Introduction to Machine Learning

KASHTANOVA VICTORIYA

INRIA, EPIONE

Plan of the lectures

1. Introduction to Machine Learning + **Practical work (PW)** (4h)
2. Supervised Learning : Classification, Regression etc + **PW** (4h)
3. Supervised Learning : More methods, Model evaluation + **PW** (4h)
4. Unsupervised Learning : Clustering + **PW** (4h)
5. Project session (4h)

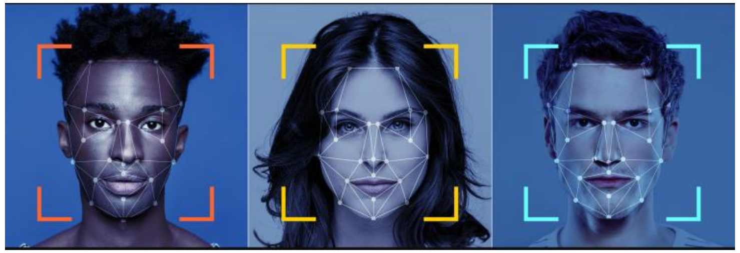
What/Where is Machine Learning ?

Where is Machine Learning ?

AlphaGo

Recommendation systems

Voice assistants



Face recognition

Self-driving cars

Photo filters

What is Machine Learning ?

Using the example of a simple code for playing chess :

1 Situation → Produses → 1 Outcome

```
if B diagonal from K AND no P around :
    if K can move :
        Move K
    else :
        Game Over
```

Code for ONLY ONE possible outcome

What is Machine Learning ?

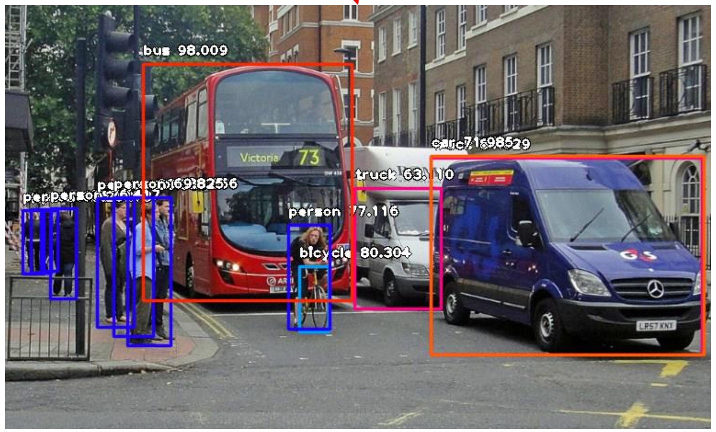
Using the example of a simple code for playing chess :

1 Situation → Produces → 1 Outcome

1 Program (100000 LOC) → Solves → 1 Problem (Play Chess)

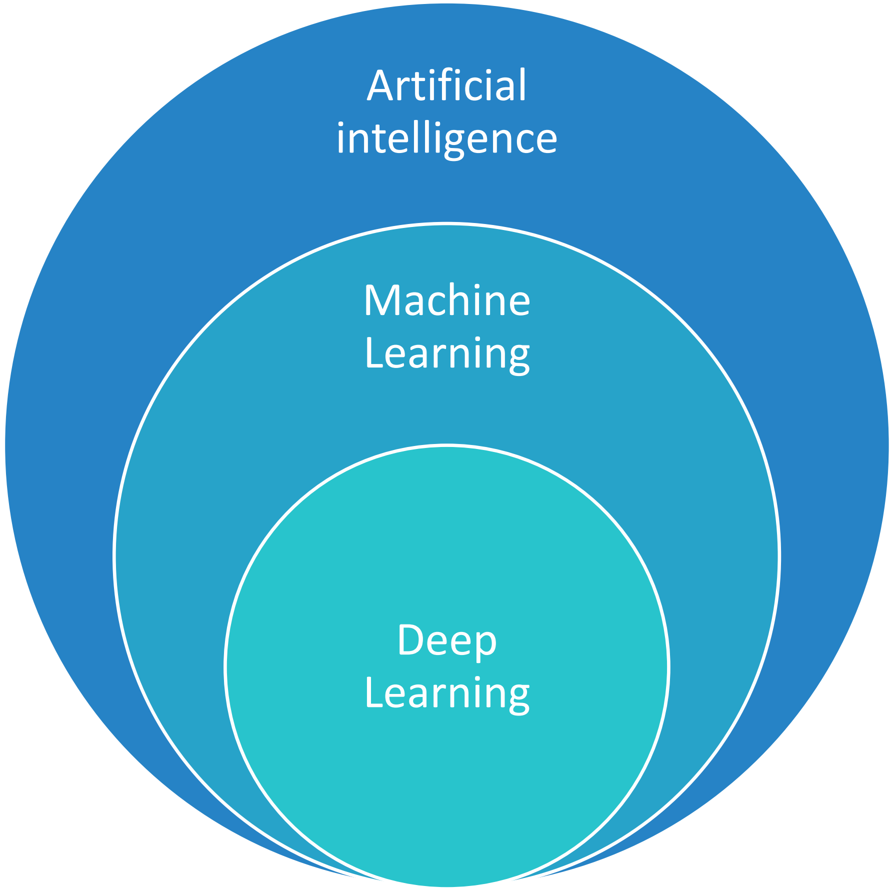
What is Machine Learning ?

