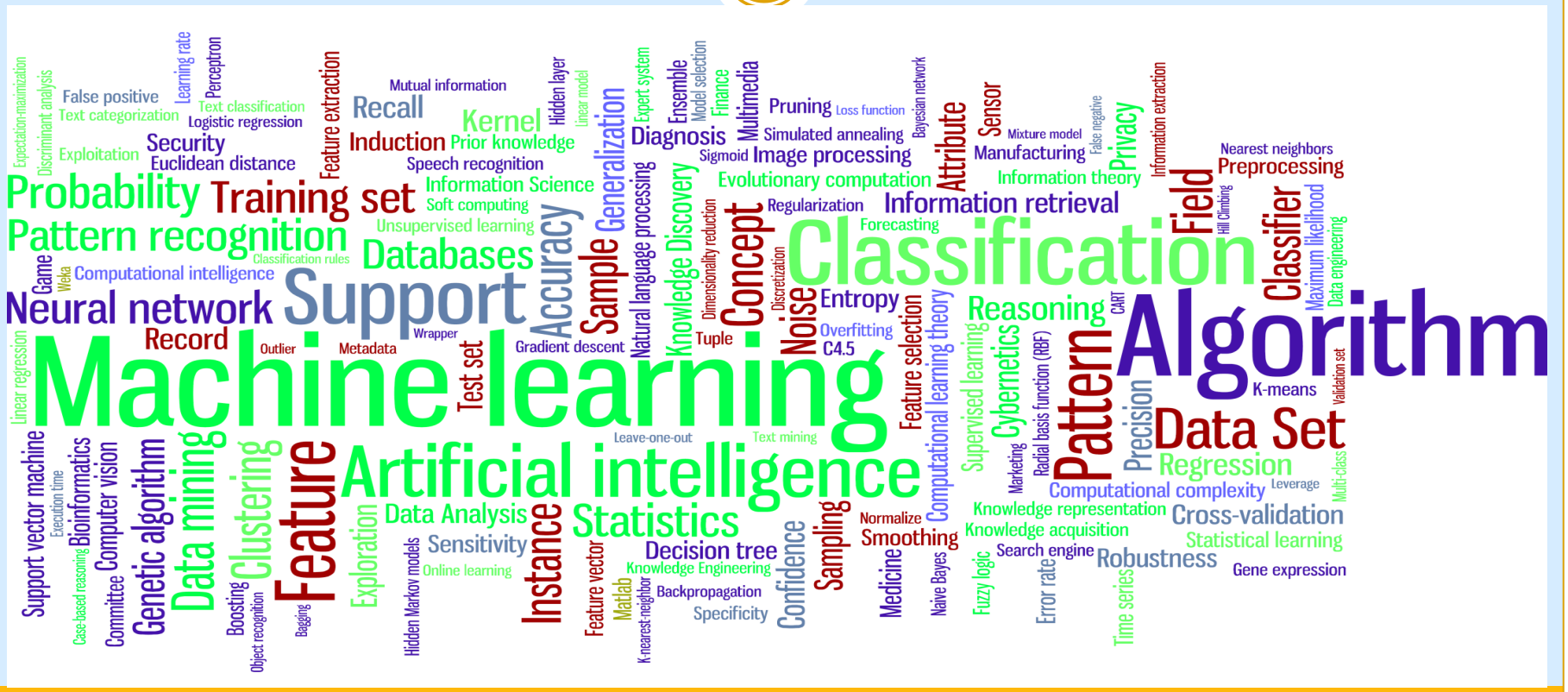
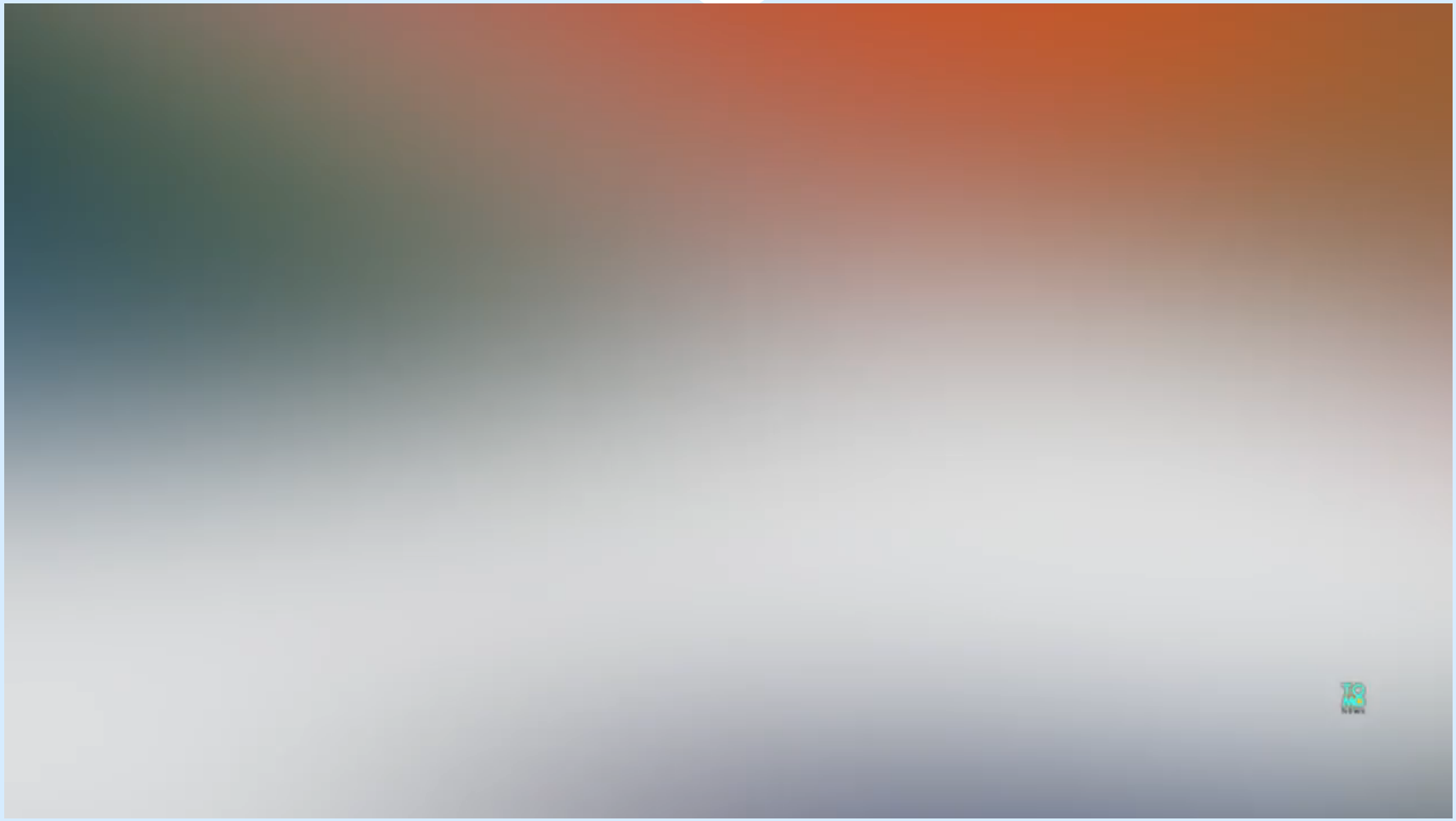
Machine Learning & MLLib



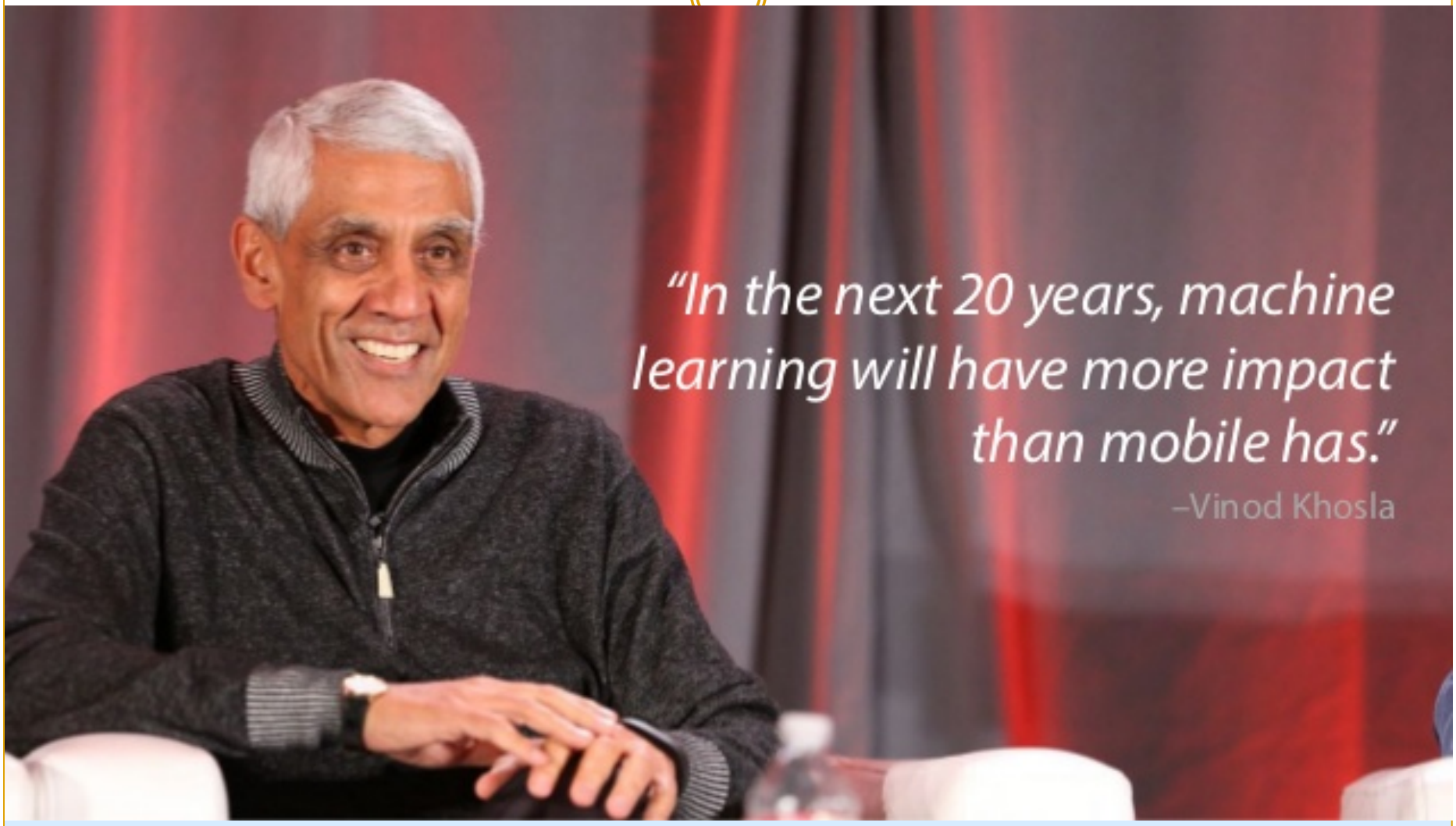
Agenda

- Introduction to Machine Learning
  - Machine Learning
  - Algorithms
- Machine Learning Using Spark MLLib



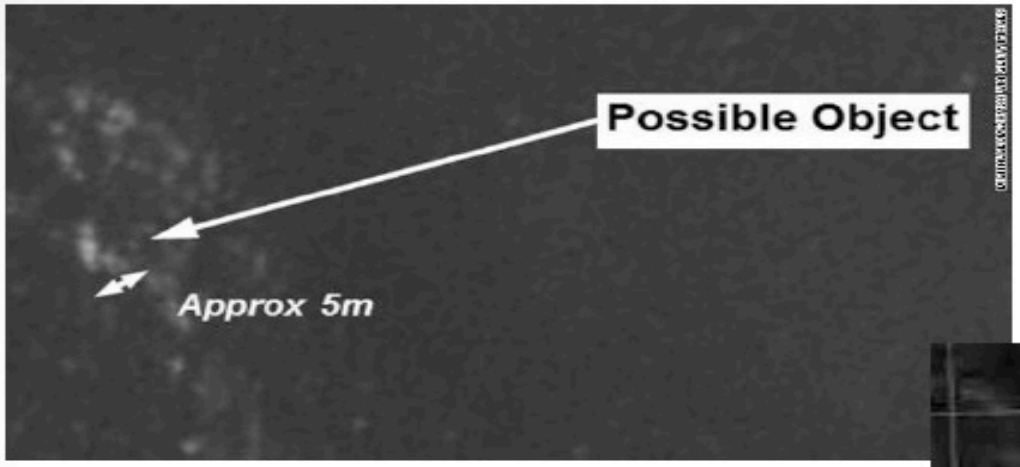
"A breakthrough in machine learning would be worth ten Microsofts"

–Bill Gates


"In the next 20 years, machine learning will have more impact than mobile has."
–Vinod Khosla

Simple Example

How about this?




UNCLASSIFIED
Object 2 Possibly Associated With MH370 Search
Indian Ocean
COORDS: 44:03:02S 091:13:27E
16 Mar 2014
0416Z/1216Local
© DigitalGlobe
Panchromatic
Possible Object
Approx 5m
16 Mar 2014
0416Z/1216Local
© DigitalGlobe
Multispectral
Possible Object
Approx 5m
N
DIGO-00718-02-14

Questions to Ponder

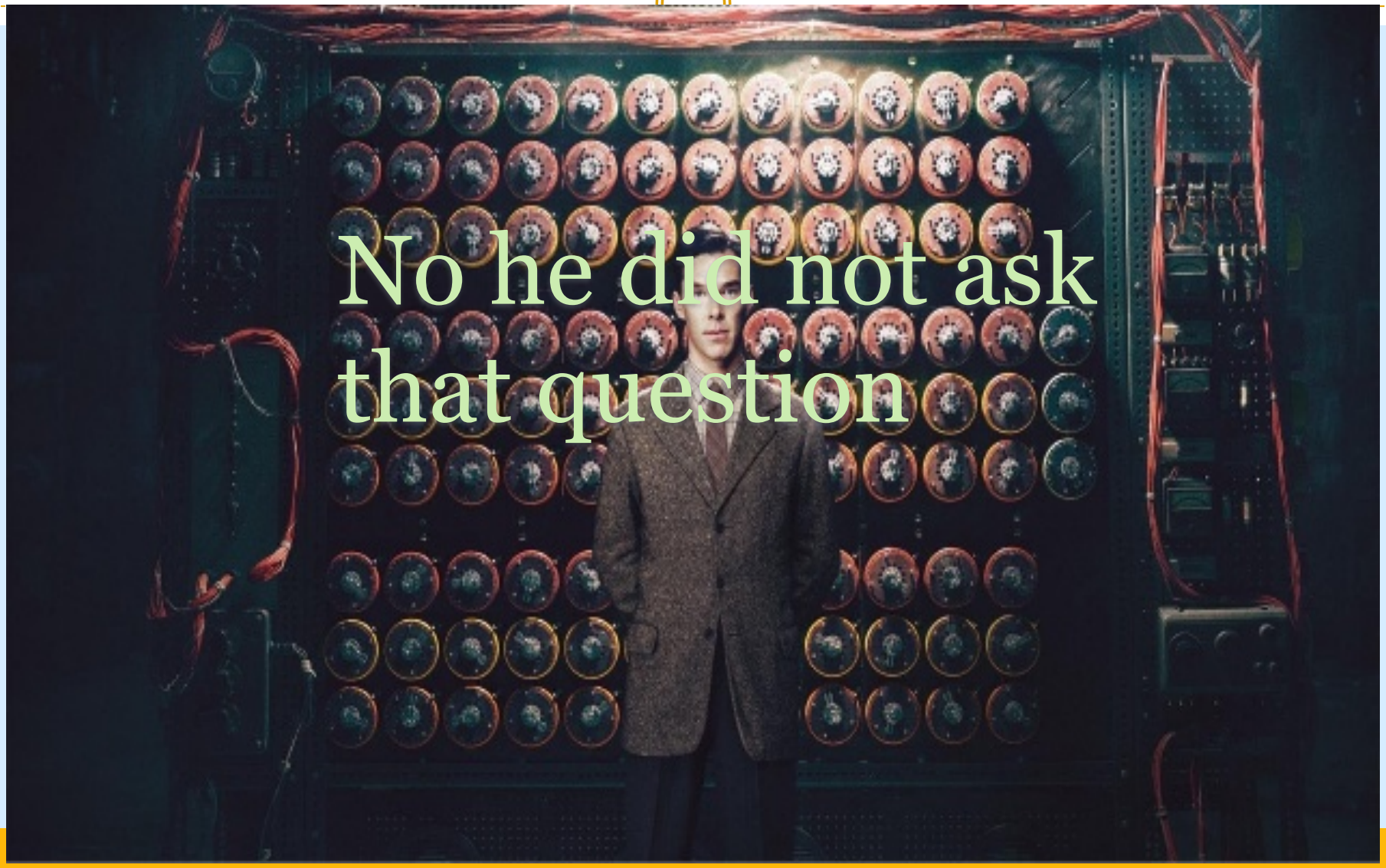
- How did your brain process the images, and group them?
- We call that part as learning
  - Learning could be visual, hearing, sense, touch etc based
  - After learning, we look at new images, and compare with groups classified during earlier phase
    - This is prediction/forecasting
- This is how humans have evolved…we are natural at this
- Machines..not so much.
- ML is that entire branch that attempts to make it possible

Idea behind ML

- So how old is ML?
  - Literally as old as computing itself
- Idea originated with a simple question:

- CAN MACHINES THINK?

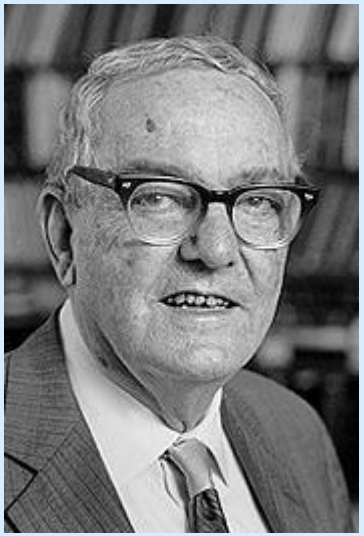
Remember this guy?



He did….in 1950!!! (Alan Turing)

Machine Learning

- **Herbert Alexander Simon**: "Learning is any process by which a system improves performance from experience."
- "Machine Learning is concerned with computer programs that automatically improve their performance through experience. "



Herbert Simon
Turing Award 1975
Nobel Prize in Economics 1978

Why Machine Learning?

- Develop systems that can automatically adapt and customize themselves to individual users.
  - Personalized news or mail filter
- Discover new knowledge from large databases (***data mining***).
  - Market basket analysis (e.g. diapers and beer)
- Ability to mimic human behavior and replace certain monotonous tasks - which require some intelligence.
  - like recognizing handwritten characters

Machine Learning

- Systems learn with data
  - Different from traditional paradigm
  - Logic learnt from data
- Computation intensive
- Ubiquitous
  - Apple (Siri), Google (Search), Amazon (Recommendations), FB
- Eventual goal is to predict or classify

Why now?

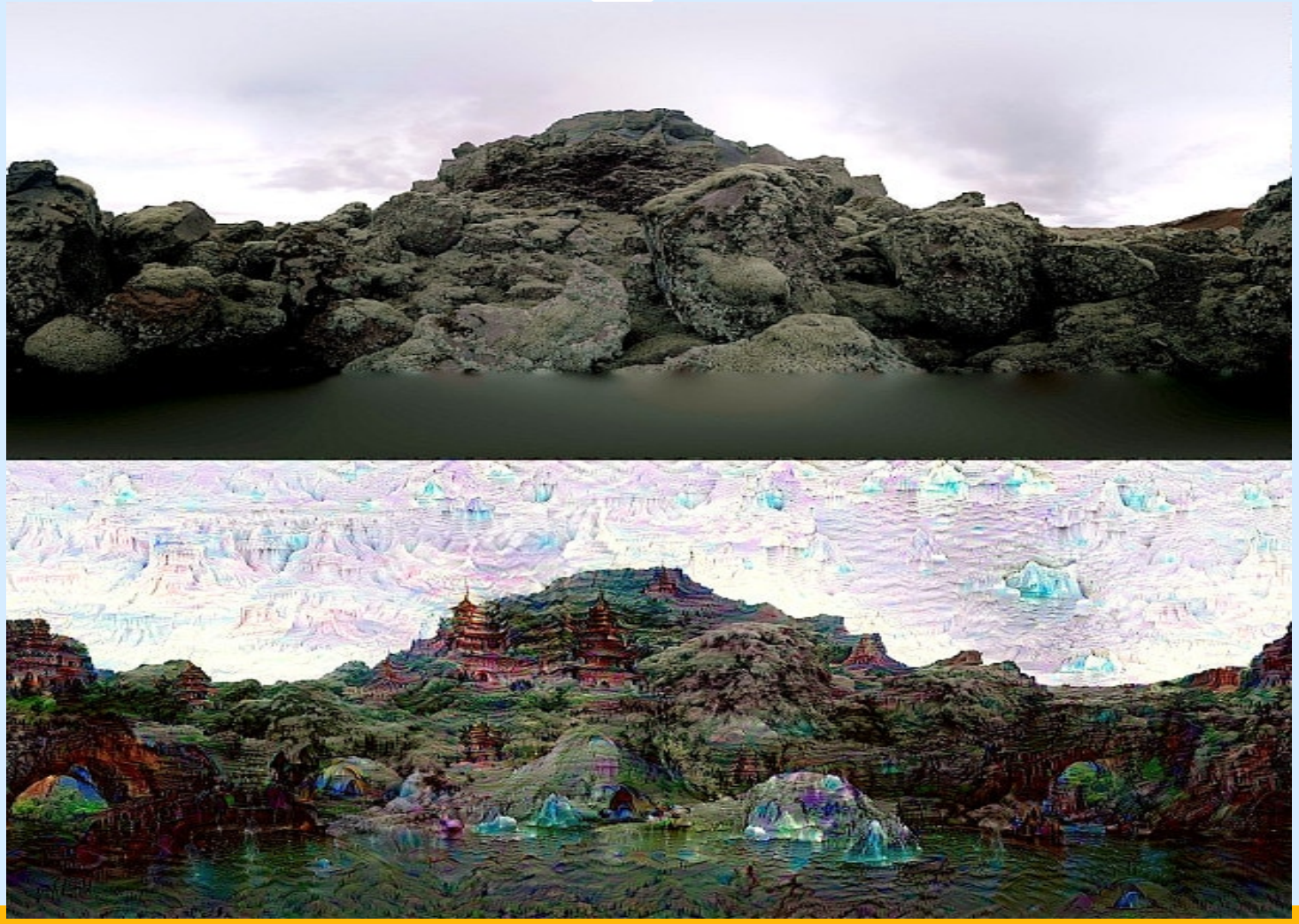
- Flood of available data (especially with the advent of the Internet)
- Increasing computational power
- Growing progress in available algorithms and theory developed by researchers
- Increasing support from industries

ML Applications

So what are the buzzwords these days?

- Machine Learning?
  - Nice
- Neural Networks?
  - Good
- Deep Learning?
  - Great
- Machine Dreaming??????????
  - ☹

Detour: Machine Dreaming



The concept of learning in a ML system

- Learning = Improving with experience at some task
  - Improve over task *T*,
  - With respect to performance measure, *P*
  - Based on experience, *E*.

