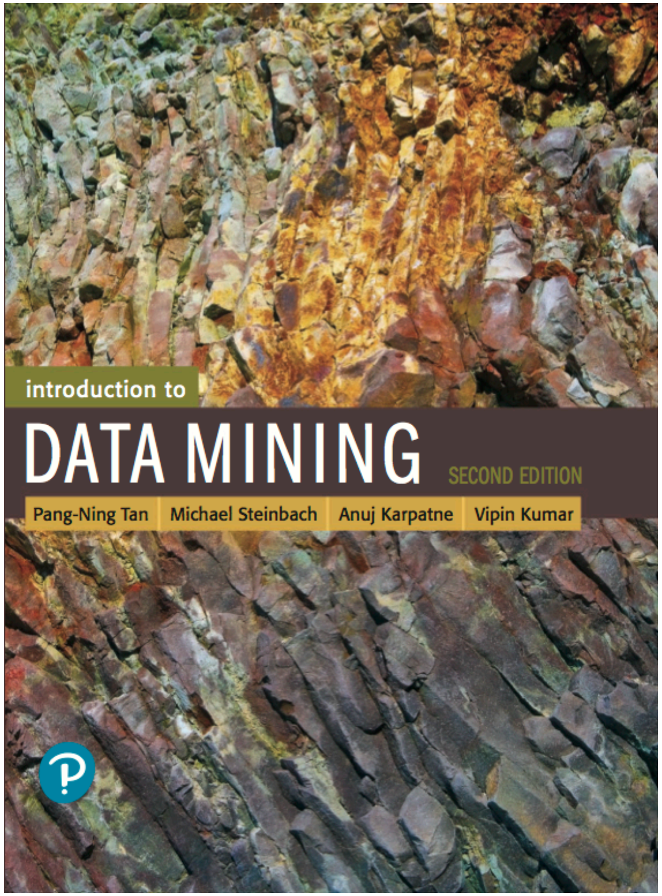
Introduction to Machine Learning

Anuj Karpatne
Assistant Professor,
Computer Science
Virginia Tech

Torgersen Hall 2160F,
karpatne@vt.edu
https://people.cs.vt.edu/karpatne/

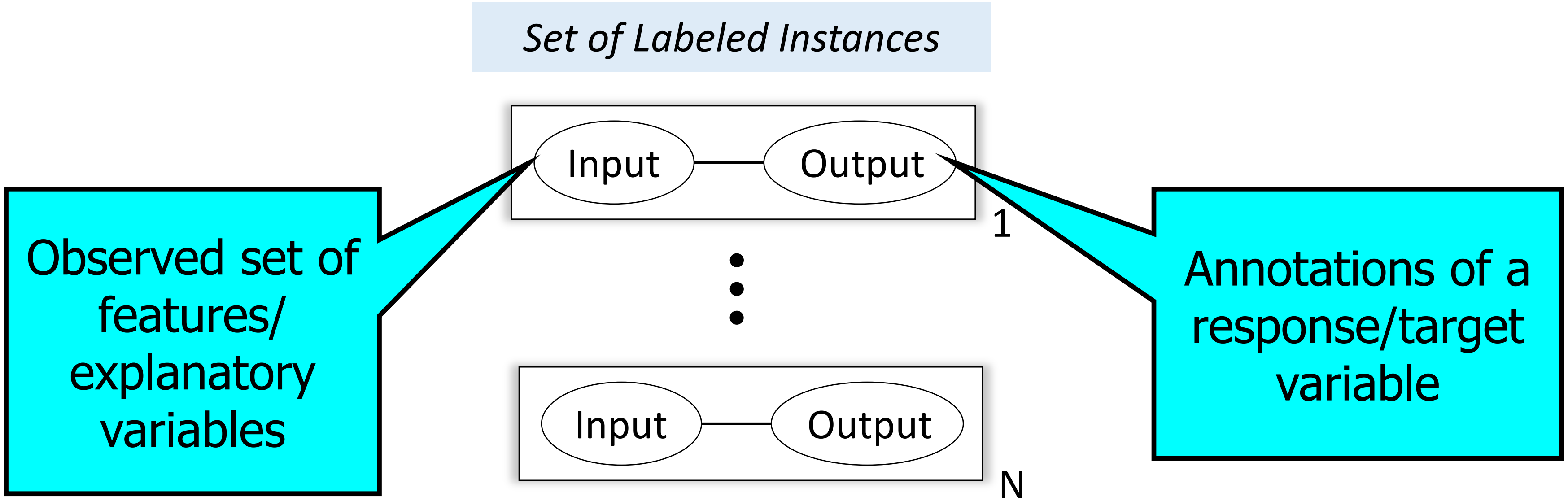
Slide materials adapted from:



https://www-users.cs.umn.edu/~kumar001/dmbook/index.php

Key Areas of Machine Learning

1. Predictive Modeling / Supervised Learning



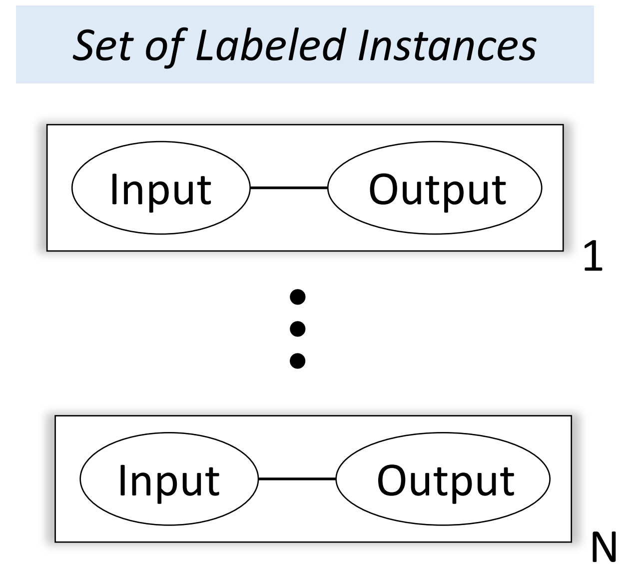
Basic Goal:

- Model relationship between input and output variables to predict the output on unseen (new) instances

Key Areas of Machine Learning

1. Predictive Modeling

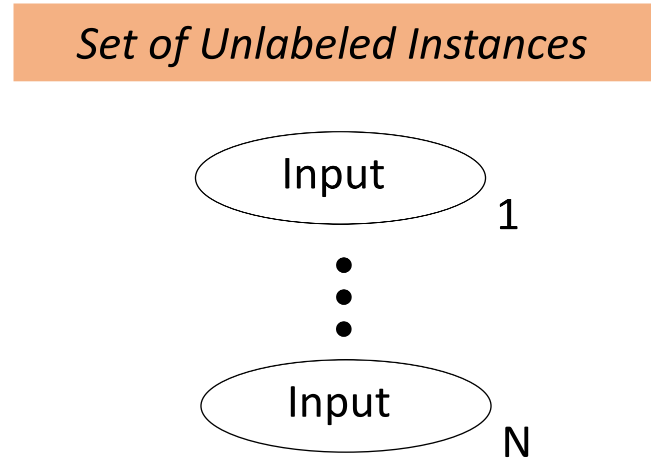
- Classification
  - Target takes discrete values: {0,1,2,…}
- Regression
  - Target takes continuous values



Key Areas of Machine Learning

1. Predictive Modeling

- Classification
- Regression



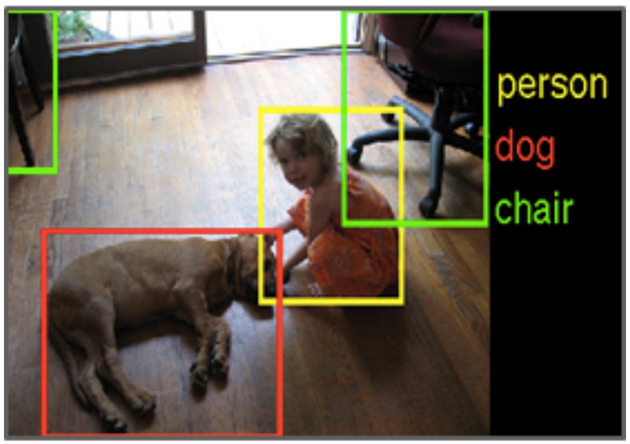
2. Descriptive Modeling / Unsupervised Learning