1 Program → Solves → **N** Problem (just change input)



What is Machine Learning ?

Machine learning (**ML**) is the study of computer algorithms that improve automatically through experience (Wikipedia)



How are the things learned ?

- Memorization
  - Accumulation of individual facts
  - Limited by :
    - Time to observe facts
    - Memory to store facts

Declarative knowledge

How are the things learned ?

- Memorization
  - Accumulation of individual facts
  - Limited by :
    - Time to observe facts
    - Memory to store facts

- Generalization
  - Deduce new facts from old facts
  - Limited by accuracy of deduction process
    - Essentially a predictive activity
    - Assumes that the past predict the future

Declarative knowledge

Imperative knowledge

How are the things learned ?

- Memorization
  - Accumulation of individual facts
  - Limited by :
    - Time to observe facts
    - Memory to store facts

Declarative knowledge

- Generalization
  - Deduce new facts from old facts
  - Limited by accuracy of deduction process
    - Essentially a predictive activity
    - Assumes that the past predict the future

Imperative knowledge

- Interested in extending to programs that can infer useful information from implicit patterns in data

Basic paradigm of ML

- Observe set of examples : training data
- Infer something about process that generated that data
- Use inference to make predictions about previously unseen data : test data
- Variations on paradigm
  - Supervised : given a set feature/label pairs, find a rule that predicts the label associated with a previously unseen input
  - Unsupervised : given a set of feature vectors (without labels) group them into “natural clusters (or create labels for groups)