Example: Spam Filter

Example: Spam Filtering
Spam - is all email the user does not want to receive and has not asked to receive

T: Identify Spam Emails

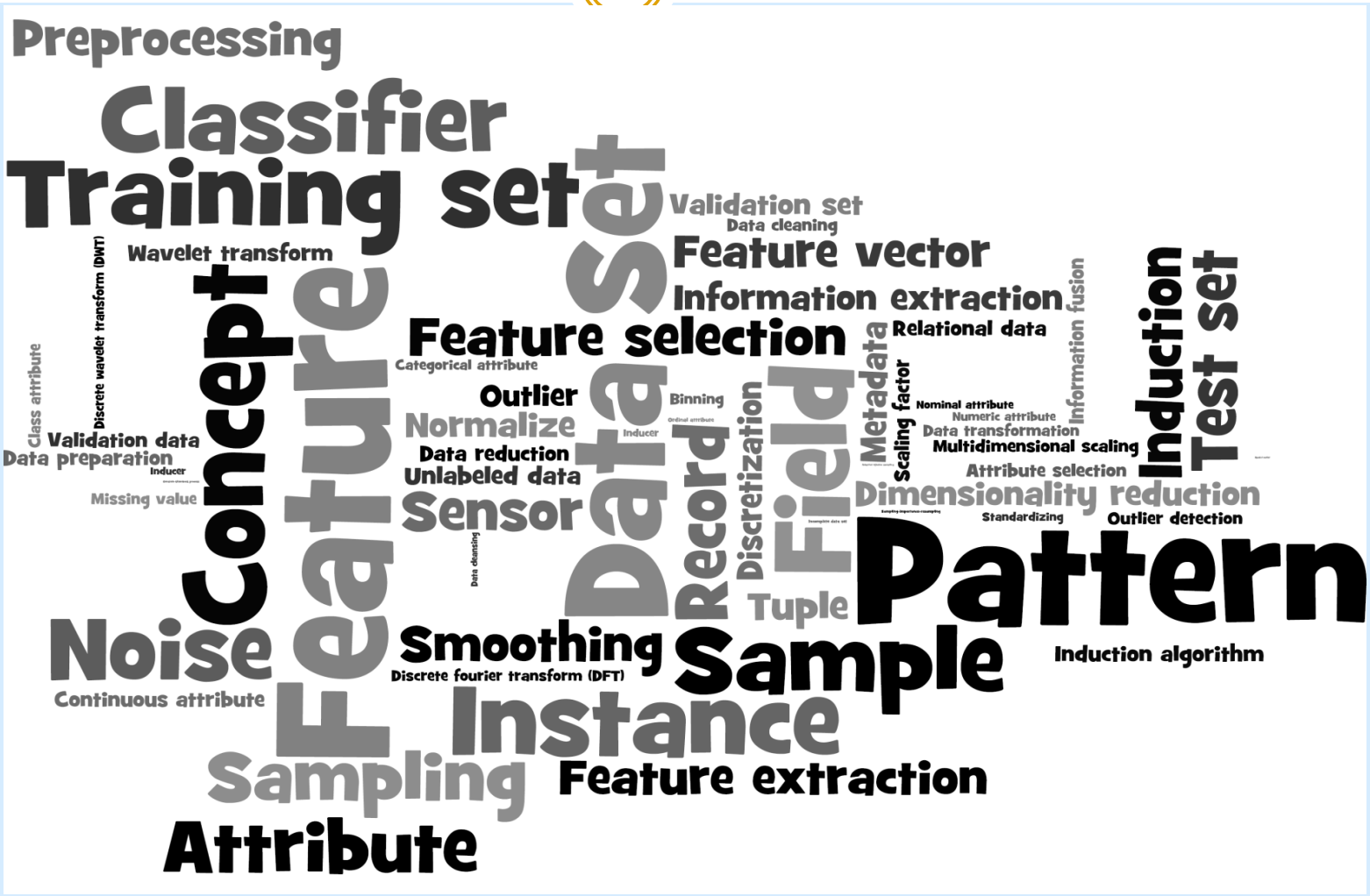
P:

% of spam emails that were filtered

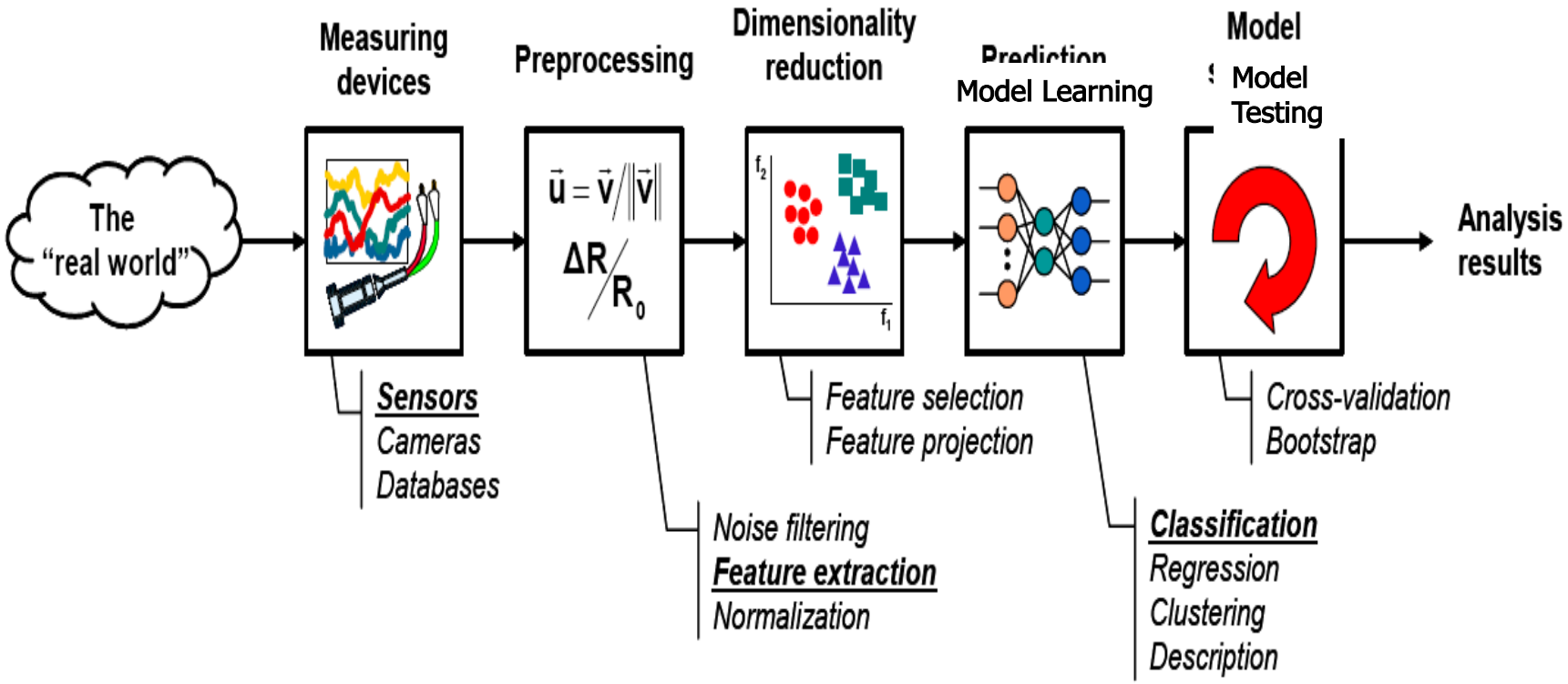
% of ham/ (non-spam) emails that were incorrectly filtered-out

E: a database of emails that were labelled by users

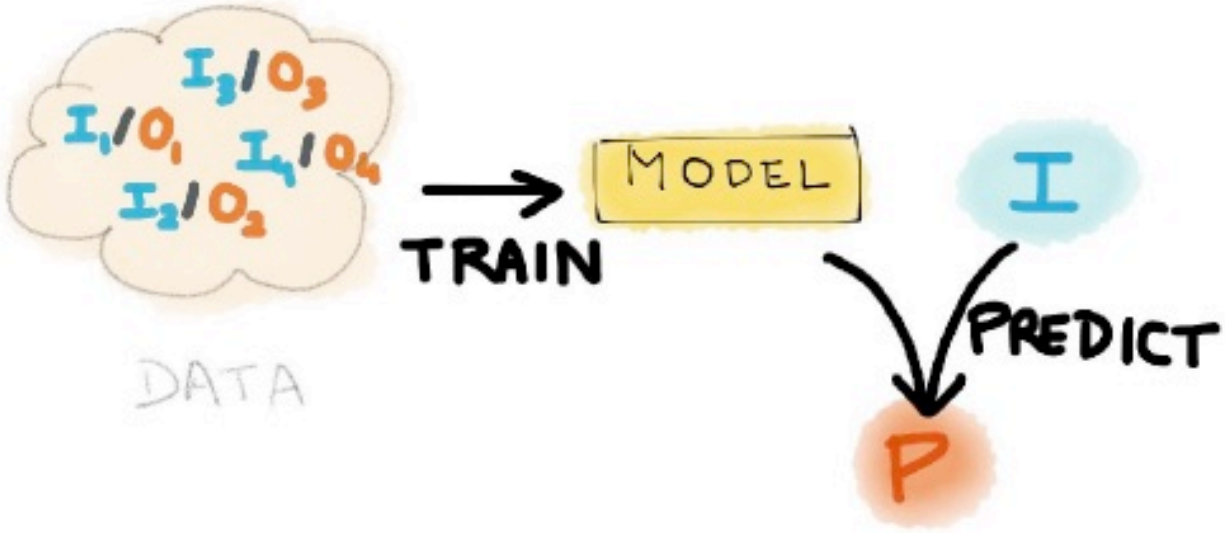
Language of ML



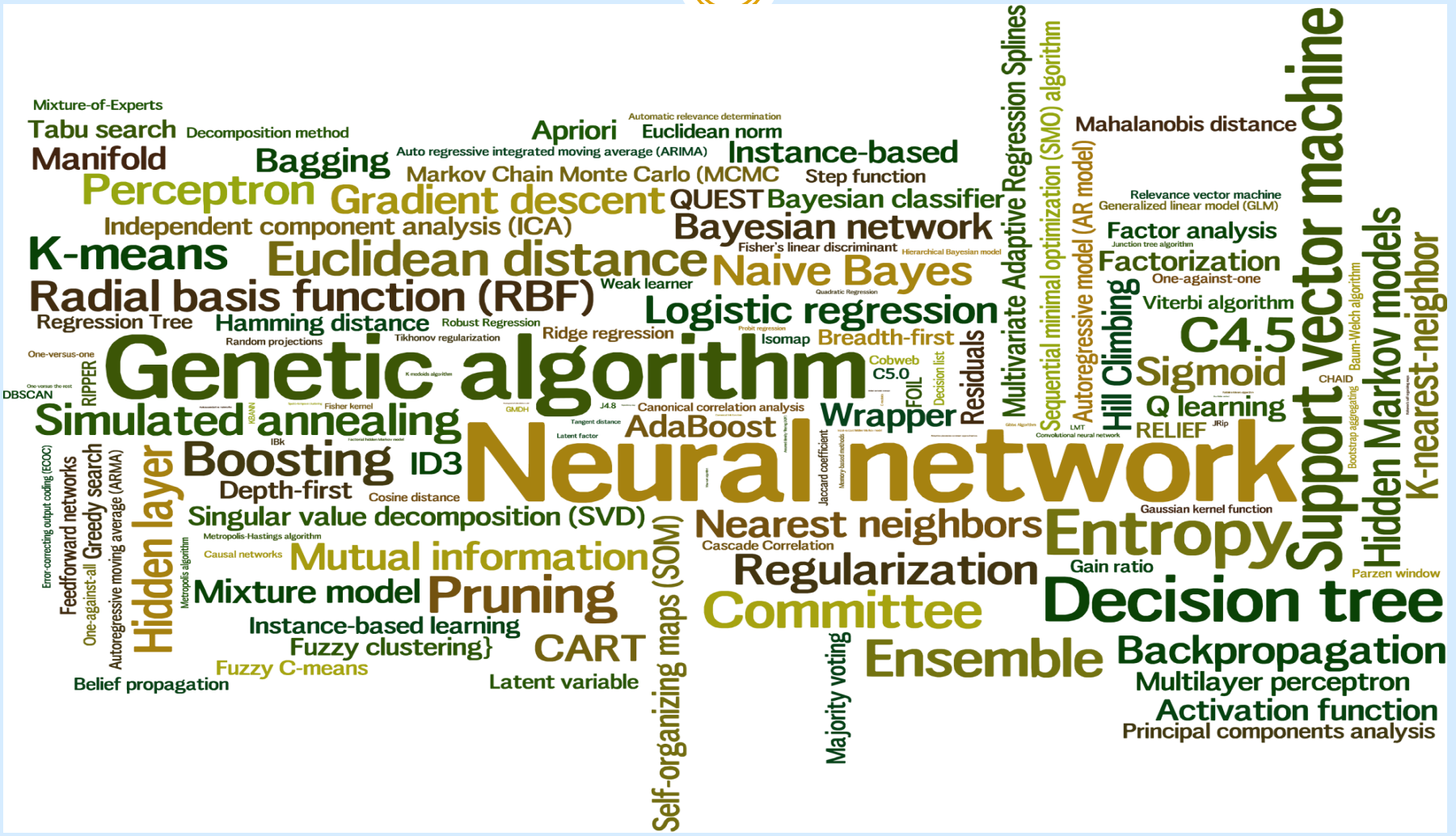
The Learning Process



Simplify....



World of ML Algos



Connect between ML and Statistics/Probability

- ML algorithms are based on a bunch of assumptions
- Eg. Data distribution: Normal, linear/non-linear
- They are a simplified representation of reality
- ML algos tell us how likely a given result is?
  - This means results come to us not as Yes/No but probability of an observation being Yes/No
  - This probability output is result of assumptions and actual algorithm
- No ML algo can exist without probability and statistical concepts

Terminology

- Feature
  - Attribute or a variable
  - Table – row is observation, and column a feature
  - Dimensionality
  - Categorical or numerical
- Label
  - Dependent variable
  - To be predicted
  - Numerical or categorical
- Model
  - Mathematical representation of relationship betn variables

Machine Learning

- Machine learning algo fits model to data
- Computationally intensive
- Train and test data
- Applications
  - Classification
  - Regression
  - Clustering
  - Recommendation

(Broad) Categories of ML

- Supervised (Classification or Prediction)
  - First, train the algorithm to create a model.
  - What is training?
  - Data comes along with output as well.
  - New data is labelled based on model created from training data.
- Unsupervised (Clustering)
  - We do not know output, just have a bunch of observations
  - What we try to do is keep them in buckets according to certain similarities
- Reinforcement based (think ANN, Deep Learning)
  - Learning based on feedback or reward

What if only this was given?

Age	Spectacle prescription	Astigmatism	Tear production rate
Young	Myope	No	Reduced
Young	Myope	No	Normal
Young	Myope	Yes	Reduced
Young	Myope	Yes	Normal
Young	Hypermetrope	No	Reduced
Young	Hypermetrope	No	Normal
Young	Hypermetrope	Yes	Reduced
Young	Hypermetrope	Yes	Normal
Pre-presbyopic	Myope	No	Reduced
Pre-presbyopic	Myope	No	Normal
Pre-presbyopic	Myope	Yes	Reduced
Pre-presbyopic	Myope	Yes	Normal
Pre-presbyopic	Hypermetrope	No	Reduced
Pre-presbyopic	Hypermetrope	No	Normal
Pre-presbyopic	Hypermetrope	Yes	Reduced
Pre-presbyopic	Hypermetrope	Yes	Normal
Presbyopic	Myope	No	Reduced
Presbyopic	Myope	No	Normal
Presbyopic	Myope	Yes	Reduced
Presbyopic	Myope	Yes	Normal
Presbyopic	Hypermetrope	No	Reduced
Presbyopic	Hypermetrope	No	Normal
Presbyopic	Hypermetrope	Yes	Reduced
Presbyopic	Hypermetrope	Yes	Normal

Example

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Supervised

```
If tear production rate = reduced then recommendation = none
If age = young and astigmatic = no
    and tear production rate = normal then recommendation = soft
If age = pre-presbyopic and astigmatic = no
    and tear production rate = normal then recommendation = soft
If age = presbyopic and spectacle prescription = myope
    and astigmatic = no  then recommendation = none
If spectacle prescription = hypermetrope and astigmatic = no
    and tear production rate = normal then recommendation = soft
If spectacle prescription = myope and astigmatic = yes
    and tear production rate = normal then recommendation = hard
If age young and astigmatic = yes
    and tear production rate = normal then recommendation = hard
If age = pre-presbyopic
    and spectacle prescription = hypermetrope
    and astigmatic = yes then recommendation = none
If age = presbyopic and spectacle prescription = hypermetrope
    and astigmatic = yes then recommendation = none
```

Supervised Learning

- Idea is to learn like humans from past experience
- But machines do not have any experience
- Experience here comes from data
- So what do we learn
  - Attempt to solve a target function
  - In the process we would come up with a formula/decision rules

Supervised and Unsupervised

- Supervised
  - Data comes with a y “output” variable.
  - Algorithm is learning under the supervision
- Unsupervised
  - There is no y variable
  - Task is try and find some sort of clusters or classes in the data

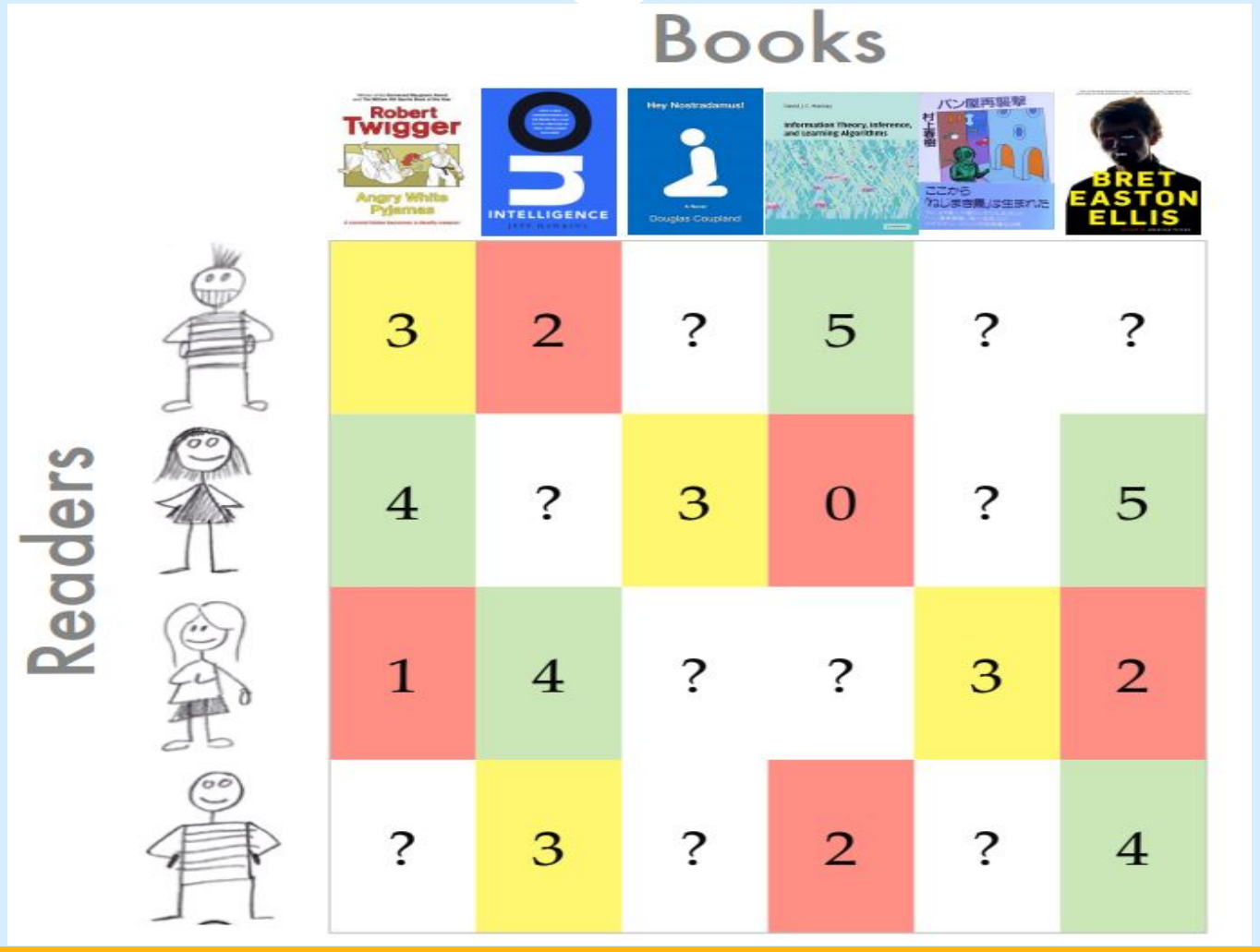
Supervised Algorithms

- Trains model with labeled dataset
- Create model based on training data and use it to classify new data
- Largely, classification and prediction
- Prediction
  - Predict numerical variable (generally continuous valued)
  - Linear regression, decision tree, ensemble
- Classification
  - Predict categorical variables (discrete or nominal)
  - Logistic regression, Neural network, Naive Bayes
- Recommendations

Classification

- “Which category of products is most interesting to this customer?”
- “Is this movie a romantic comedy, documentary, or thriller?”
- “Is this review written by a customer or a robot?”
- “Will the customer buy this product?”
- “Is this email spam or not spam?”

Recommendation



Recommendation

- “Which movies should be recommended to a user?”
- “If the user just listened to a song, which song would he like now?”
- “Which news articles are relevant for a user in a particular context?”
- “Which advertisements should be displayed for a user on a mobile app?”
- “Which products are frequently bought together?”

Two step process

- It is a two-step process:
  - First, generate model using training dataset
  - Second, once the model is generated then predict new observations using that model

Before you run an algo, you do feature engg

- Derive new features from the initial data set:

1. Aggregations: Count, Sum, Min, Max, Avg, Std
2. Temporal: Elapsed time, Trends
3. Continuous to Categorical: Converting real values to enumerations.
4. Categorical to Binary: Converting enumerations to binary features.
5. Domain-specific derived features.

You have a huge menu to choose from

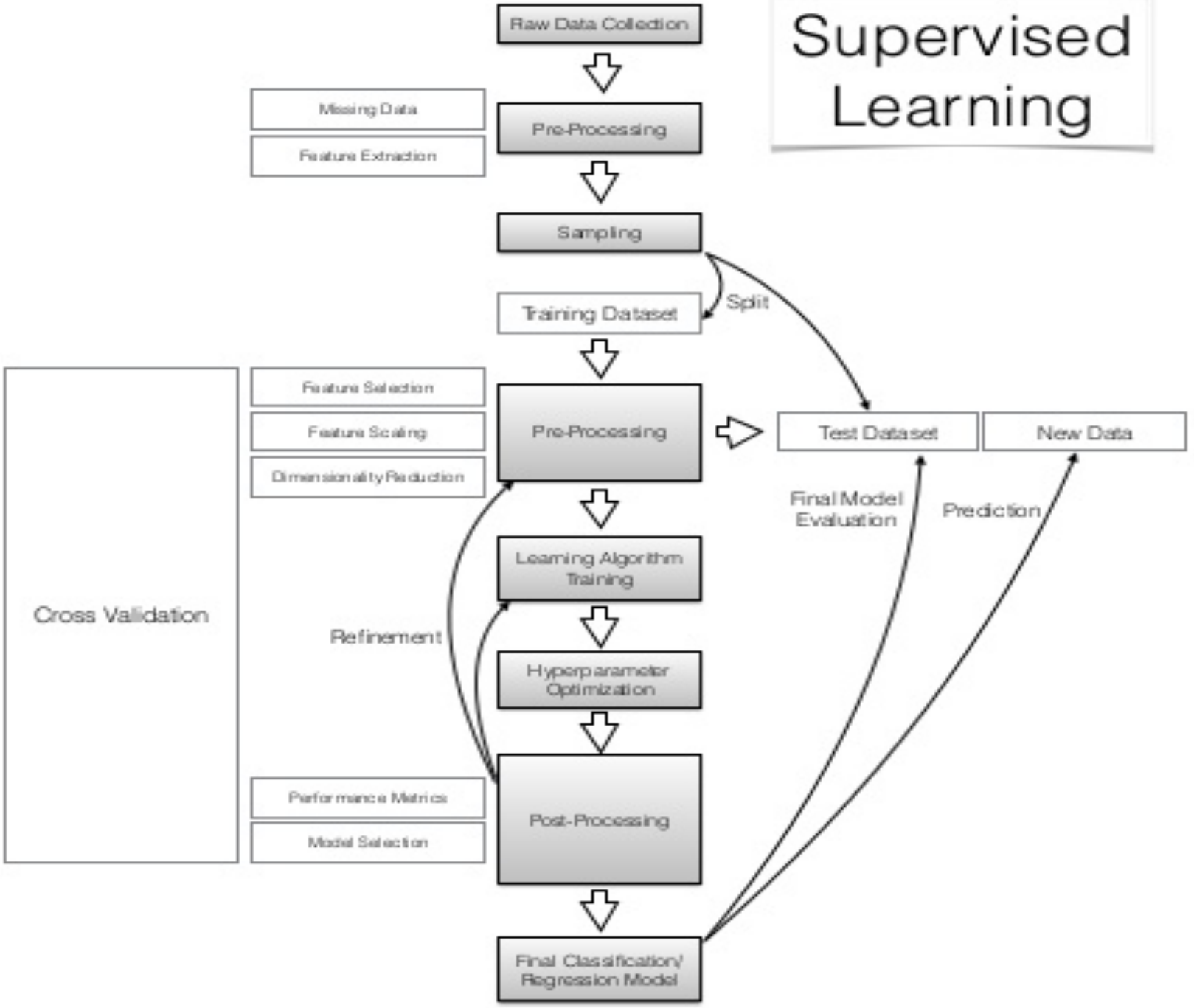
Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised