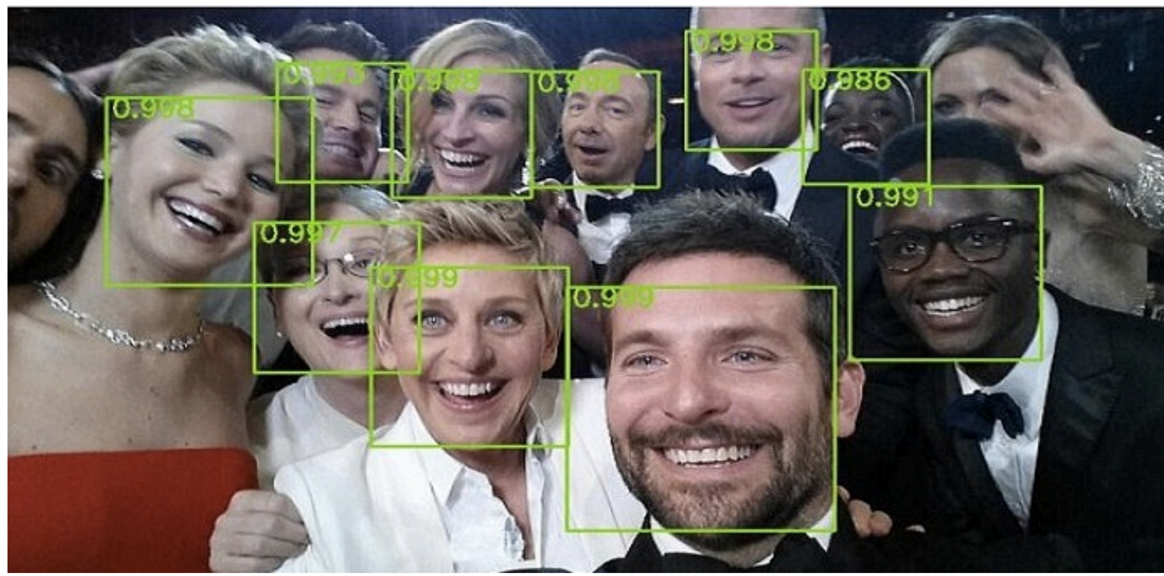
- Find human-interpretable patterns from “unlabeled” data

- **Dimensionality Reduction**
  - Find low dimensional data representations
- **Generative Modeling**
  - Learn a model to generate synthetic samples from a data distribution
- **Clustering and Anomaly Detection**
  - Find groups with similar properties
  - Find unusual instances

Classification: Illustrative Examples

- Object Recognition
  - Given the pixel values of an image region *(features)*, identify the type of object *(class)*



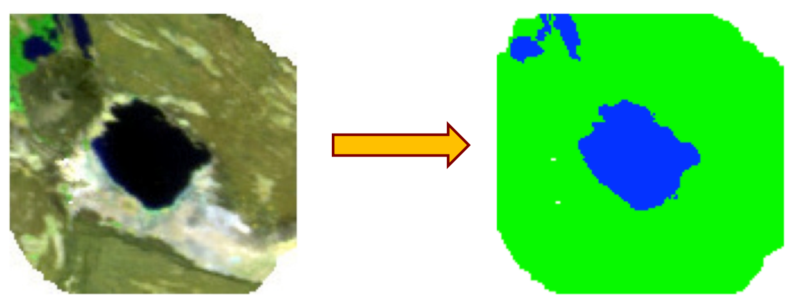


Classification: Illustrative Examples

- Image Recognition
  - Given the pixel values of an image region *(features)*, identify the type of object *(class)*
- Spam Filtering
  - Given the message header and content of an email *(features)*, classify spam or no spam *(class)*

Classification: Illustrative Examples

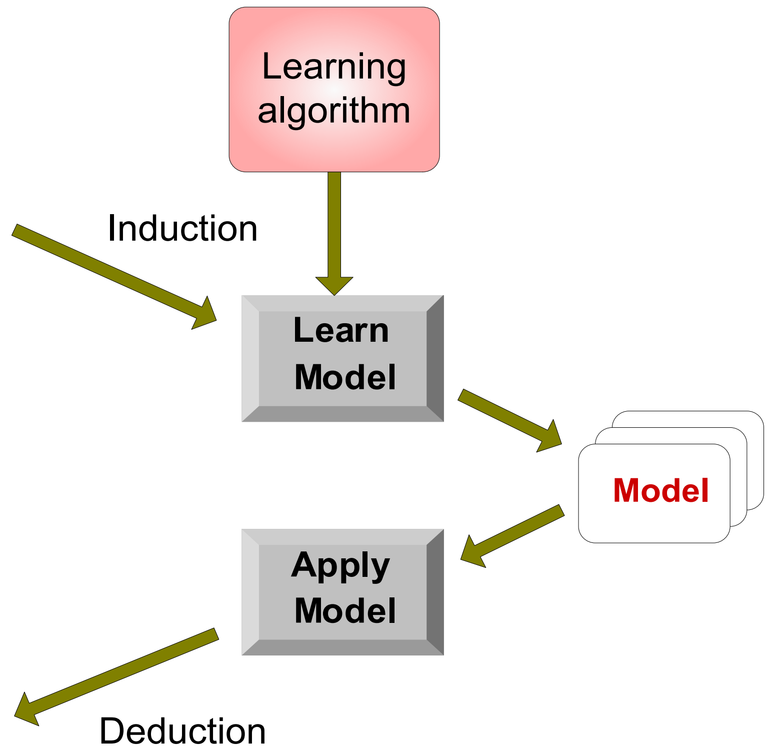
- Image Recognition
  - Given the pixel values of an image region *(features)*, identify the type of object *(class)*
- Spam Filtering
  - Given the message header and content of an email *(features)*, classify spam or no spam *(class)*
- Land Cover Mapping
  - Given the multi-spectral values *(features)*, classify land cover: water, vegetation, urban, etc. *(class)*



Predictive Modeling: General Approach

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

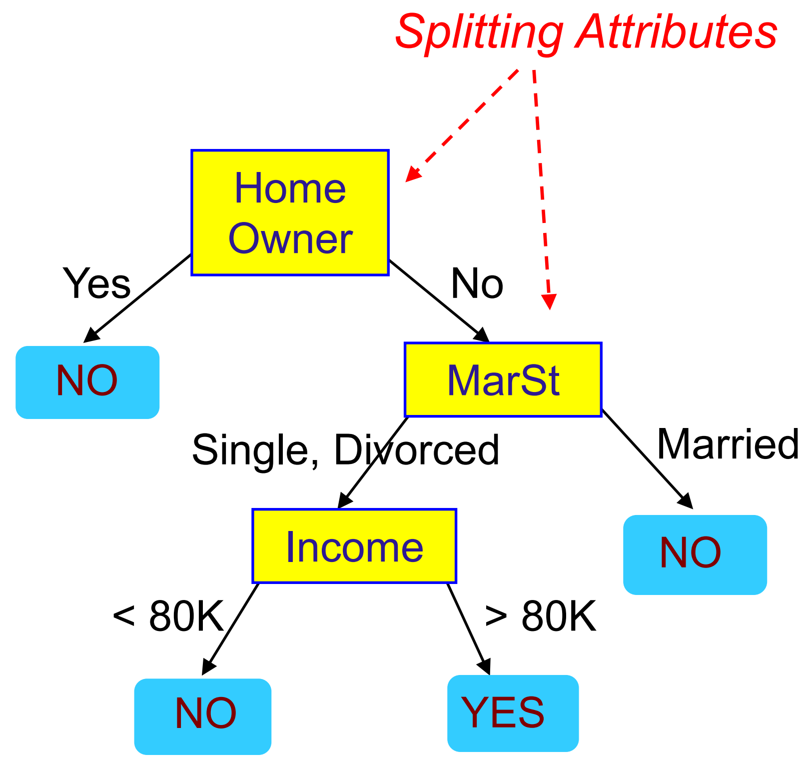
Two Modeling Choices:

- Choice of Model Design (linear/non-linear/…)
- Choice of Learning Algorithm

Example of Classification Model:
Decision Tree

	categorical	categorical	continuous	class
ID	**Home Owner**	**Marital Status**	**Annual Income**	**Defaulted Borrower**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Training Data

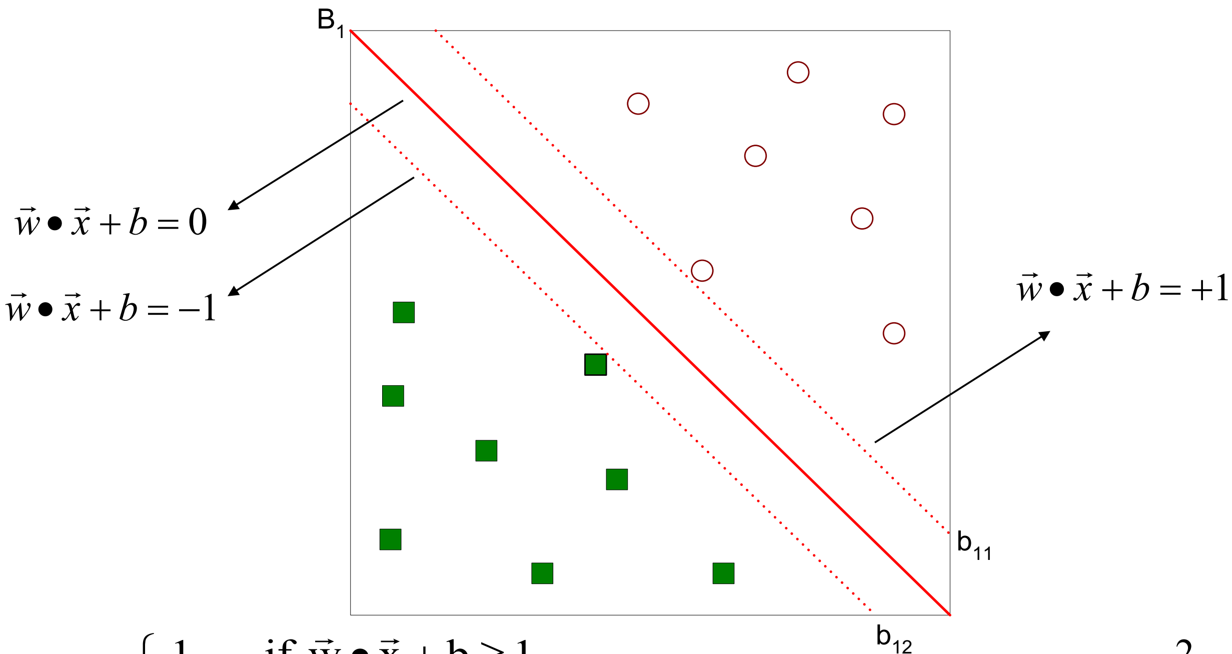


Model: Decision Tree

Design choice: Number of nodes in tree (size)

http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Example of Classification Model:
Support Vector Machines (SVMs)

