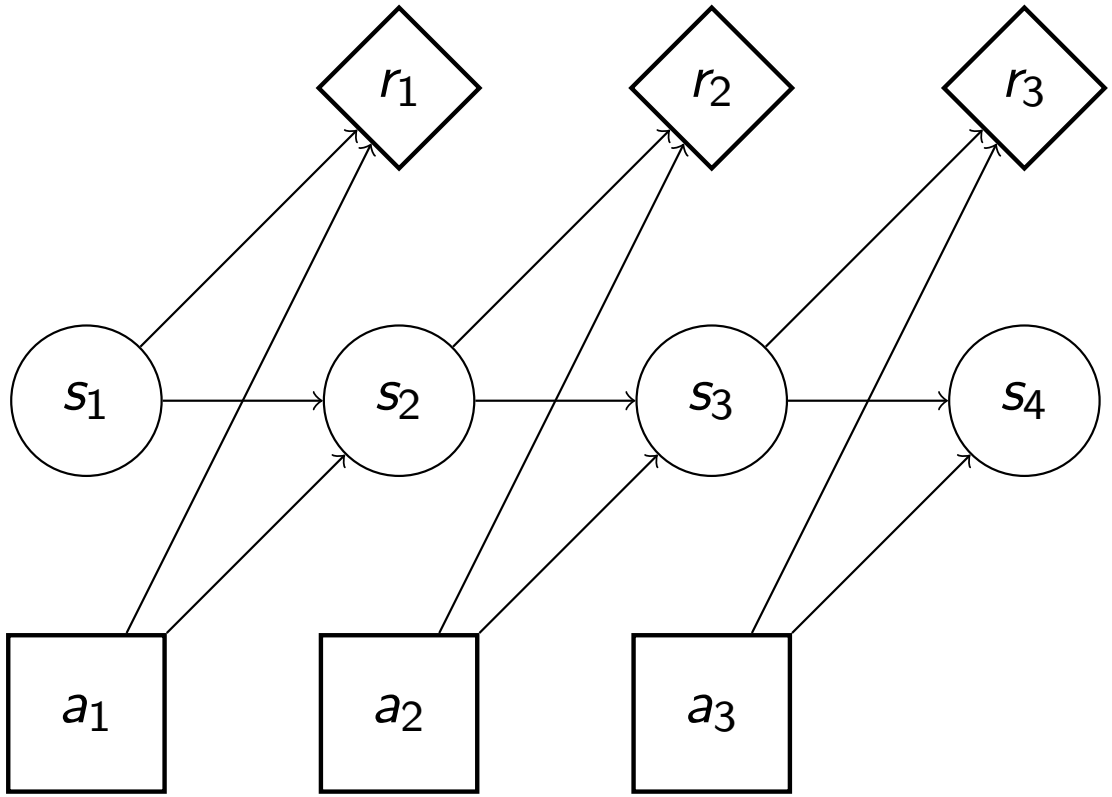


Definition

$$P(s_{t+1}|s_t, ..., s_1) = P(s_{t+1}|s_t) \tag{1}$$

- s_t is the state of the markov process at time t.

A *Markov Decision Process (MDP)* adds some further structure to the problem.



Grid world is a canonical example used in reinforcement learning.

