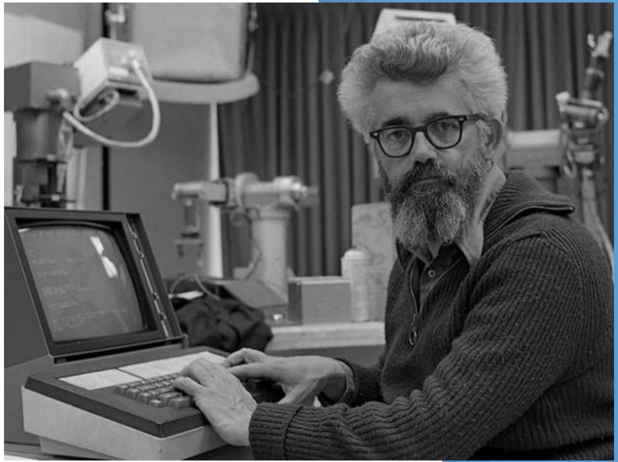
Introduction to AI and Machine Learning

Nirand Pisutha-Arnond

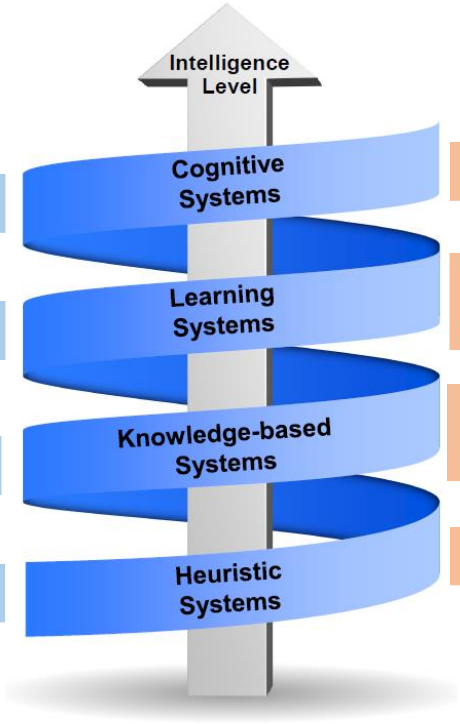
Department of Industrial Engineering
Faculty of Engineering

What is Artificial Intelligence ?

- “The science and engineering of making intelligent machines, especially intelligent computer programs”.
  - John McCarthy



Four Phases of AI Research



Phase 4 –2010 -Present

Hybrid Architectures

Phase 3 –1990 -2010

Machine Learning

Phase 2 –1970 -1990

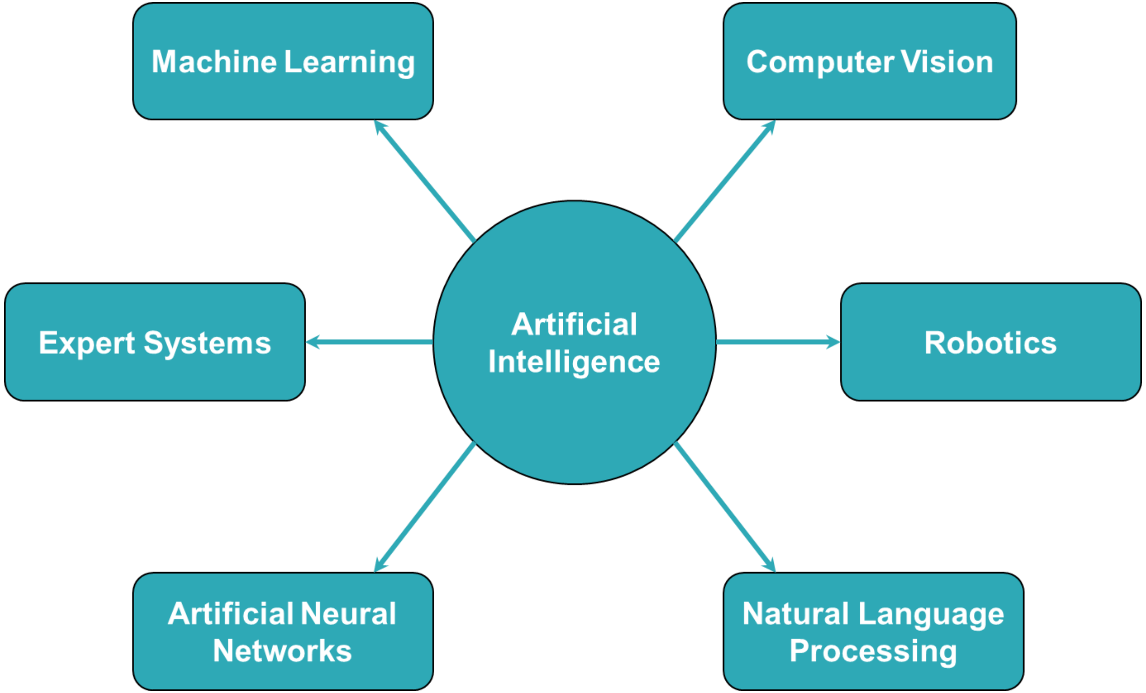
Knowledge Representation and Reasoning

Phase 1 –1950 -1970

Heuristic Search

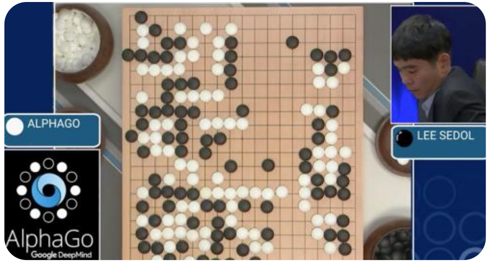
Branches of Artificial Intelligence

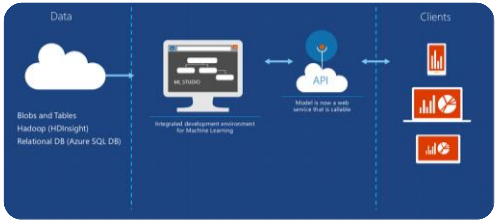
We will learn more about this today.

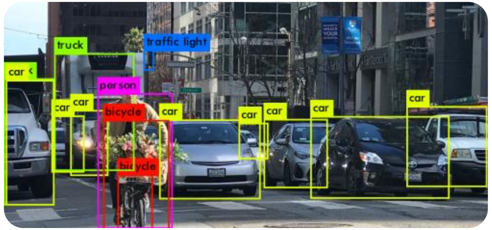


Applications of AI

- Game Playing
- Expert Systems
    - Chatbot
    - Personal Assistant
- Data Analytics
- Object Detection
- Self-Driving Cars







Machine Learning

What is Machine Learning?

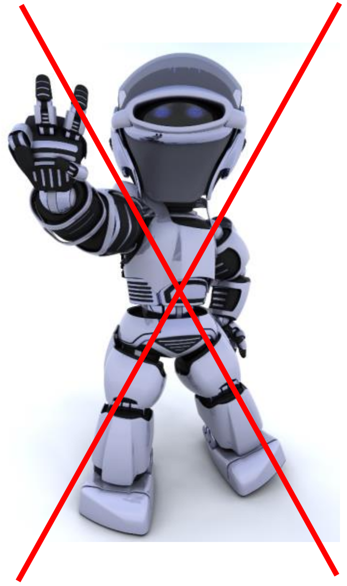
- Subfield of artificial intelligence
  - Concerned with techniques that allow computers to “learn”.
  - Without being explicitly programmed.

What is Machine Learning?

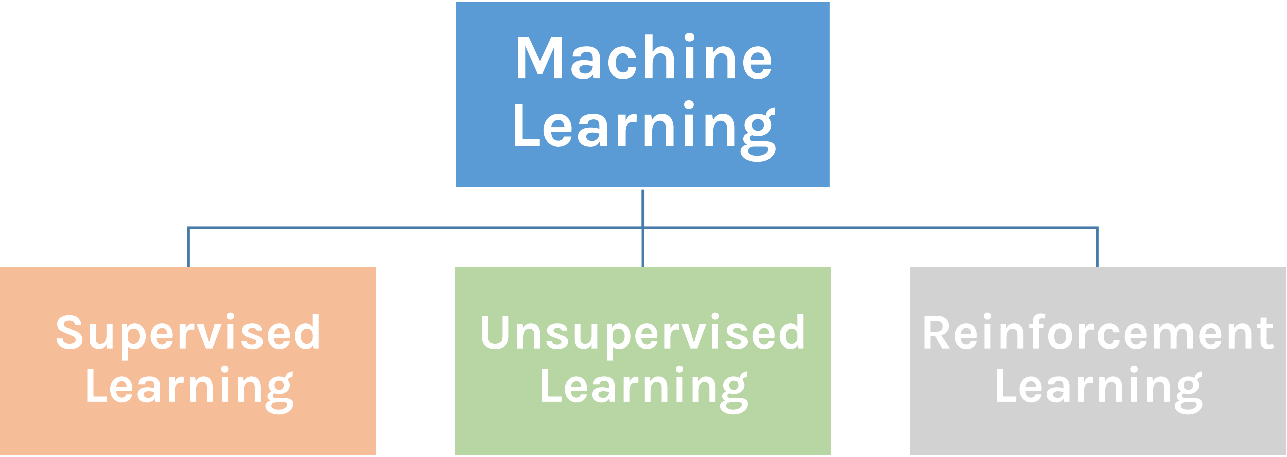
Learn from experience

Learn from ~~experience~~ *Data*

Follow instructions



Types of Machine Learning



Task driven
(Regression / Classification)

Data driven
(Clustering)

Algorithm learns to react to an environment

We will learn more about this today.