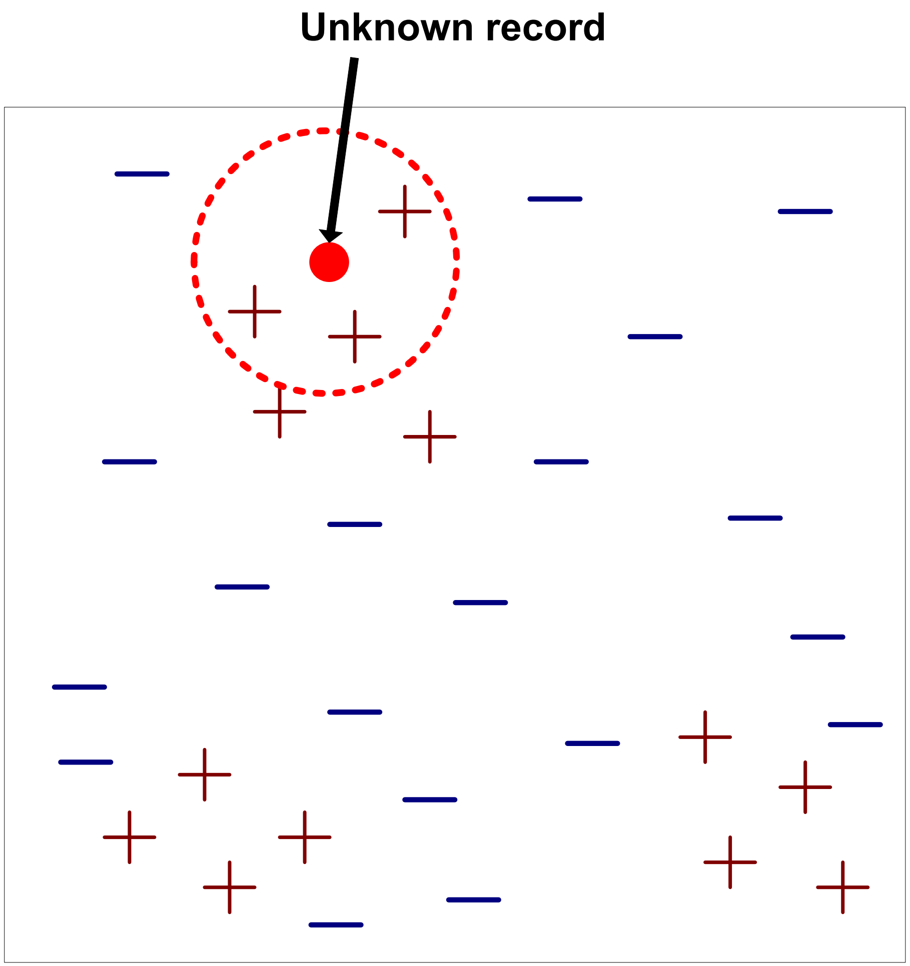


$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \geq 1 \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}} + \mathrm{b} \leq -1 \end{cases}$$

$$\text{Margin} = \frac{2}{\|\vec{w}\|}$$

- Linear hyperplane (decision boundary) to separate the classes
- Non-linear version:
  - Learn decision boundaries in a higher-dimensional transformed space
  - Non-linear mapping to transformed space modeled using kernel functions

Example of Classification Model:
k-Nearest Neighbor (kNN) Classifier



- To classify an unknown record:
  - Compute distance to other training records
  - Identify *k* nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Example of Classification Model:

Naïve Bayes and Probabilistic Graphical Models

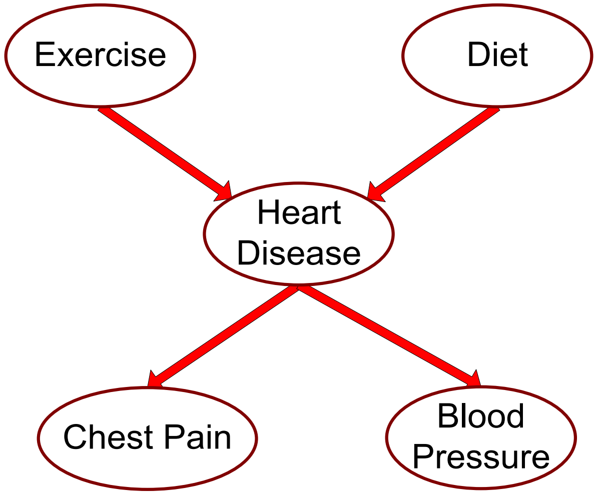
Bayes Theorem:

Conditional Prior

$$\boxed{P(Y \mid X_1 X_2 \ldots X_d) = \frac{P(X_1 X_2 \ldots X_d \mid Y) P(Y)}{P(X_1 X_2 \ldots X_d)}}$$

Posterior Evidence

- Naïve Bayes Model:
  - Assume **conditional independence** among attributes X_i when class is given:
    - $P(X_1, X_2, \ldots, X_d \mid Y_j) = P(X_1 \mid Y_j)\, P(X_2 \mid Y_j) \ldots P(X_d \mid Y_j)$
- Probabilistic Graphical Models:
  - Provides graphical representation of probabilistic relationships among a set of random variables
  - **Directed edges: Bayesian Networks**, Undirected edges: Markov Random Fields

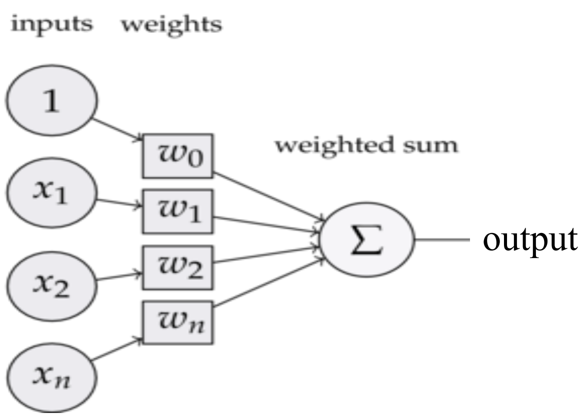


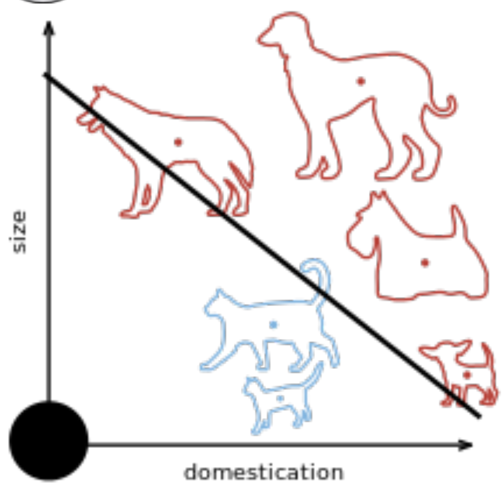
Example of Classification Model:
Artificial Neural Networks

Perceptron (1970s) → Deep Learning (~2010+)

Perceptron (1970s)

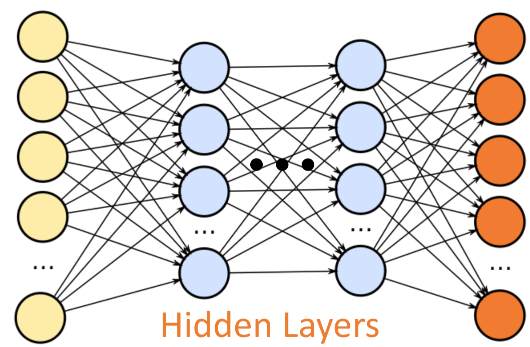
- Single processing unit
- Can only learn linear decision boundaries