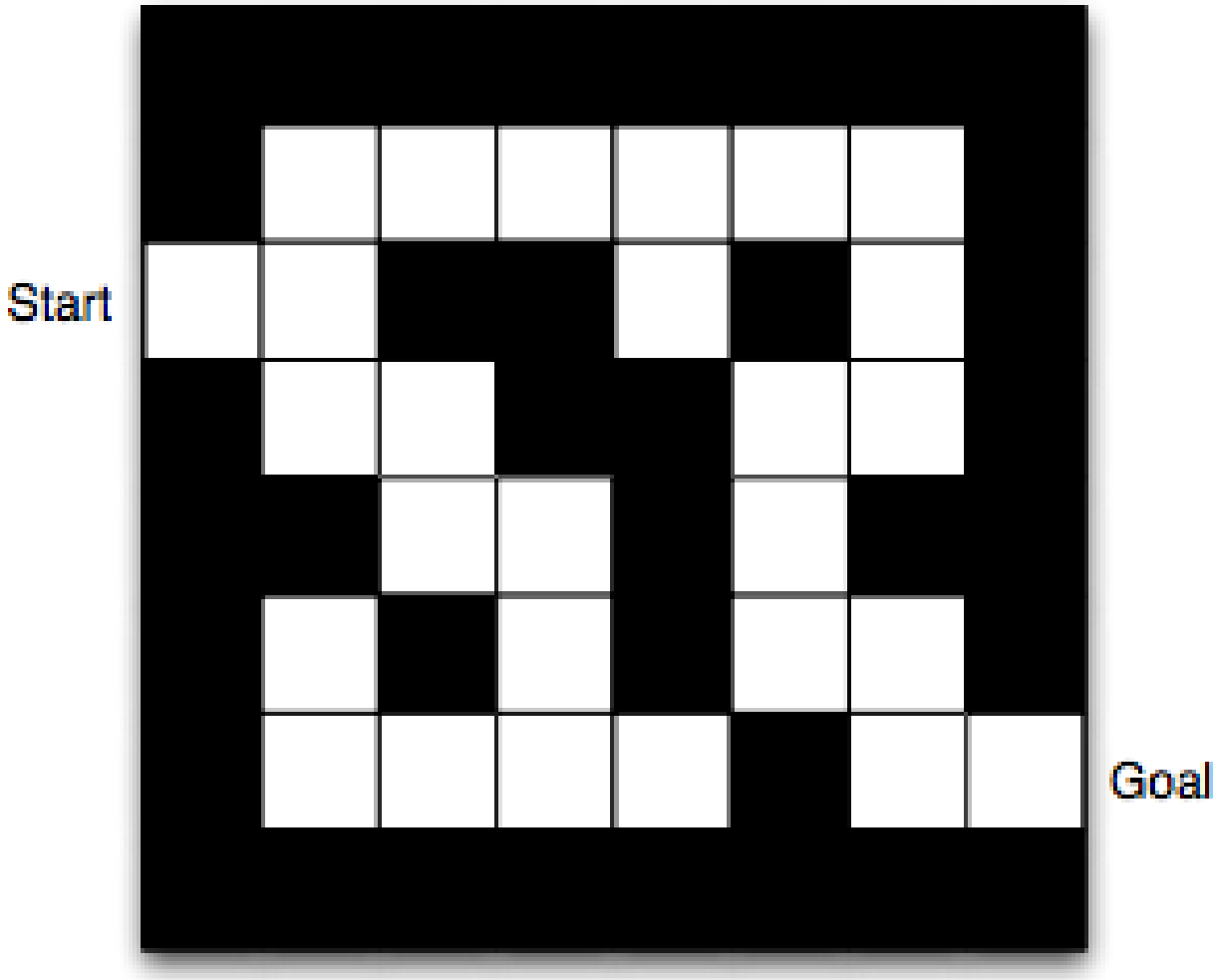


Image from David Silver.

We need to find a policy to navigate through the maze.