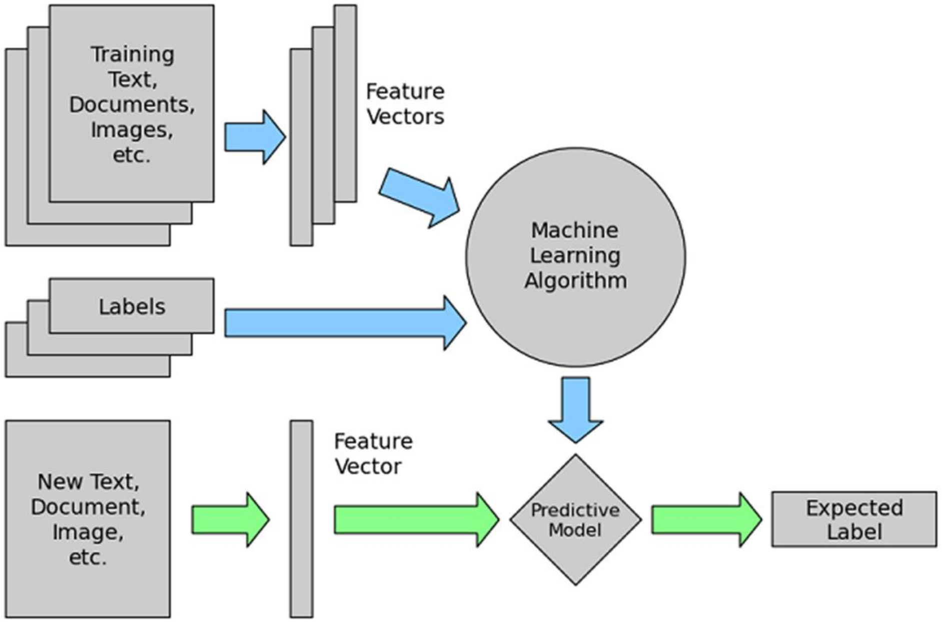
Supervised Learning Concept

Known Data Set (Training Data) → Model → Predictions

A set of
- Input data
- Response values

Creates response values from input data.

Detailed Concept



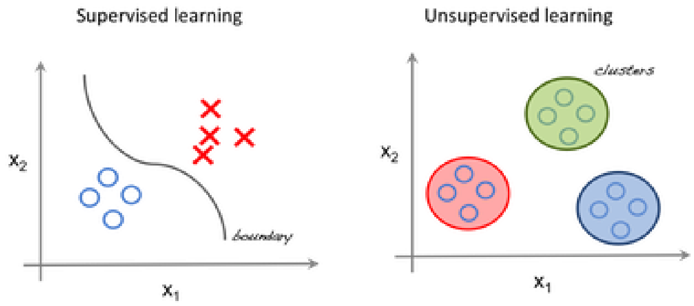
Types of Supervised Learning

Supervised learning can be separated into two general categories of algorithms:

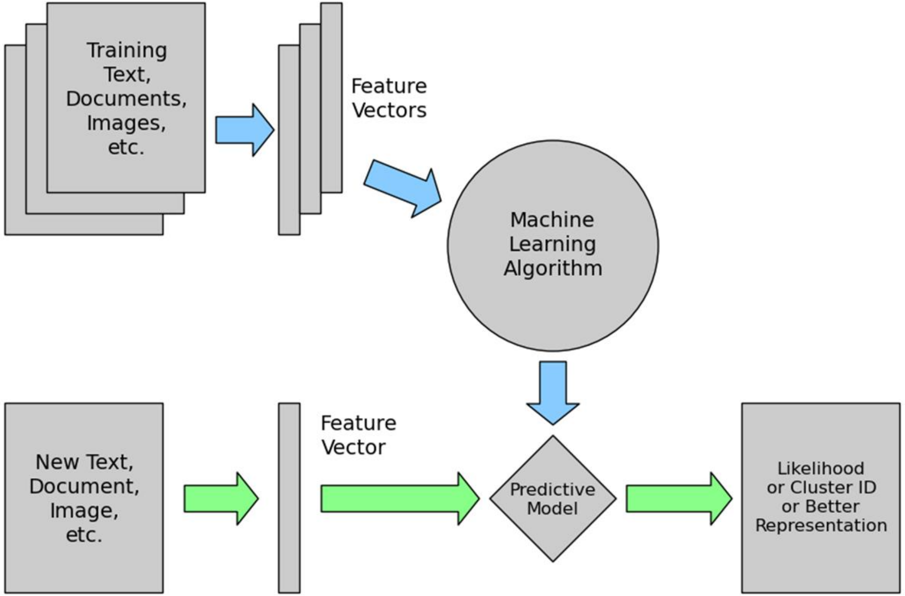
- **Classification:**
  - Categorical response values, where the data can be separated into specific “classes”
- **Regression**
  - Continuous-response values

Unsupervised Learning

- Operates on **unlabeled** examples.
  - Correct responses are not provided
- The algorithm tries to identify **similarities** between the inputs
  - Inputs that have something in common are categorized together.
- This is called **clustering**.



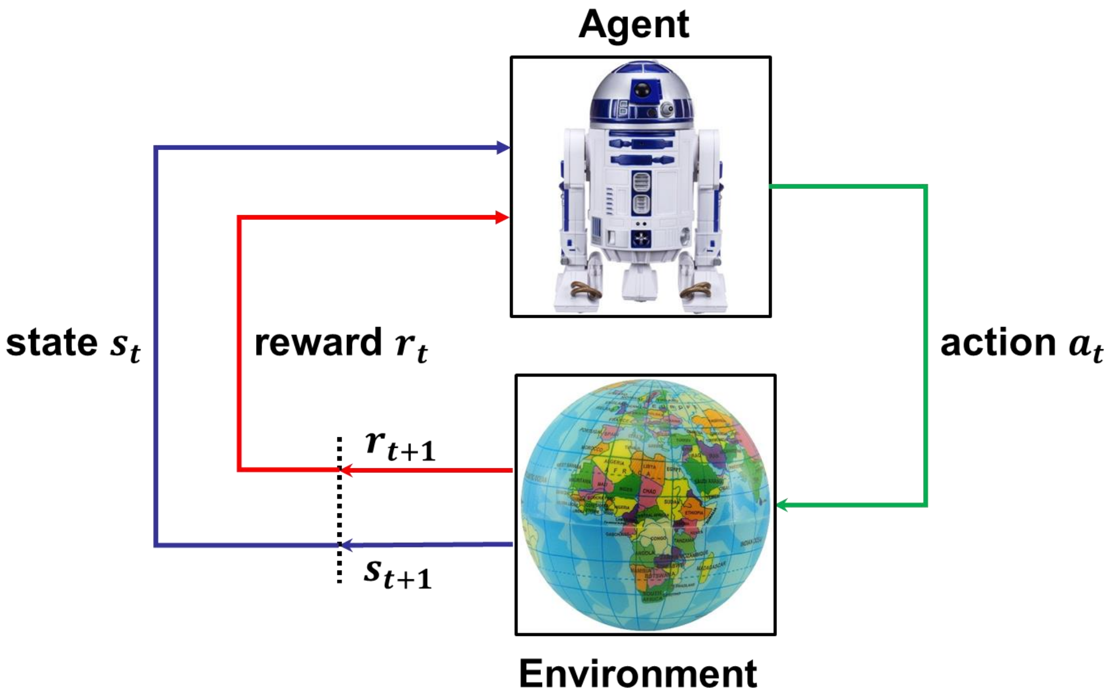
Unsupervised Learning



Reinforcement Learning

- Type of ML that interacts with the **environment**
  - Learns which sequence of actions yields the most favorable results.
- The learner is a decision-making **agent** that takes actions in an environment
  - Receives reward (or penalty) for its actions.
- After a set of trial-and-error runs, it should learn the best **policy**
  - The sequence of actions that maximize the total reward.

Reinforcement Learning



Mario Reinforcement Learning

https://youtu.be/qv6UVOQ0F44

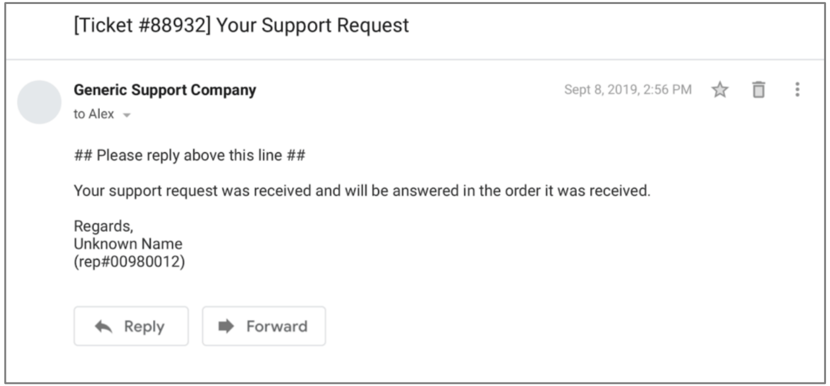
Supervised Learning

Case Study: Spam Detector

- Classification problem
- Analyzes text from email
  - Then determine whether the email is a spam or not.