Basic paradigm of ML

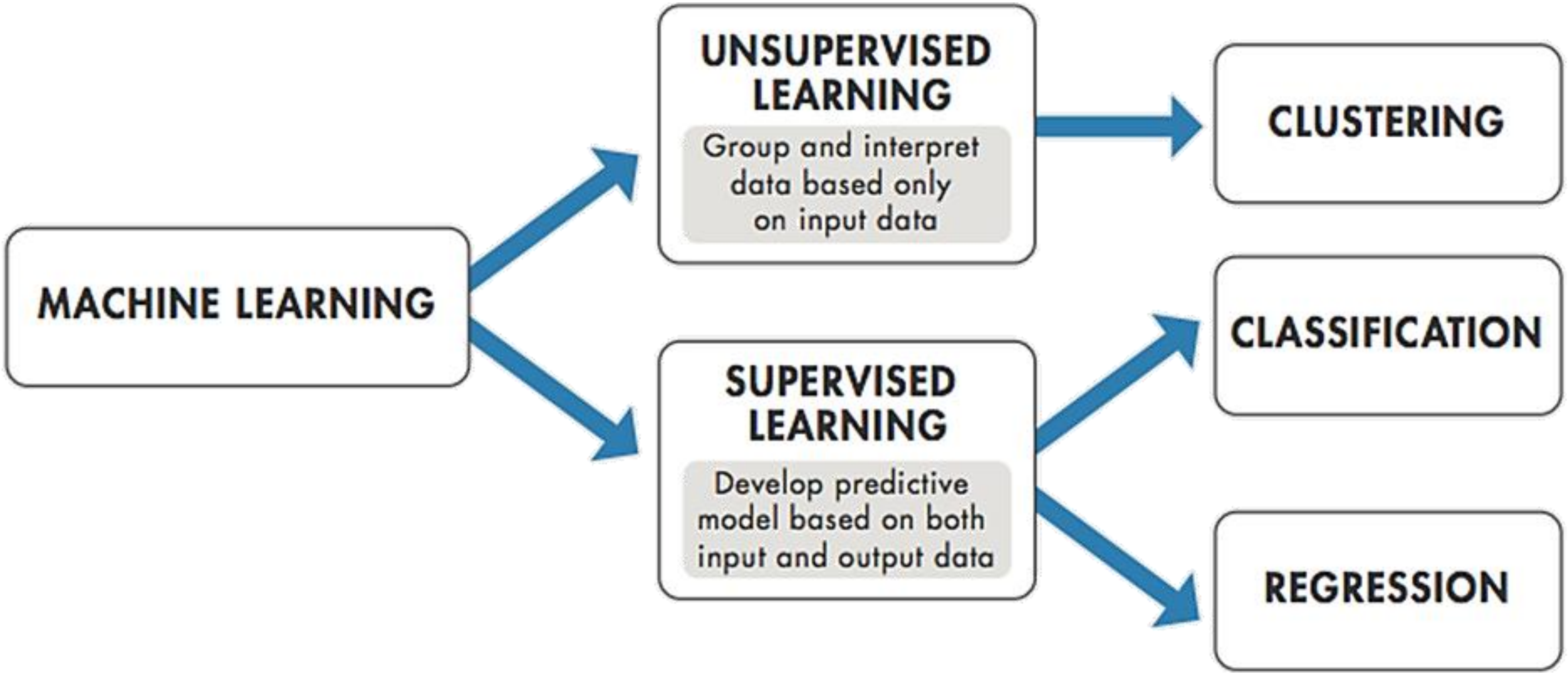
- Observe set of examples : training data
- Infer something about process that generated that data
- Use inference to make predictions about previously unseen data : test data
- Variations on paradigm
  - Supervised : given a set feature/label pairs, find a rule that predicts the label associated with a previously unseen input
  - Unsupervised : given a set of feature vectors (without labels) group them into “natural clusters (or create labels for groups)

Benign and Malignant neoplasms with information of neoplasms size and cell density