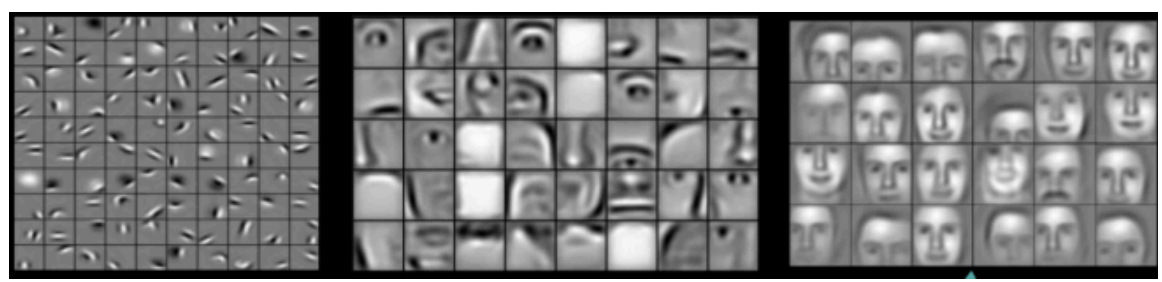




Deep Learning (~2010+)

- Composition of large number of processing units
- Can learn highly complex decision boundaries
- Feedforward neural networks, multi-layer perceptrons



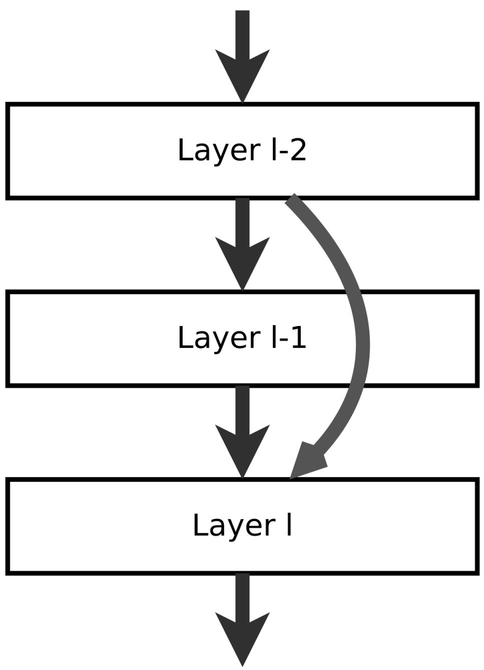


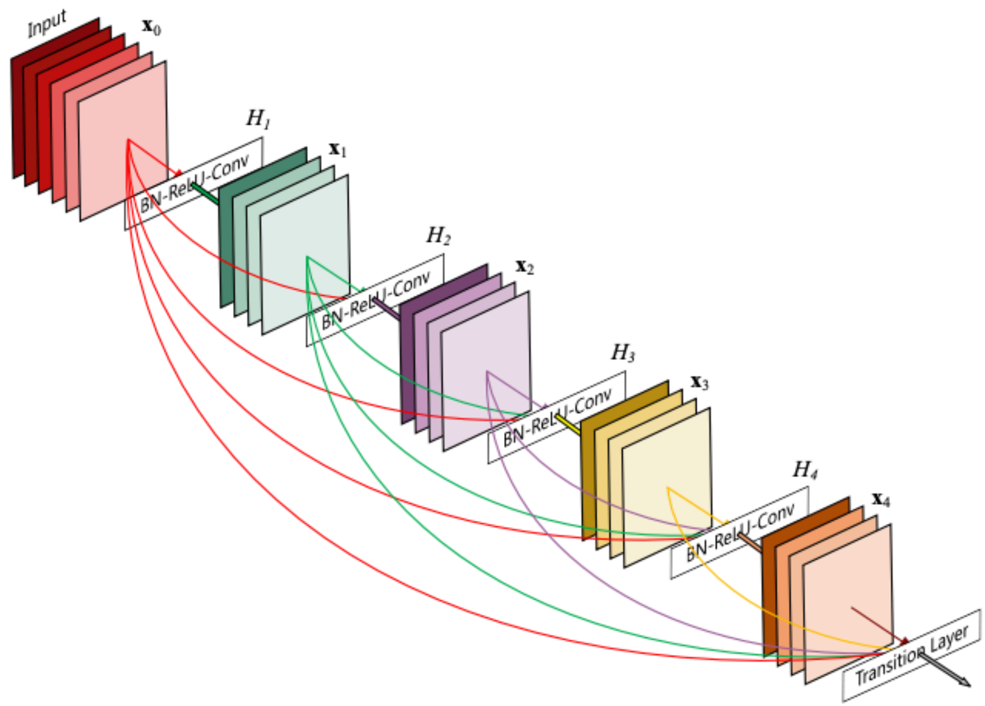
Design choice: Number of layers, type of connections, …

Deep Learning Architectures:

Going beyond Fully Connected Architectures

- Residual Connections:
  - Enable learning of “very” deep neural networks by only learning residuals of last layer
- Dense Connections:
  - Include all shortcut connections to encourage feature reuse

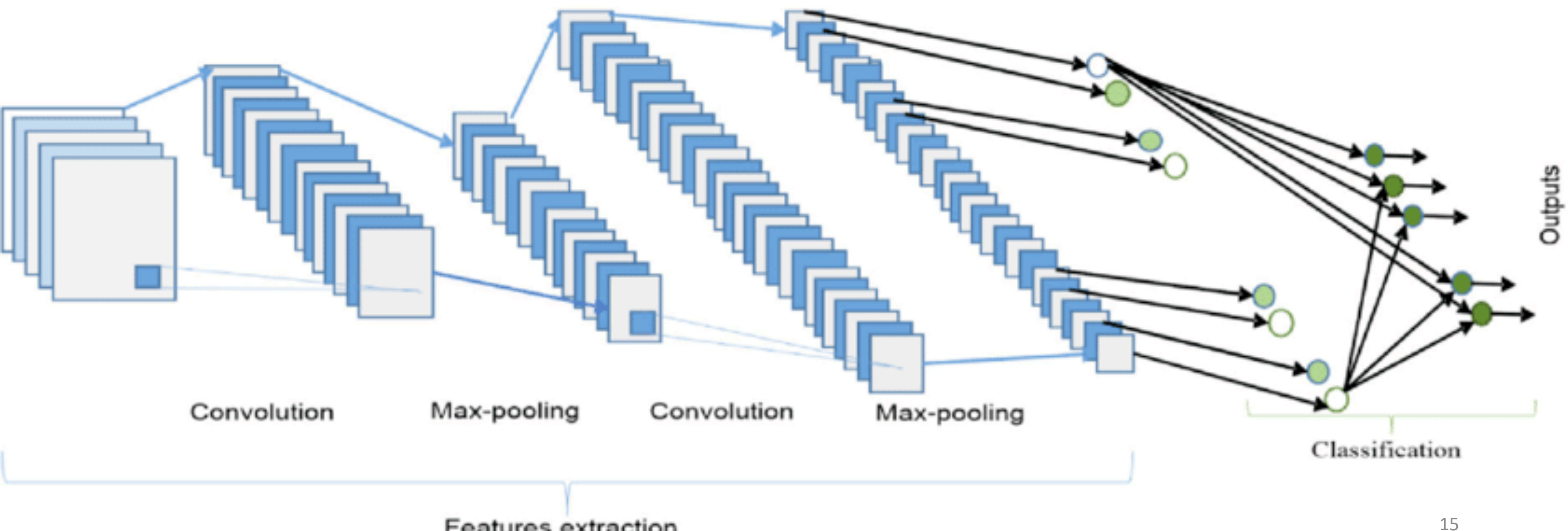




https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

Deep Learning for Image Data:
Convolutional Neural Networks

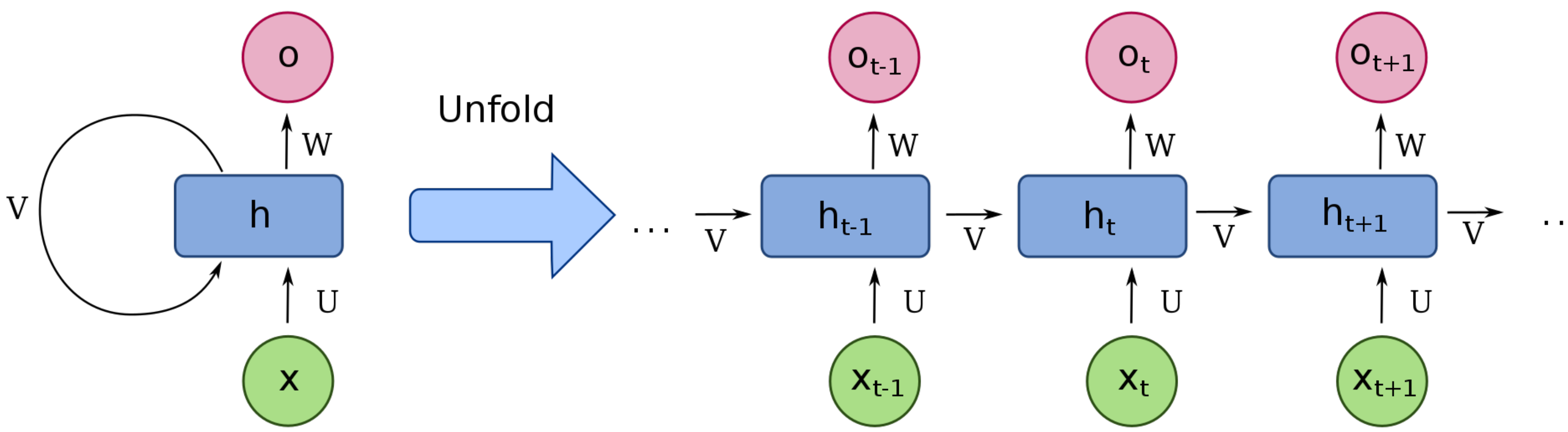
- Basic idea: Non-linear operators only need to be applied **locally** around a pixel of an image using a “convolution kernel”
- Two types of layers:
  - Convolution layers produces feature maps of similar size as input image
  - Pooling layers reduce the size of feature maps using sub-sampling



https://cs231n.github.io/understanding-cnn/

Deep Learning for Sequence Data:
Recurrent Neural Networks

- Basic idea: Use information extracted from previous time-steps for making prediction at a current time-step
- Variants: Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Transformer Networks