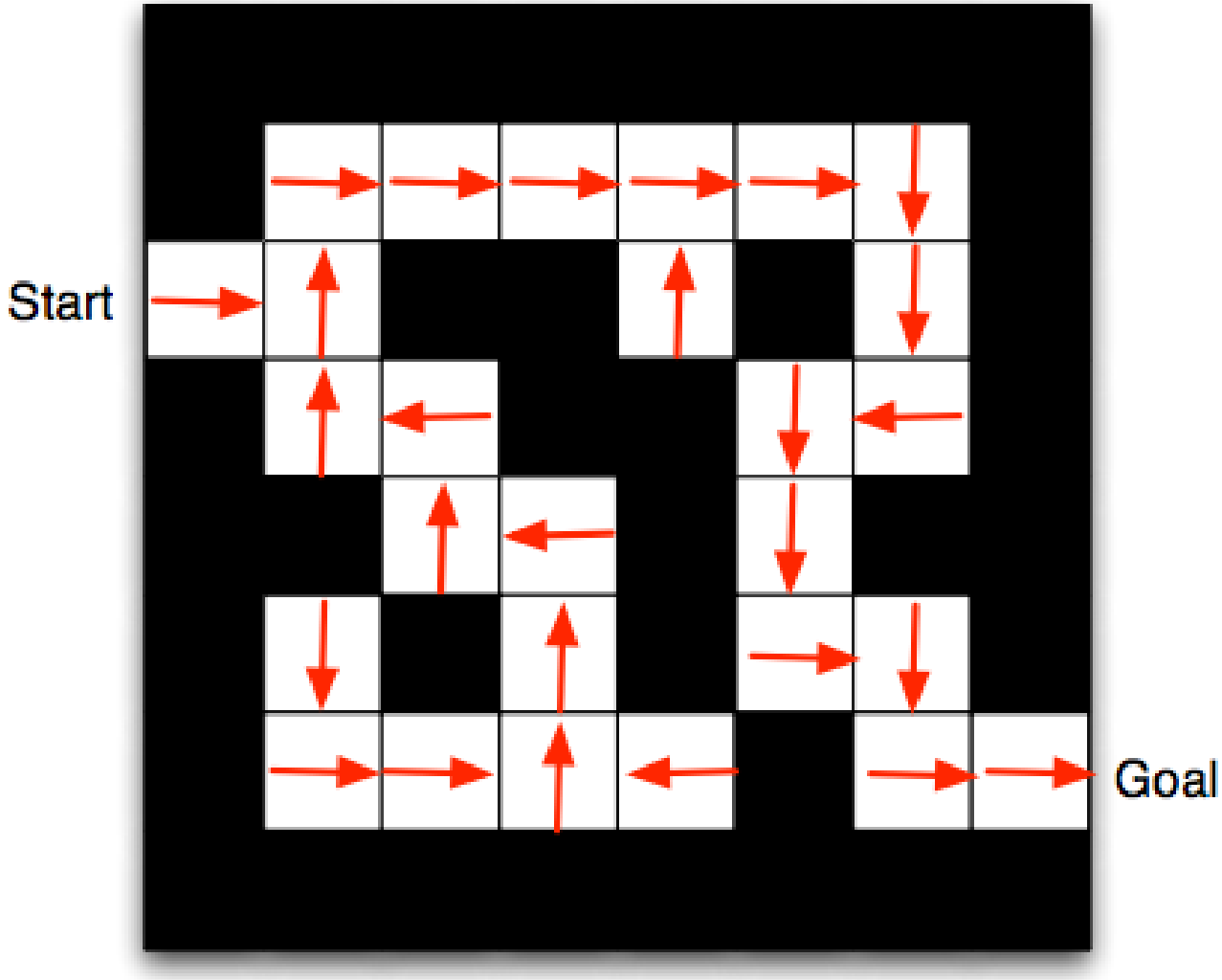


Image from David Silver.

The policy is based on maximizing a value function, that is dependent on the state and action pair (S, A).