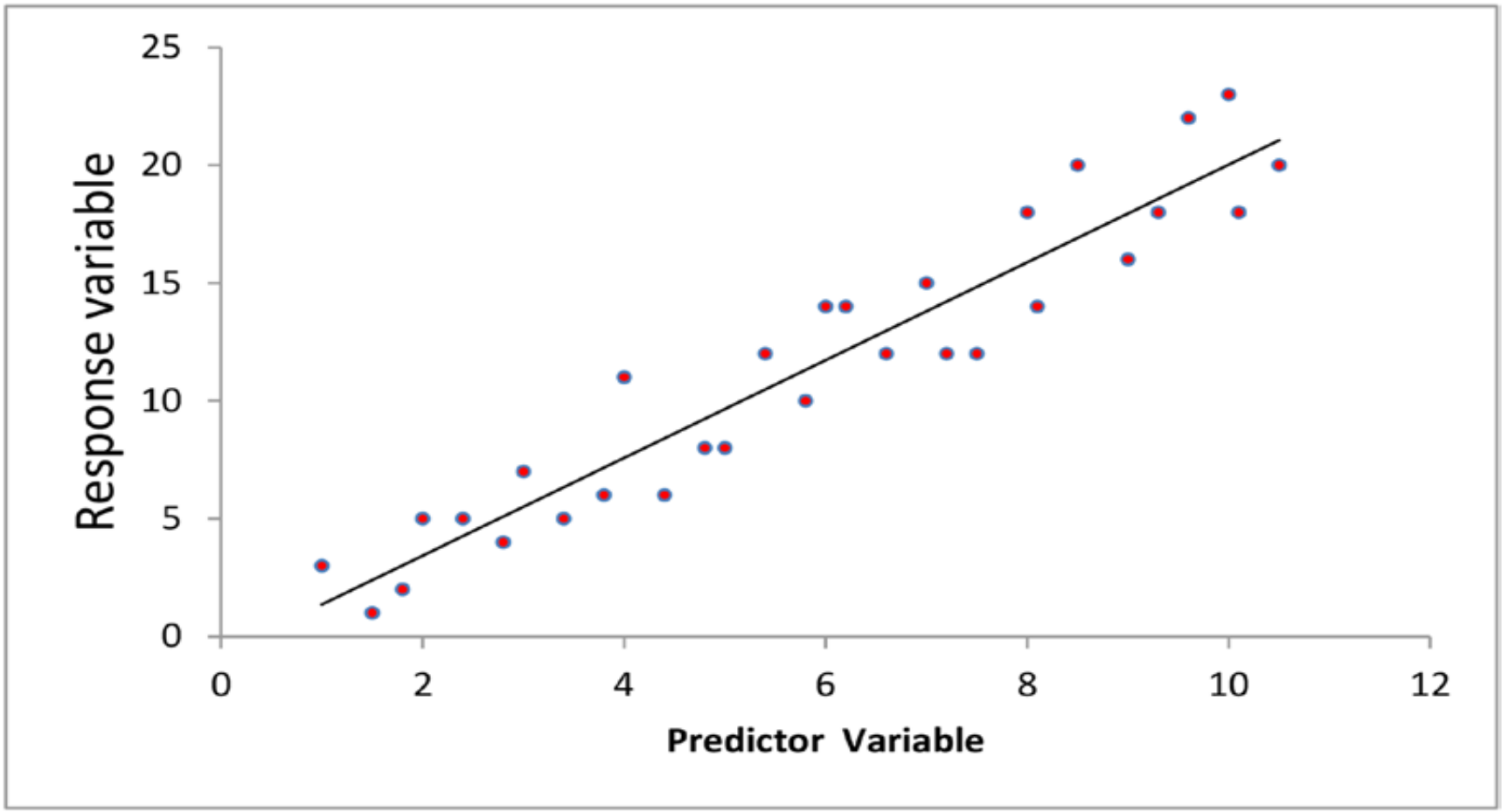
Continuous	• Clustering & Dimensionality Reduction
	  ○ SVD
	  ○ PCA
	  ○ K-means
		• Regression
		  ○ Linear
		  ○ Polynomial
		• Decision Trees
		• Random Forests

Categorical	• Association Analysis
	  ○ Apriori
	  ○ FP-Growth
	• Hidden Markov Model
		• Classification
		  ○ KNN
		  ○ Trees
		  ○ Logistic Regression
		  ○ Naive-Bayes
		  ○ SVM

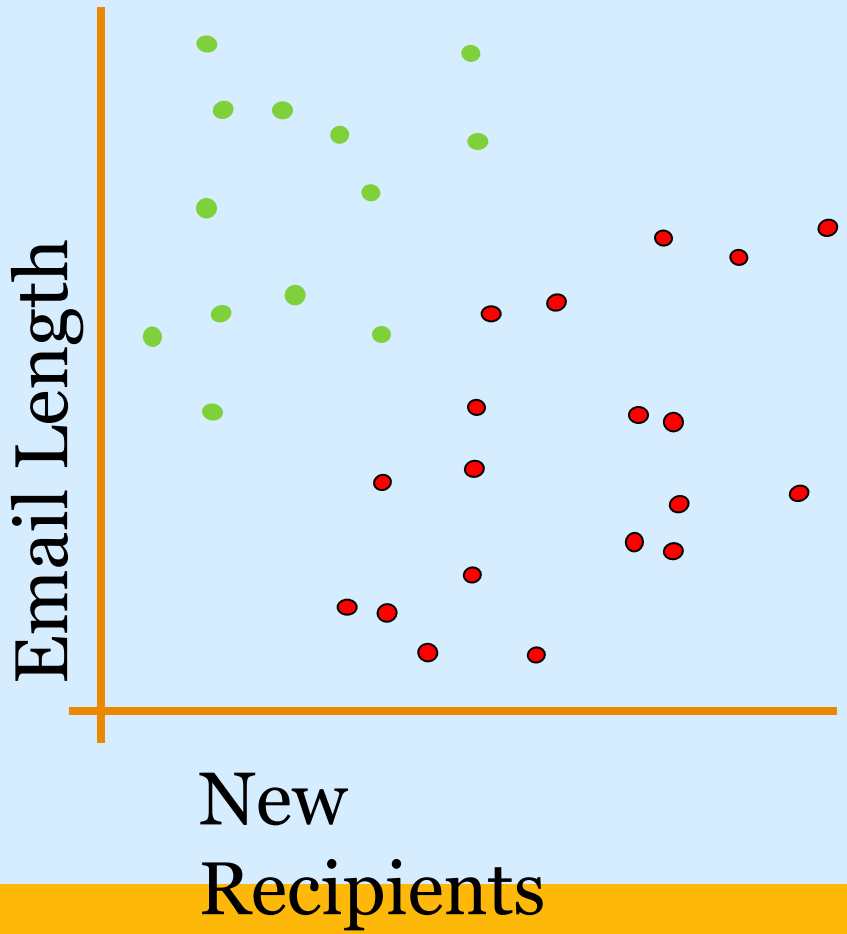
Supervised Learning



Linear Regression

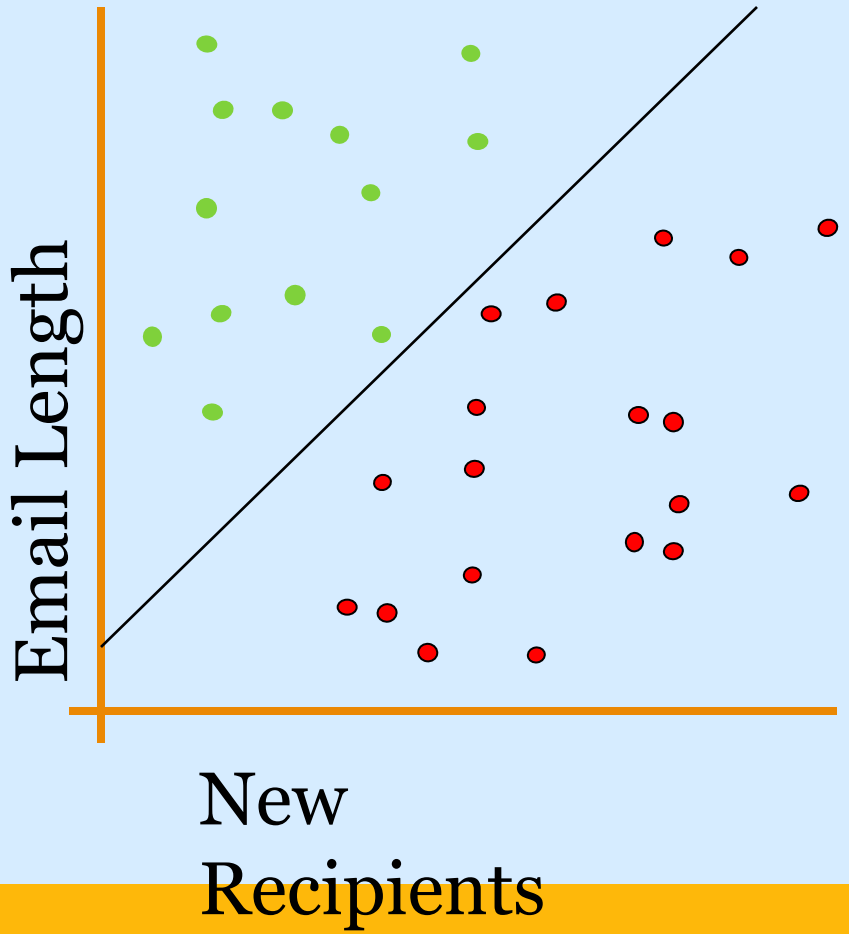


Classifiers



How would you classify this data?

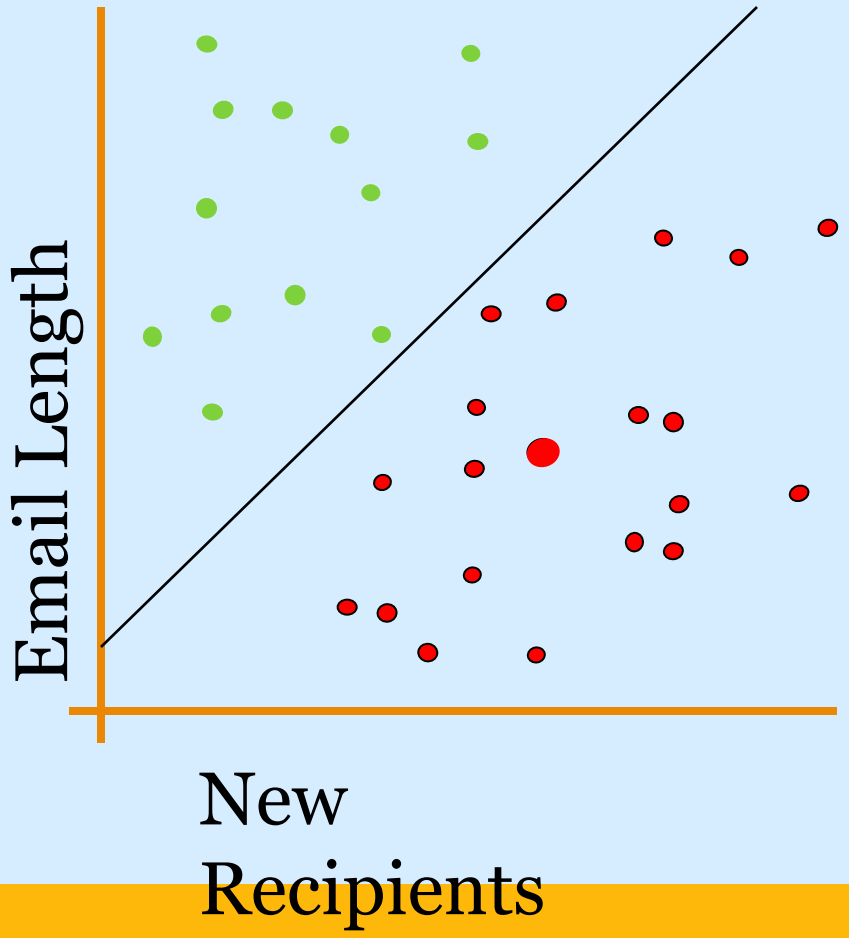
Classifiers



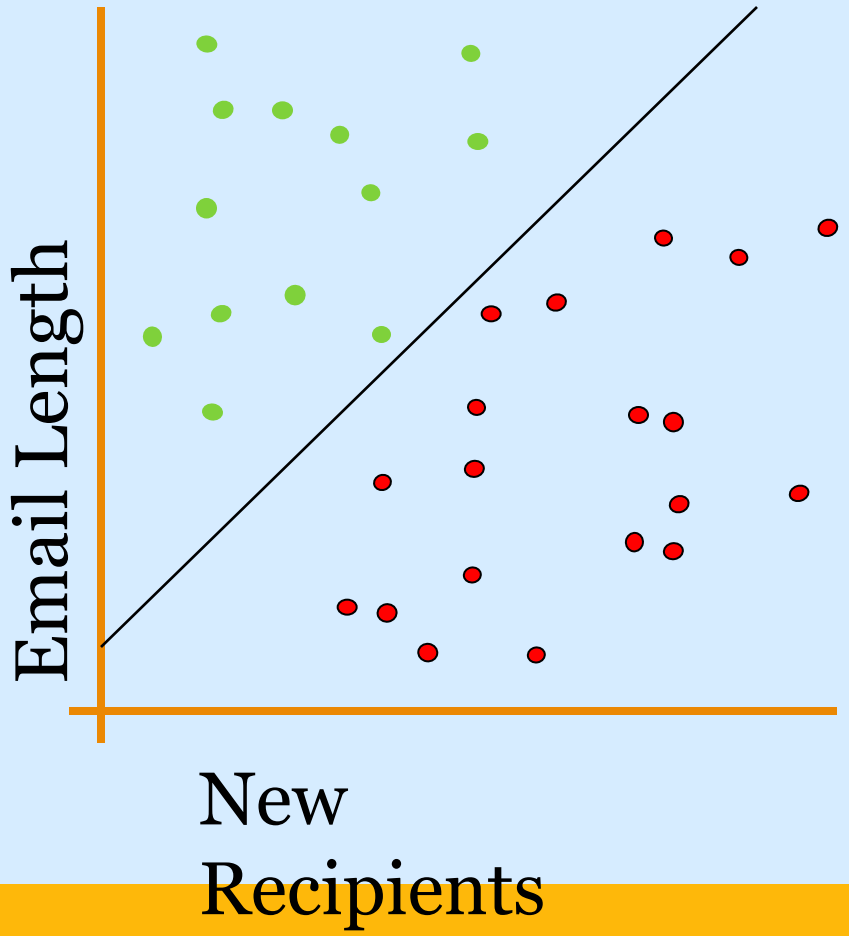
How would you classify this data?

When a new email is sent

1. We first place the new email in the space
2. Classify it according to the subspace in which it resides

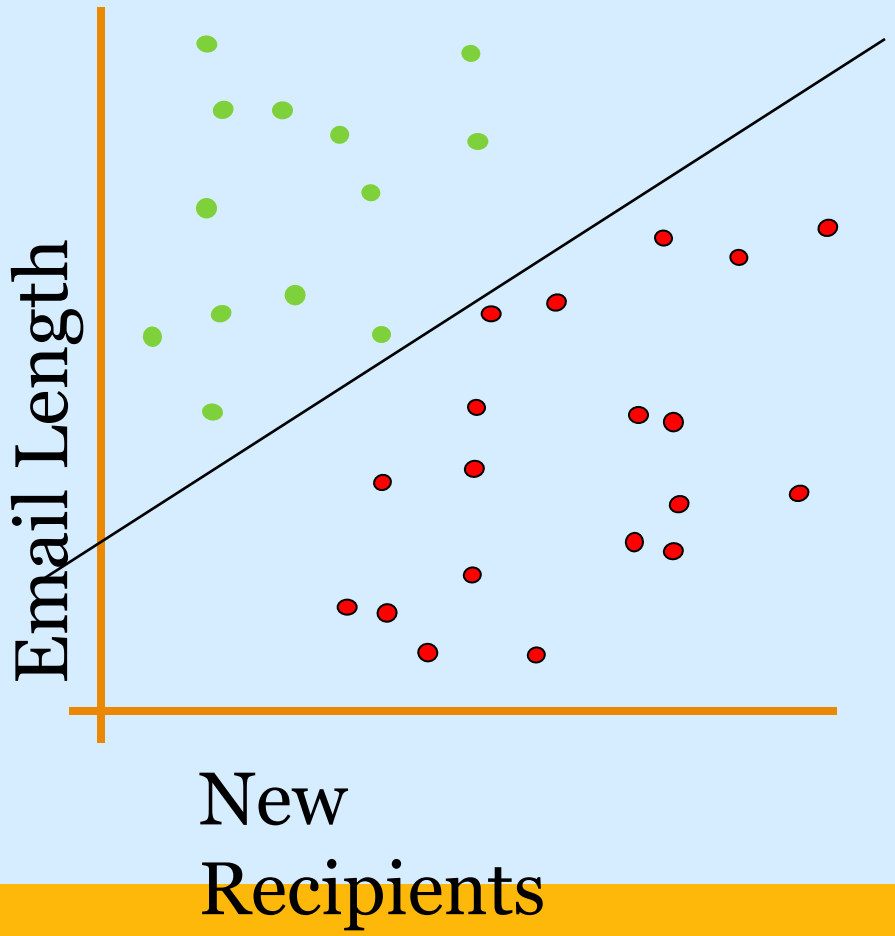


Linear Classifiers



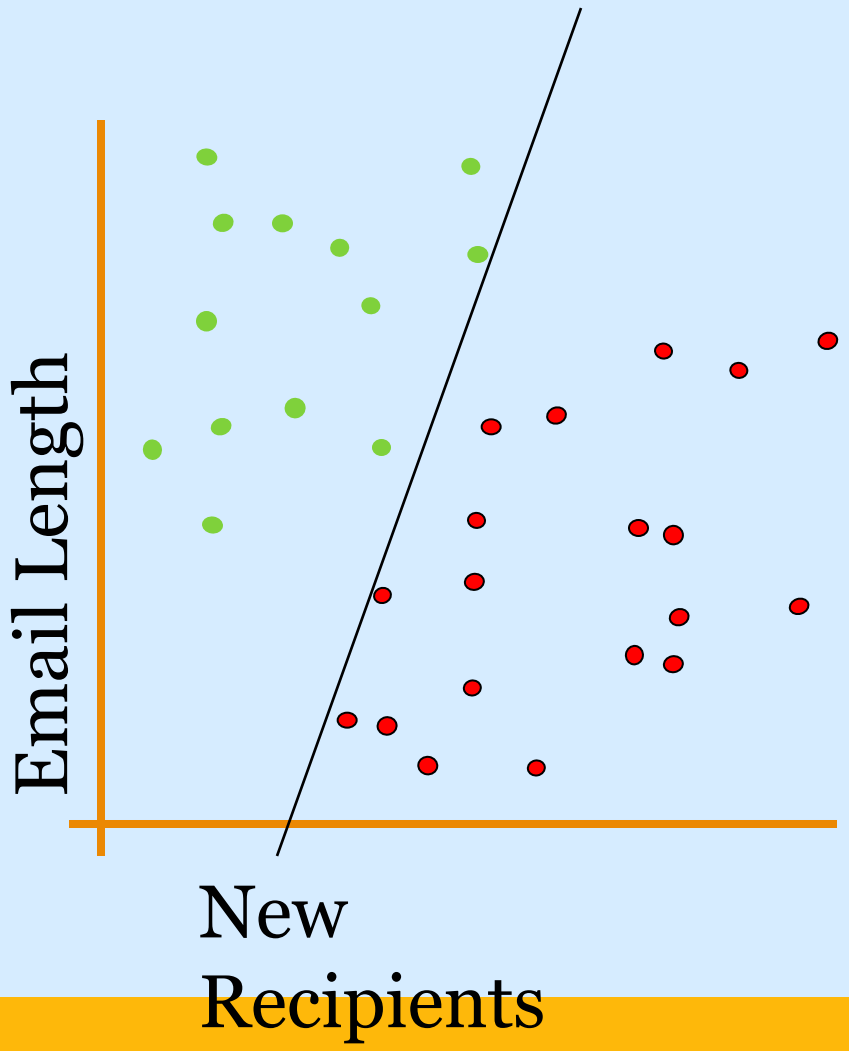
How would you classify this data?

Linear Classifiers



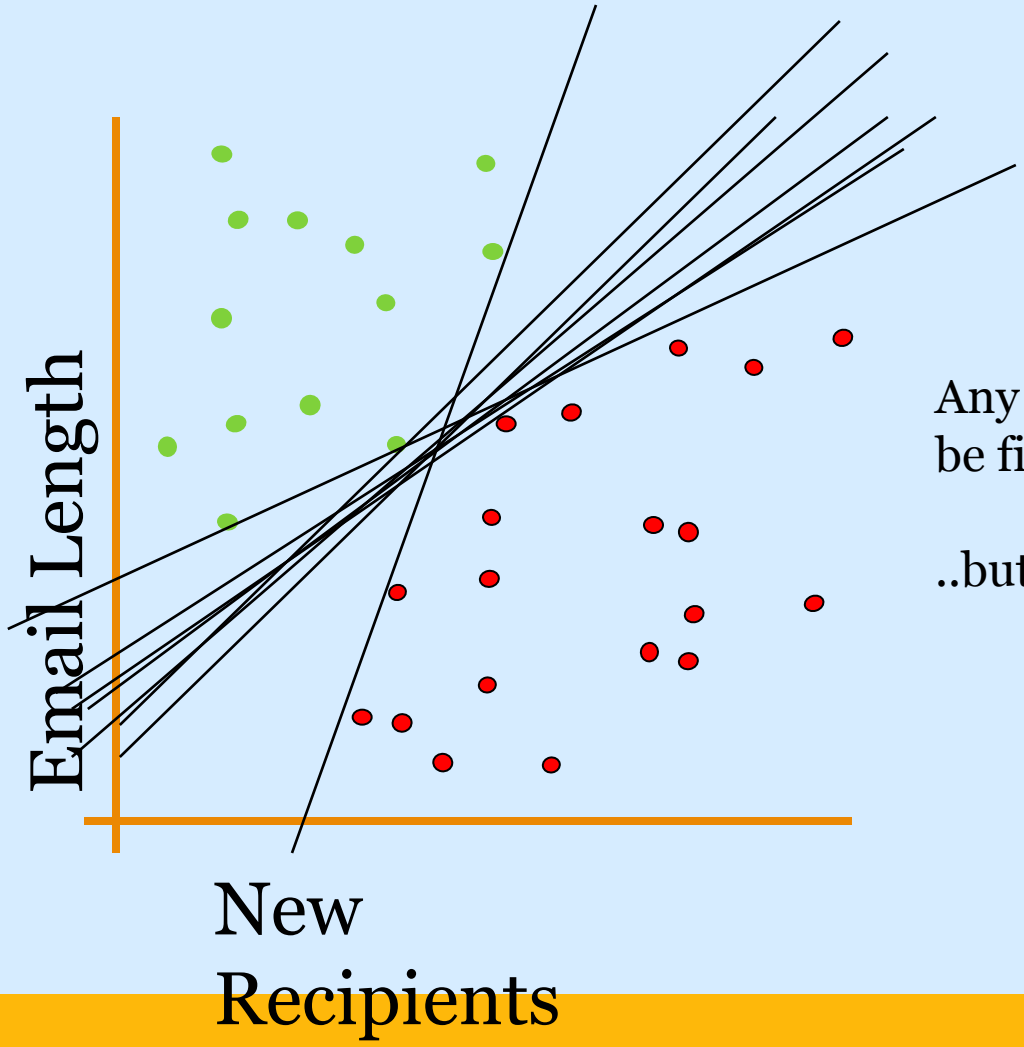
How would you classify this data?

Linear Classifiers



How would you classify this data?

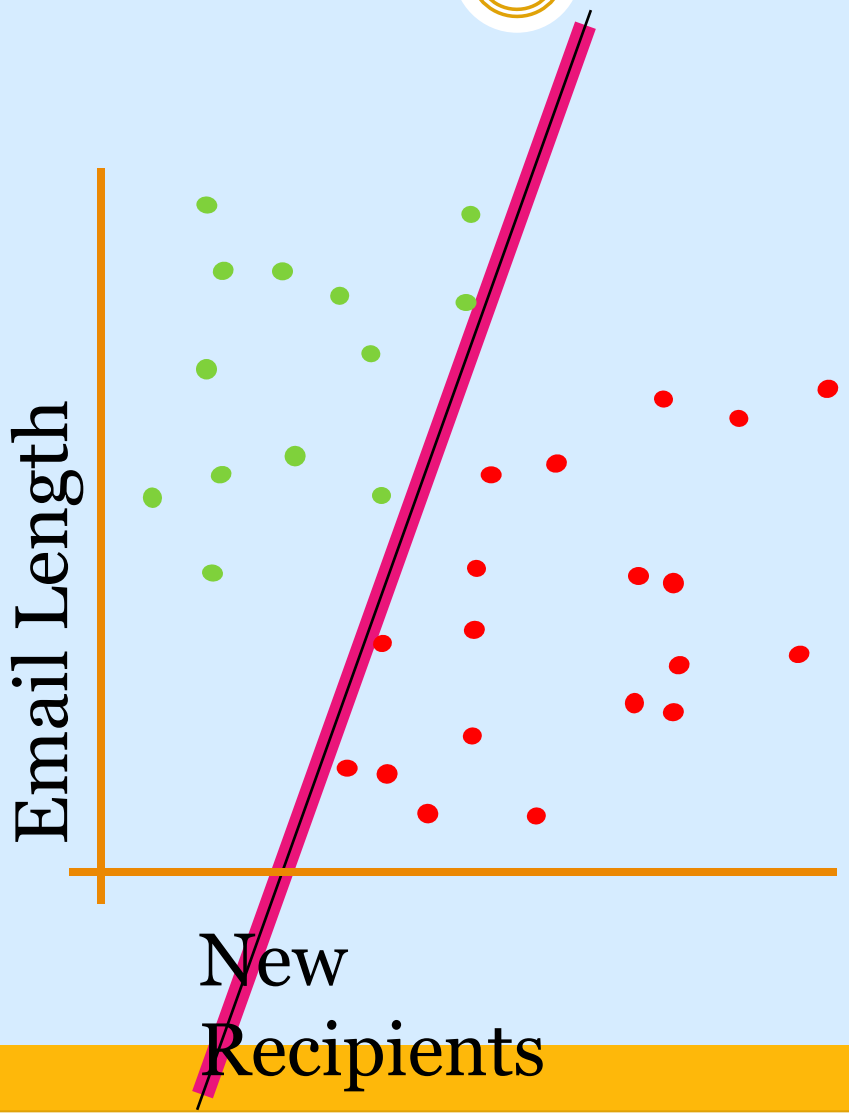
Linear Classifiers



Any of these would be fine..