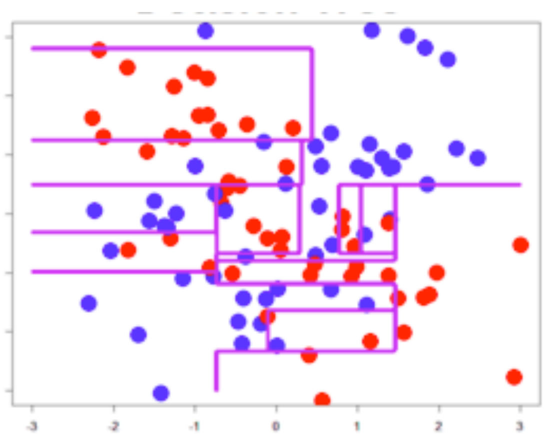
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks
https://towardsdatascience.com/transformers-141e32e69591

Classification Models

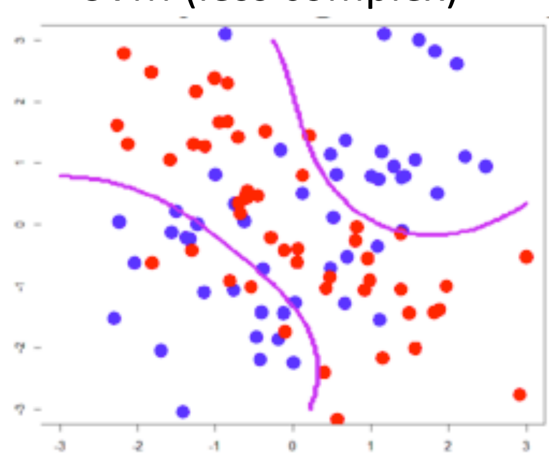
- Decision Trees
- Support Vector Machines (SVM)
- Nearest-neighbor Classifier
- Naïve Bayes and Probabilistic Graphical Models
- Artificial Neural Networks

Models with varying ***complexity***:
Capacity to represent complex boundaries

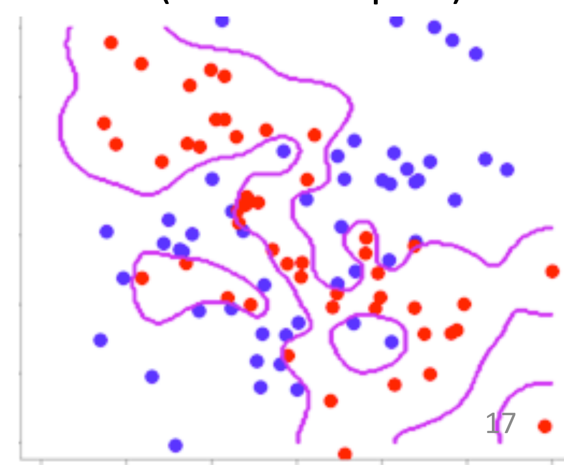
Decision Tree



SVM (less complex)



SVM (more complex)

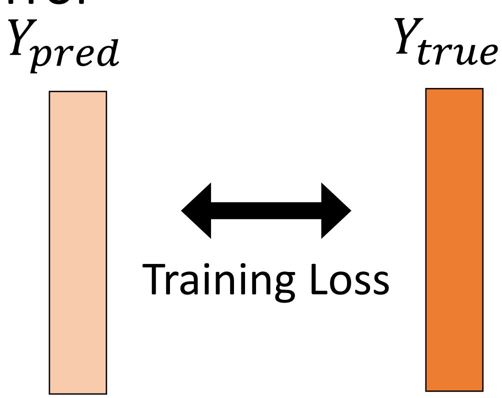


Learning Algorithms for Classification

- Criteria for selecting a suitable model:
  - Good *generalization performance*:
    - Model should perform well on unseen instances encountered outside the training set

- However, we can only measure the performance on the training set during model building!

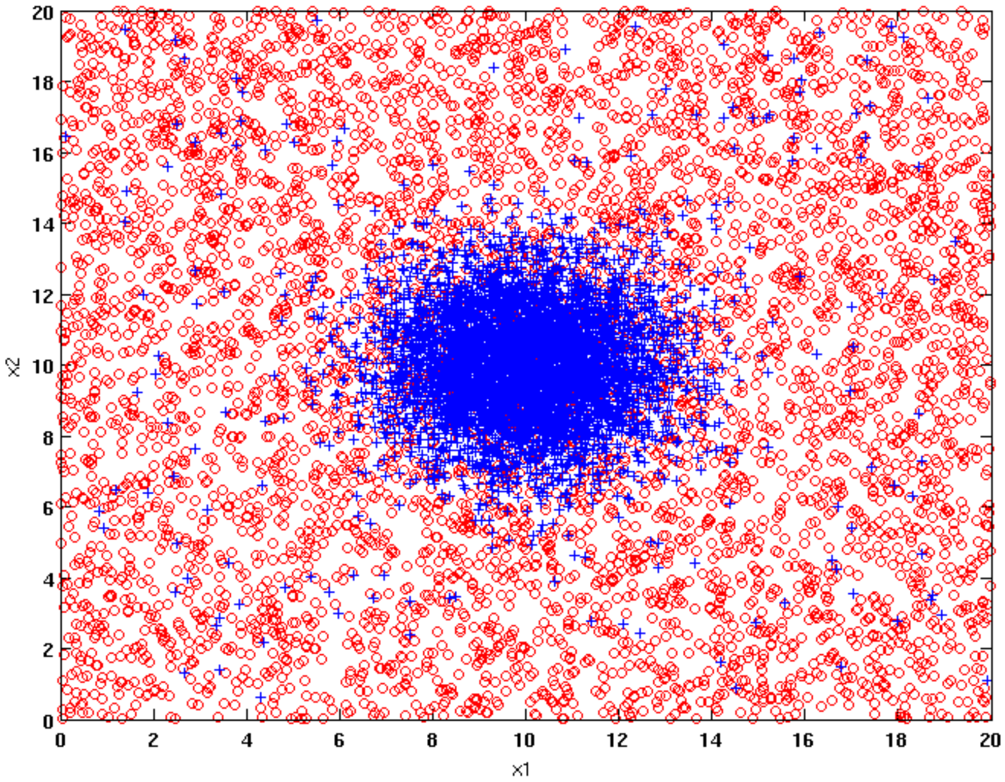
- Naïve Approach: Use training error (or loss) as an estimate of generalization error



Model = *argmin* Training Loss

Complex models (almost always) show lower training error

Assessing Generalization Performance



Two class problem:

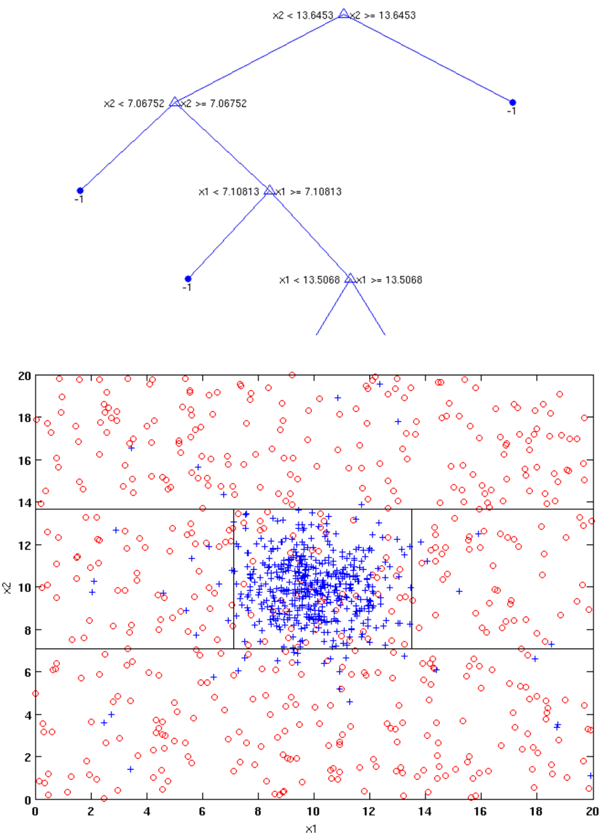
+ : 5200 instances

o : 5200 instances

10 % of the data used for training and 90% of the data used for testing

Assessing Generalization Performance

Decision Tree T1
(Less Complex)