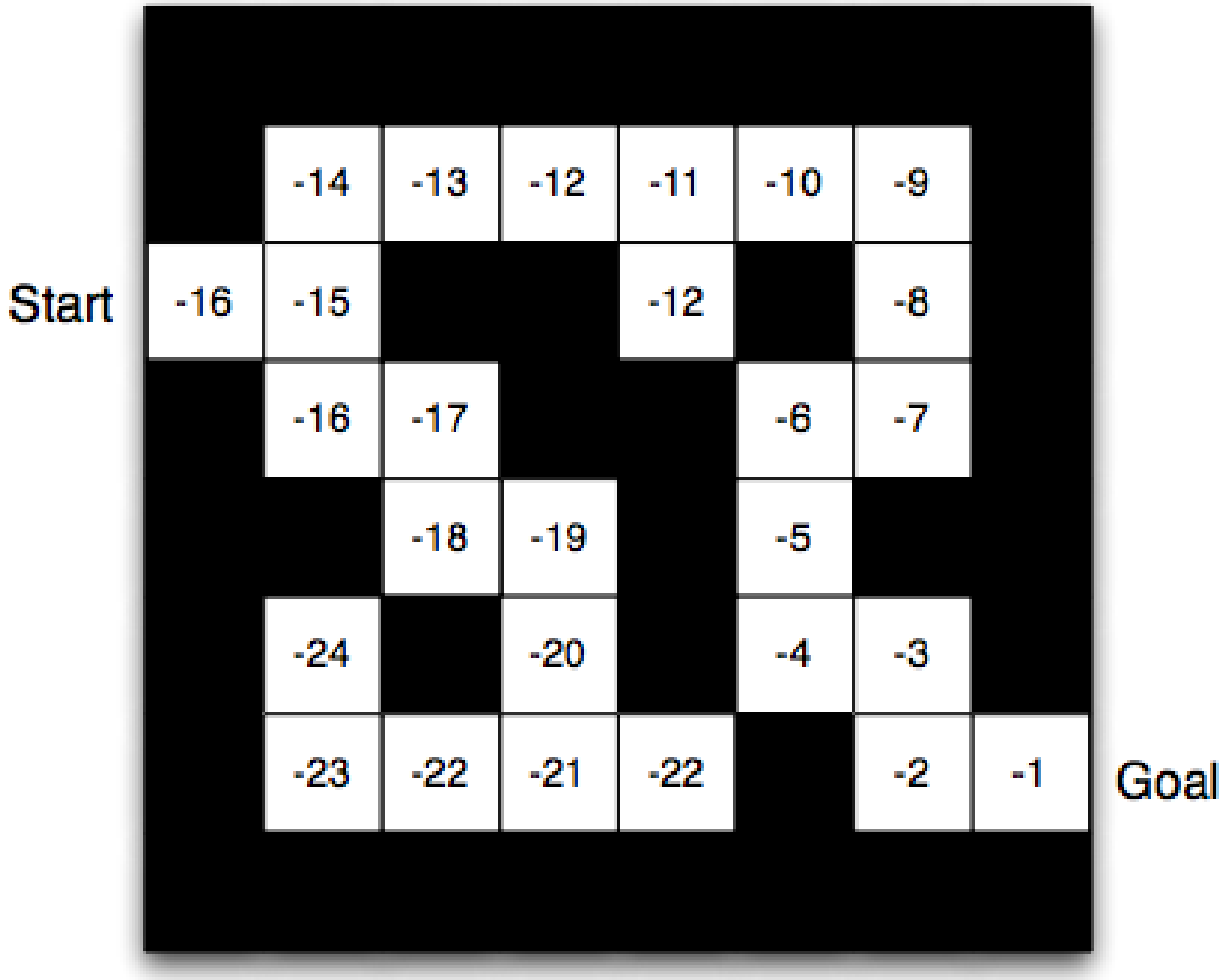


Image from David Silver.

Meta Algorithms

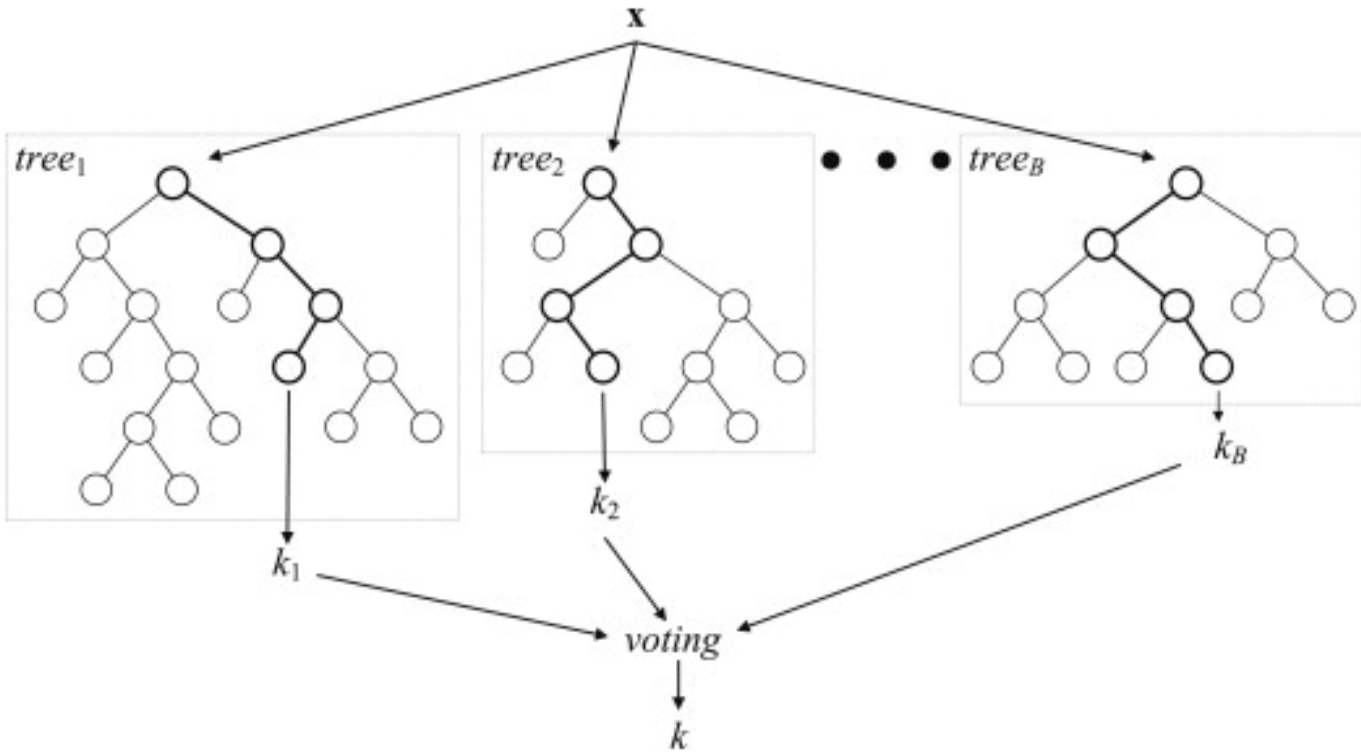
Several of the most successful recent algorithms make use of *ensemble methods* to reduce variance.