Training Data (Raw)

A lot of these examples


[Ticket #88932] Your Support Request

Generic Support Company Sept 8, 2019, 2:56 PM

to Alex

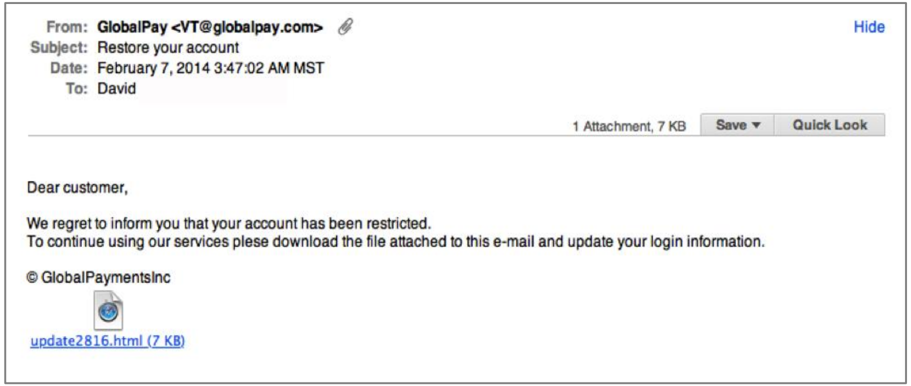
Please reply above this line

Your support request was received and will be answered in the order it was received.

Regards,
Unknown Name
(rep#00980012)

Reply Forward

Not Spam


From: GlobalPay <VT@globalpay.com> Hide
Subject: Restore your account
Date: February 7, 2014 3:47:02 AM MST
To: David

1 Attachment, 7 KB Save Quick Look

Dear customer,

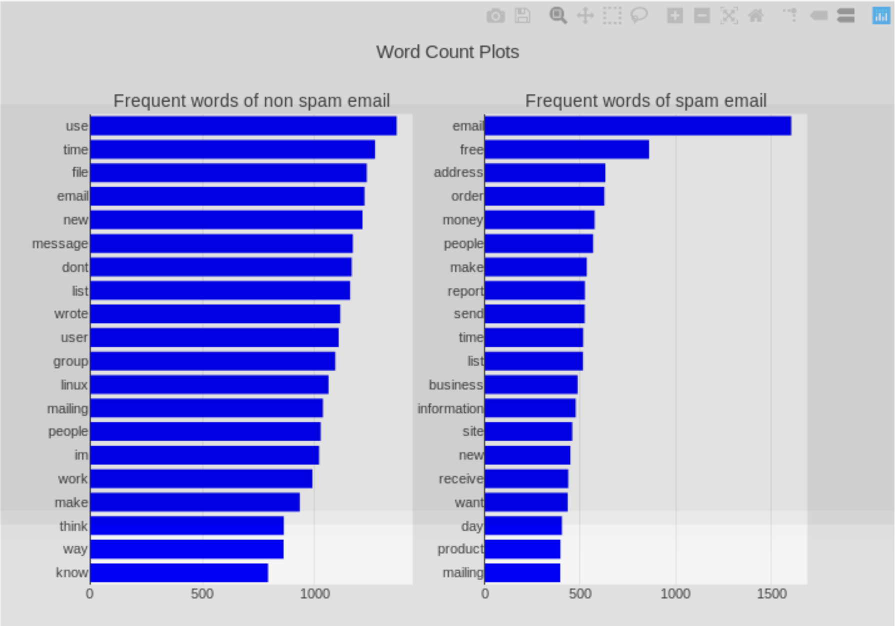
We regret to inform you that your account has been restricted.
To continue using our services plese download the file attached to this e-mail and update your login information.

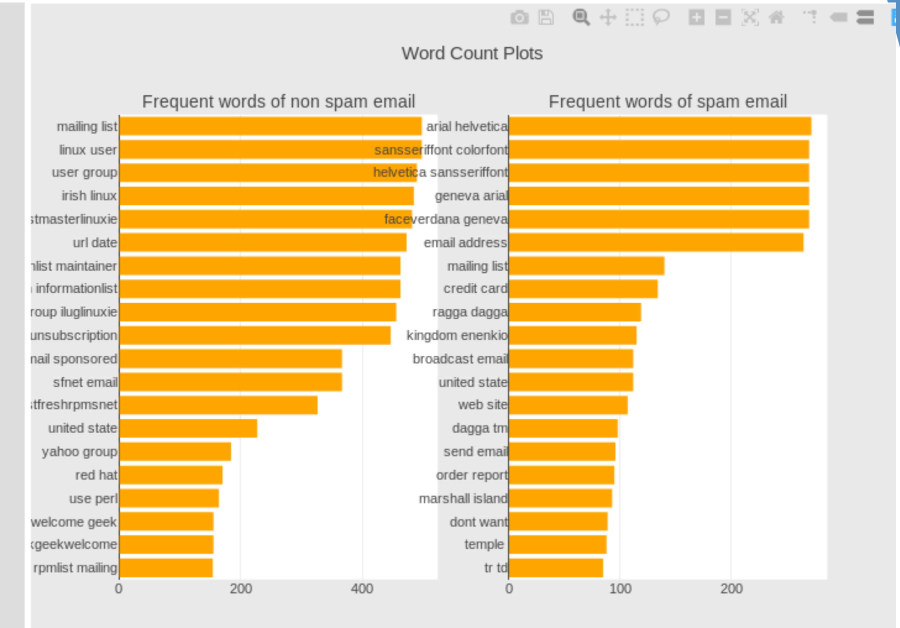
© GlobalPaymentsInc

update2816.html (7 KB)

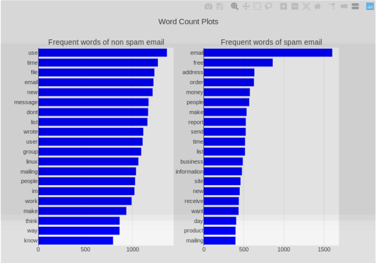
Spam

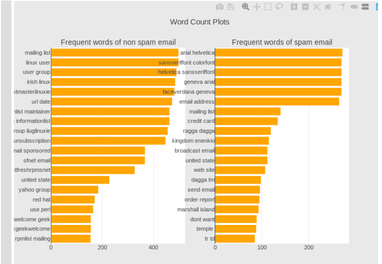
Exploratory Data Analysis





Training Data (Structured)





No.	Contain "Use"	Contain "Email"	Not Spam?
1	✓	✗	✓
2	✗	✓	✗
3	✓	✓	✗
…	…	…	…

Attributes

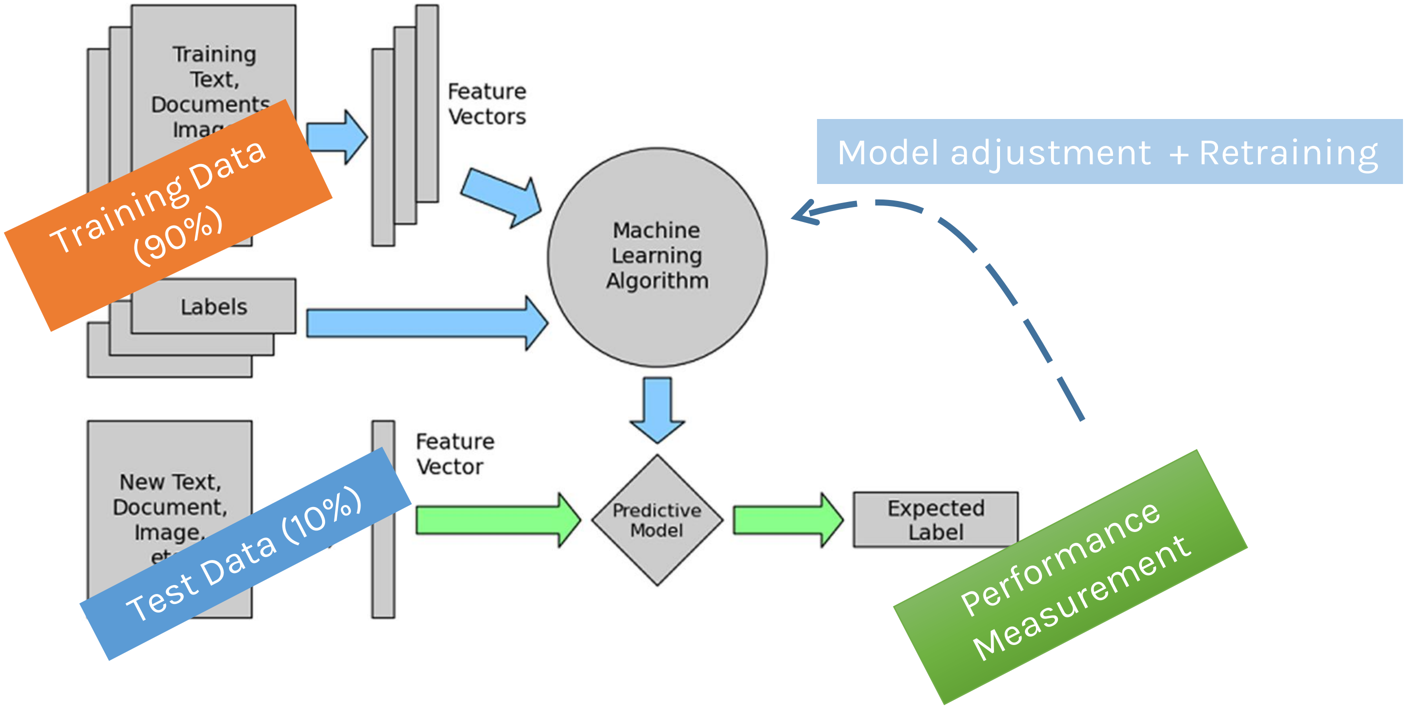
Response (Class)