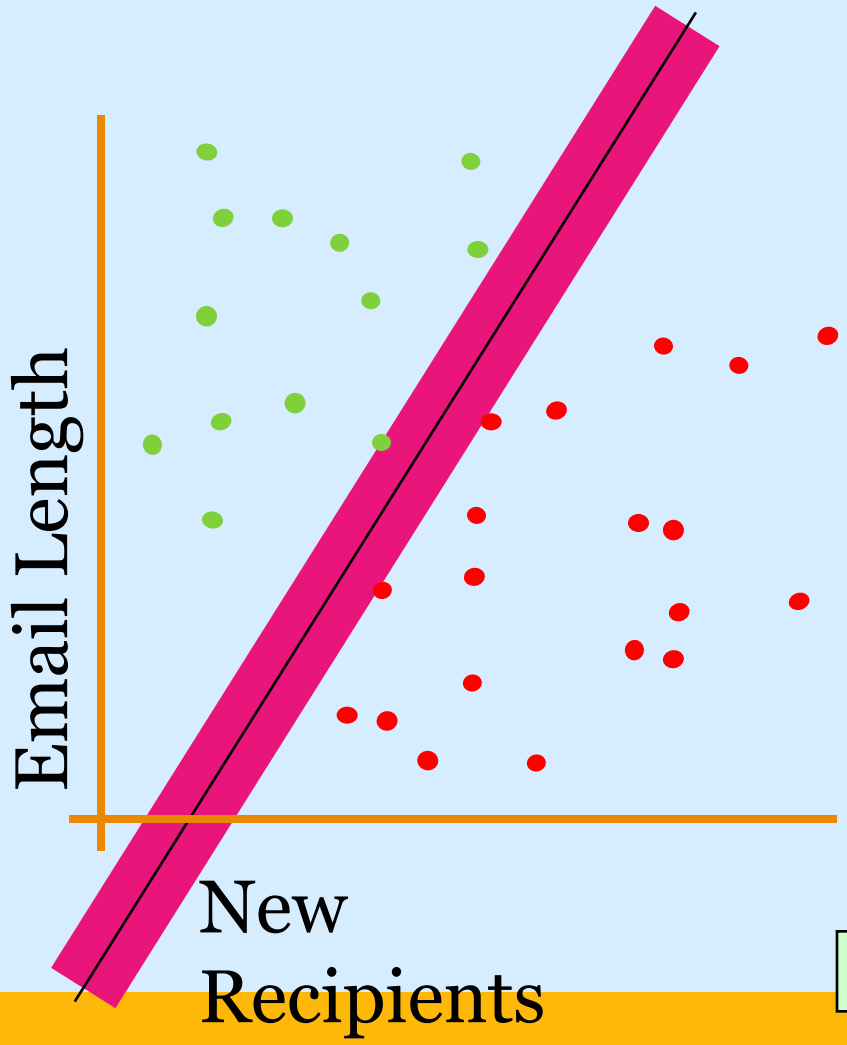
..but which is best?

Classifier Margin



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

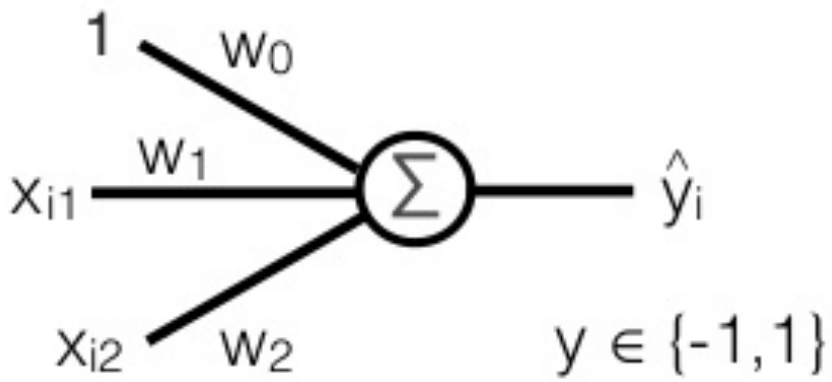
Maximum Margin



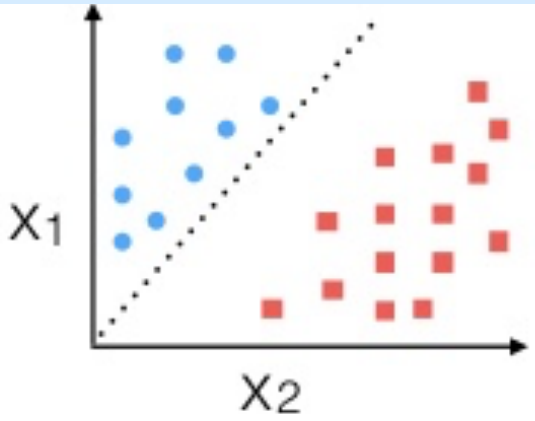
The maximum margin linear classifier is the linear classifier with the, maximum margin. This is the simplest kind of SVM (Called an LSVM)

Linear SVM

General Algorithms for this class



$y \in \{-1,1\}$



$$\hat{y} = \mathbf{w}^T\mathbf{x} = w_0 + w_1x_1 + w_2x_2$$

$$\hat{y}_i \begin{cases} 1 \text{ if } w^Tx_i \geq \theta \\ -1 \text{ otherwise} \end{cases}$$

w_j = weight
x_i = training sample
y_i = desired output
$\hat{y}_i$ = actual output
t = iteration step
η = learning rate

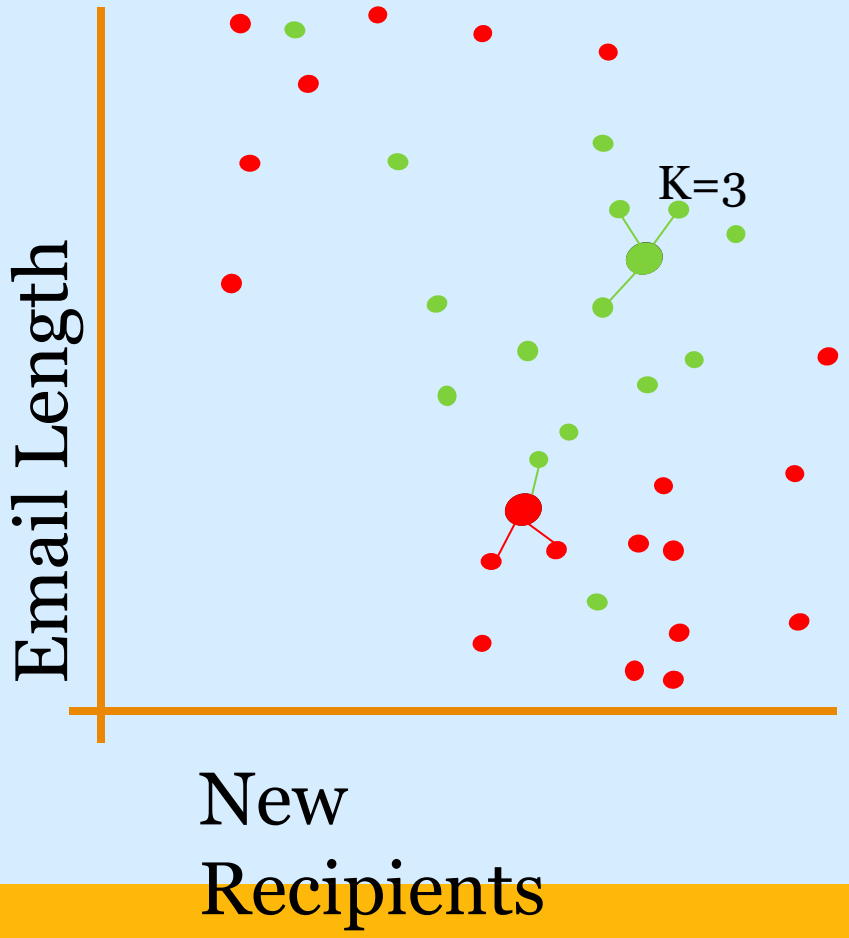
update rule:

$$w_j(t+1) = w_j(t) + \eta(y_i - \hat{y}_i)x_i$$

until
t+1 = max iter
or error = 0

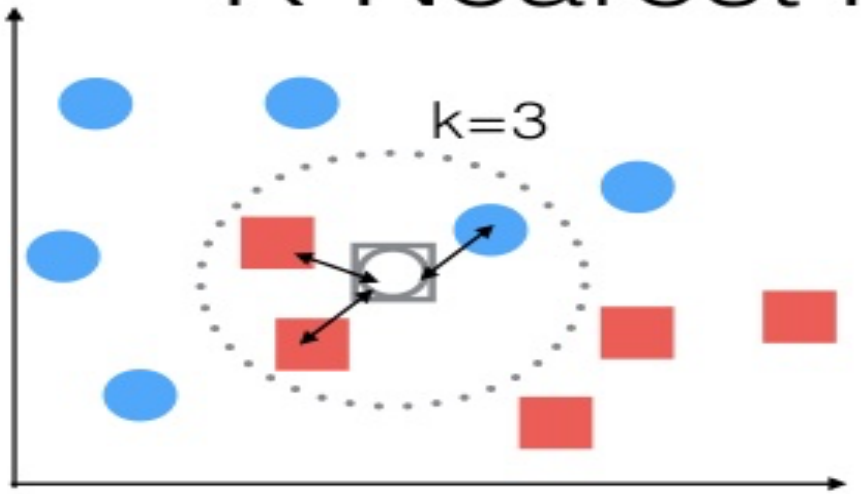
Another classifier: k-Nearest Neighbors

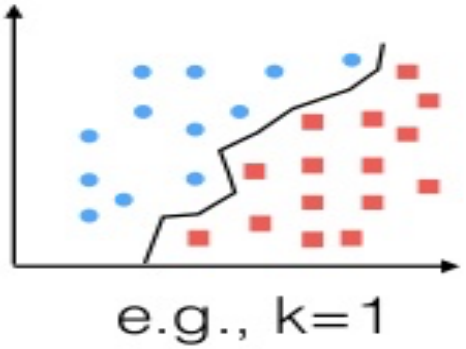
- What should be k?
- Which distance measure should be used?



Non-Parametric Classifiers

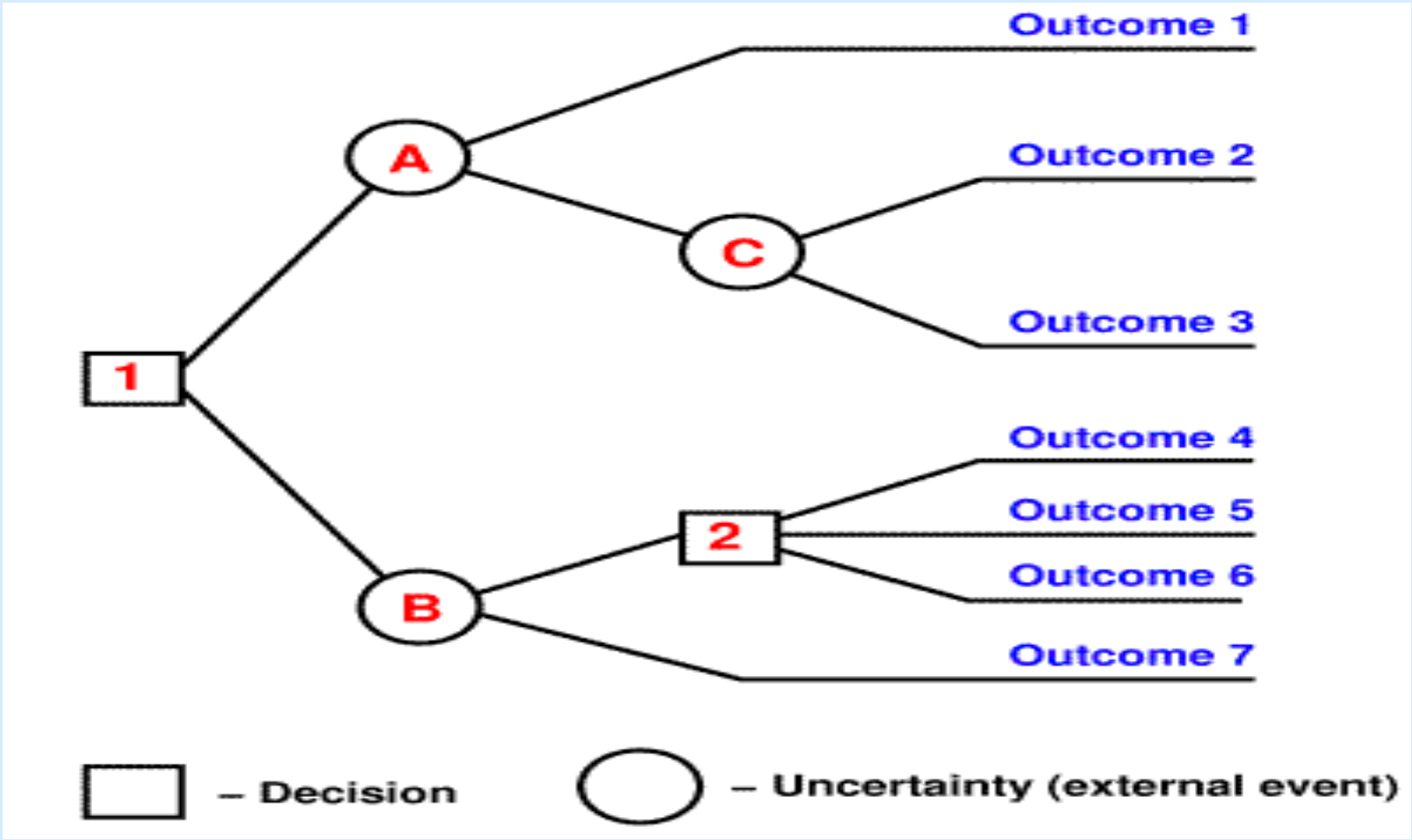
Non-Parametric Classifiers: K-Nearest Neighbor