- Bagging (bootstrap aggregation)
- Boosting

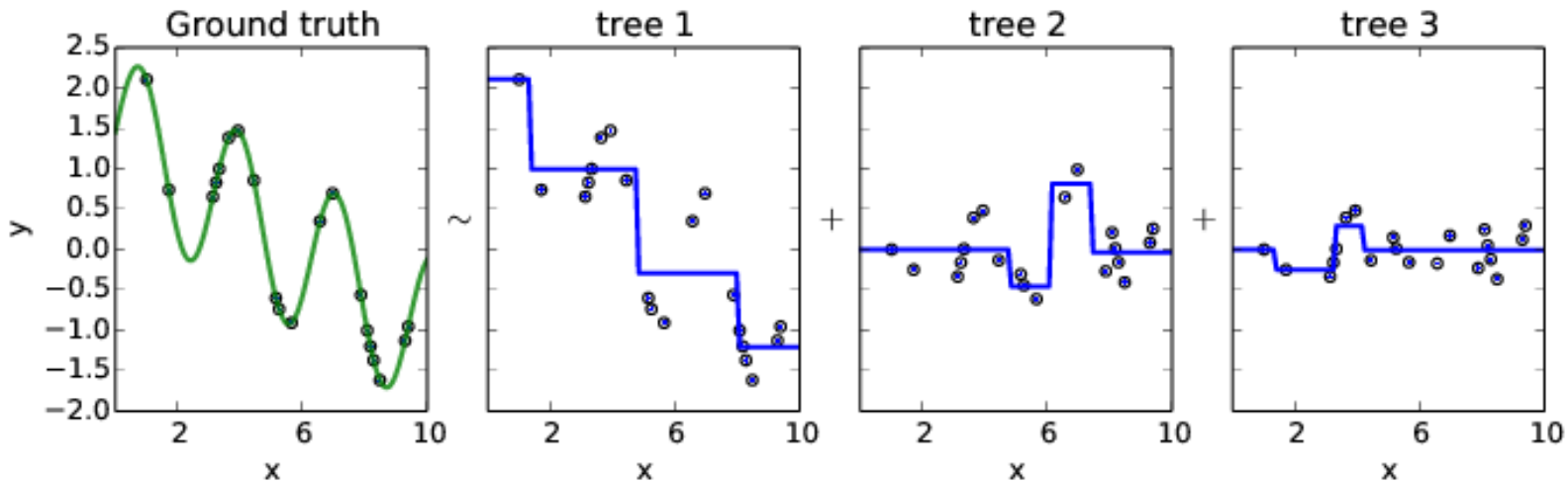
Other recent advances are a result of combining multiple techniques, such as Deep Reinforcement Learning, which combines a multilayer neural network with reinforcement learning.

Random Forests (Breiman 2001) is an ensemble method, using a bagging of decision trees.