Test Data

No.	Contain "Use"	Contain "Email"	Spam?
1	✓	✗	✓
2	✗	✓	✗
3	✓	✓	✗
…	…	…	…
…	…	…	…
…	…	…	…

Training Data (90%)

Test Data (10%)



Quality Data = Better Model

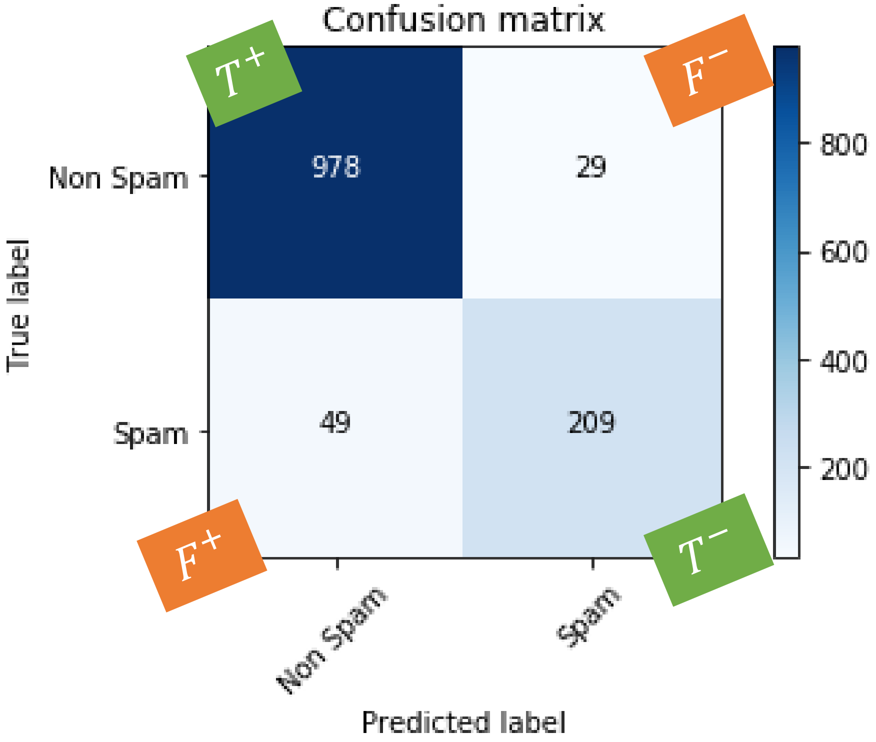
Performance Analysis

- **Confusion matrix**
  - True positive → T^+
  - True negative → T^-
  - False positive → F^+
  - False negative → F^-

$$F^+$$

(เลือก)ทายผิด

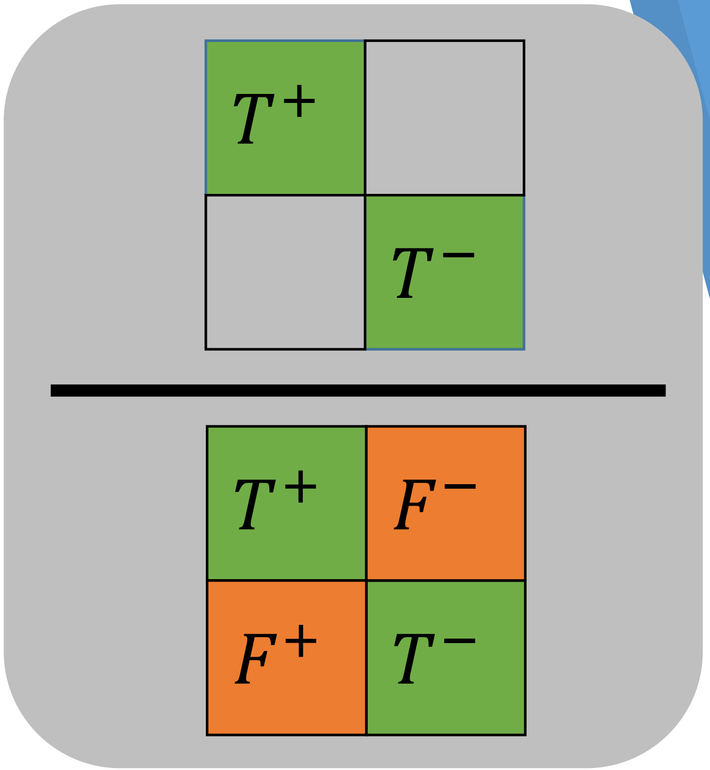
ทายว่า True (not spam)



Performance Analysis

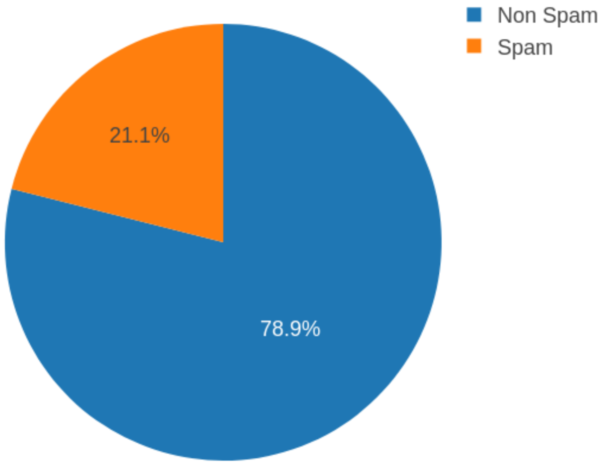
- **Accuracy** = $\frac{T^{+}+T^{-}}{T^{+}+T^{-}+F^{+}+F^{-}}$

- Accuracy may not be the best indicator.
  - For imbalanced class



Accuracy Problem

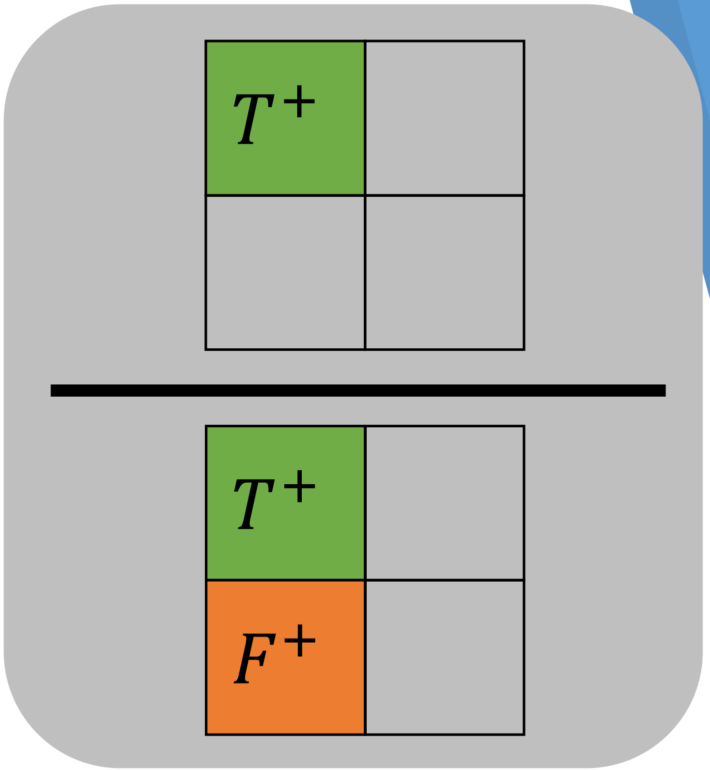
- Let's assume that in real life
  - Spam email is about 20%.
- For un-bias data collection
  - We should obtain similar distribution.
- In this case, I can build a model with 80% accuracy
  - By simply predicting every email as non-spam.