Find canonical model of neoplasms type, by statistics

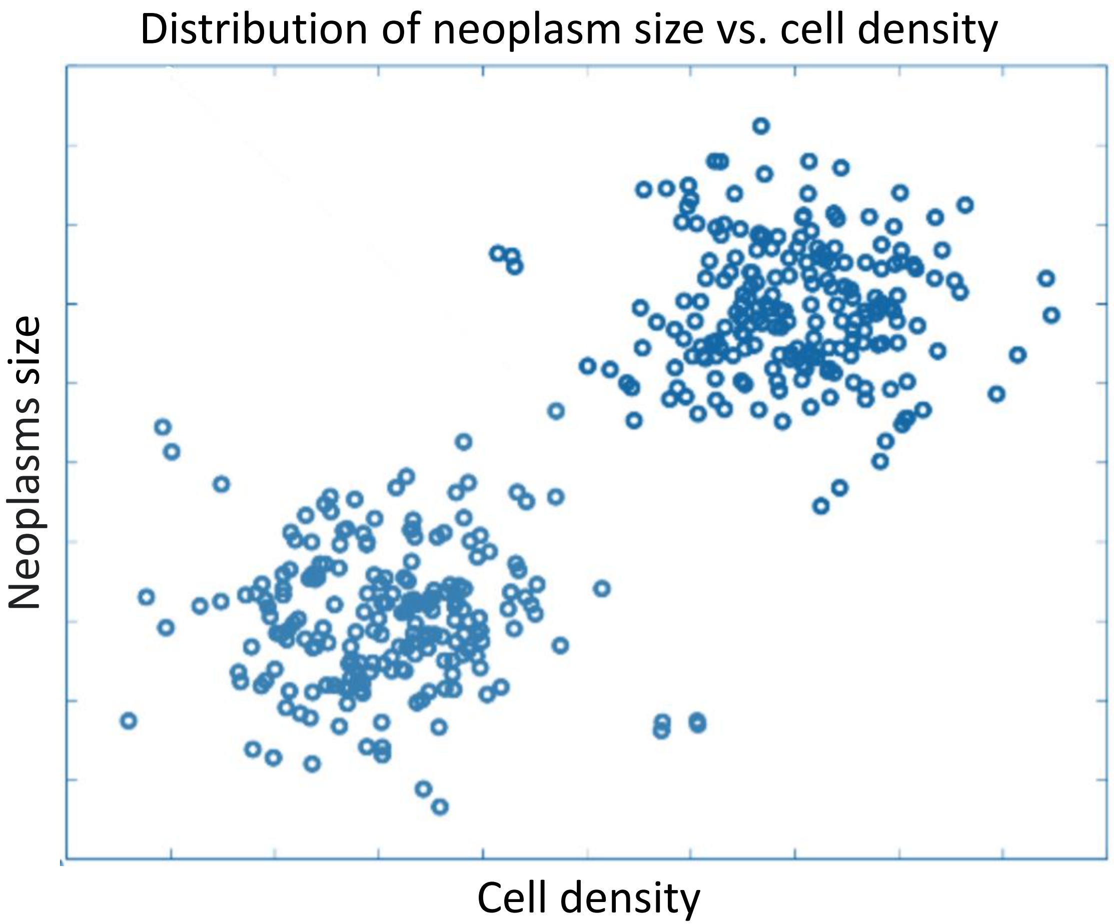
Predict type of new neoplasms

Machine learning methods



Some examples of Classifying and Clustering

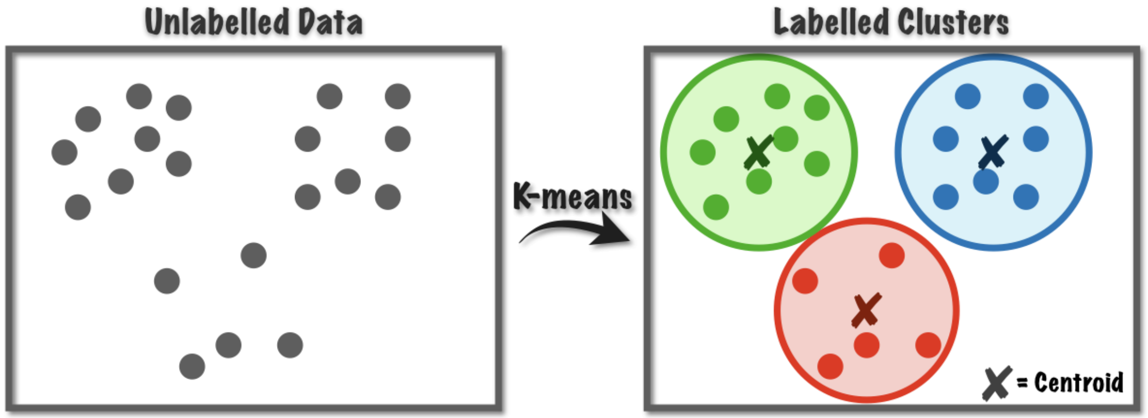
Unlabeled data : Breast cancer



<u>Suppose :</u> There are two types of neoplasms (Benign and Malignant)