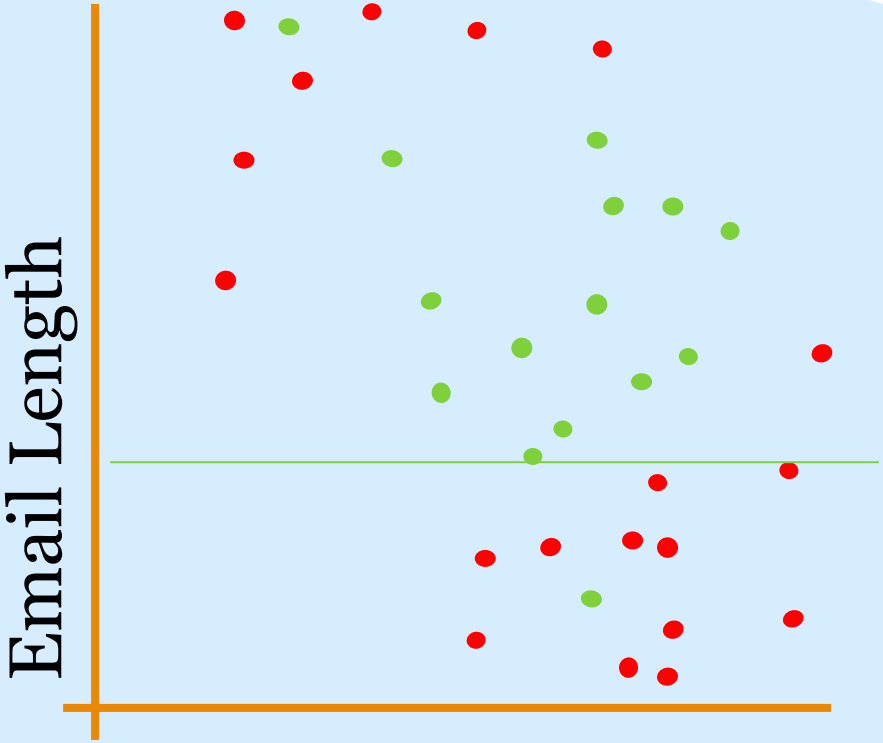


- Simple!
- Lazy learner
- Very susceptible to curse of dimensionality

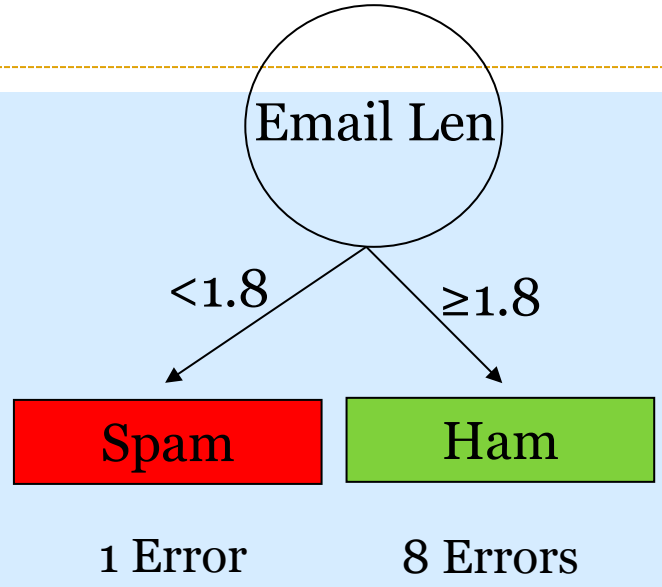
Decision Tree



Top Down Induction of Decision Trees

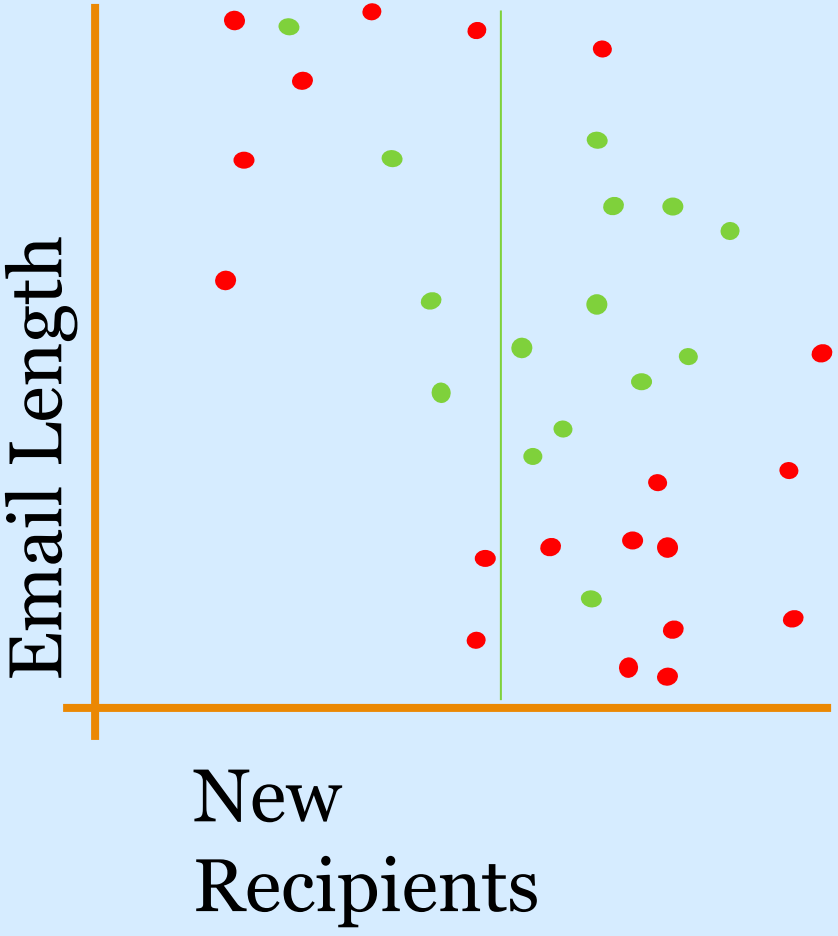


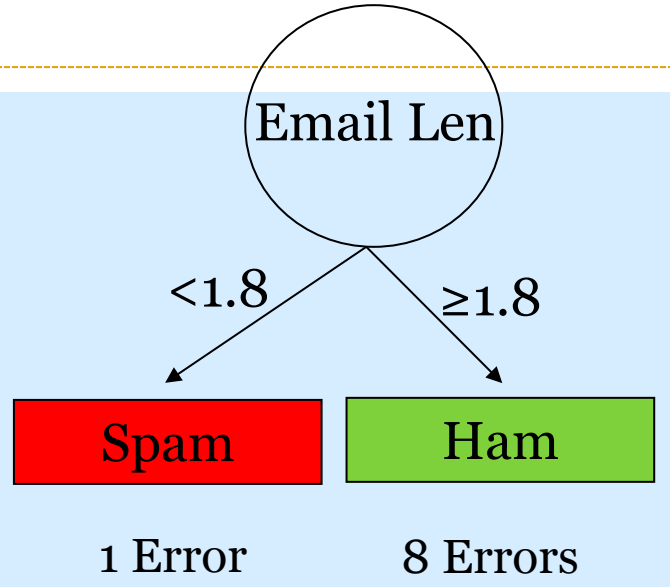
New Recipients

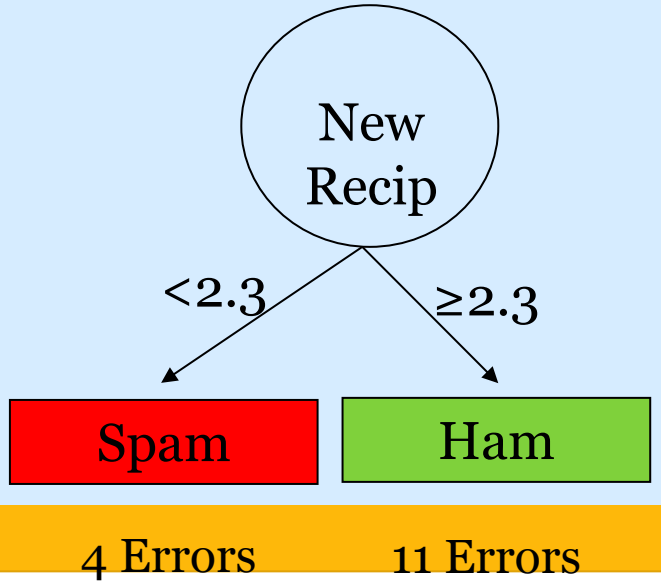


A single level decision tree is also known as Decision Stump

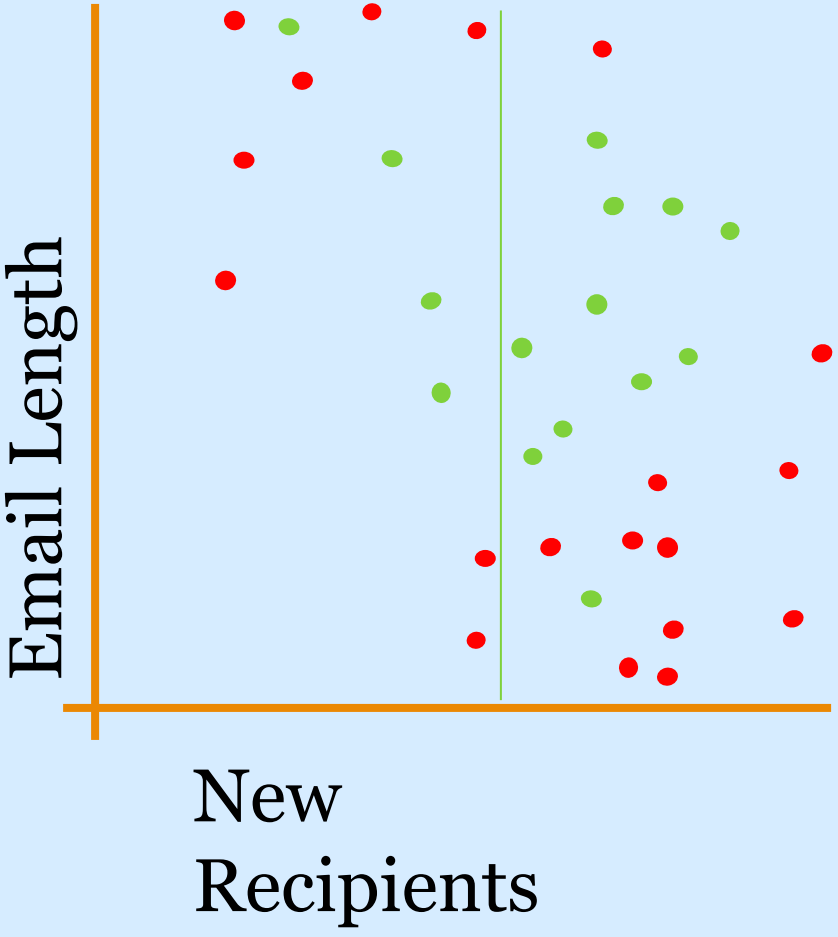
Top Down Induction of Decision Trees

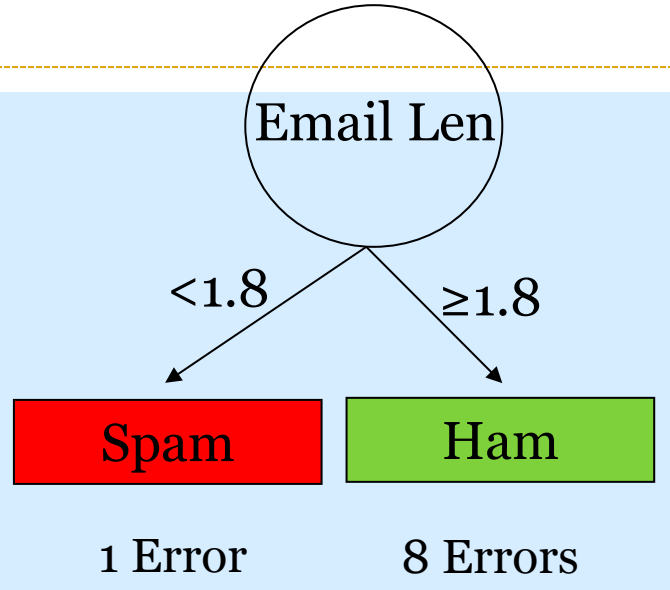


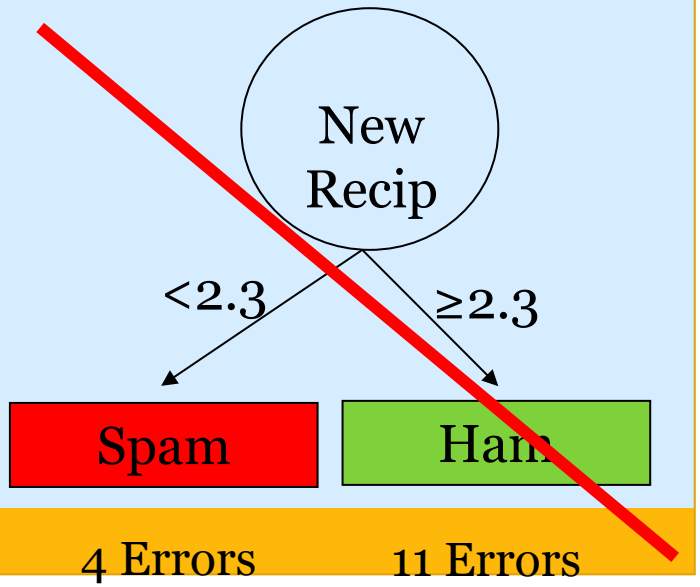




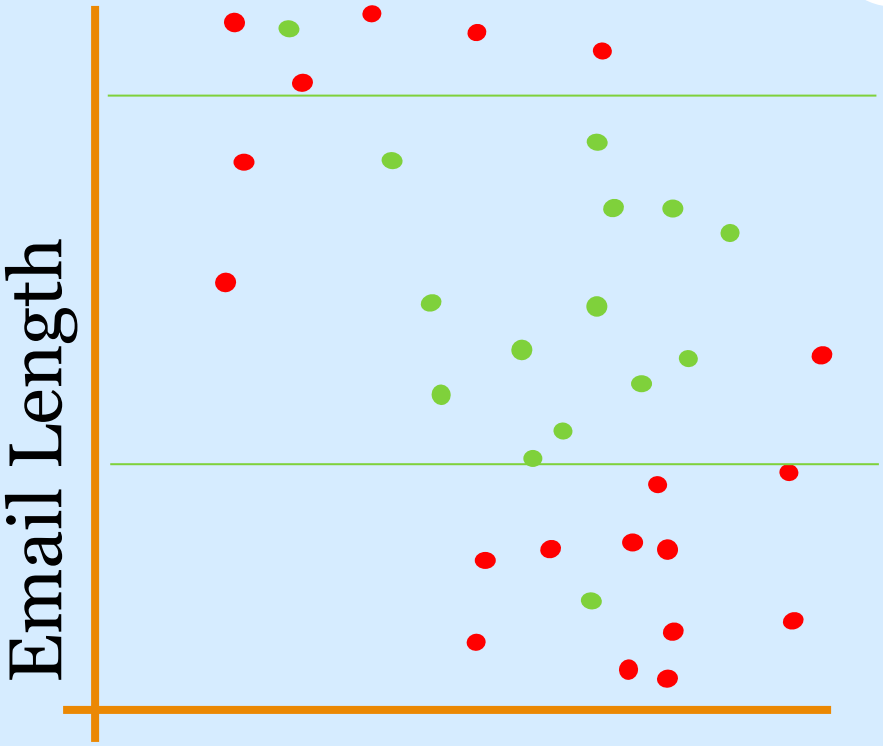
Top Down Induction of Decision Trees

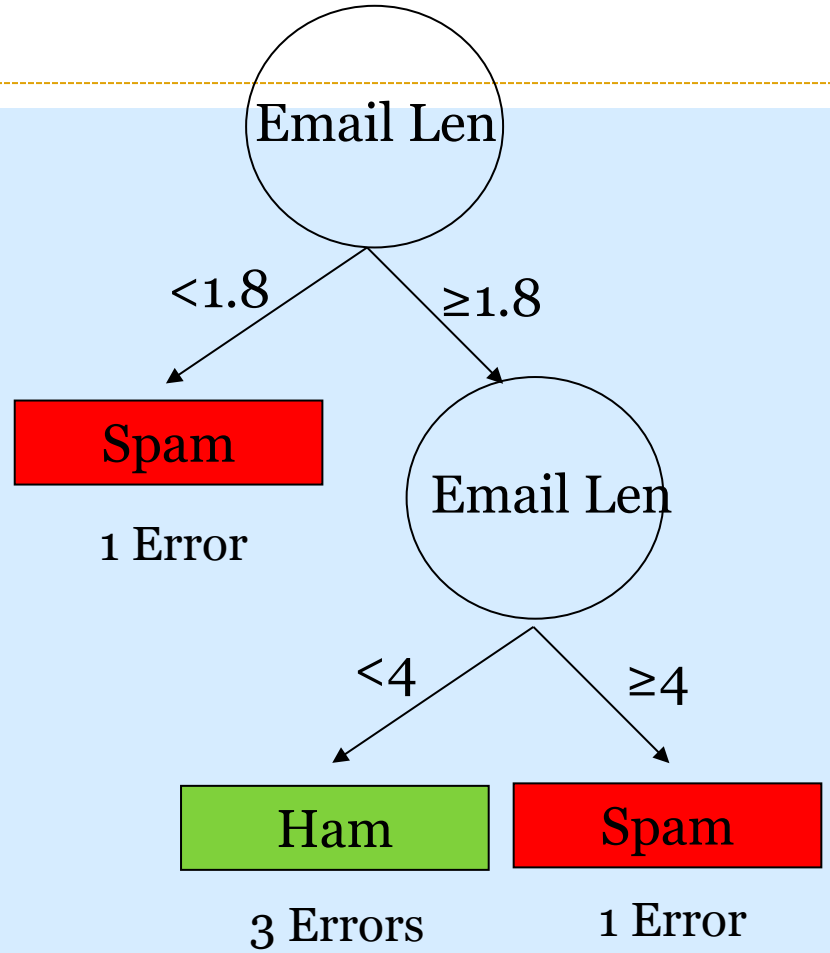




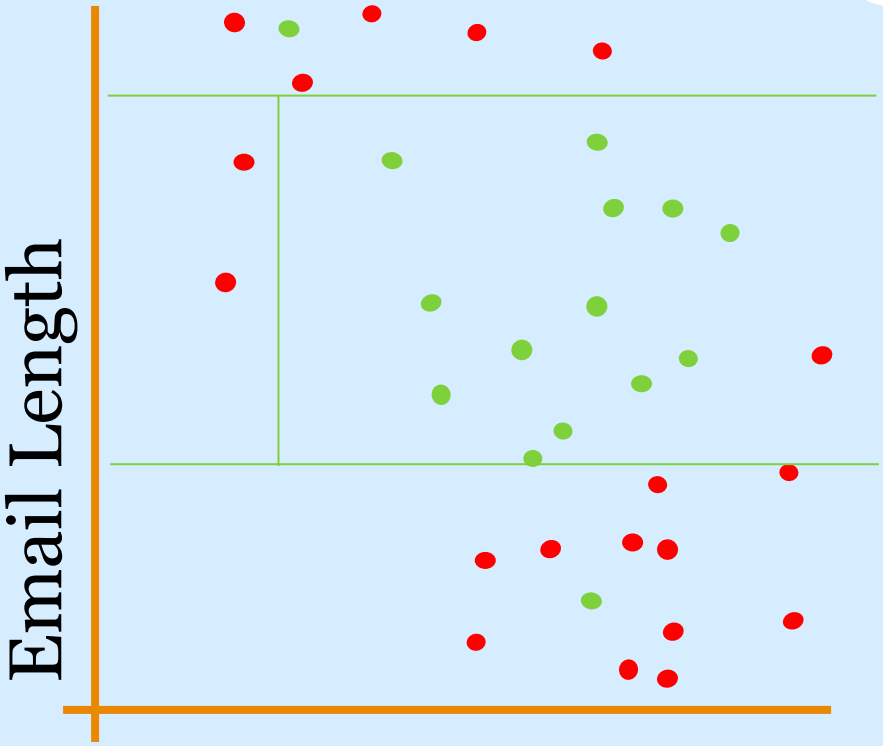


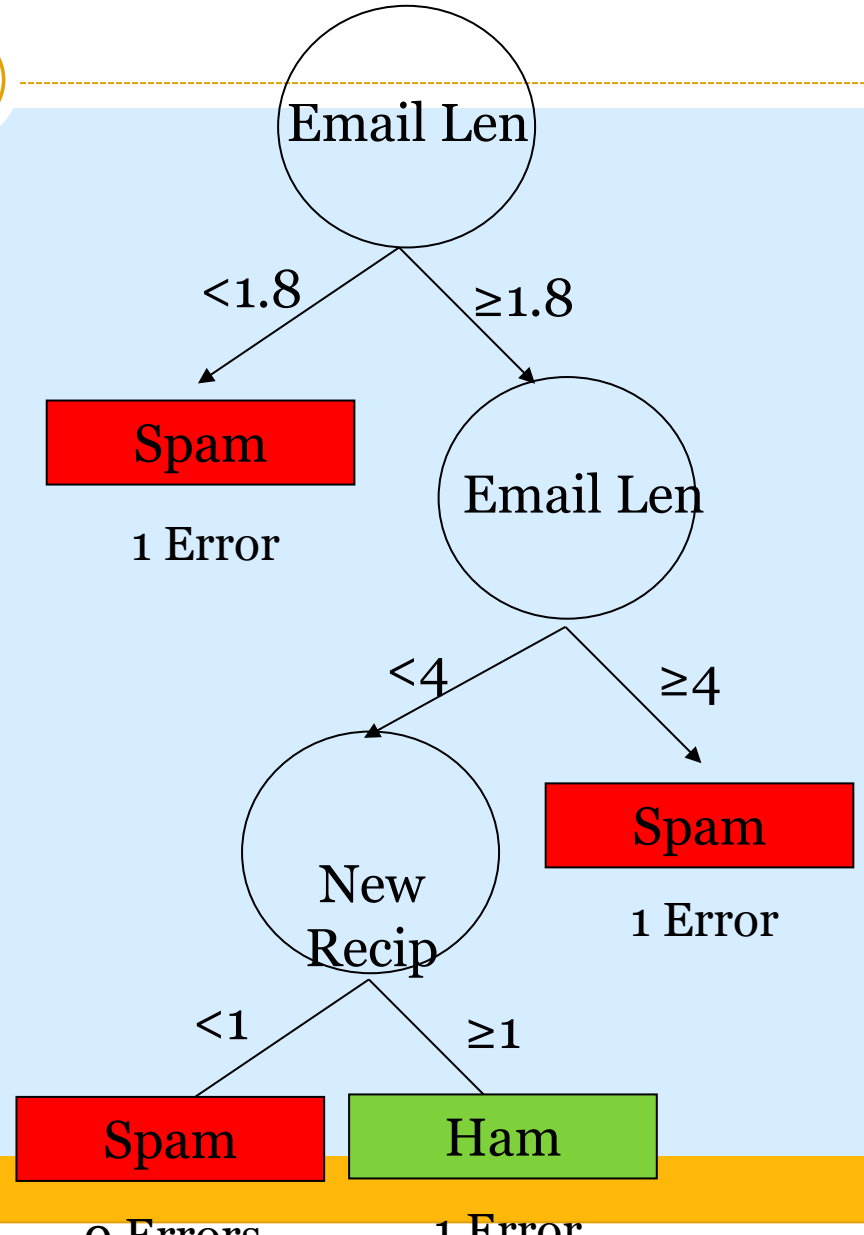
Top Down Induction of Decision Trees



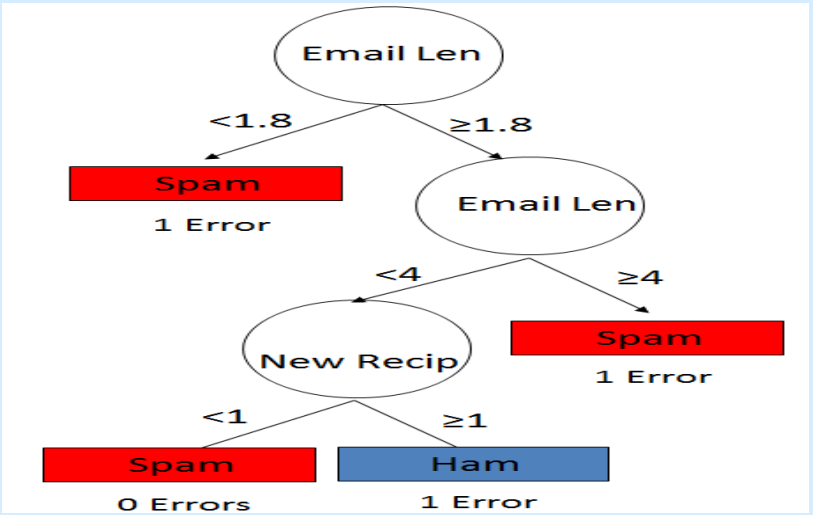


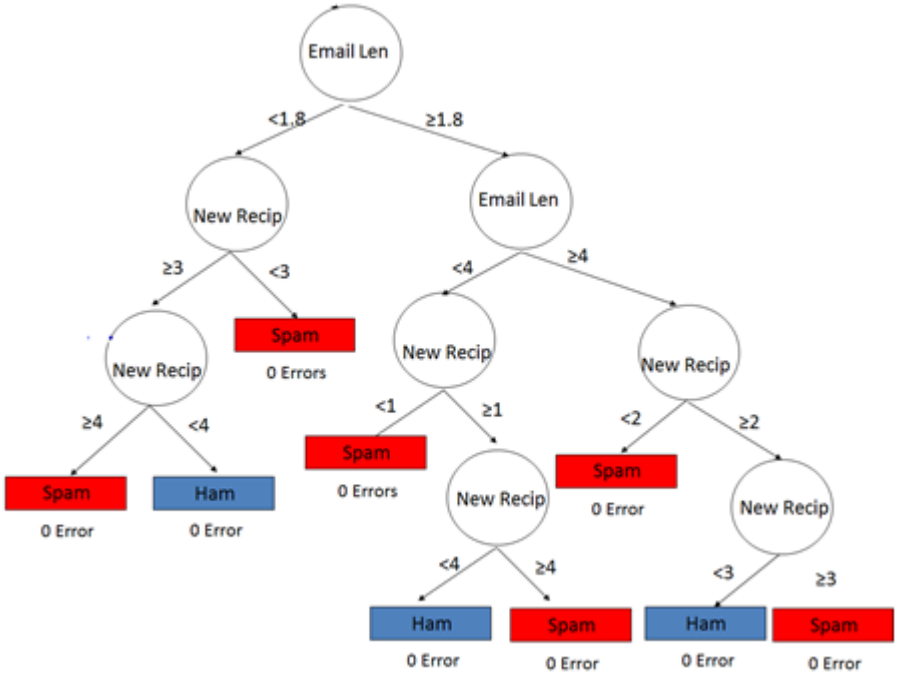
Top Down Induction of Decision Trees

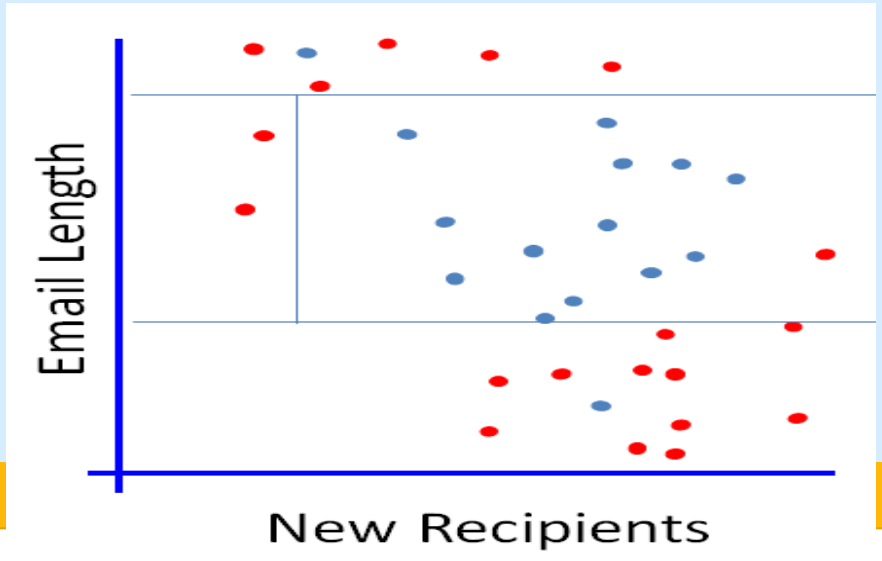


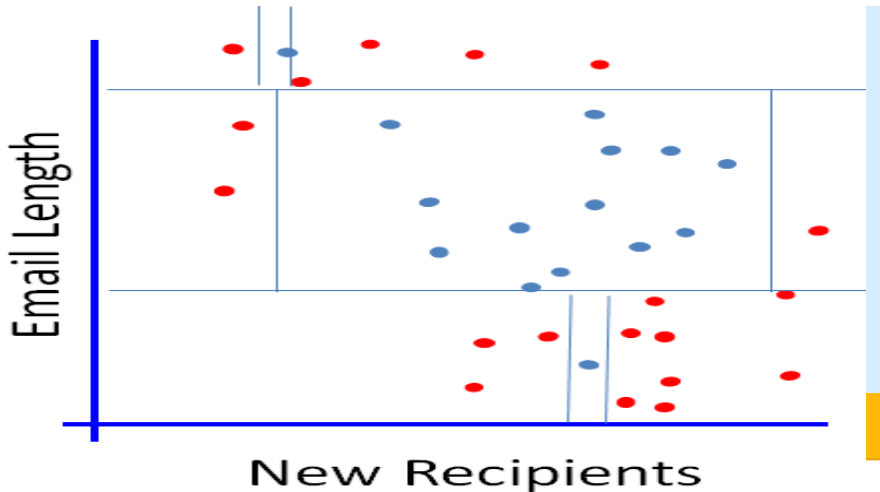


Which One?


Email Len
<1.8
≥1.8
Spam
1 Error
Email Len
<4
≥4
New Recip
Spam
1 Error
<1
≥1
Spam
Ham
0 Errors
1 Error


Email Len
<1.8
≥1.8
New Recip
Email Len
≥3
<3
<4
≥4
New Recip
Spam
New Recip
New Recip
0 Errors
≥4
<4
<1
≥1
<2
≥2
Spam
Ham
Spam
New Recip
Spam
New Recip
0 Error
0 Error
0 Errors
0 Error
<4
≥4
<3
≥3
Ham
Spam
Ham
Spam
0 Error
0 Error
0 Error
0 Error


Email Length
New Recipients

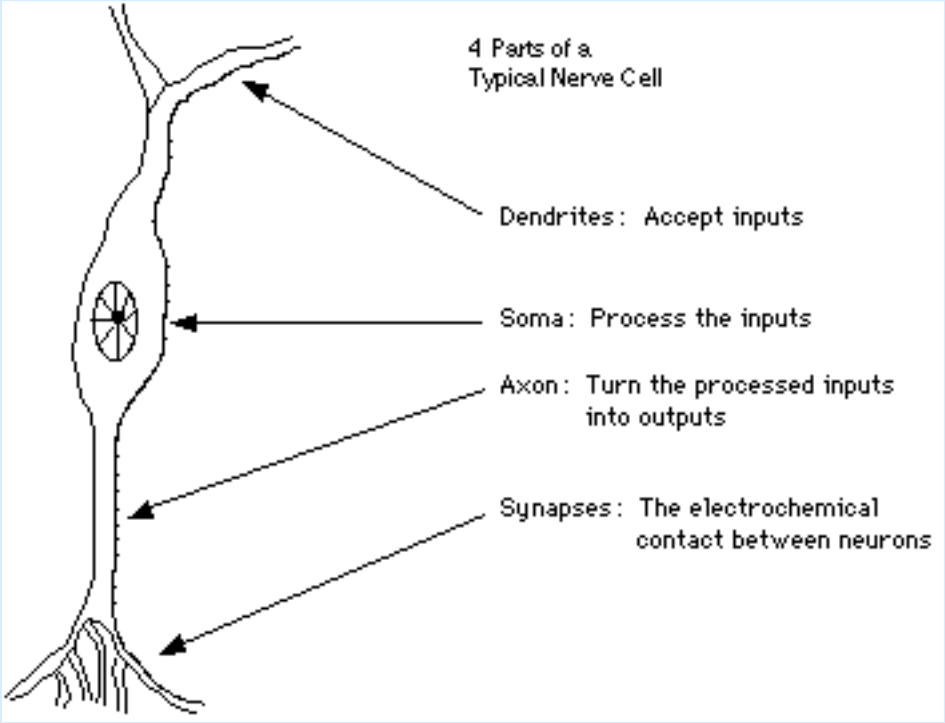

Email Length
New Recipients

Moving Forward

- Earlier algorithms discussed form general branch of Machine Learning
- ML is practice of using algorithms to parse data, learn from it, then determine or make predictions
- Another approach is mimicking human brain
- Entire branch of Aritifical Intelligence based on this approach is known as Artificial Neural Networks and Deep Learning
- It is composed of a large number of highly interconnected processing elements called neurons

Let us simply understand brain processing

- Four parts of a nerve cell:
- Dendrite: Accepts the input
- Soma: Processes the input
- Axon: Turns processed input into output
- Synapses: Connection between neurons



Our brain is a huge neural network

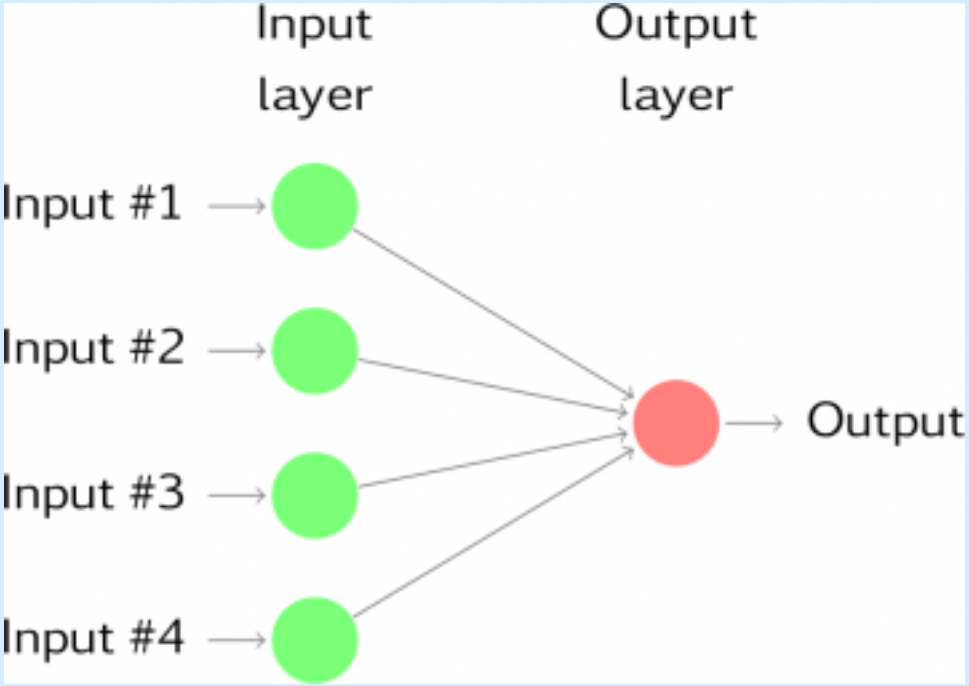
Moving Forward

- Artificial Neural Network (ANNs) are programs designed to solve any problem by trying to mimic the structure and the function of our nervous system.
- Neural networks are based on simulated neurons, Which are joined together in a variety of ways to form networks.
- Neural network resembles the human brain in the following two ways: -
  - A neural network acquires knowledge through learning.
  - A neural network's knowledge is stored within the interconnection strengths known as synaptic weight.

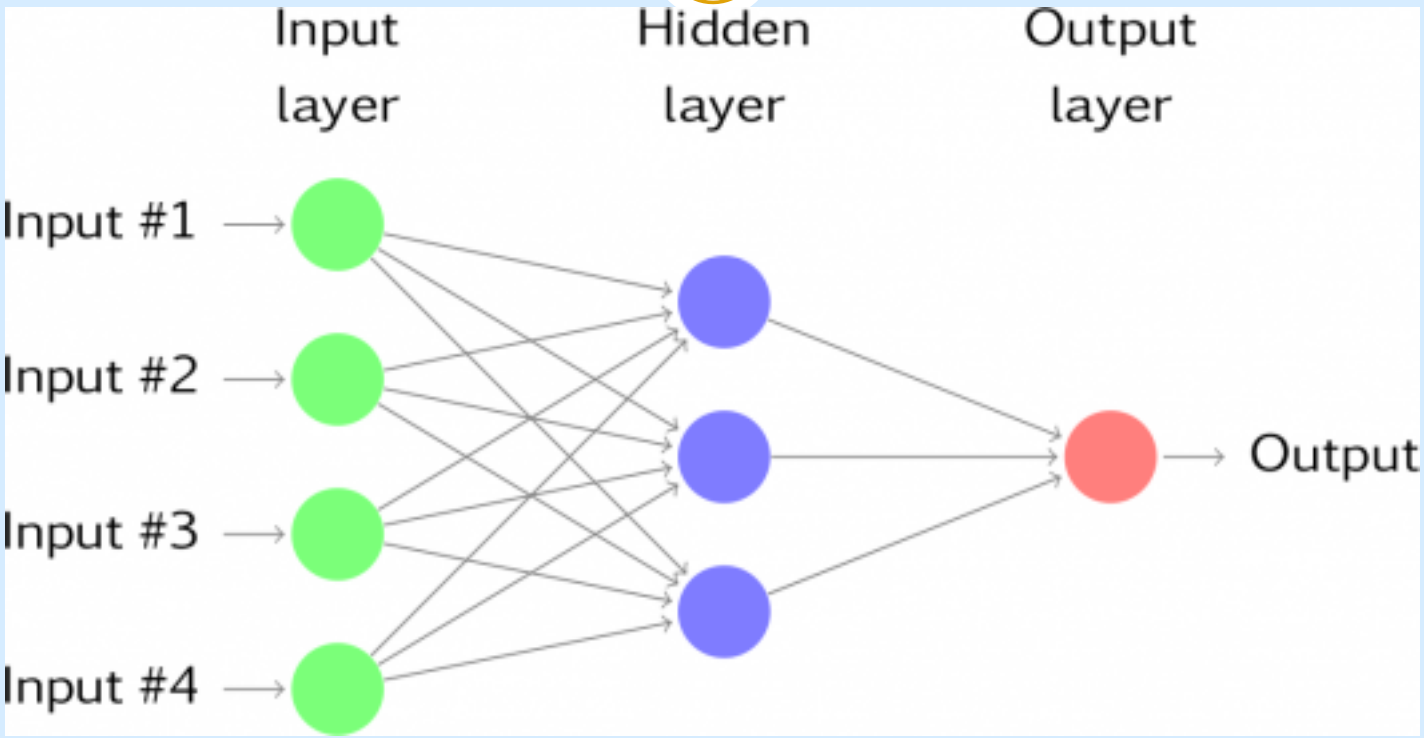
Artificial Neural Network (ANN)

- An ANN is based on the same idea
- There are inputs to the network (x1, x2 …)
- Each of these inputs is multiplied by a connection weight (w1, w2 …)
- The products are then summed and fed to a function to generate output
- Sum = w1x1 + w2x2 + ..