Performance Analysis

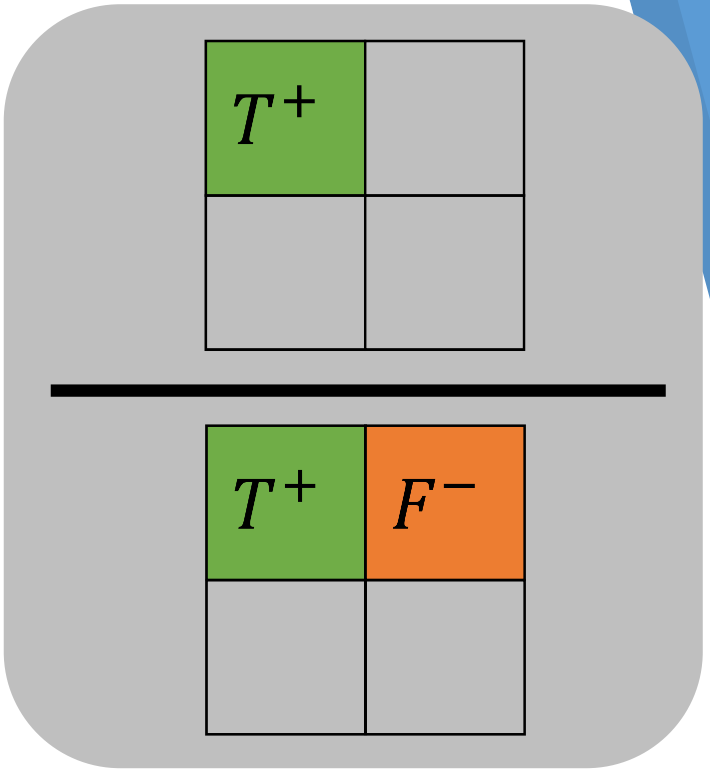
- **Precision** = $\frac{T^{+}}{T^{+}+F^{+}}$

- What proportion of positive identifications was actually correct?
- When a model predicts something as positive, how accurate is the model?



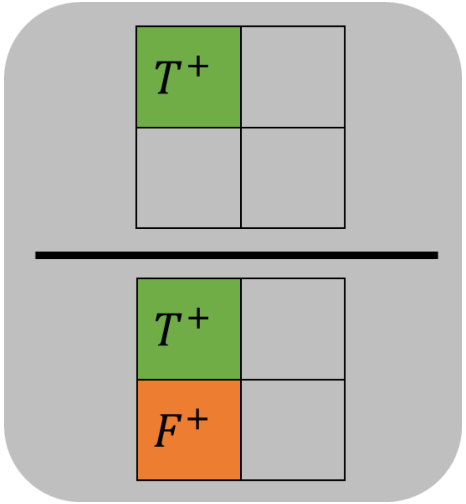
Performance Analysis

- **Recall** = $\frac{T^{+}}{T^{+}+F^{-}}$
  - Also called "True Positive Rate"
- What actual proportion of actual positives was identified correctly?
- Evaluating how well a model in finding all the positive samples.



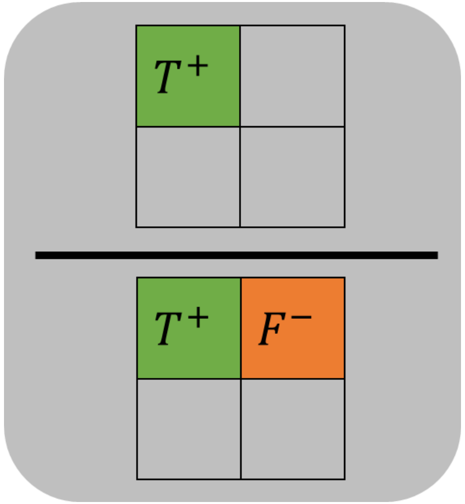
Precision VS Recall

Precision



ความถูกต้องของการทำนายว่า "ถูก"

Recall



ความสามารถในการหาตัวที่ถูก

Performance of a Spam Filter

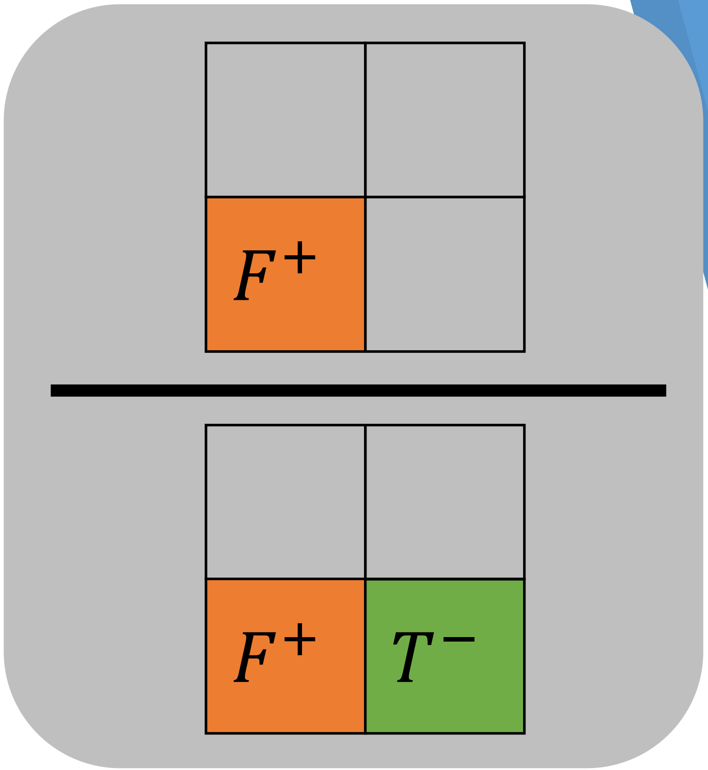
Measurement	Value
Accuracy	93.2%
Precision	87.82%
Recall	81.01%

The model might not be doing a good enough job in discovering the spam email.

Performance Analysis

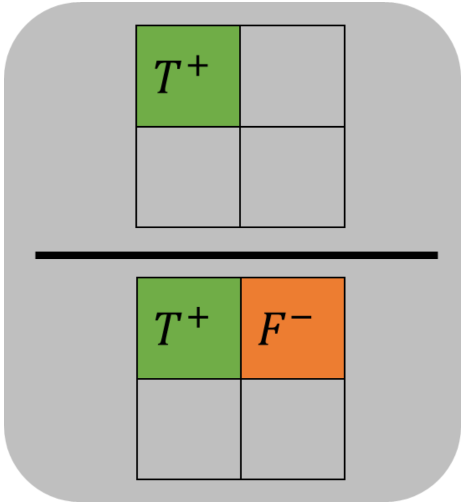
- **False Positive Rate** = $\frac{F^+}{F^+ + T^-}$

- Proportion of negative cases incorrectly identified as positive cases



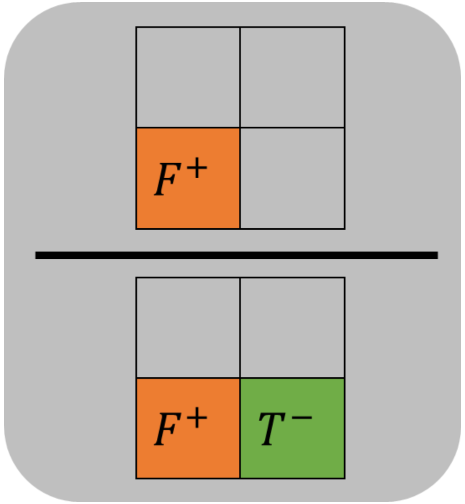
True Positive VS False Positive

True Positive Rate (Recall)



ความสามารถในการหาตัวที่ถูก

False Positive Rate



ความ ‘ไม่สามารถ’ ในการหาตัวที่ถูก

Performance Analysis

- **True Negative Rate** = 1 – **False Positive Rate**
  - Also called **Specificity**
- **False Negative Rate** = 1 – **True Positive Rate**
  - Also called **Miss Rate**

Performance Analysis

- **F-Measure** = $\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

- **Mean absolute error (MAE)**
  - Average of absolute difference between prediction and observation.

- **Root mean squared error (RMSE)**
  - Similar to MAE but use the root of squared sum.

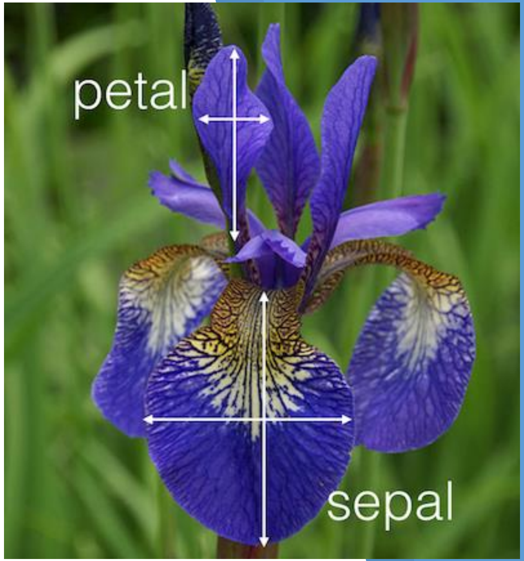
Performance Analysis

- **K-value**
  - Agreement of the prediction with the true class
  - Value of 0
    - Model is no better than guessing
  - Value between 0 and 1
    - Model is better than guessing
- **ROC Area**
  - Probability that a randomly chosen positive instance is ranked above the randomly chosen negative instance.
  - Value of 0.5
    - Random ranking
  - Value close to 1
    - Correct ranking
  - Value close to 0
    - Anti-learning