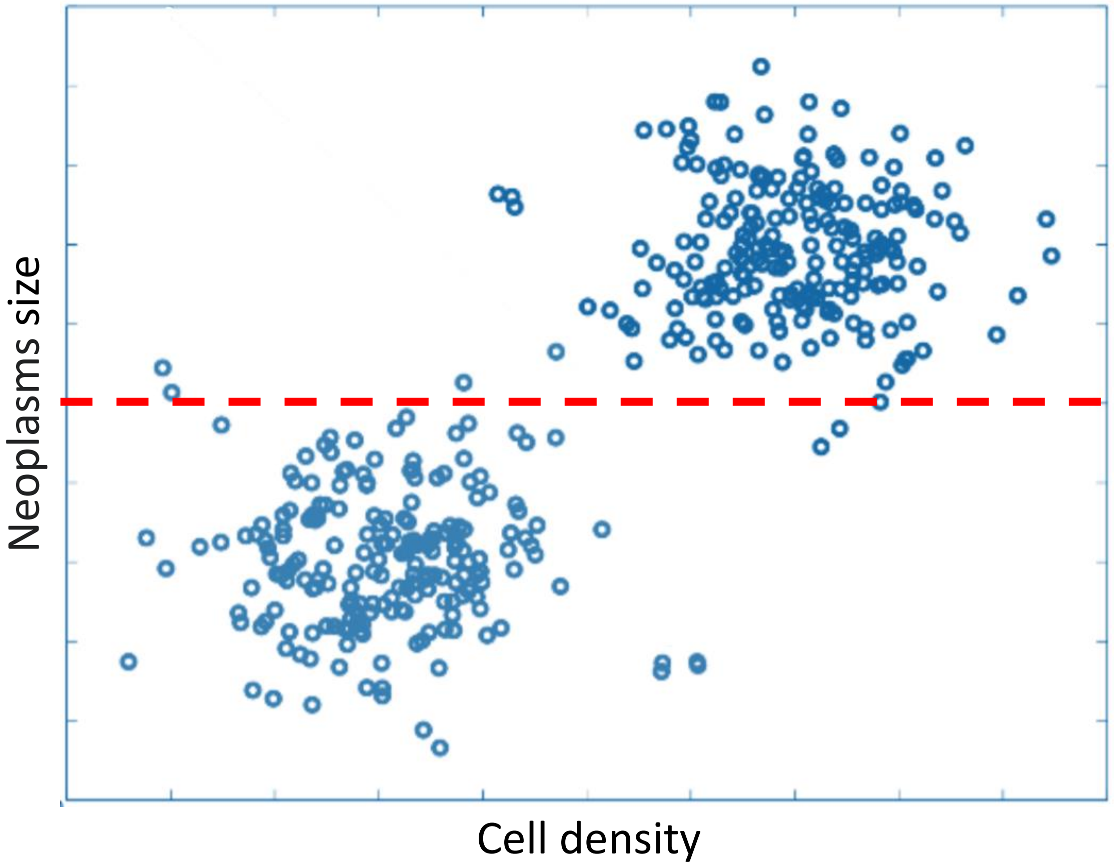
Task : Clustering examples into groups

- Want to decide on “similarity” of example, with goal of separating into distinct, “natural” groups
  - Similarity is a distance measure
- Suppose we know that there are K different groups in our training data, but don’t know labels (here K=2)
  - Construct the groups by minimizing of distance between samples in same cluster (objective function)