Neural Network



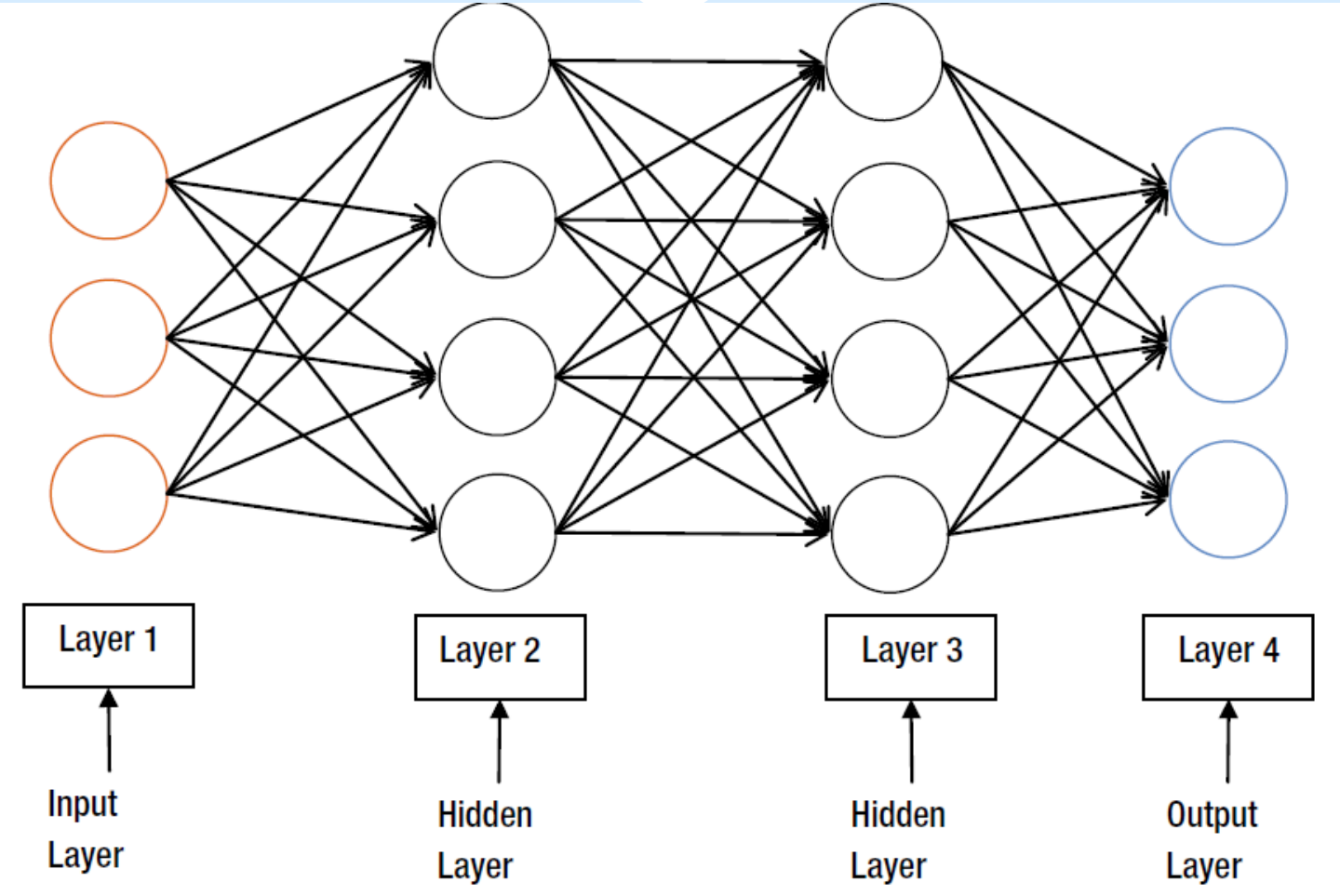
Let us add complexity



$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} x_i$$

$$s(z) = \frac{1}{1 + e^{-z}}$$

Neural Network



Illustrative Example

- For example, a 3-input neuron is taught to output 1 when the input (X1,X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001

X 1:		0	0	0	0	1	1	1	1
X 2:		0	0	1	1	0	0	1	1
X 3:		0	1	0	1	0	1	0	1
O U T:		0	0	0/ 1	0/ 1	0/ 1	1	0/ 1	1

Illustrative Example

- Take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000 which belongs in the 0-taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1).

X 1:		0	0	0	0	1	1	1	1
X 2:		0	0	1	1	0	0	1	1
X 3:		0	1	0	1	0	1	0	1
OUT:		0	0	0	0/1	0/1	1	1	1

Typical Structure of an ANN

- The algorithm is a learning rule which suggests a way of modifying weights to represent a function from input to output
- The network architecture is a feedforward network where computational units are structured in a multi-layered network: an input layer, one or more hidden layer(s), and an output layer
- The units on a layer have full connections to units on the adjacent layers, but no connection to units on the same layer. This leads us to the idea of back propagation

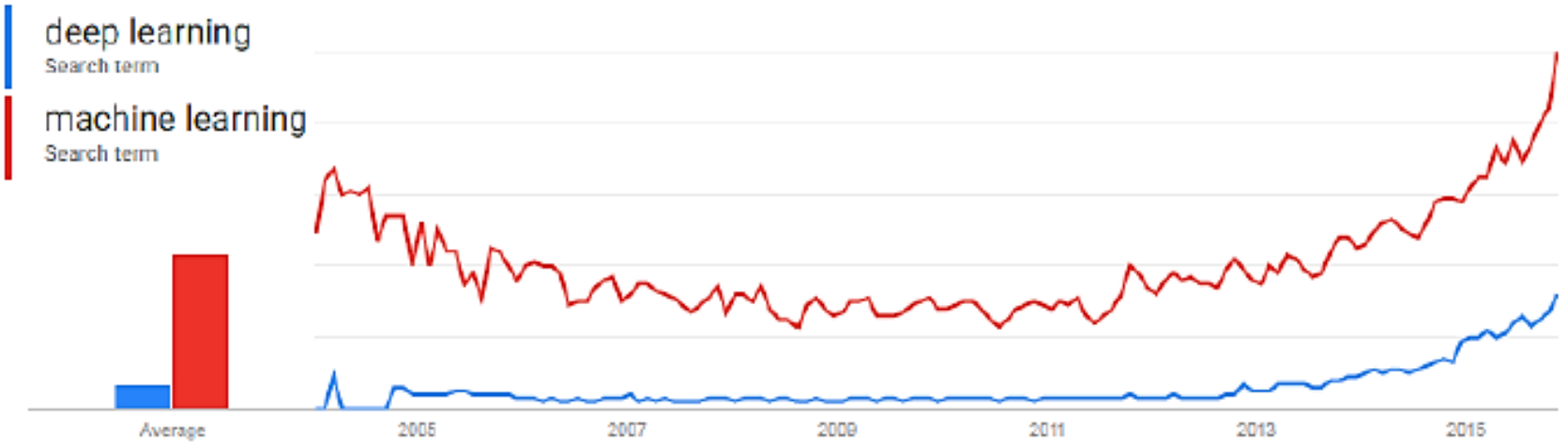
Backpropagation

- Calculate the difference (error) between the expected and actual output value
- Adjust the weights in order to minimize the error
- Minimize the error by performing a gradient decent on the error surface
- The amount of the weight change for each input pattern in an epoch is proportional to the error

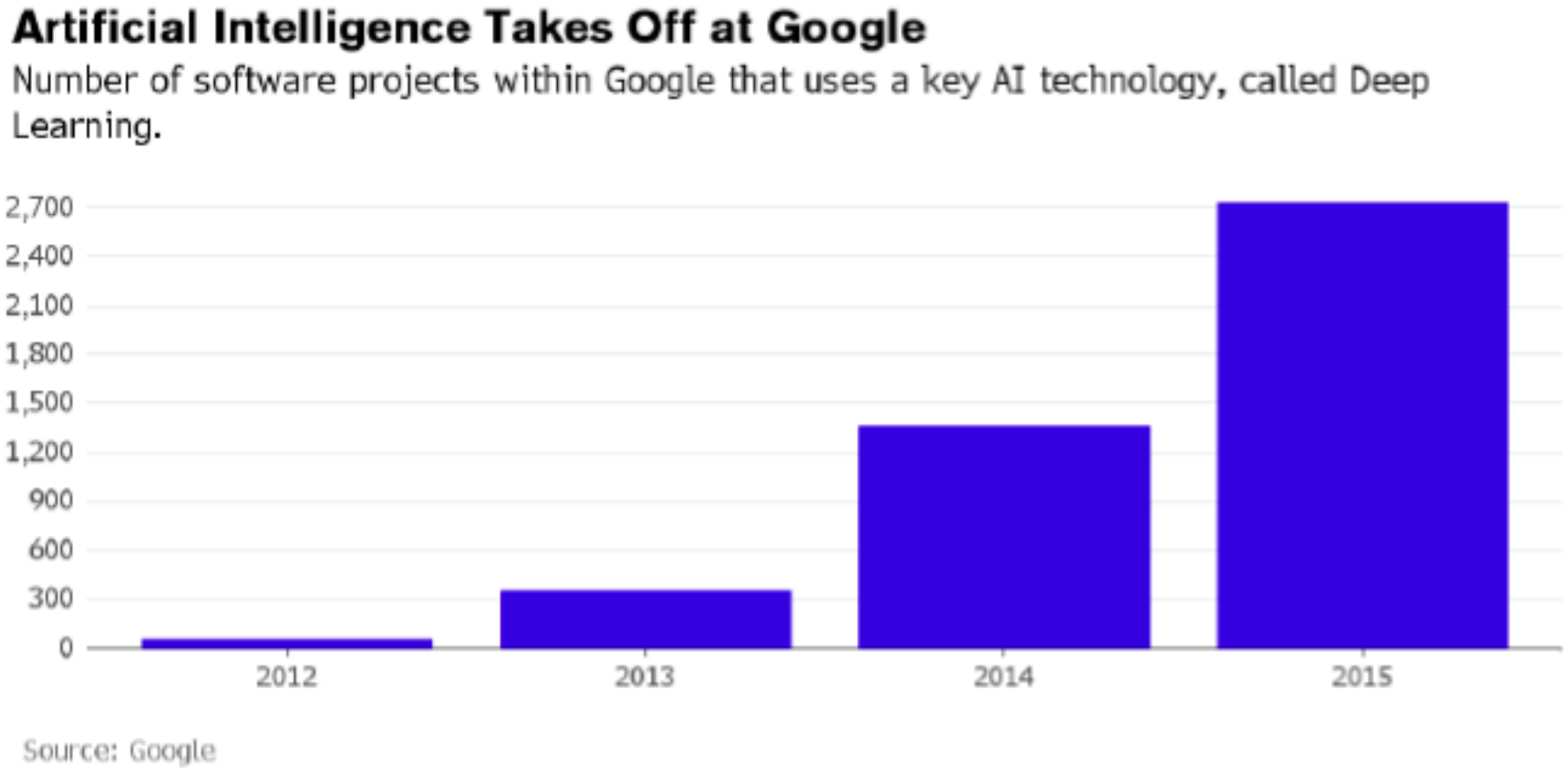
Why use Neural Networks?

- ability to derive meaning from complicated or imprecise data
- extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques
- Adaptive learning
- Real Time Operation

Deep Learning



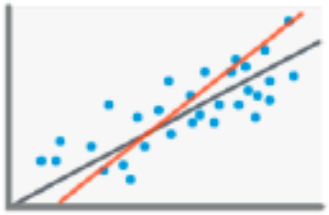
Deep Learning



Why this hype about Deep Learning?

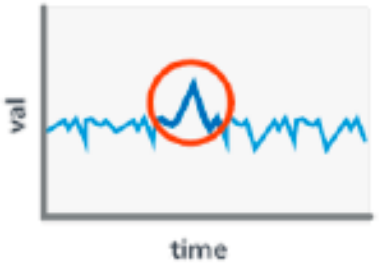
- To understand this let us understand our learning till now:
- Supervised Learning: Learning with a labeled training set. *Example: email spam detector with training set of already labeled emails*
- Unsupervised Learning: Discovering patterns in unlabeled data *Example: cluster similar documents based on the text content*
- Reinforcement Learning: learning based on feedback or reward *Example: learn to play chess by winning or losing*

Classification
(supervised – predictive)



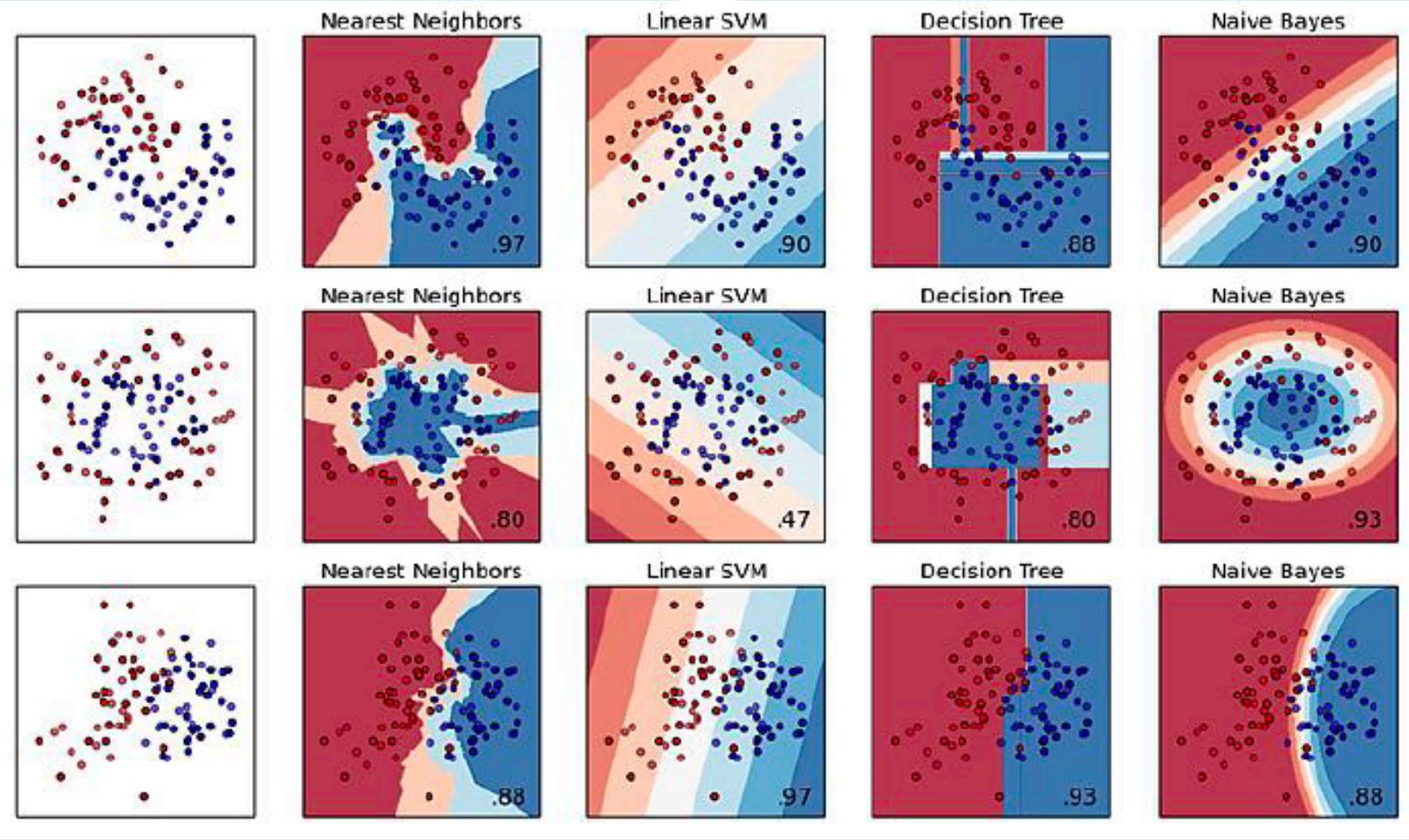
Regression
(supervised – predictive)

Clustering
(unsupervised – descriptive)



Anomaly Detection
(unsupervised– descriptive)

Comparison of algorithmic accuracy



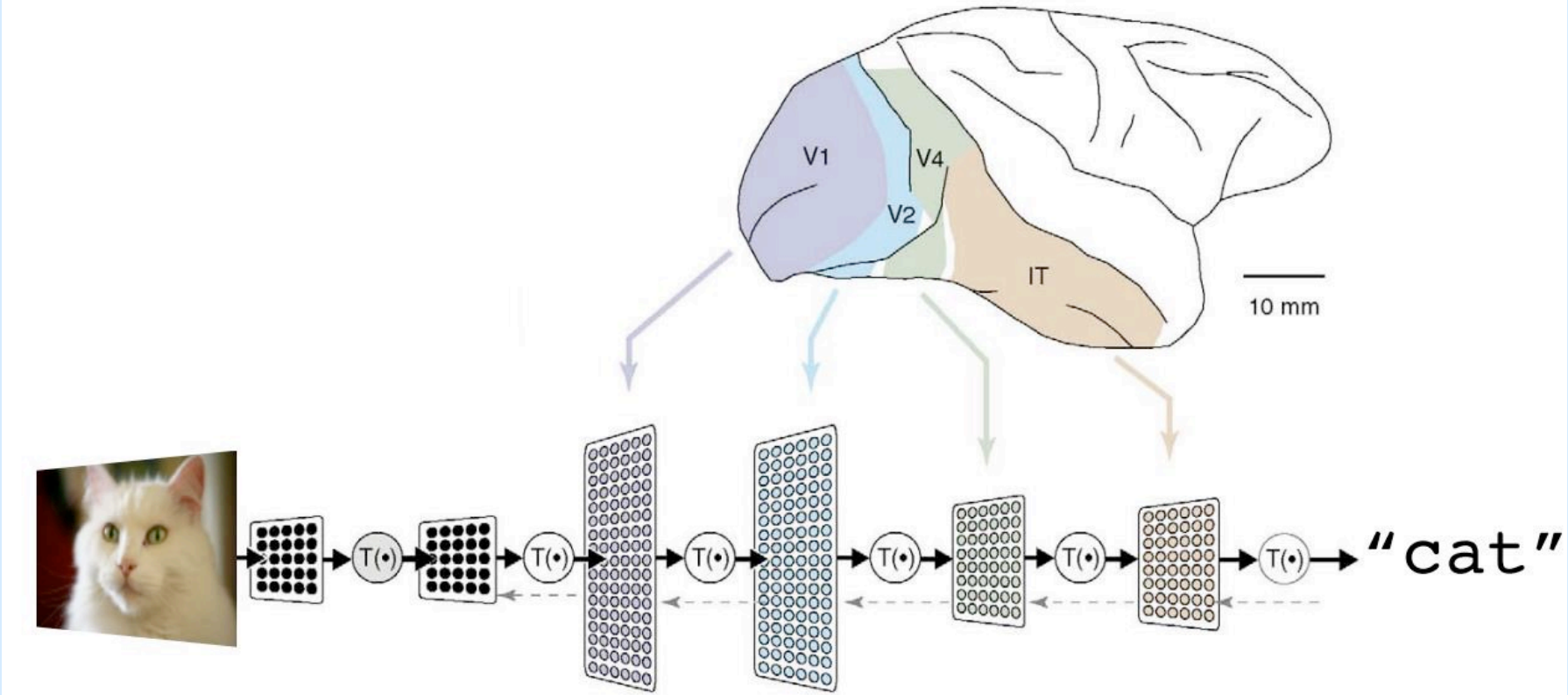
Deep Learning

- So what is the idea behind Deep Learning?
- Part of the machine learning field of learning representations of data. Exceptionally effective at learning patterns.
- Utilizes learning algorithms that derive meaning out of data by using a hierarchy of multiple layers that mimic the neural networks of our brain.
- If you provide the system tons of information, it begins to understand it and respond in useful ways.

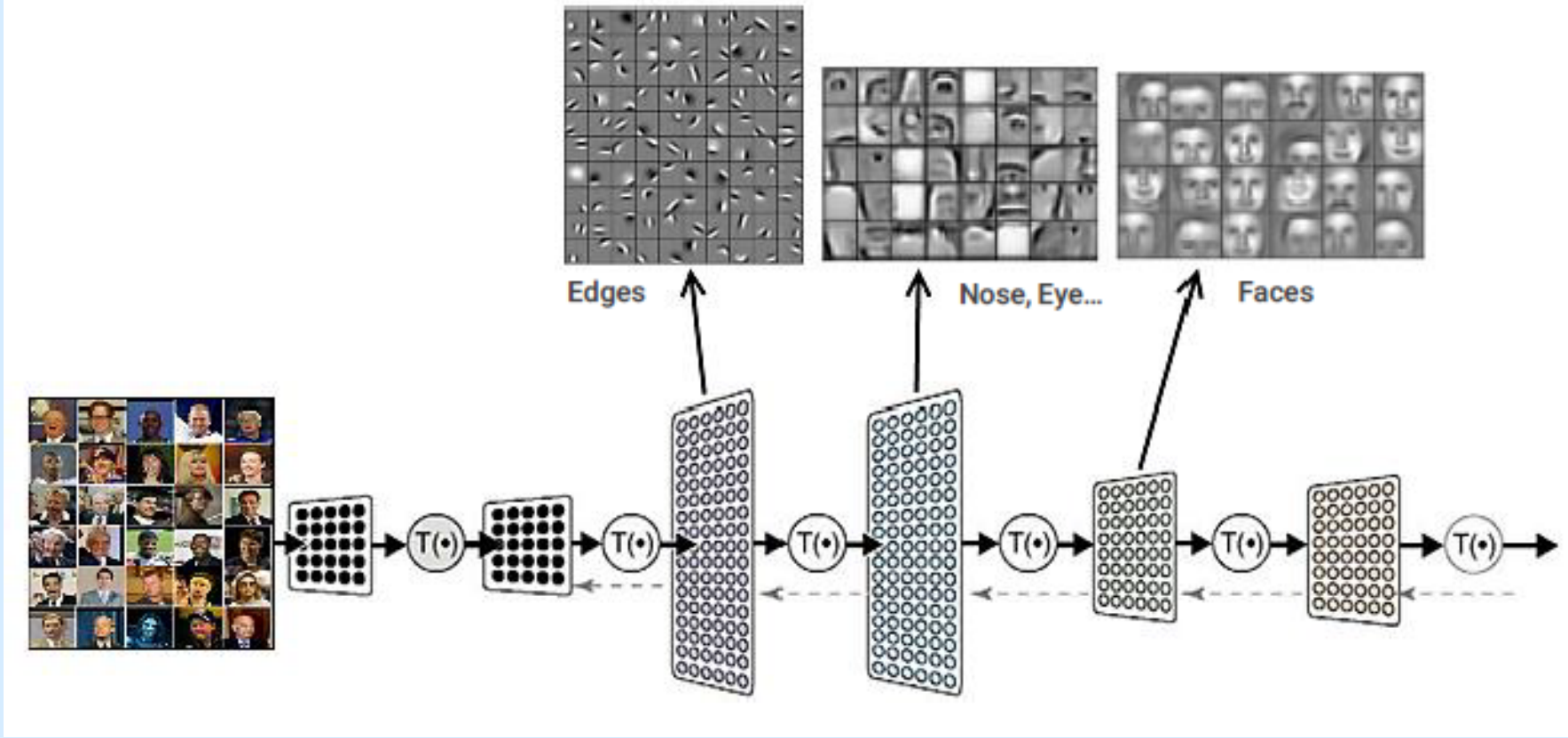
Deep Learning

- Conceptually, similar to ANN
- Hierarchy of layers (think layers in ANN)
- Each layer transforms image/input to abstract representations
- Output layer combines these abstract layers to form predictions
- Popular algos: Convolutional Neural Networks, TensorFlow, Theano, DGCAN