Machine Learning with Weka

Activity 1 – Loading Data

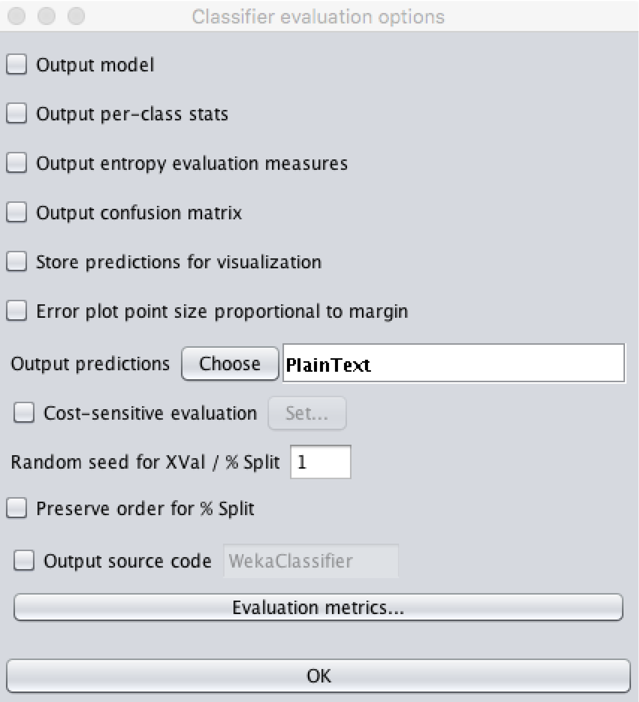
- Use text editor to view data/iris.arff
- Use Weka Explorer
- Analyze iris.arff
  - Preprocess tab
  - Visualize tab

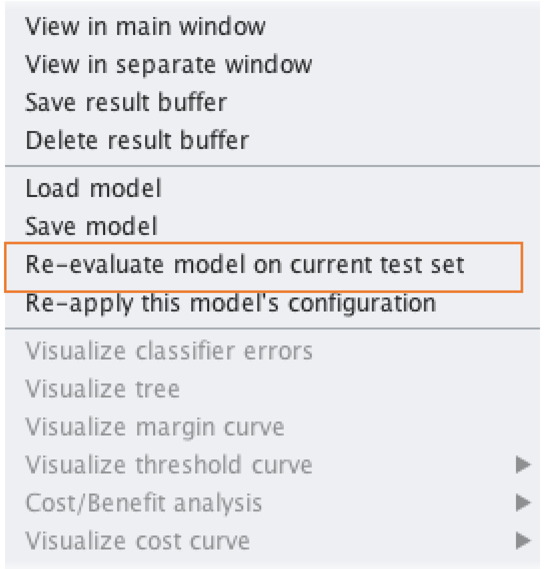


Activity 2 - Training

- Train the model with rules.ZeroR
  - Zero Rule Algorithm
- Train the model with trees.J48
  - C4.5 Algorithm (Decision Tree)
- Train the model with laze.Ibk
  - K-Nearest Neighbors Algorithm
  - Try K=1 and K=11

Activity 3 – Make Predictions





Machine Learning Algorithms

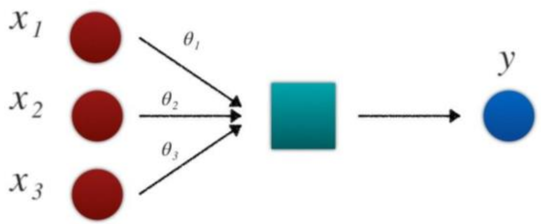
Weka Group	Description
bayes	Algorithms that use Bayes Theorem in some core way, like **Naive Bayes**.
function	Algorithms that estimate a function, like **Linear Regression**.
lazy	Algorithms that use lazy learning, like **k-Nearest Neighbors**.
meta	Algorithms that use or combine multiple algorithms, like **Ensembles**.
misc	Implementations that do not neatly fit into the other groups, like running a saved model.
rules	Algorithms that use rules, like **Zero Rule**.
trees	Algorithms that use decision trees, like **Random Forest**.

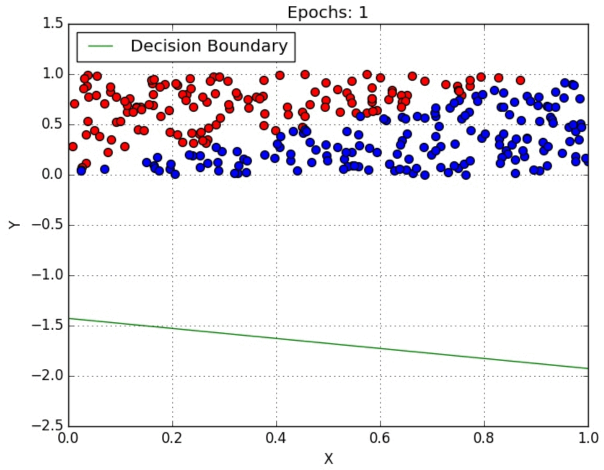
Popular Algorithms

Classification Algorithm	Regression Algorithm
• Logistic Regression • Naive Bayes • Decision Tree • k-Nearest Neighbors • Support Vector Machines	• Linear Regression • k-Nearest Neighbors • Decision Tree • Support Vector Machines • Multi-Layer Perceptron

Logistic Regression

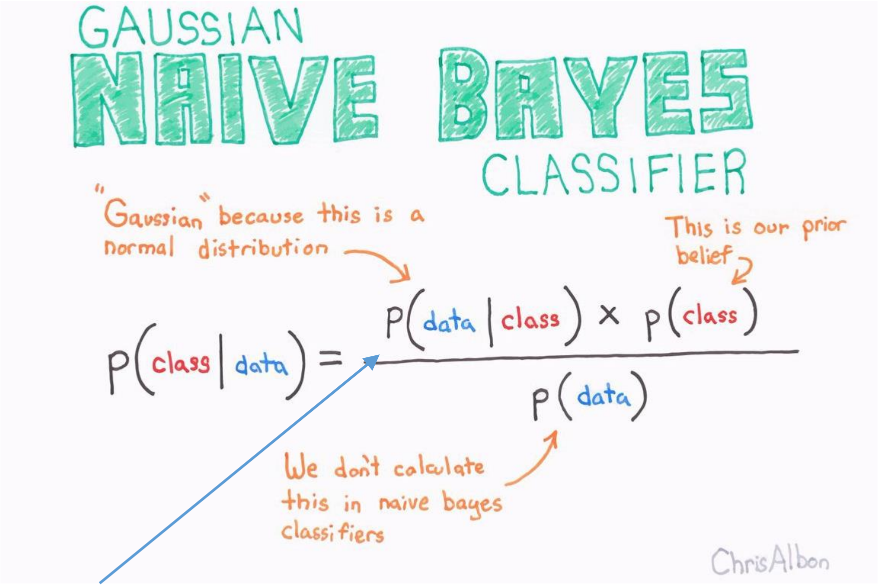
- The algorithm learns a coefficient for each input value
- Input are linearly combined into a regression function
  - And transformed using a logistic function.
- Fast and simple technique
  - Can be very ineffective on some problems.





Naïve Bayes Classifier

- Uses a simple implementation of **Bayes Theorem**
- Prior probability for each class is calculated from the training data
    - Assumed to be independent of each other.

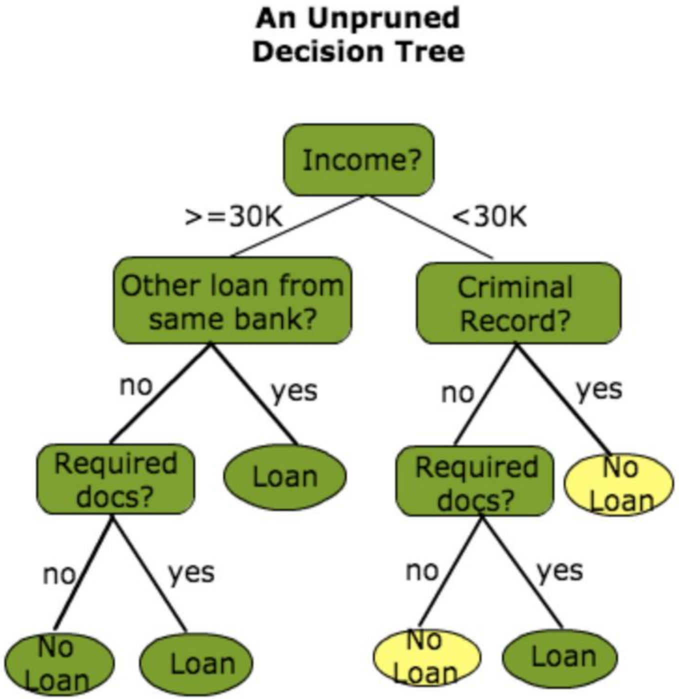


Prior probability

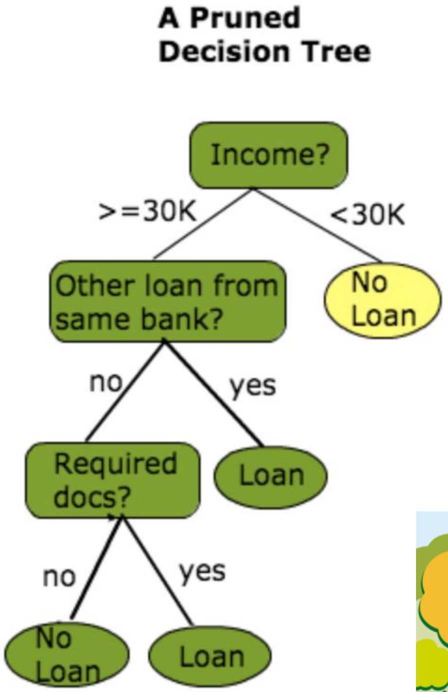
Decision Tree

- Also called **Classification And Regression Trees** (CART).
- Creates a tree to evaluate an instance of data.
  - start at the root of the tree and moving town to the leaves.
- The process of creating a decision tree works by greedily selecting the best split point in order to make predictions
  - Repeating the process until the tree is a fixed depth.
- After the tree is constructed, it is pruned in order to improve the model’s ability to generalize to new data.