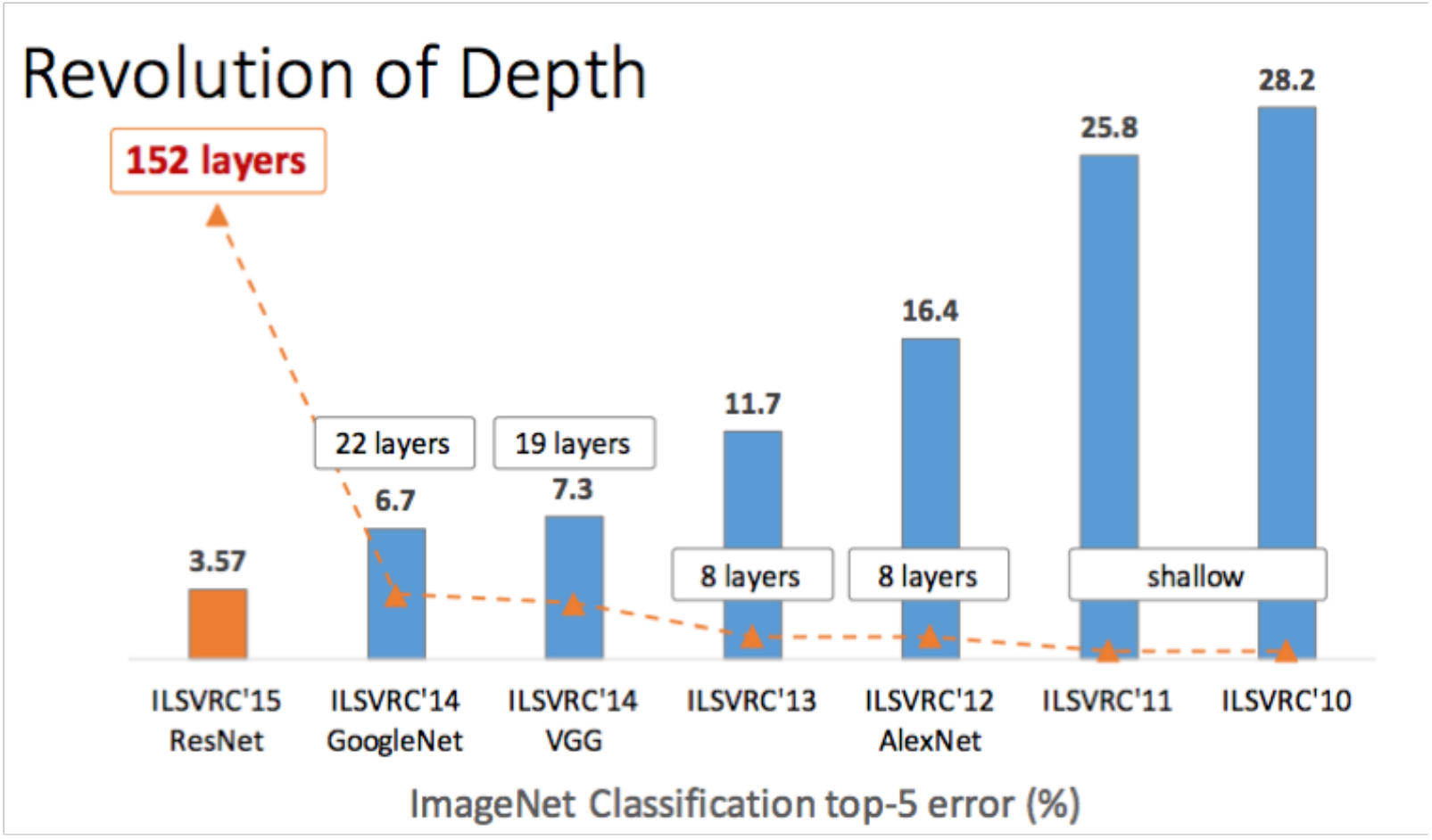
Random forests differ in only one way from this general scheme: at each split, they select a random subset of the features ("feature bagging"); this improves performance over bagged trees by decorrelating each tree.



Gradient boosting machines (GBM) also uses decision trees, but uses boosting: combining many weak learners (Friedman 1999). GBM refits each subsequent tree on the residuals of the prior fit, thus resulting in less correlated trees.



Deep Learning employs multiple levels (hierarchy) of representations, often in the form of a large and wide neural network.



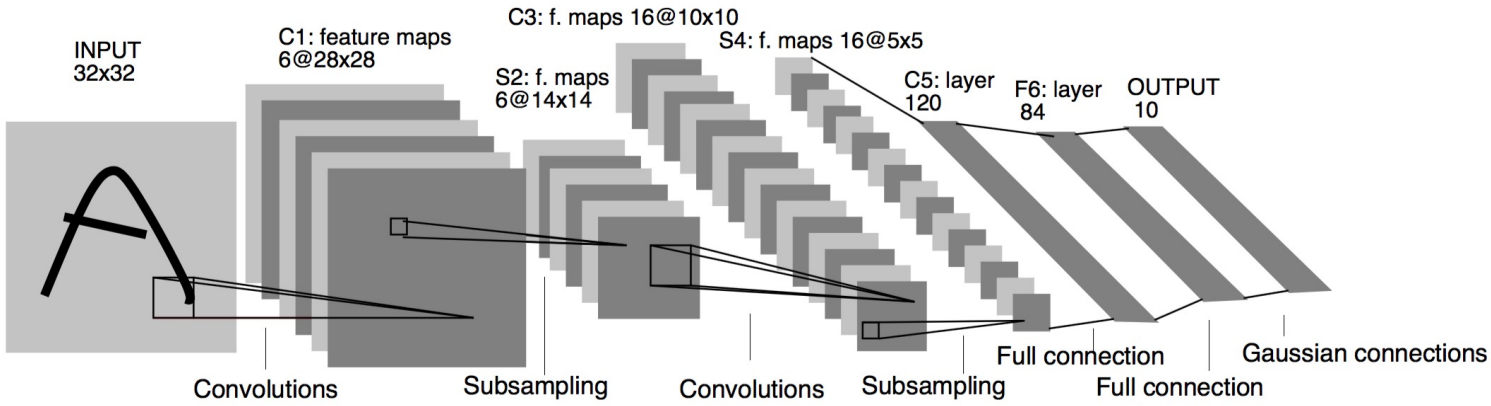


Figure : LeNET (1998), Yann LeCun et. al.