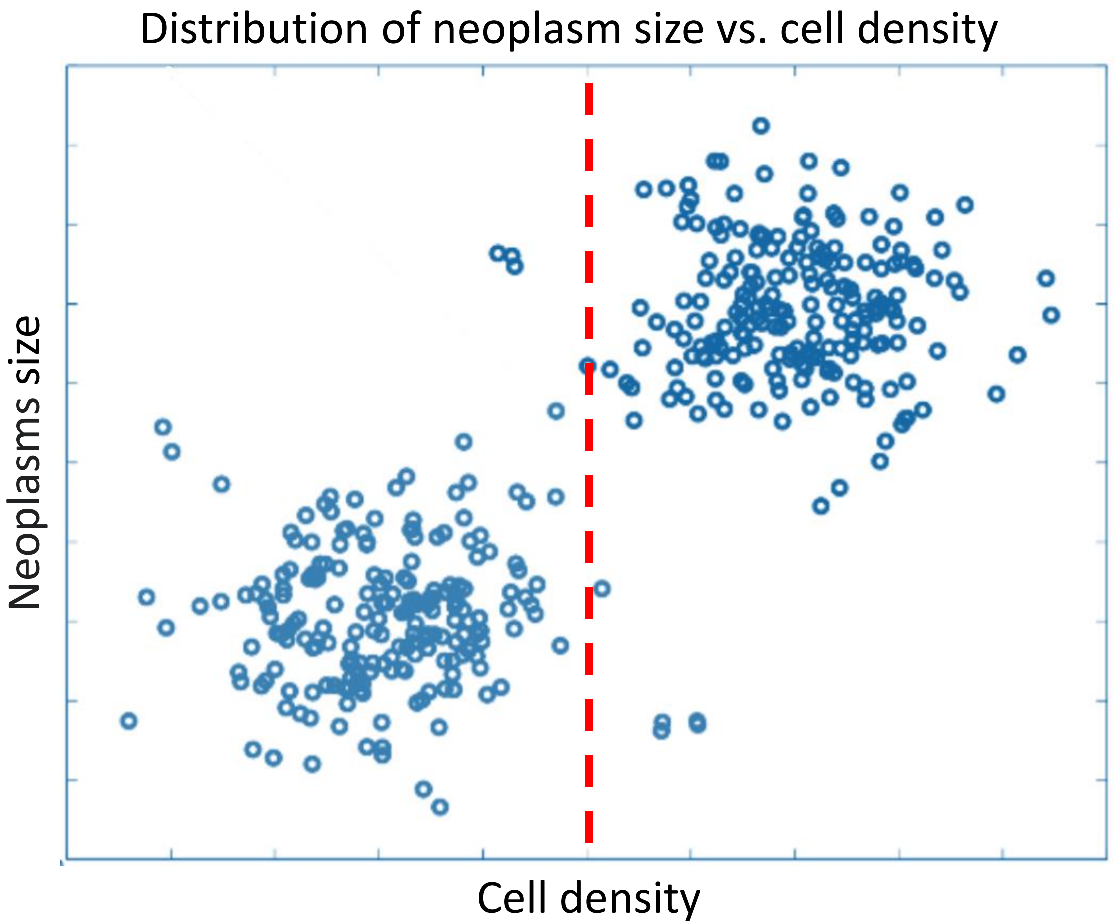
Similarity based on Neoplasm size

Distribution of neoplasm size vs. cell density



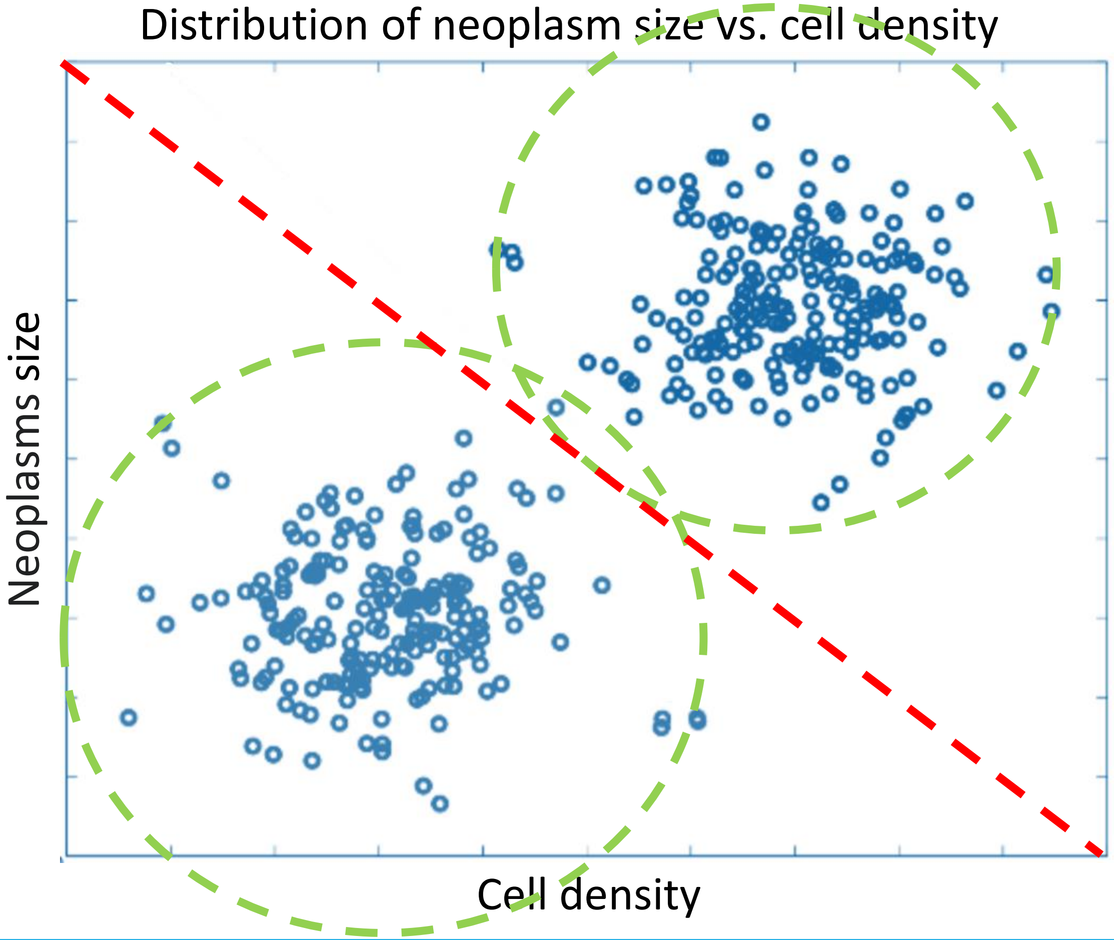
Suppose : There are two types of neoplasms (Benign and Malignant)