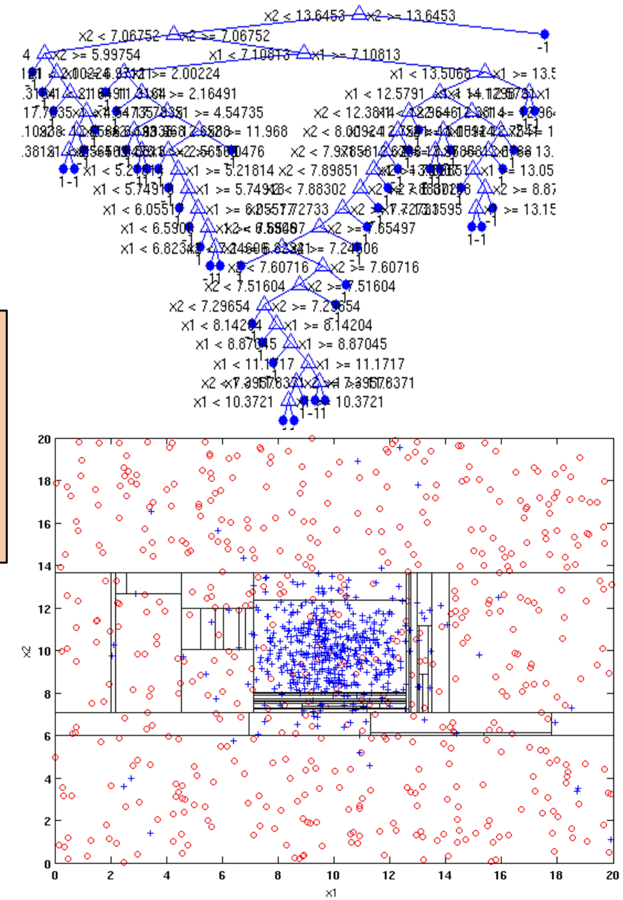
Is T2 better than T1?

- **Not Really!**

Phenomena of Overfitting:

- When model is too complex, training error is small but test error is large

Decision Tree T2
(More Complex)



Training Error: 10%

Test Error: 10%

Training Error: 5%

Test Error: 18%

Ensuring Generalization Performance

- **Trade-off** training error (loss) with model complexity

Model = *argmin* Training Loss + λ Model Complexity

Basis of several ML principles such as structural risk minimization, bias-variance trade-off, …

- Learning Algorithms:
  - Regularization (using statistical norms of parameters as loss)
  - Using Priors (in probabilistic frameworks)
  - Constrained Optimization Methods
  - …

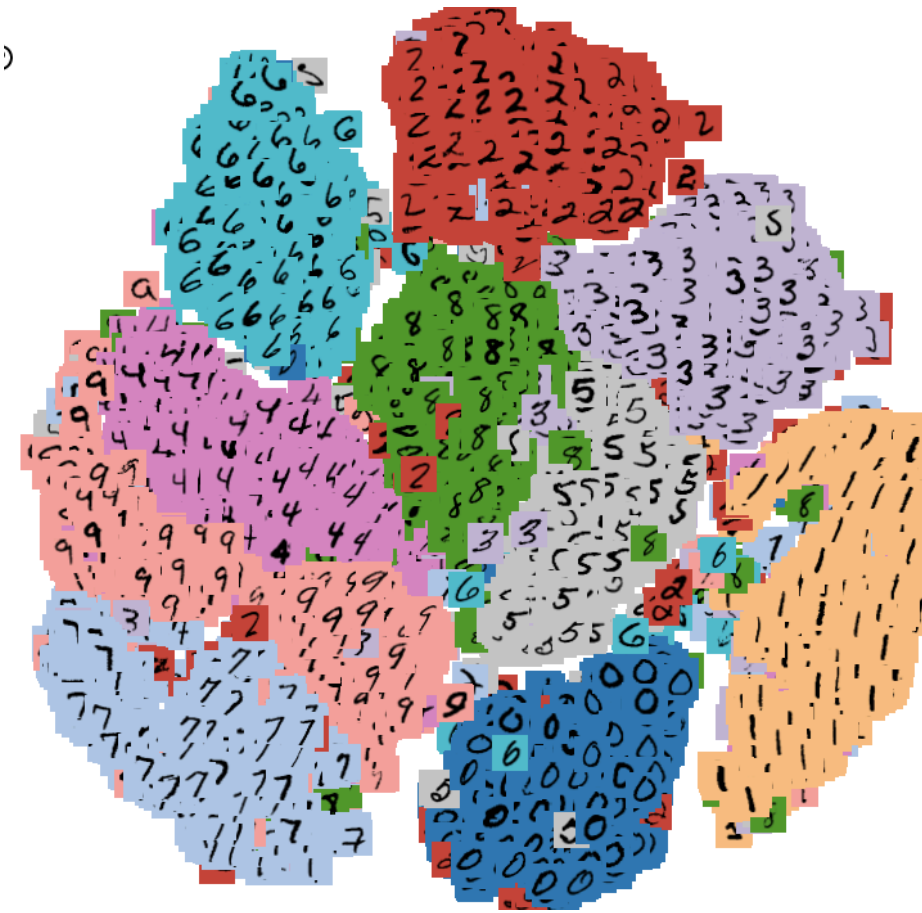
Great interactive tutorial on bias-variance trade-off:
http://www.r2d3.us/visual-intro-to-machine-learning-part-2/

Dimensionality Reduction

- Find a low-dimensional representation of data that is easy to visualize and ingest in ML algorithms