Pruning



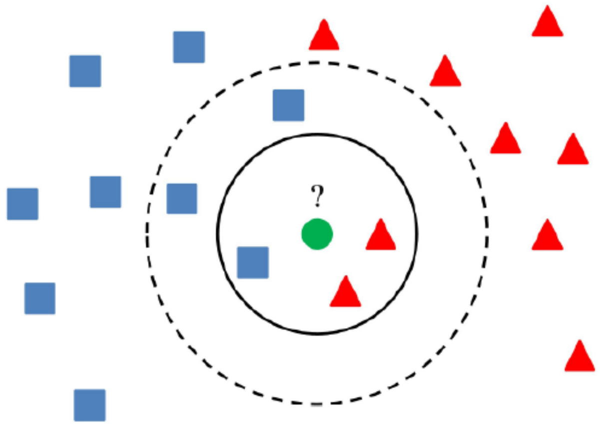
An Unpruned Decision Tree
Income?
>=30K
<30K
Other loan from same bank?
Criminal Record?
no
yes
no
yes
Required docs?
Loan
Required docs?
No Loan
no
yes
no
yes
No Loan
Loan
No Loan
Loan



A Pruned Decision Tree
Income?
>=30K
<30K
Other loan from same bank?
No Loan
no
yes
Required docs?
Loan
no
yes
No Loan
Loan

K Nearest Neighbor

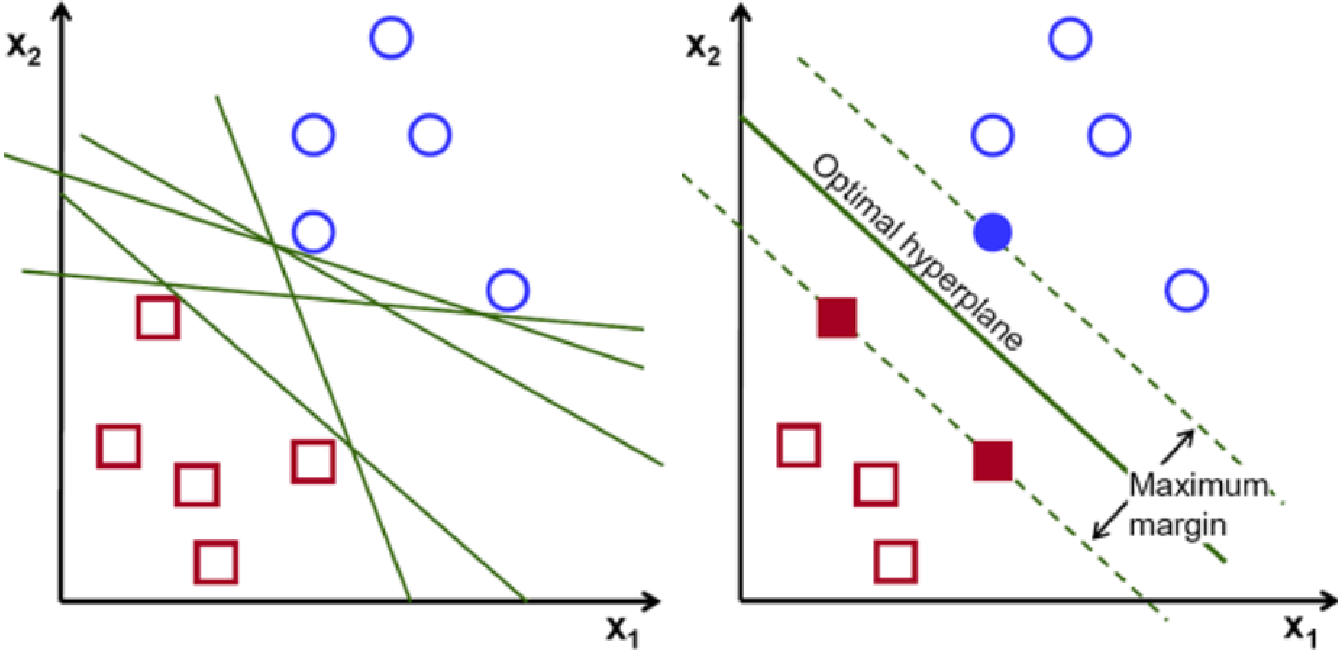
- Store the entire training dataset
  - Querying it to locate the k most similar training patterns when making a prediction.
- There is no model other than the raw training dataset
  - The only computation performed is the querying of the training dataset when a prediction is requested.



Support Vector Machines

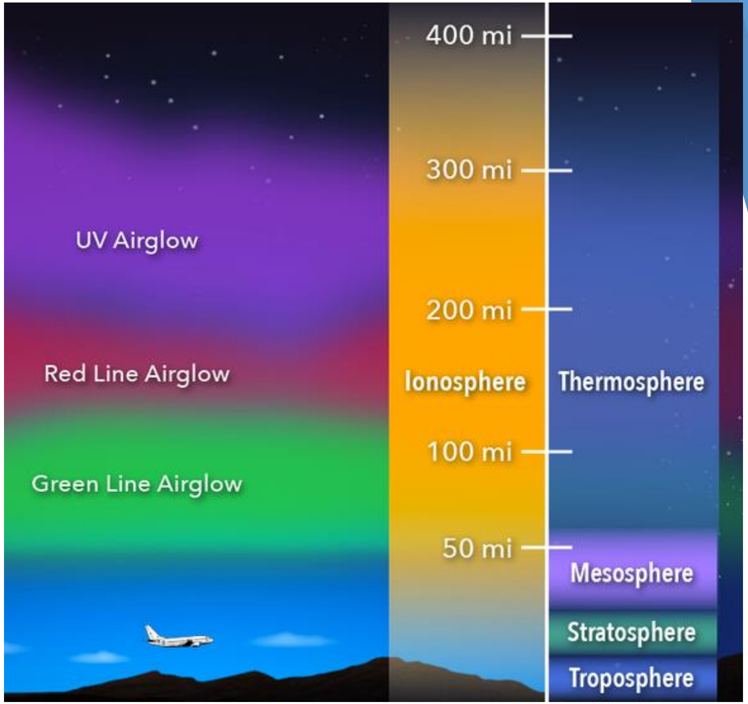
- Works by finding a line that best separates the data into the two groups.
  - Using an optimization process
- Normally a line cannot be drawn to neatly separate the classes
  - A margin is added around the line to relax the constraint.
  - Allowing some instances to be misclassified but allowing a better result overall.
- Data can be projected into a higher dimensional space
  - In order to draw complex lines and shapes.

Support Vector Machines



Activity 4

- Perform training on **ionosphere.arff** data
- Radar data collected by a system in Canada.
  - A phased array of 16 high-frequency.
- The targets were free electrons in the ionosphere.
  - "Good" radar returns are those showing evidence of some type of structure in the ionosphere.
  - "Bad" returns are those that do not.

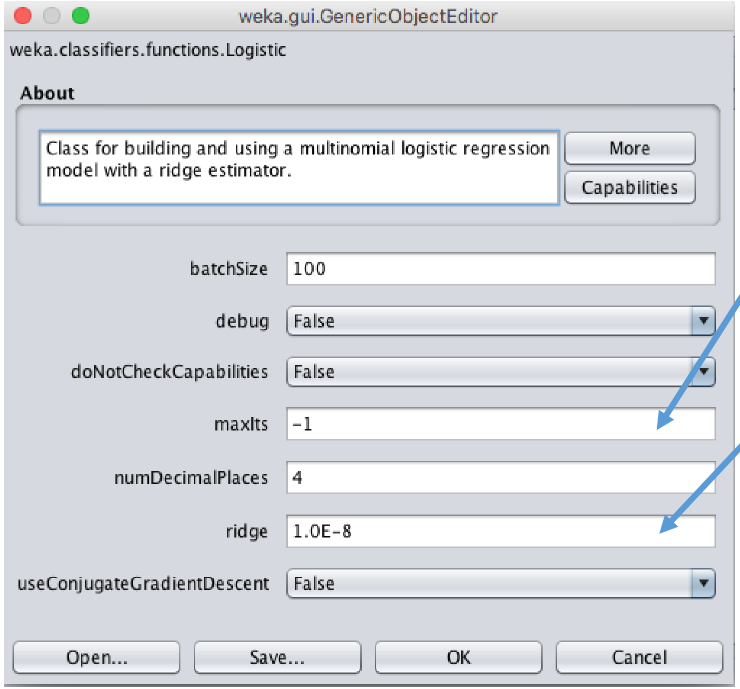


Activity 4

Algorithm	Weka Name
Logistic Regression	function.Logistic
Naive Bayes	bayes.NaiveBayes
Decision Tree	trees.REPTree
K-Nearest Neighbors	lazy.IBk (Instance Based - K)
Support Vector Machines	function.SMO (Sequential Minimal Optimization)

Top results are in the order of 98% accuracy.