Similarity based on Cell density



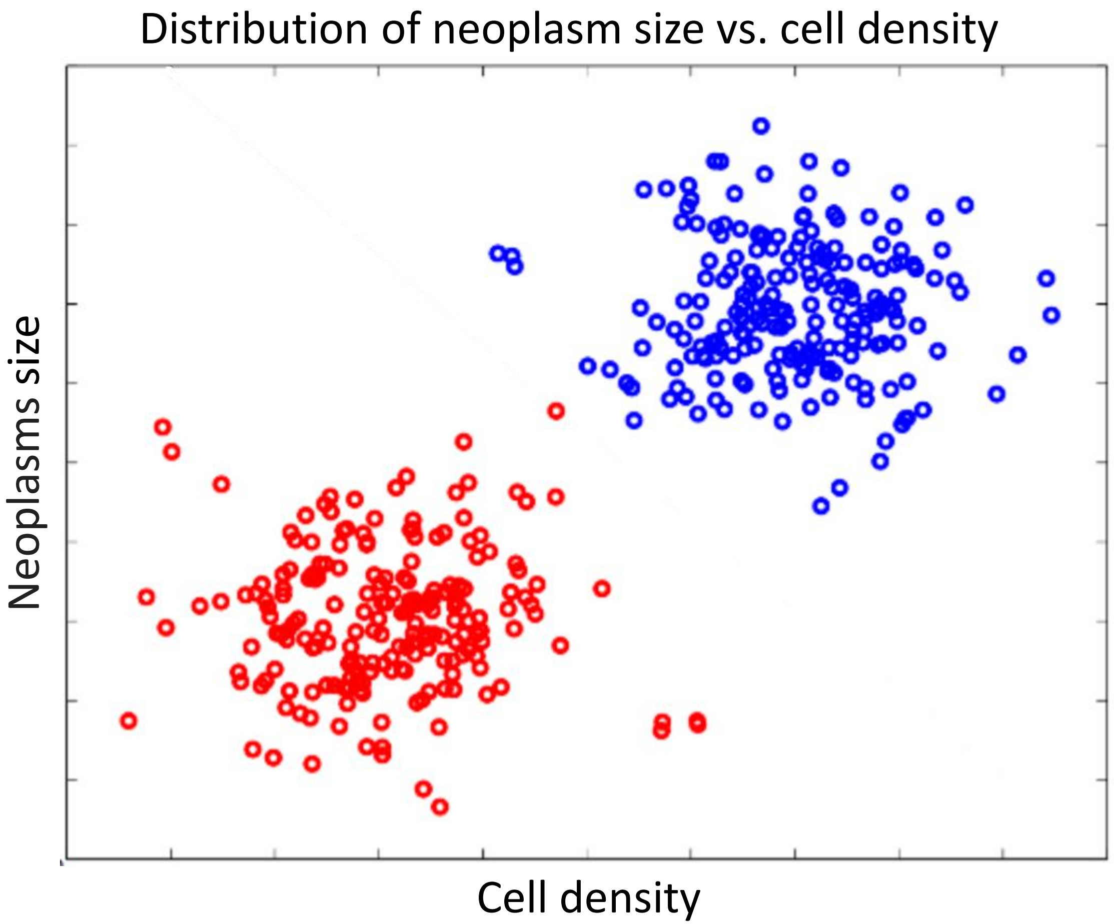
Suppose : There are two types of neoplasms (Benign and Malignant)

Cluster into two groups using both attributes



Suppose : There are two types of neoplasms (Benign and Malignant)