Deep Learning

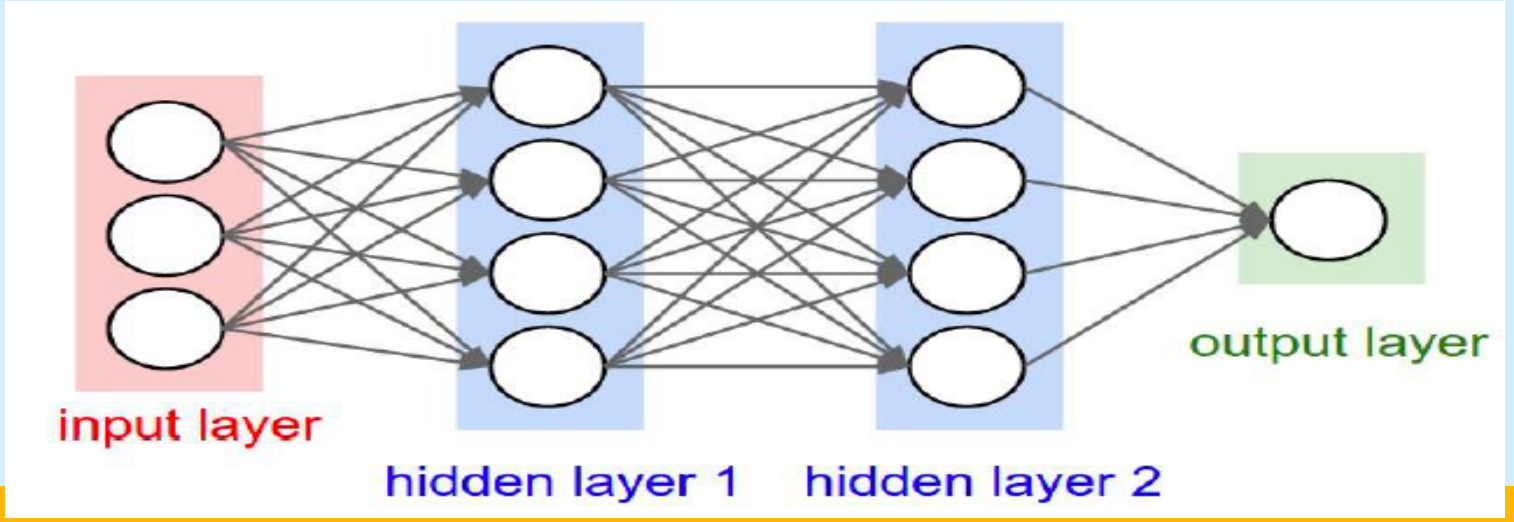


Deep Learning



Still very similar to ANN

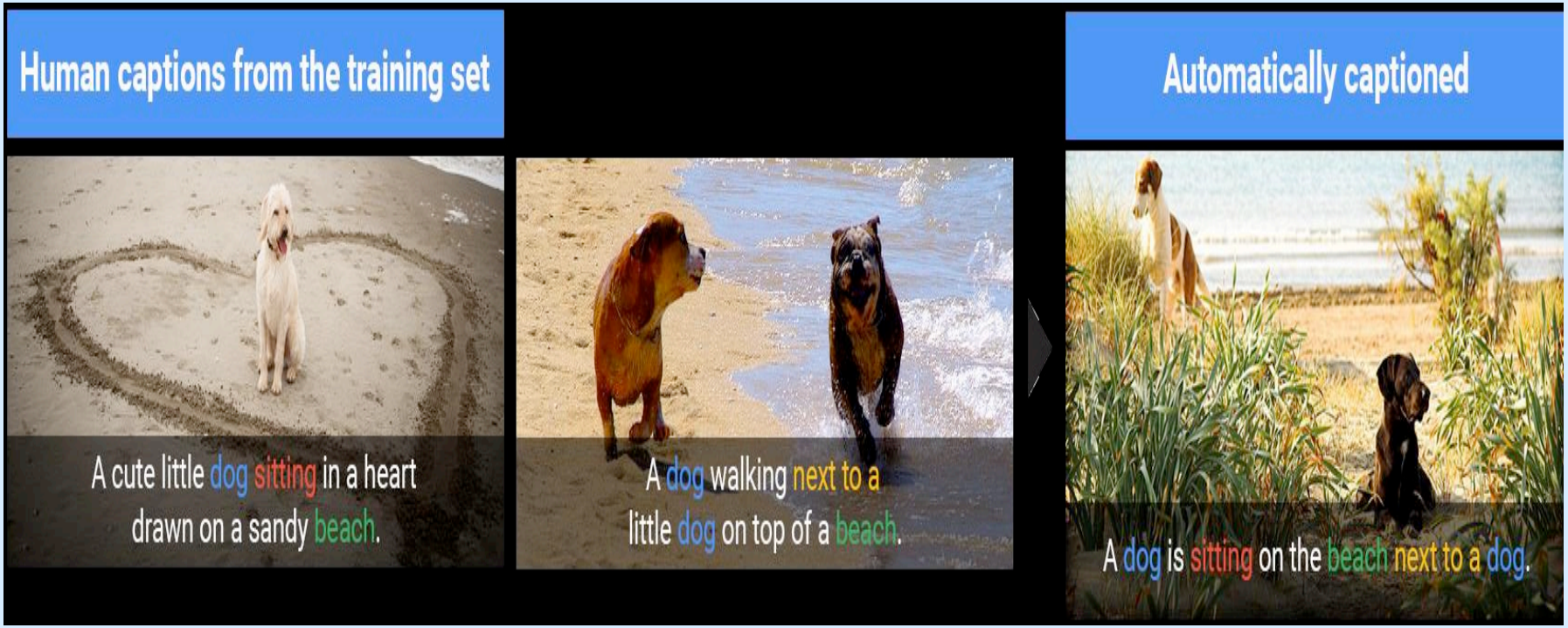
- Consists of one input, one output and multiple fully-connected hidden layers in-between. Each layer is represented as a series of neurons and progressively extracts higher and higher-level features of the input until the final layer essentially makes a decision about what the input shows.



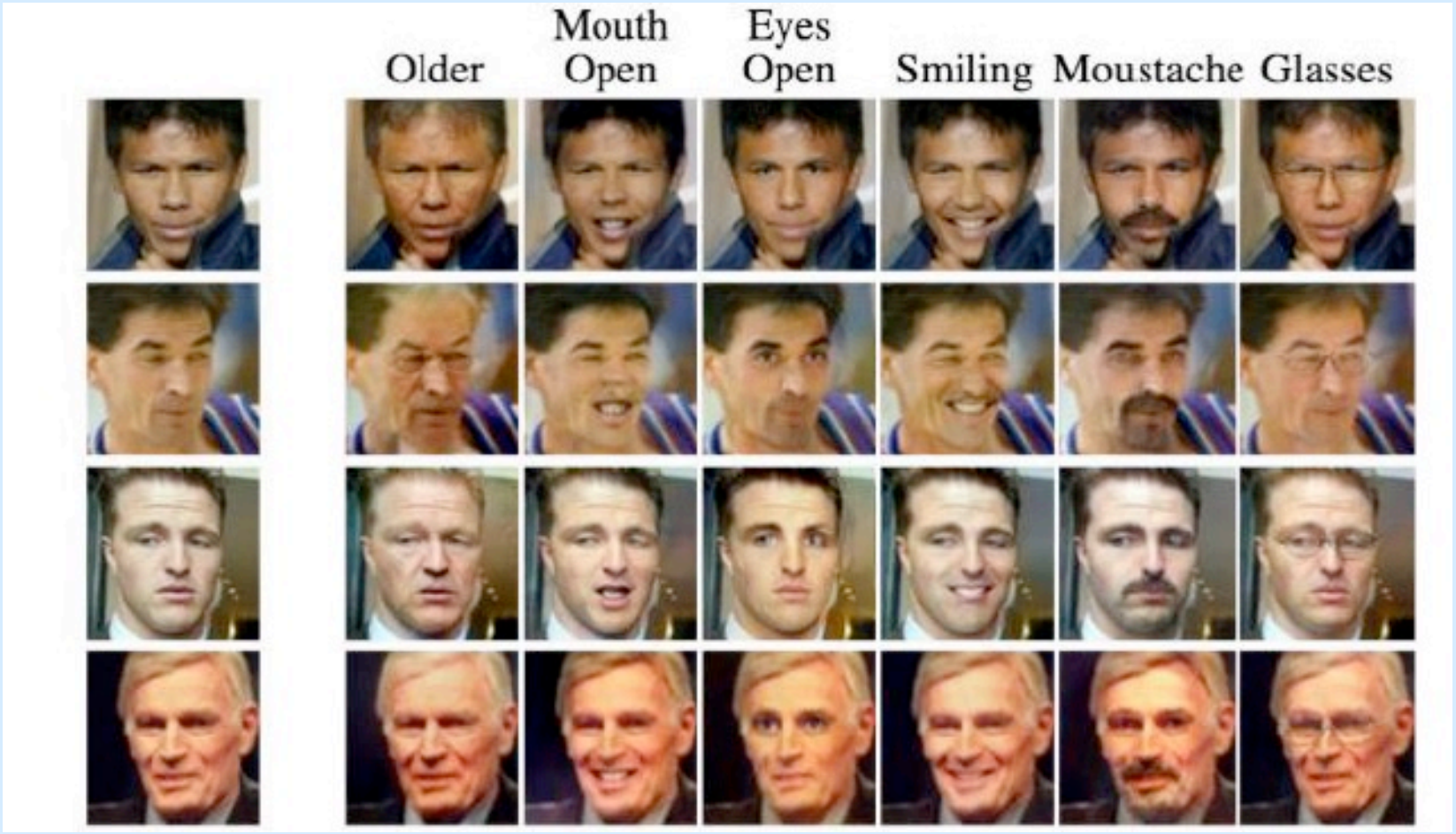
Training Process

- Learns by generating an error signal that measures the difference between the predictions of the network and the desired values and then using this error signal to change the weights(or parameters) so that predictions get more accurate.
-

Deep Learning

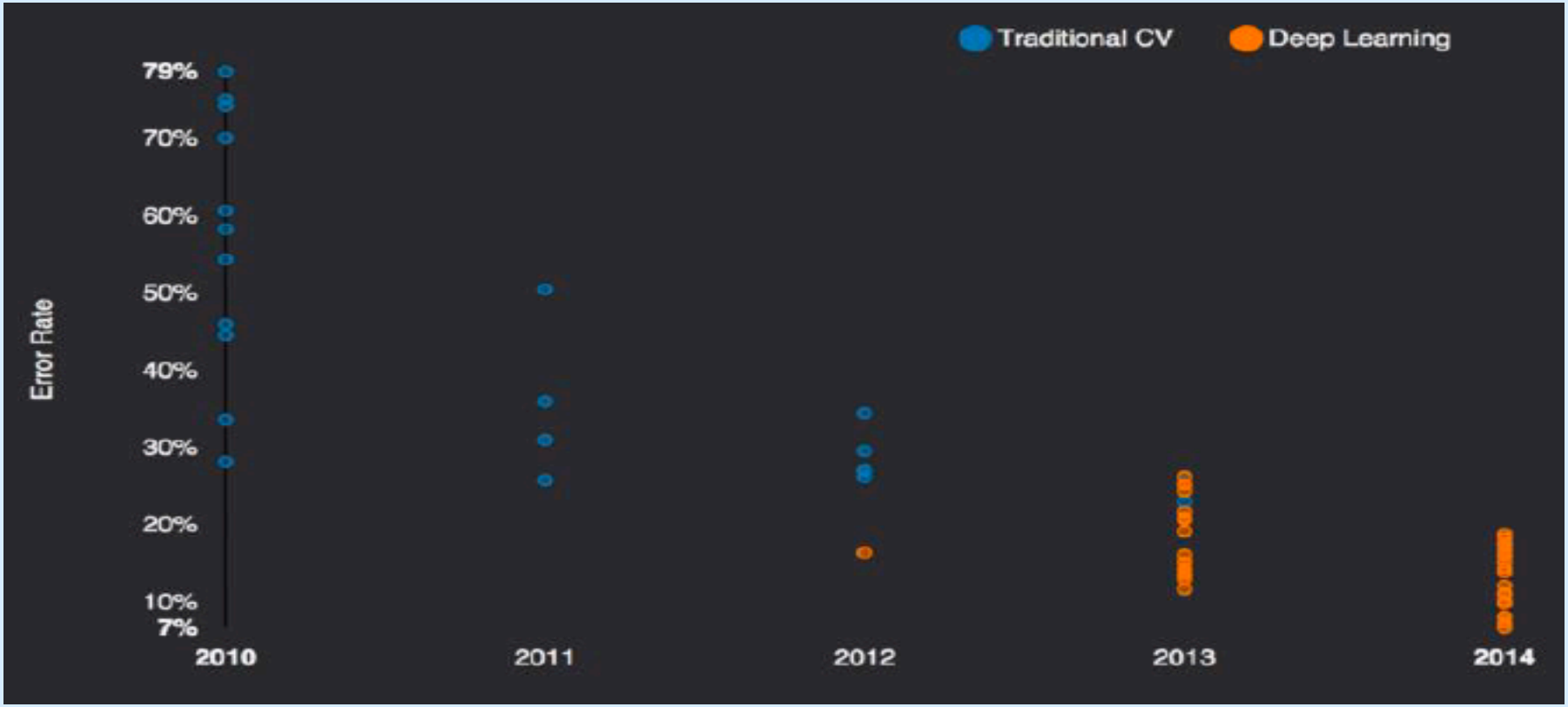


Deep Learning



Applications of Deep Learning

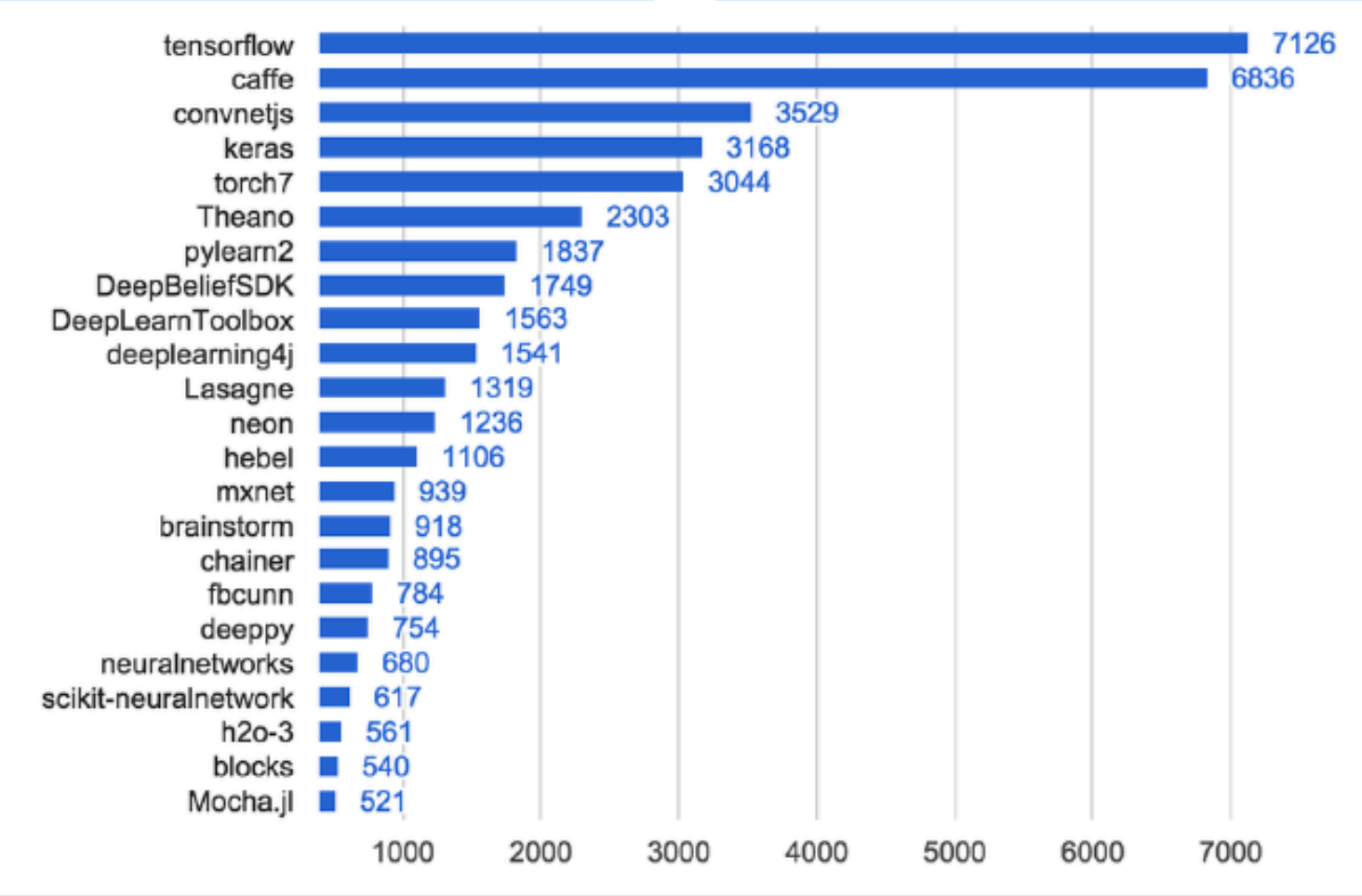
- Speech recognition, Computer Vision, NLP



Usage Requirements

- Large dataset with good quality input-output mappings
- Measurable and describable goals
- Enough computing power – generally GPUs are needed
- Excels in tasks where the basic unit *(pixel, word)* has very little meaning in itself, but the combination of such units has a useful meaning

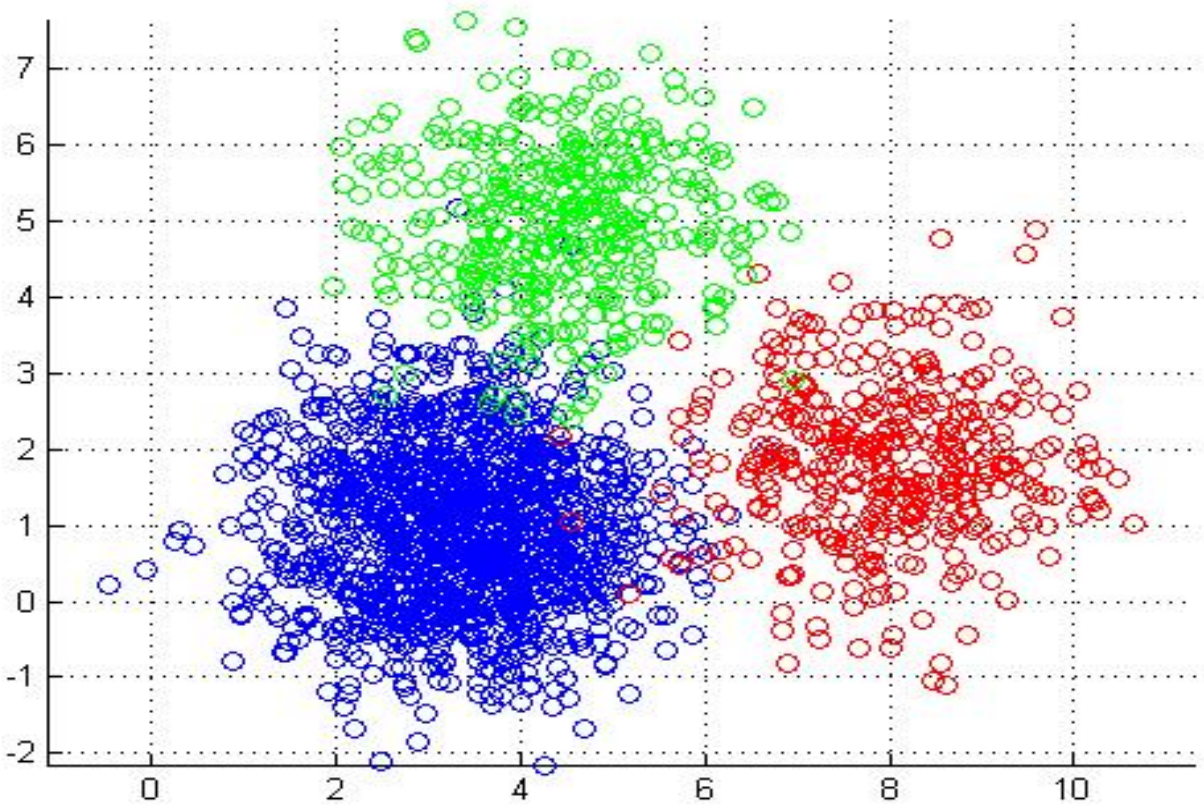
Some Popular Packages



Deep Learning

Unsupervised Algorithm

- Think Clustering
- Used when dataset is unlabelled
- Goal is to cluster observations in meaningful groups
- Try to place observations in different buckets
  - Learning patterns inherent in data
  - PCA, SVD, k-means

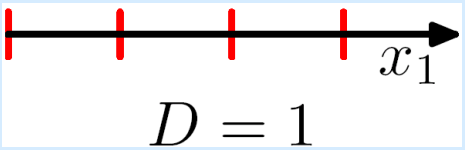

7
6
5
4
3
2
1
0
-1
-2
0
2
4
6
8
10

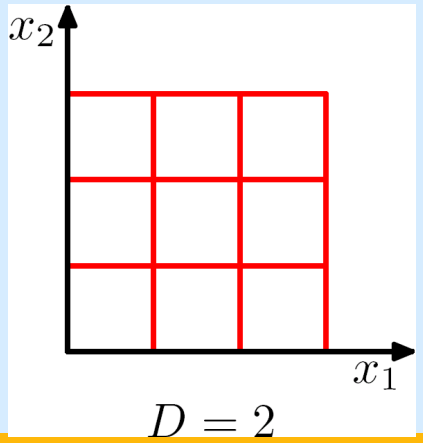
Applications

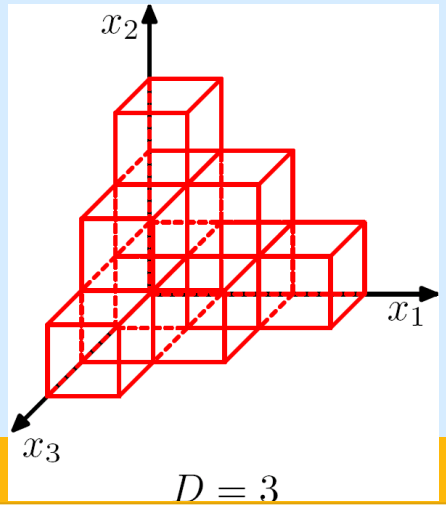
- “How to group customers for targeted marketing purposes?”
- “Which neighborhoods in a country are most similar to each other?”
- “What groups of insurance policy holders have high claim costs?”
- “How to group the products in a store based on their attributes?”
- “How to group pictures based on their description?”

Dimensionality

- What is that?







PCA

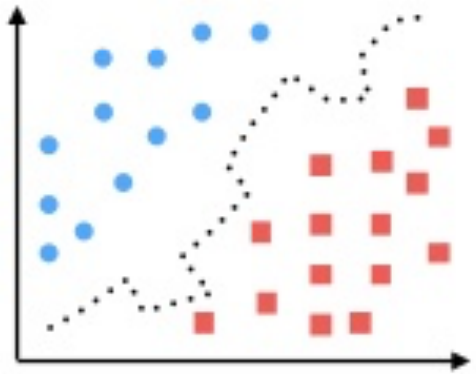
- Basically, reduce number of dimensions
- If you have 50 columns, are all of them needed?
  - Or can we squeeze info of those columns into 3 columns?
  - The three columns would be some combination of 50 columns
- This tremendously helps training of models
- Generally used as a step before actual regressions/classification

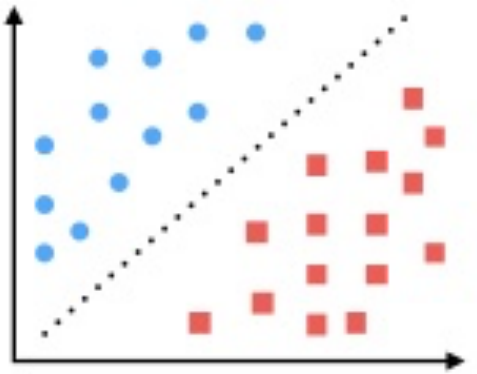
While **dimensionality reduction** is an important tool in machine learning/data mining, we must always be aware that it can distort the data in misleading ways.

Above is a two dimensional projection of an intrinsically three dimensional world….

Original photographer unknown

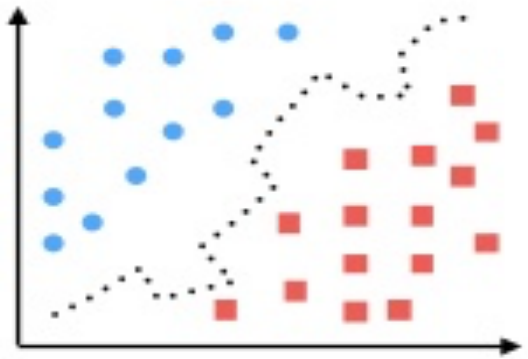
Problem of over-fitting

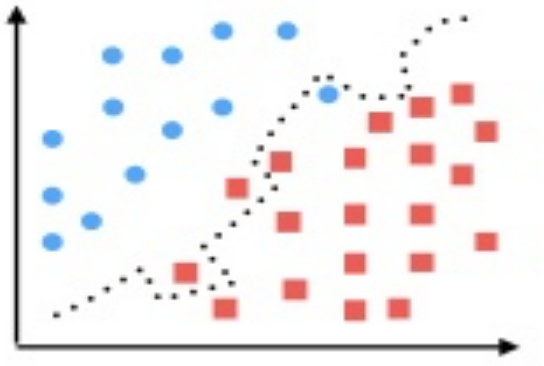


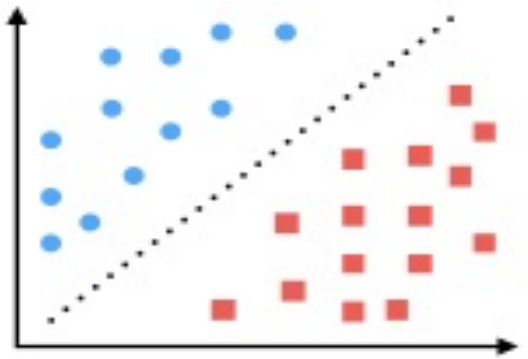


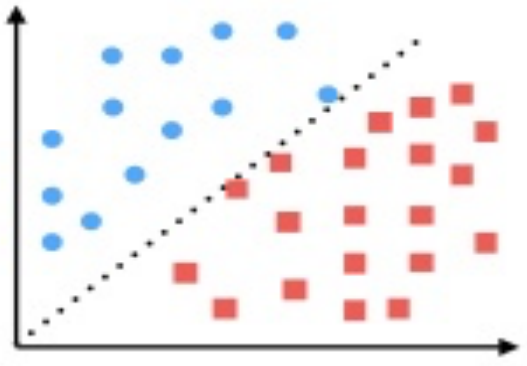
How well does the model perform on unseen data?