MNIST Data

10,000 images with 784 features per image

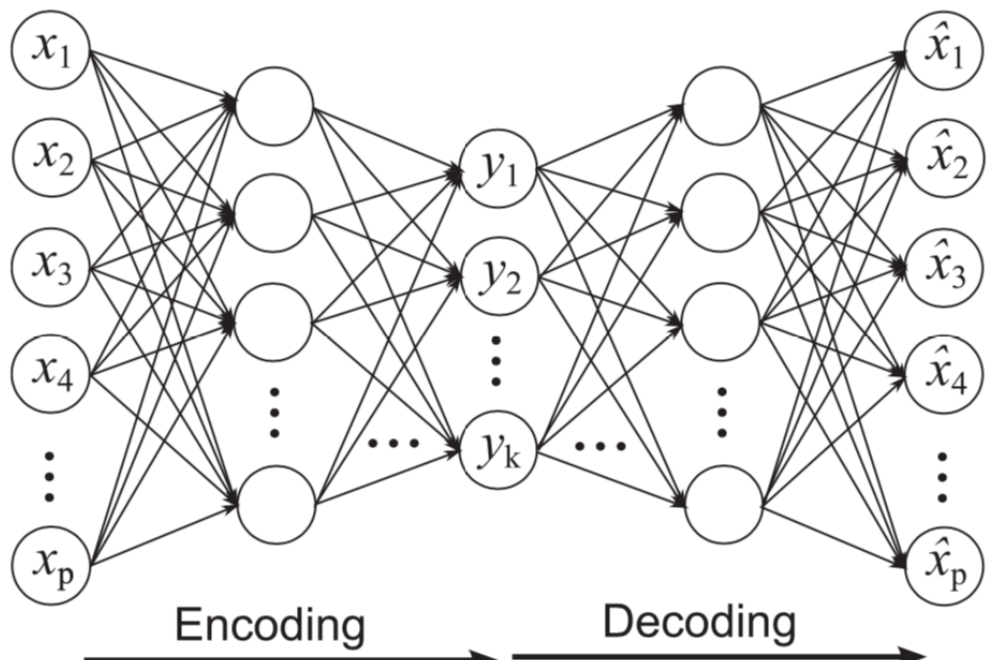


2-dimensional view

http://projector.tensorflow.org/

Autoencoders

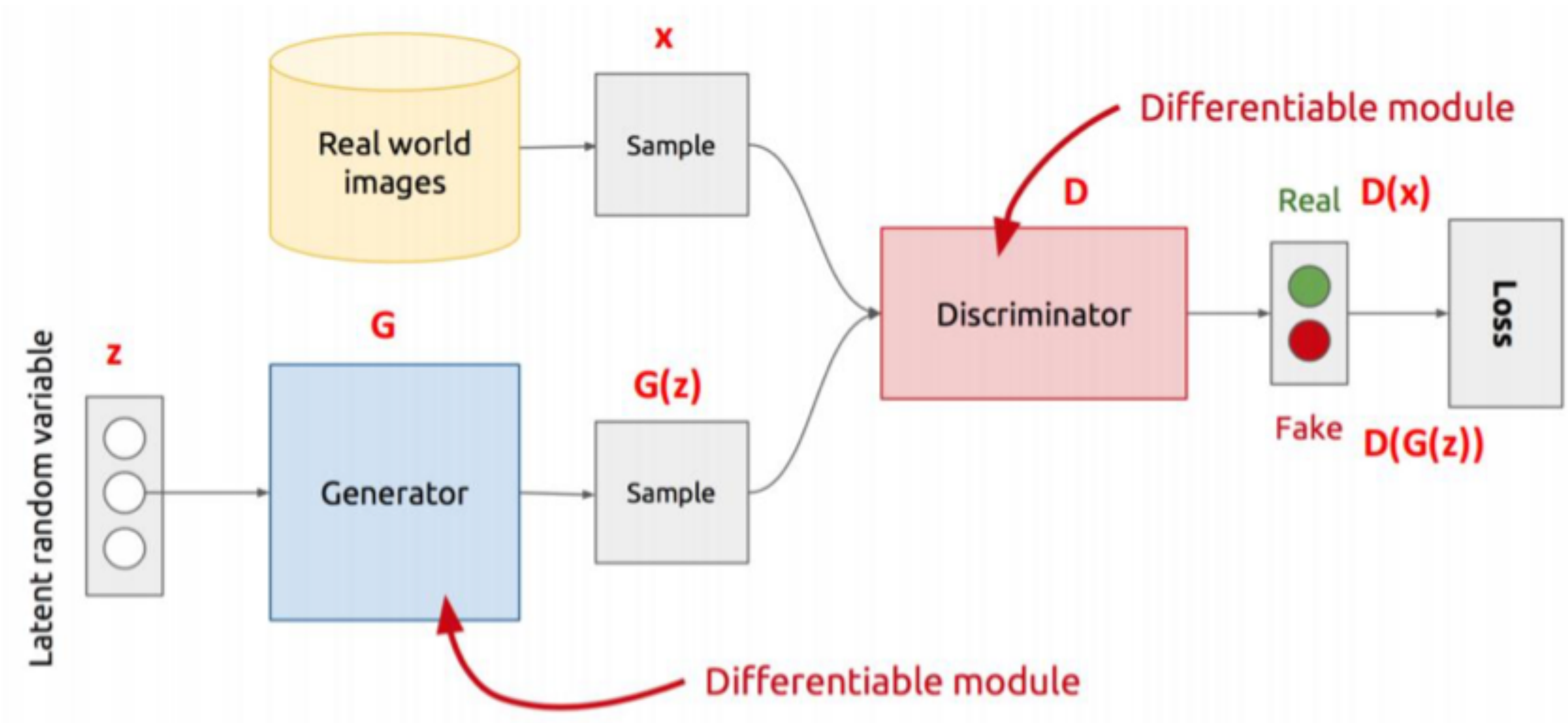
- Objective: learn a latent representation of length k that minimizes reconstruction error over a data set with p attributes (k < p)
- Variants: Variational Autoencoders (VAE)



Generative Modeling:

Generative Adversarial Networks

- GANs can create new data instances that resemble training data, using two parts
  - The **Generator** learns to generate plausible data.
  - The **Discriminator** learns to distinguish the generator's fake data from the real data.
- Variants: Wassertien GAN, conditional GANs (cGANs), pix2pix, cycleGANs, …

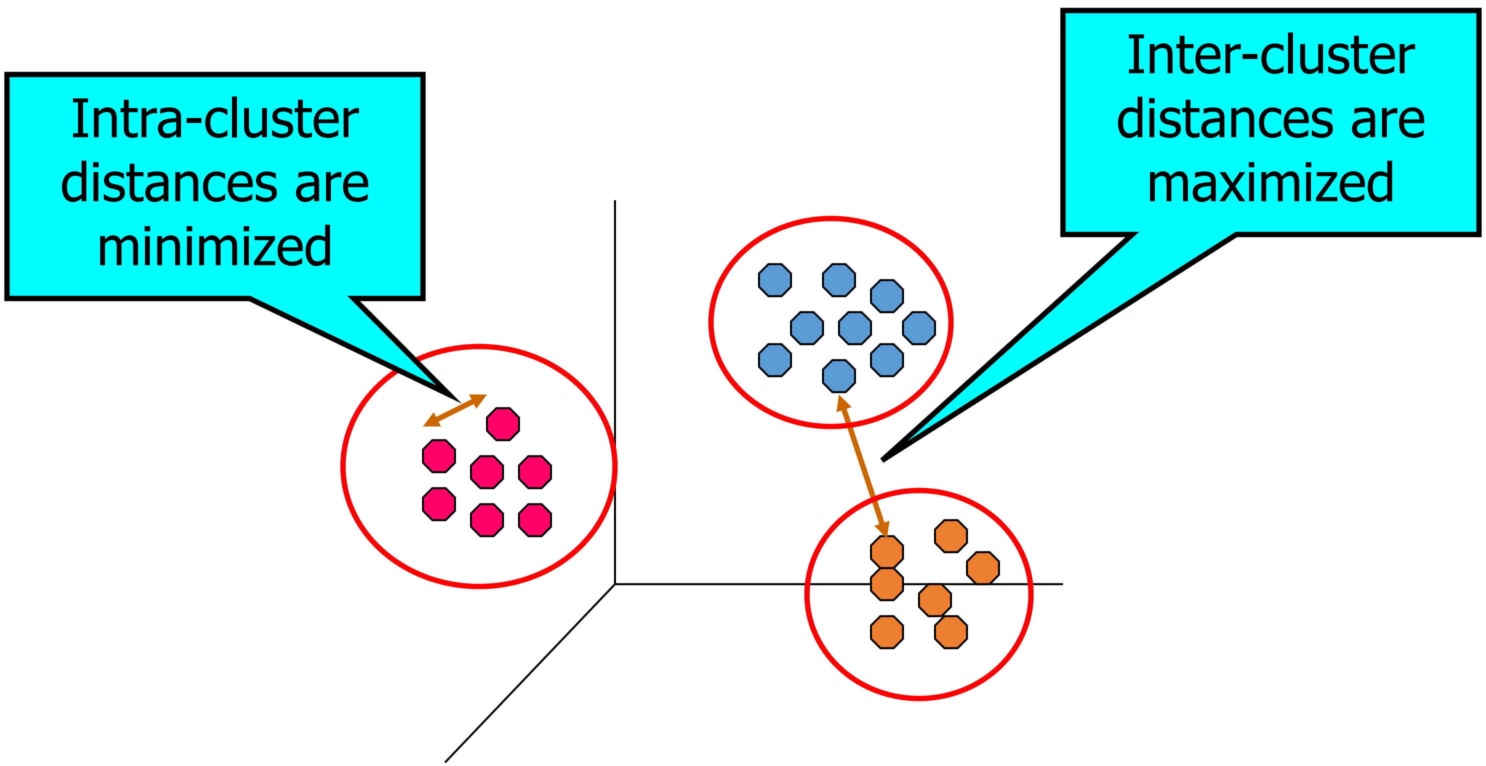


Examples of Images generated by Progressive GANs

Visit the following link to generate faces of people that don’t exist
https://www.thispersondoesnotexist.com/

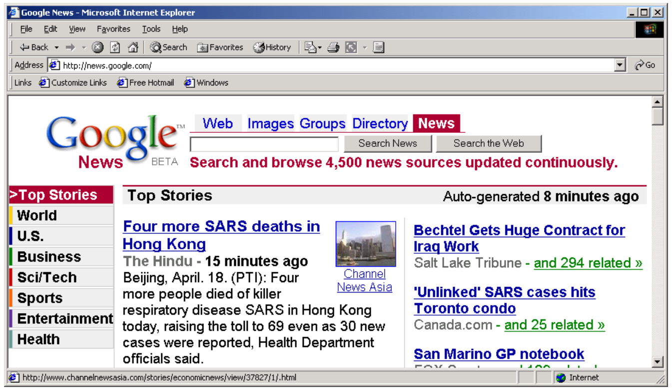
Clustering

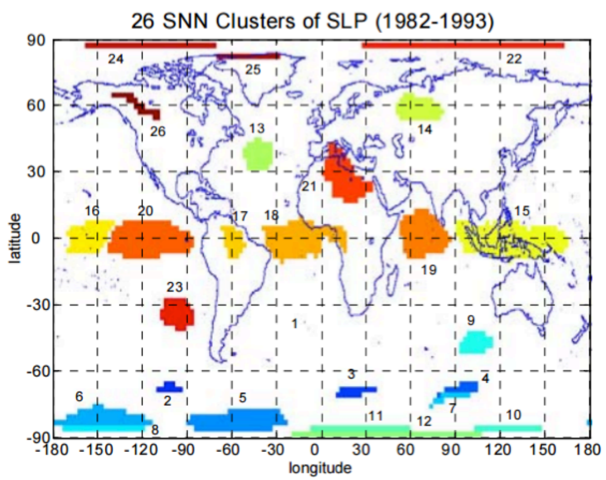
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