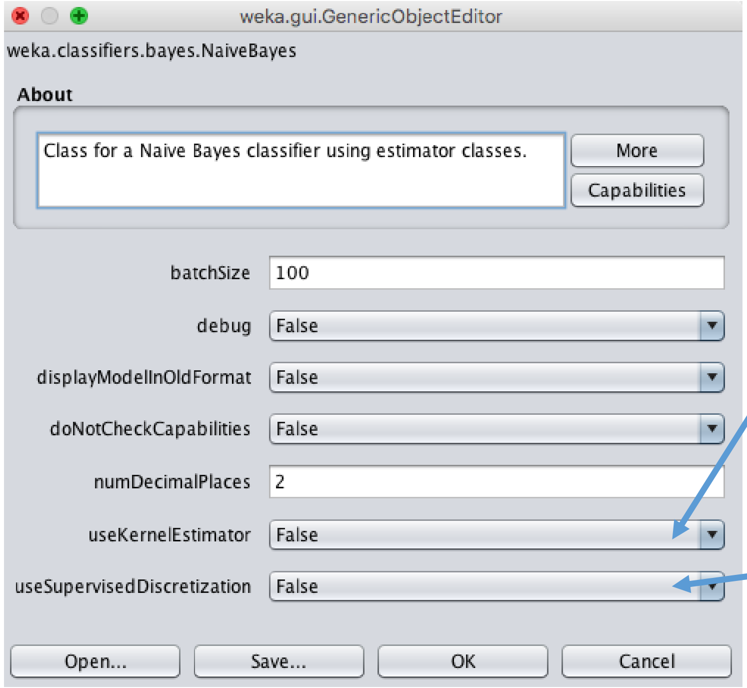
Options: Logistic Regression



- Runs for a fixed number of iterations
- By default (-1), runs until the algorithm has converged.

- The implementation uses a ridge estimator which is a type of regularization.
- This method minimizes the coefficients learned by the model.
- The ridge parameter defines how much pressure to put on the algorithm to reduce the size of the coefficients.
- Setting this to 0 will turn off this regularization.

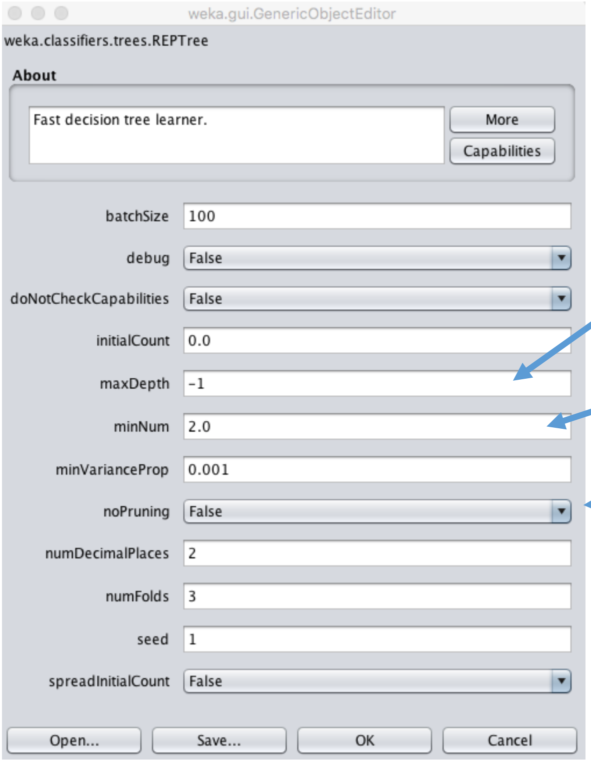
Options: Naive Bayes



- By default, the algorithm uses a Gaussian distribution for numerical attributes.
- Use this option to instead use Kernal Estimator.

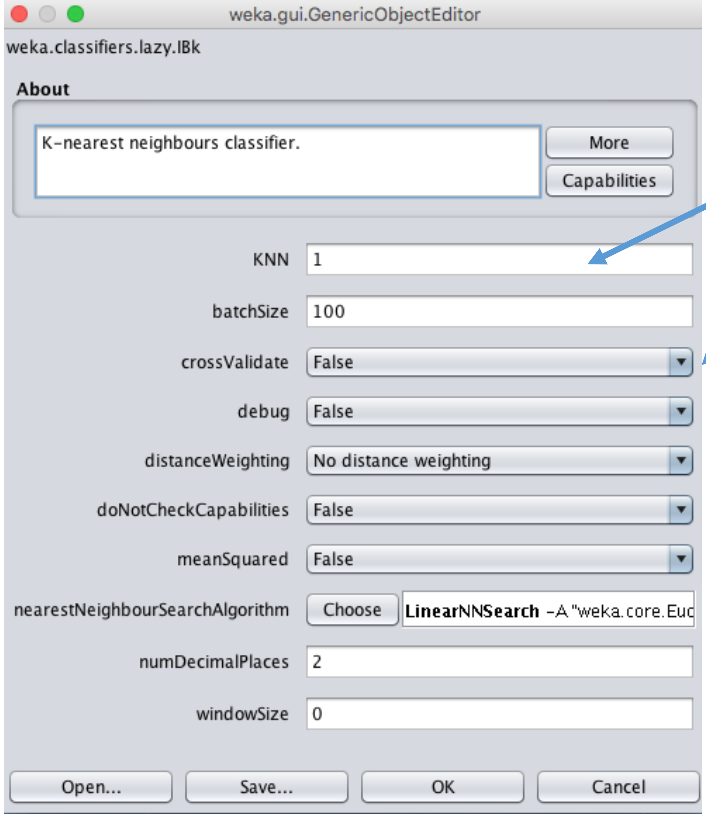
- Alternatively, you can automatically converts numerical attributes to nominal attributes.

Options: Decision Tree

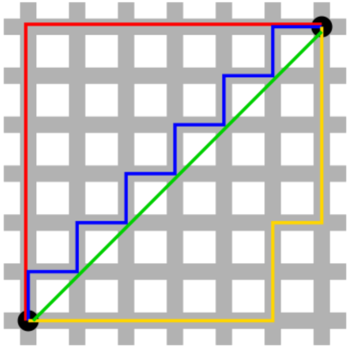


- The depth of the tree is defined automatically, but a depth can be specified in the maxDepth attribute.
- Minimum number of instances supported by the tree in a leaf node when constructing the tree from the training data.
- You can also choose to turn of pruning by setting the noPruning parameter to True, although this may result in worse performance.

Options: K-Nearest Neighbors

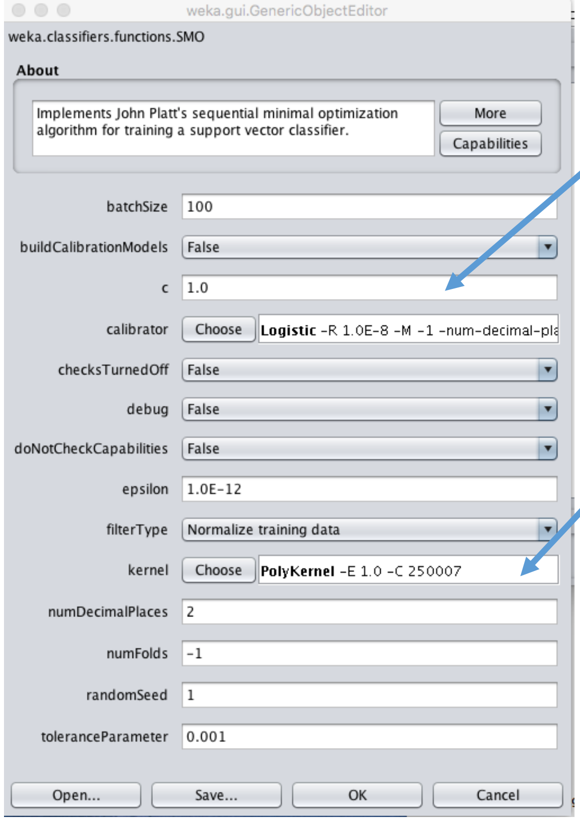


- The size of the neighborhood (Common values for k are 3, 7, 11 and 21.)
- Automatically discover a good value for k using cross validation inside the algorithm (True)
- Distance measure used.



- **Green:** Euclidian distance.
- **Red, blue, yellow:** Manhattan distances.

Options: SVM



- Complexity parameter controls how flexible the process for drawing the line to separate the classes can be. A value of 0 allows no violations of the margin, whereas the default is 1.

- Type of Kernel to use. The default is a Polynomial Kernel that will separate the classes using a curved or wiggly line.
- The simplest kernel is a Linear kernel that separates data with a straight line or hyperplane.
- A powerful kernel is the RBF Kernel or Radial Basis Function Kernel that is capable of learning closed polygons and complex shapes to separate the classes.