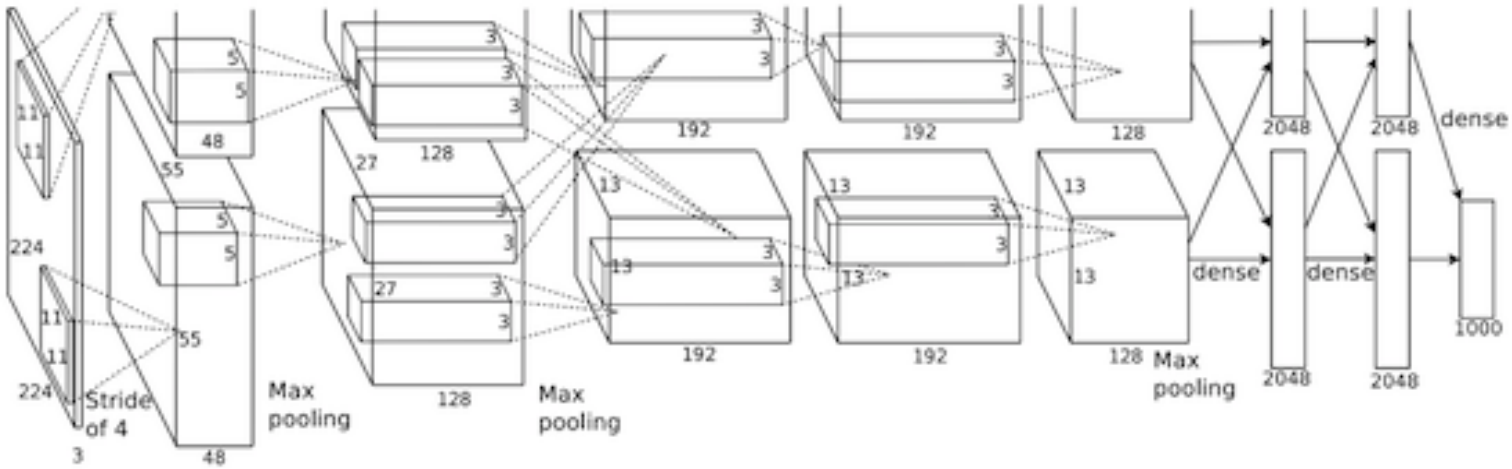


Figure : AlexNET (2012), Alex Krizhevsky, Ilya Sutskever and Geoff Hinton

Source: *Andrej Karpathy*

Generative adversarial networks (Goodfellow 2014) is an example of dueling neural networks which can learn about different kinds of information (e.g. photos, video, music) and generate new versions.

- Generator network: simulate fake data.
- Descriminator network: distinguish between fake and real data.

Generalization remains one of the greatest challenges to existing machine learning models.

Transfer Learning is a topic within machine learning that tries to generalize from one problem to another.

Pathnet (Deepmind 2017) uses multiple deep neural networks trained on different problems, and finds that a network already trained on a problem can learn more quickly on a new task.