Overfitting



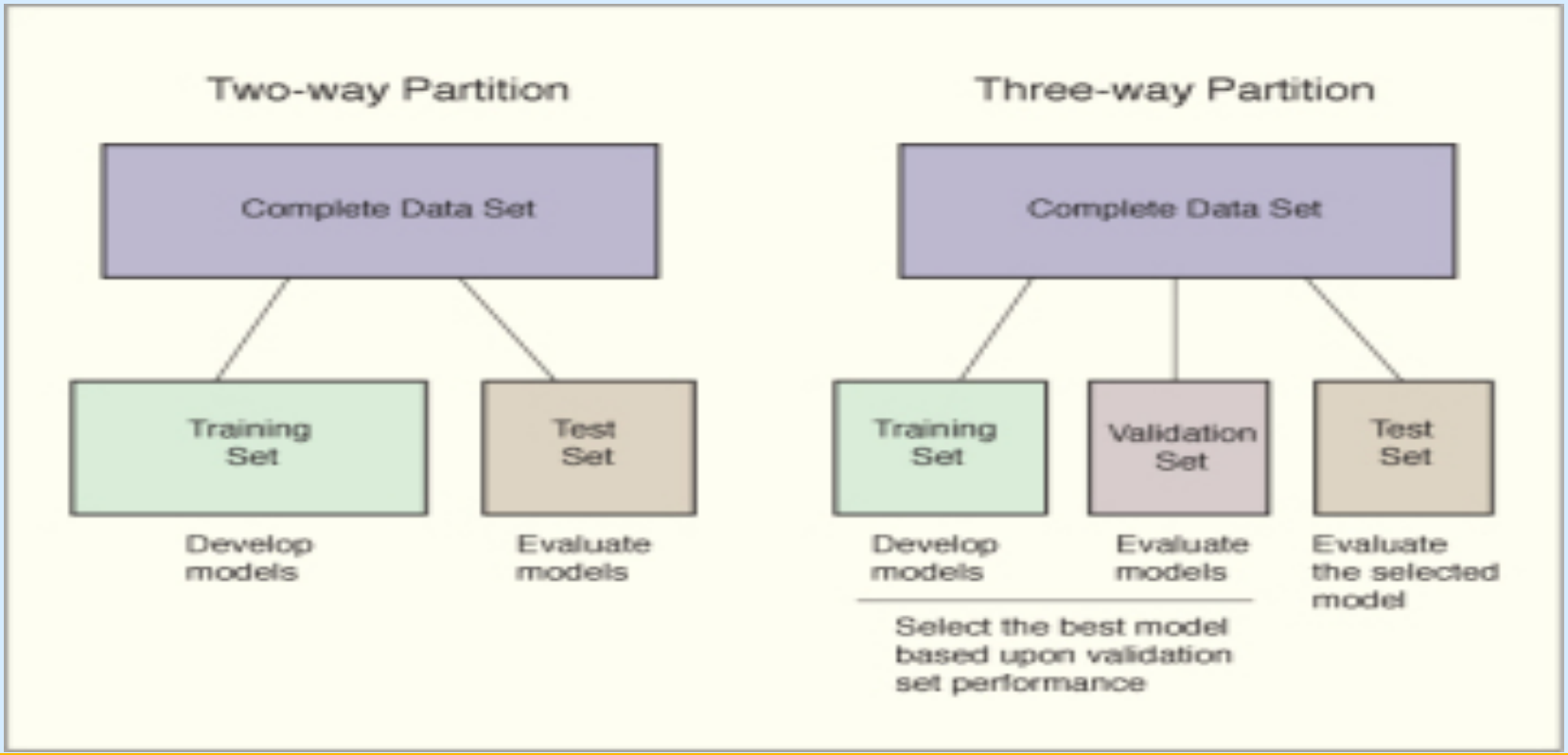






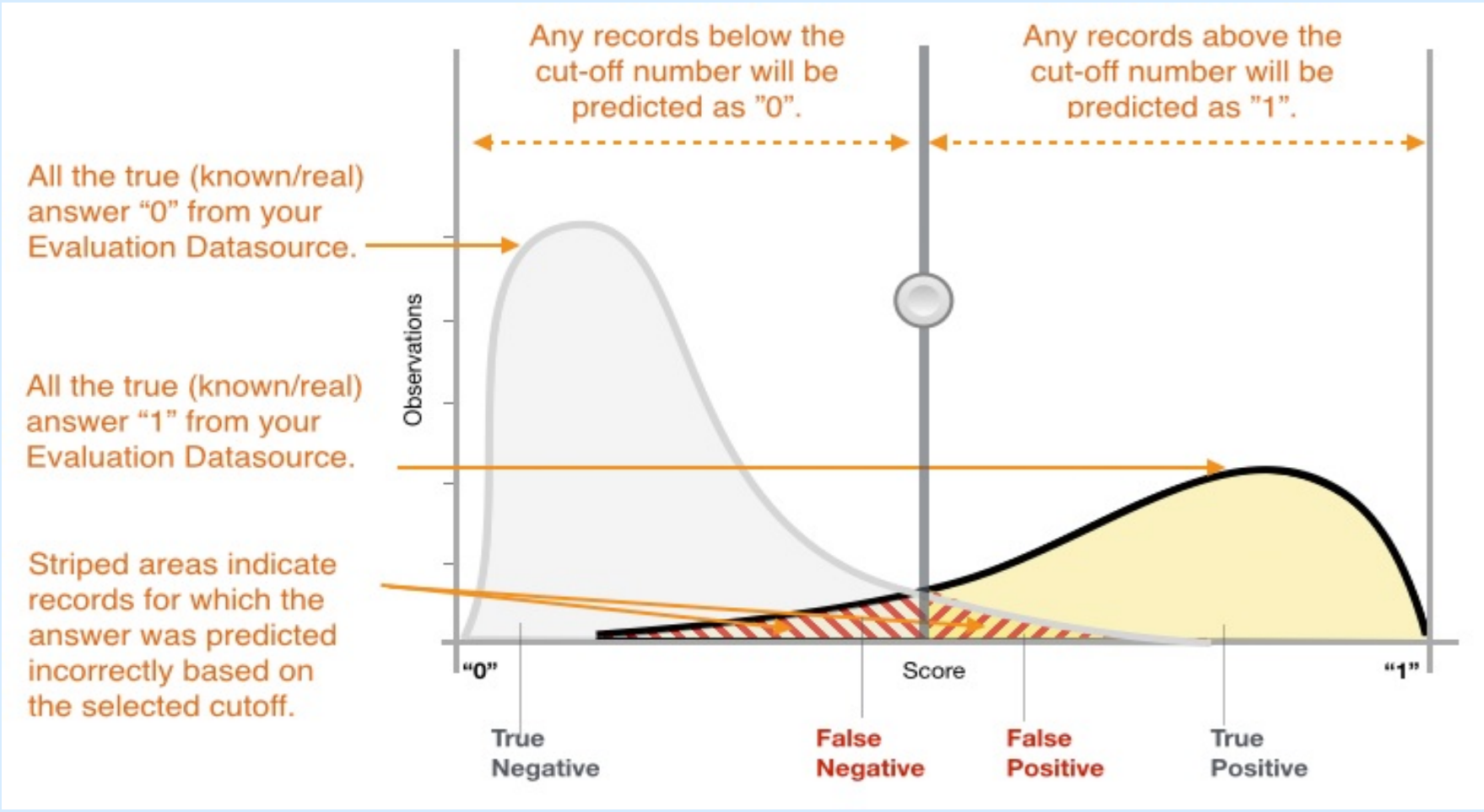
Evaluation of Accuracy

- Question: How to evaluate the predictive accuracy of your model?
- Answer: Partition the data set into Train and Test sets.
- Train is like the “past” you learn from, and Test is like the “future” you predict.



Some ways of evaluating – Binary Classification

Predicted Value (predicted by the test)		Actual Value (as confirmed by experiment): positives	Actual Value (as confirmed by experiment): negatives
	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative


Any records below the cut-off number will be predicted as "0".
Any records above the cut-off number will be predicted as "1".
All the true (known/real) answer "0" from your Evaluation Datasource.
All the true (known/real) answer "1" from your Evaluation Datasource.
Striped areas indicate records for which the answer was predicted incorrectly based on the selected cutoff.
Observations
"0"
Score
"1"
True Negative
False Negative
False Positive
True Positive

Evaluating Classification Accuracy

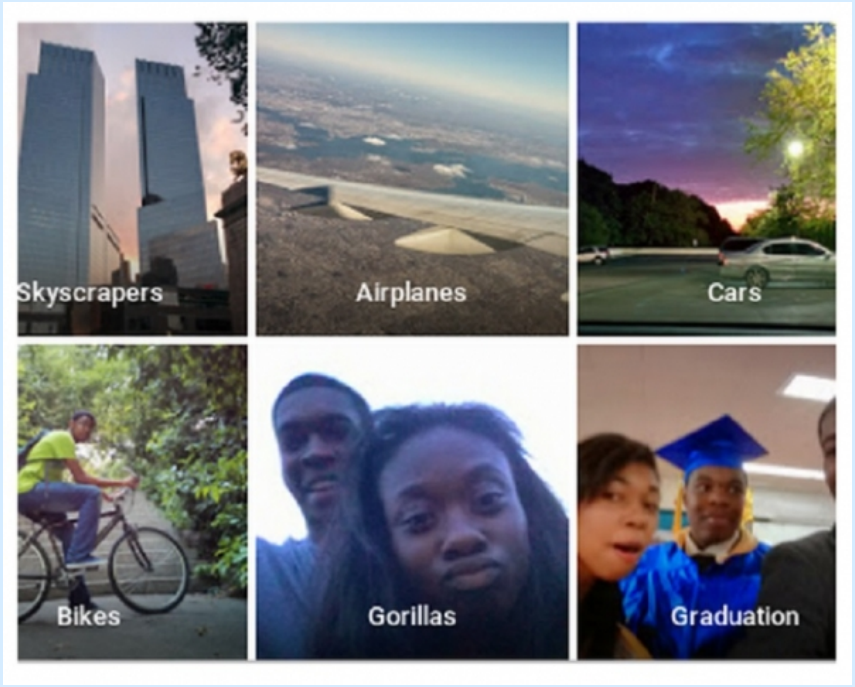
- Simply put, accuracy = (no of correct classifications)/total observations
- There are other things also to consider
- Efficiency
  - Time to construct the model
  - Time to use the model
- Scalability
- Interpretable
  - Can you take the model to a non-technical business audience and convince them to deploy it in business?
- Often times, a succinct model is a good model

Evaluating Continuous Predictions

- One popular metric is Root Mean Squared Error (RMSE)
- Idea is very simple. We want to know quality of prediction
  - For each observation: actual-predicted and then square it
  - Sounds similar to deviation?
  - Well it actually is
  - Deviation is error from the actuals
  - That's what we care for

Is ML all great?

- In May 2015 Flickr released an automatic image tagging capability that mistakenly labeled a black man for an ape.
- Soon afterwards, Google came up with a photo labeling tool similar to Flickr, which made similar mistakes. Black men were tagged as gorillas.
- A recent Carnegie Mellon University study showed that Google displayed ads in a way that discriminated based on the gender of the user.



Fundamental Assumptions of Learning

- Training data distribution is similar to test data distribution
- This does not work out in real life
  - Data is representation of reality
  - As reality changes, data also changes
- If the violations are more severe, this would imply poor prediction results even if your model works very well on training data

So whom to blame

"No Free Lunch" :(

D. H. Wolpert. The supervised learning no-free-lunch theorems. In Soft Computing and Industry, pages 25–42. Springer, 2002.

Our model is a simplification of reality

⇩

Simplification is based on assumptions (model bias)

⇩

Assumptions fail in certain situations

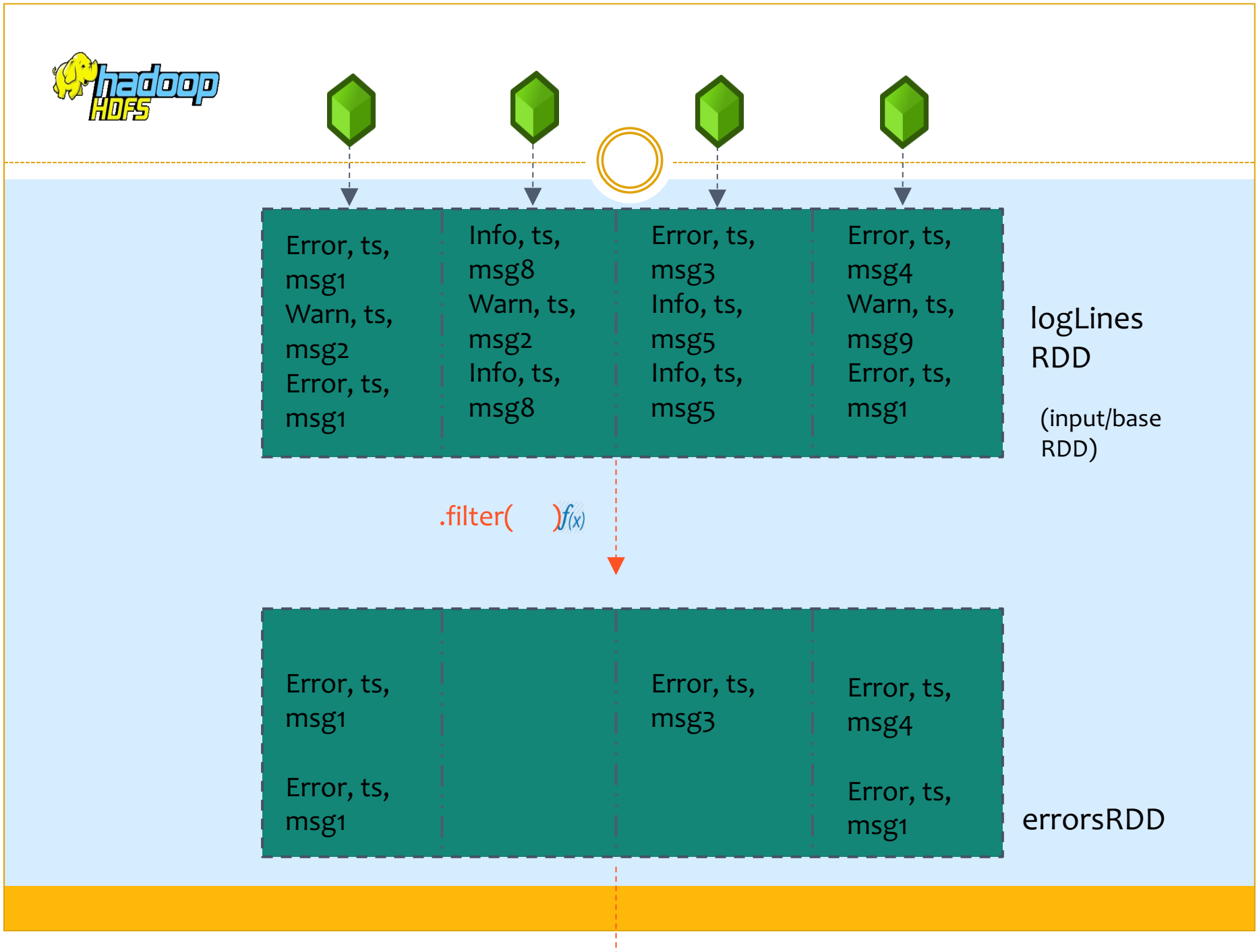
Roughly speaking:

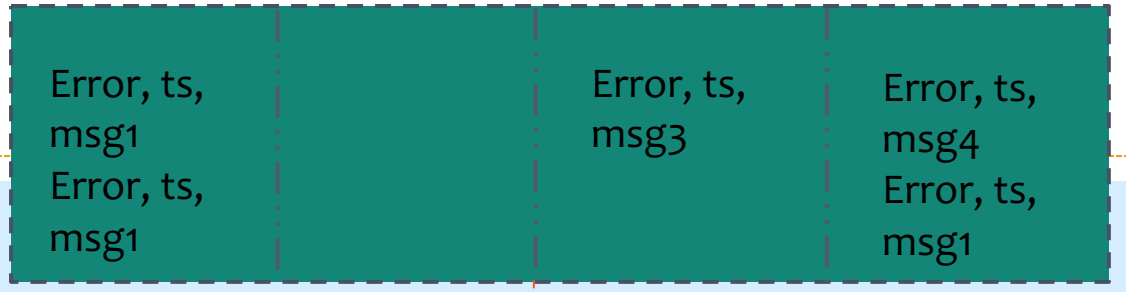
"No one model works best for all possible situations."

Quick Review of Spark

Spark Core Components

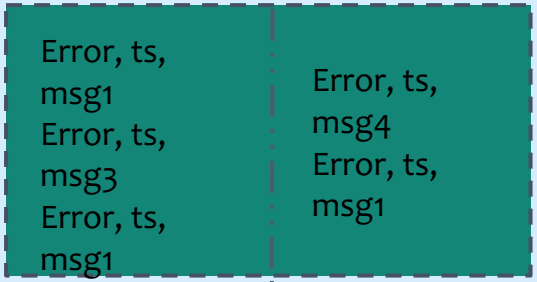
- Three building blocks
- SparkContext
  - Entry point to the Spark program
  - This is what you call from pyspark or from your own application
- RDD
  - Resilient Distributed Dataset
  - Container for your data
- Operations – Actions and Transformations


hadoop
HDFS
Error, ts, msg1
Warn, ts, msg2
Error, ts, msg1
Info, ts, msg8
Warn, ts, msg2
Info, ts, msg8
Error, ts, msg3
Info, ts, msg5
Info, ts, msg5
Error, ts, msg4
Warn, ts, msg9
Error, ts, msg1
logLines RDD
(input/base RDD)
.filter( )f(x)
Error, ts, msg1
Error, ts, msg1
Error, ts, msg3
Error, ts, msg4
Error, ts, msg1
errorsRDD



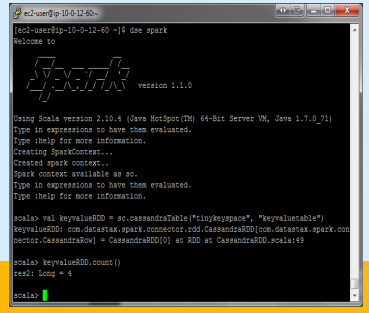
errorsRDD

.coalesce(2)



cleanedRDD

.collect( )



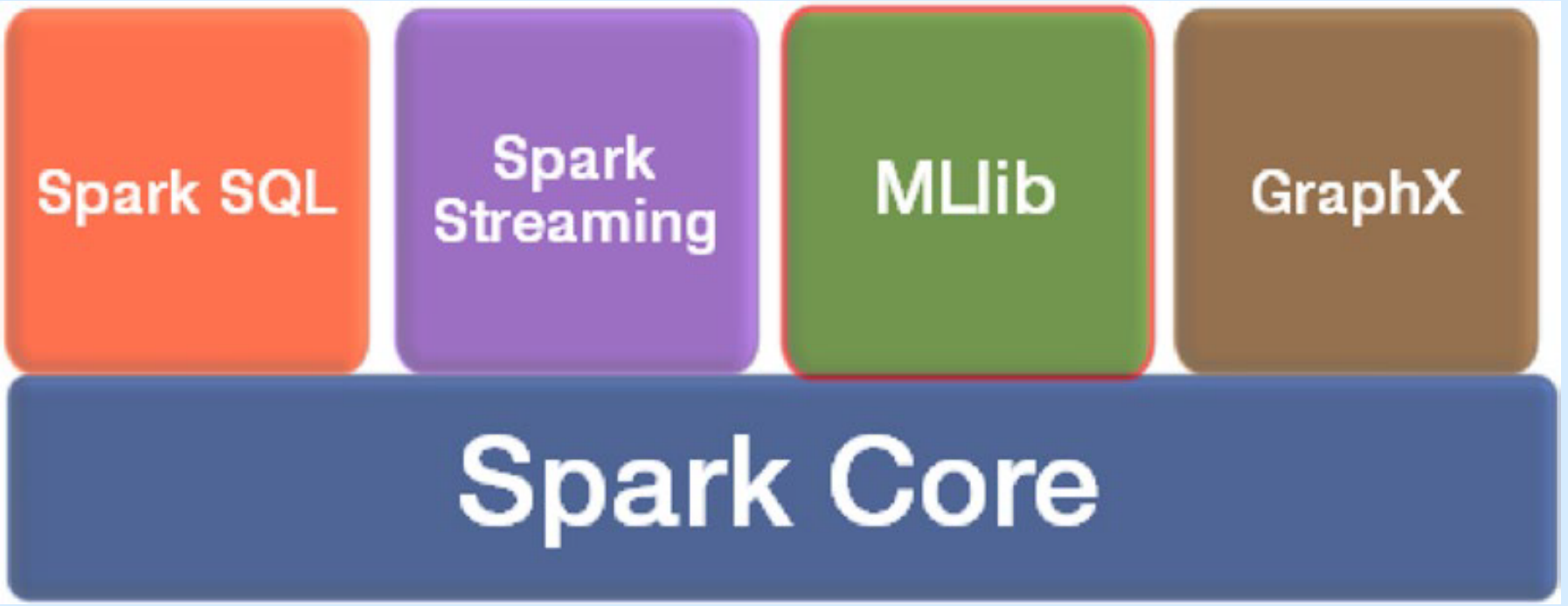
Driver

Technical Stuff Starts Now

Machine Learning in Spark

- Two libraries
  - MLLib
  - Spark ML (Pipelines API)
- Scalable (Overcome the problem of data fitting one machine only)
- MLLib is more mature
- Commonly used ML algos and statistical analysis tools
- Machine Learning on large datasets
- Scalable, faster
- Both batch and streaming data

MLLib & Spark ML



Statistical Utilities

- Summary stats
- Correlation
- Hypothesis Testing
- Sampling
- Random data

Algorithms

- Regression
  - Linear and Logistic, SVM, Naive Bayes, Decision Tree, Random Forest
- Clustering
  - K-means, Gaussian, LDA, PCA, SVD
- Data Mining and Feature Extraction
  - TF-IDF, Word2Vec, Normalization, Item set mining, Association Rules, Collaborative Filtering
- APIs for Deep Learning and ANN

Algorithms

- Alternating Least Squares
- Lasso
- Ridge Regression
- Logistic Regression
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- K-Means
- Gradient descent
- L-BFGS
- Random data generation
- Linear algebra
- Feature transformations
- Statistics: testing, correlation
- Evaluation metrics

Collaborative Filtering for Recommendation

Prediction

Clustering

Optimization

Many Utilities

API

- Data Types – MLLib operates on these data types
- Vector
- LabeledPoint
- Rating

Vector

- Represents one observation in a dataset
- Represents element in n-dimensional space
- Used for representing features
- Vector of length n represents observation with n features
- Two types of vectors
  - DenseVector
  - SparseVector

Data Type - Vector

- Dense
  - Stores a value at each position in vector
  - Use if your dataset does not have too many zeros
    - import org.apache.spark.mllib.linalg._
    - val denseVector = Vectors.dense(1.0, 0.0, 3.0)
- Sparse
  - Stores only non-zero values
    - import org.apache.spark.mllib.linalg._
    - val sparseVector = Vectors.sparse(10, Array(3, 6), Array(100.0, 200.0))

LabeledPoint

- An observation in a labeled dataset
- Contains both label and features (Vector)
- Primary RDD abstraction
- You should transform dataset into RDD of LabeledPoints to do ML on Spark
- Can represent both categorical and numerical values

LabeledPoint Example

- import org.apache.spark.mllib.linalg.Vectors
- import org.apache.spark.mllib.regression.LabeledPoint
- val positive = LabeledPoint(1.0, Vectors.dense(10.0, 30.0, 20.0))
- val negative = LabeledPoint(0.0, Vectors.sparse(3, Array(0, 2), Array(200.0, 300.0)))

Rating

- Used with recommendation algorithms
- Represents user rating
- Must transform dataset into an RDD of Ratings to run a recommendation algorithm
- Has three fields
  - User
  - Product
  - Rating
- import org.apache.spark.mllib.recommendation._
- val rating = Rating(100, 10, 3.0)

Feature Extraction and Basic Statistics

- Several functions for basic operations
- Scaling, normalization, summary statistics, correlations, sampling etc
- Random Number Generators
- TF-IDF

Algorithms and Models

- Model represented by a class
- ML algorithm also represented by a class
- Two key methods
  - Train
  - Predict

Regression Algorithm

- Many different regression methods available
  - LinearRegression, IsotonicRegression, RideRegression
- Train method fits a linear regression model to data
  - Returns an object of type LinearRegressionModel
- Takes RDD of LabeledPoints and returns regression model

- Let us move to hands-on