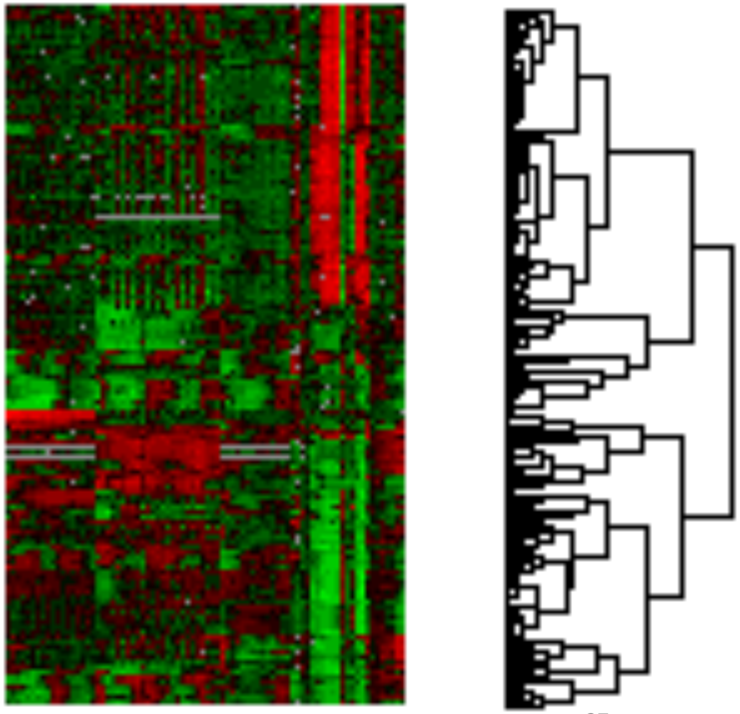
Clustering: Illustrative Examples

- **Understanding**
  - Group related documents for browsing
  - Group genes that have similar functionality
  - Group regions with similar climate activity
- **Summarization**
  - Reduce the size of large data sets





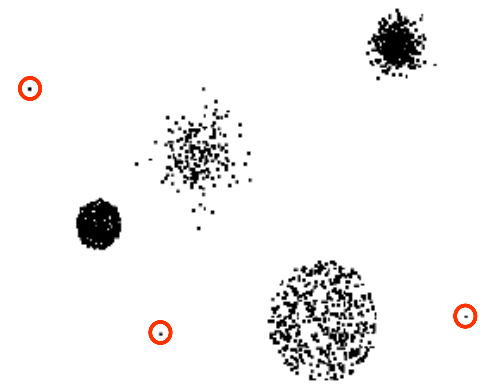
Clusters found using Sea Level Pressure Data



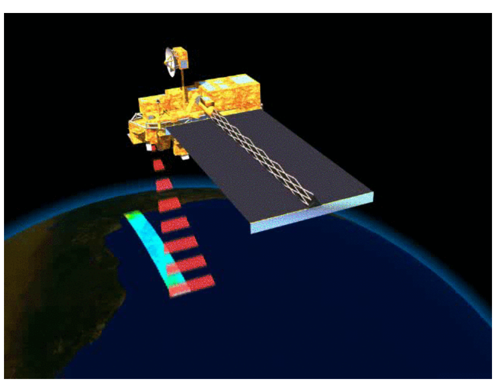
Courtesy: Michael Eisen

Anomaly Detection

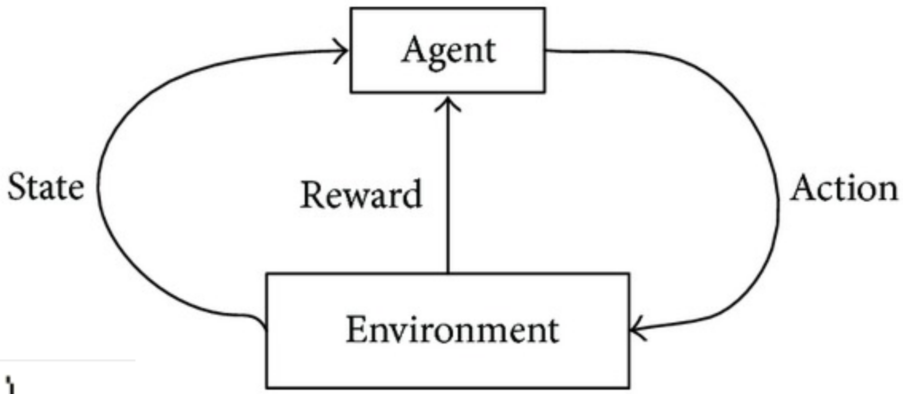
- Detect significant deviations from normal behavior

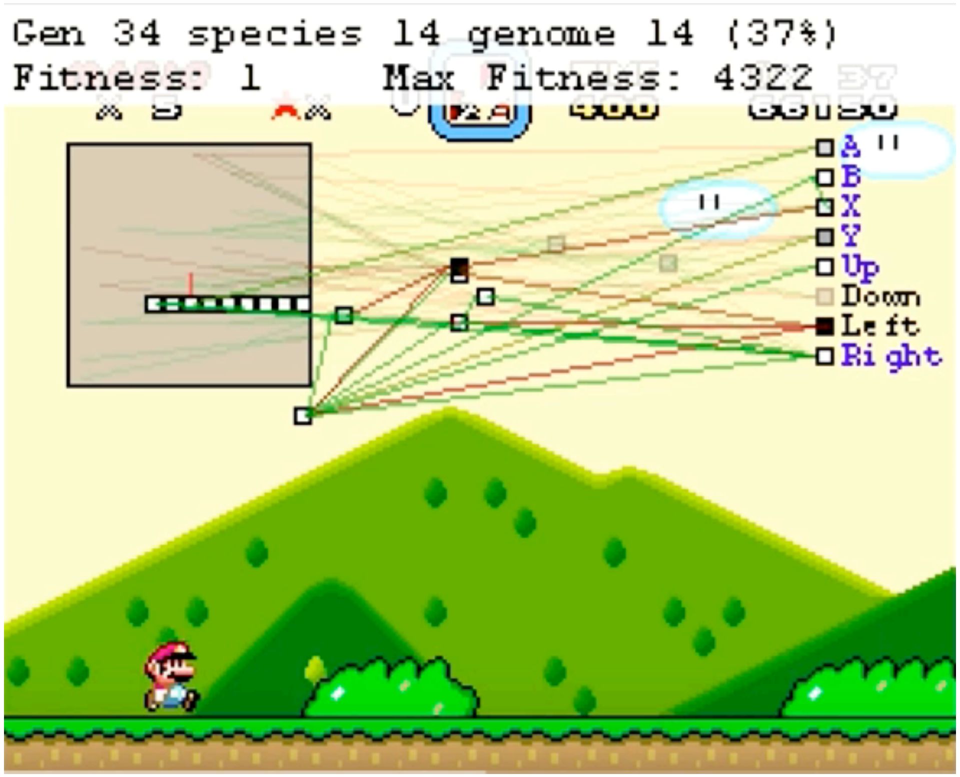


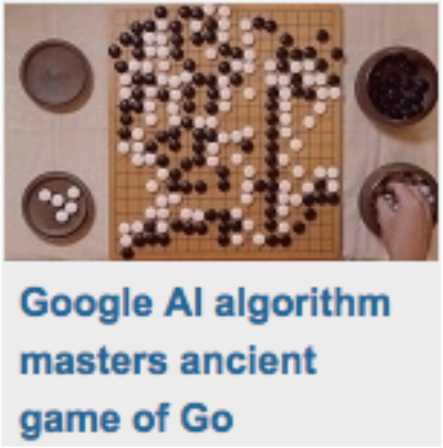
- Applications:
  - Credit Card Fraud Detection
  - Network Intrusion Detection
  - Detecting changes in the Global Forest Cover



Additional Topics: Reinforcement Learning







MarI/O: http://pastebin.com/ZZmSNaHX

Some General Guidelines in Machine Learning

1. Identify type of problem
   - Classification, regression, clustering, anomaly detection, …
2. Select relevant features
   - Feature selection often guided by domain insights
3. Obtain training labels (if needed)
4. Select type of learning algorithm
5. Design evaluation setup
   - Partition data into training and testing and measure test performance
   - Perform sensitivity analysis of learned patterns/models
   - Ensure physical interpretability of discovered results

Great resource for coding ML in Python:
http://www.cse.msu.edu/~ptan/dmbook/software/

Breakout Session (~10 to 15 mins)

- You will be assigned to smaller groups of size 3 to 4
- Goal: To start thinking about potential project ideas involving ML
- Suggested "Ice-breaker" Questions:
  - What are some examples of scientific problems where you can see opportunities to apply ML?
  - What kind of ML formulations (e.g., classification, regression, etc.) can be used?
  - What challenges do you think you will face that will need us to move beyond black-box ML?
- Use Google Docs to capture your conversations using the template:
  - https://docs.google.com/document/d/1PqqduKaQ0lY8RJb7eSlsDB_FhLhiOlKcD_1g1B8PTNs/edit?usp=sharing (also available on course webpage)
- Reach out to me anytime by clicking on "Ask for Help (?)"
- Share discussion highlights with the class at the end of session

Follow-up Assignment 1

- Tell us about one scientific problem where you can see an application of ML
  - What type of ML formulation?
    - Classification, Regression, Dimensionality Reduction, Generative Modeling, Clustering, Anomaly Detection, Reinforcement Learning, Optimization, …
  - Where will you find data?
  - What challenges will black-box ML face in this problem and what kind of scientific knowledge can be used?
    - In SGML theme areas, e.g., in selection of features, design and learning of ML models, or analysis of results
- Will be available on Canvas by tonight (due Aug 31)

What is Coming Up in Next Class?

- Introduction to SGML Theme Areas

- Suggested Readings for Next Week:
  - A. Karpatne, G. Atluri, J. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided Data Science: A New Paradigm for Scientific Discovery from Data**,"** *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 29(10), 2318–2331, 2017.

  - Willard, Jared, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. "Integrating physics-based modeling with machine learning: A survey." arXiv preprint arXiv:2003.04919 (2020).