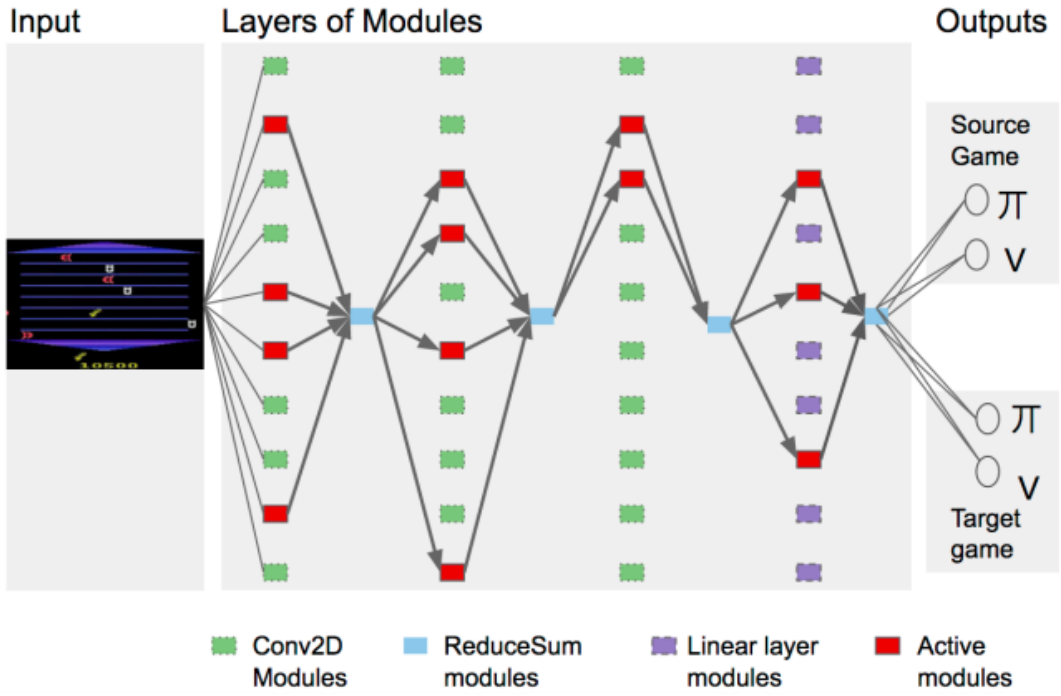
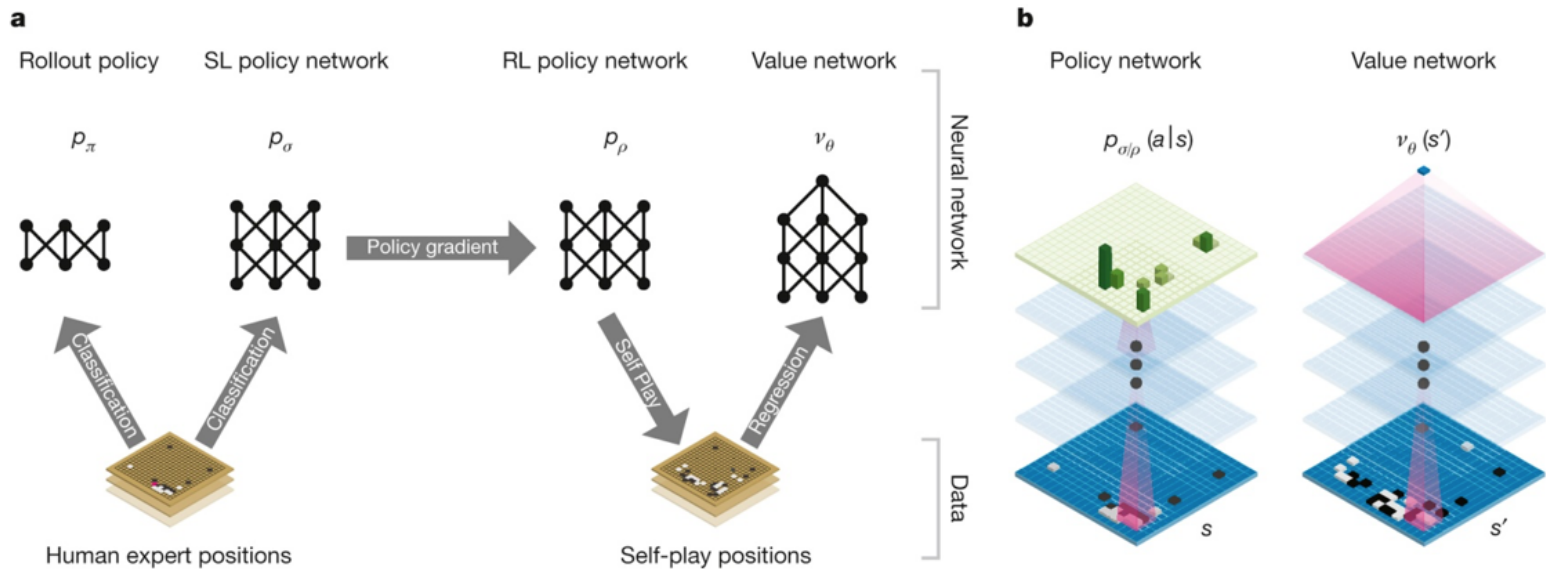


Figure : "It is a neural network algorithm that uses agents embedded in the neural network whose task is to discover which parts of the network to re-use for new tasks."

AlphaGo used several neural networks.



Final Thoughts

Fully automonomous (general) artificial intelligence is evolving quickly...

...but machine learning benefits from *domain expertise*.

- *Wicked problems* vs. tame problems (Rittel and Webber 1973)
- *No Free Lunch Theorem* (Wolpert 1996)
- *Occam's Razor*

Always important to make black box learning as "light box" as possible: don't turn your model into a "weapon of math distruction" (WMD).

At the end of the day, model objectives and constraints come from researchers: we are still responsible for the results.