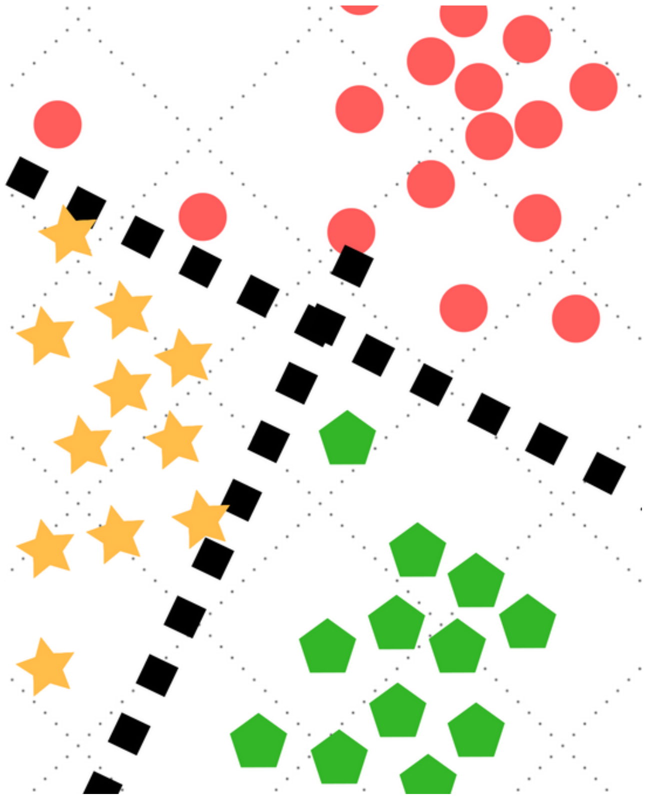
Suppose data was labeled



<u>Know :</u> There are two types of neoplasms (Benign and Malignant)

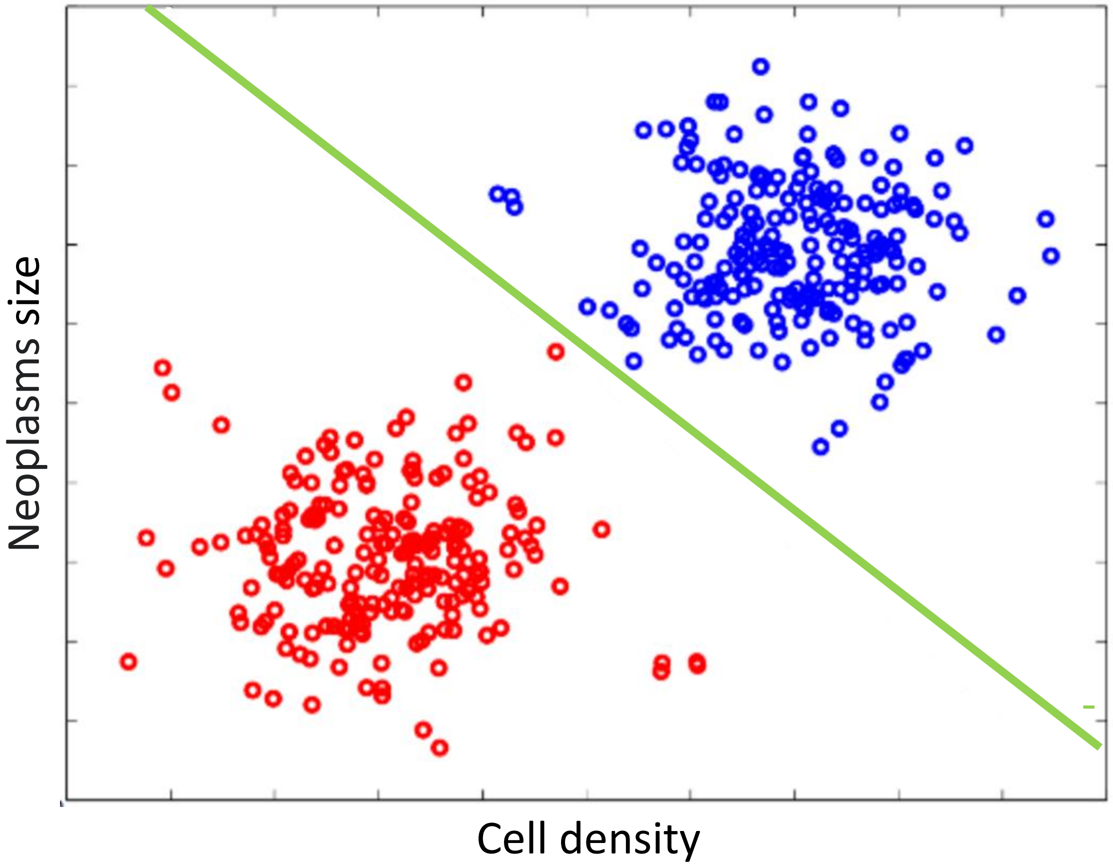
Task : Finding classifier surfaces

- Given labeled groups in feature space, want to find subsurface in that space that separates the groups
  - Subject to constraints on complexity of subsurface
- In this example, have 2D space, so find line (or connected set of line segments) that best separates the two groups
  - When examples well separated this is straightforward
  - When examples in labelled groups overlap, may have to trade off false-positives and false-negatives



Suppose data was labeled

Distribution of neoplasm size vs. cell density

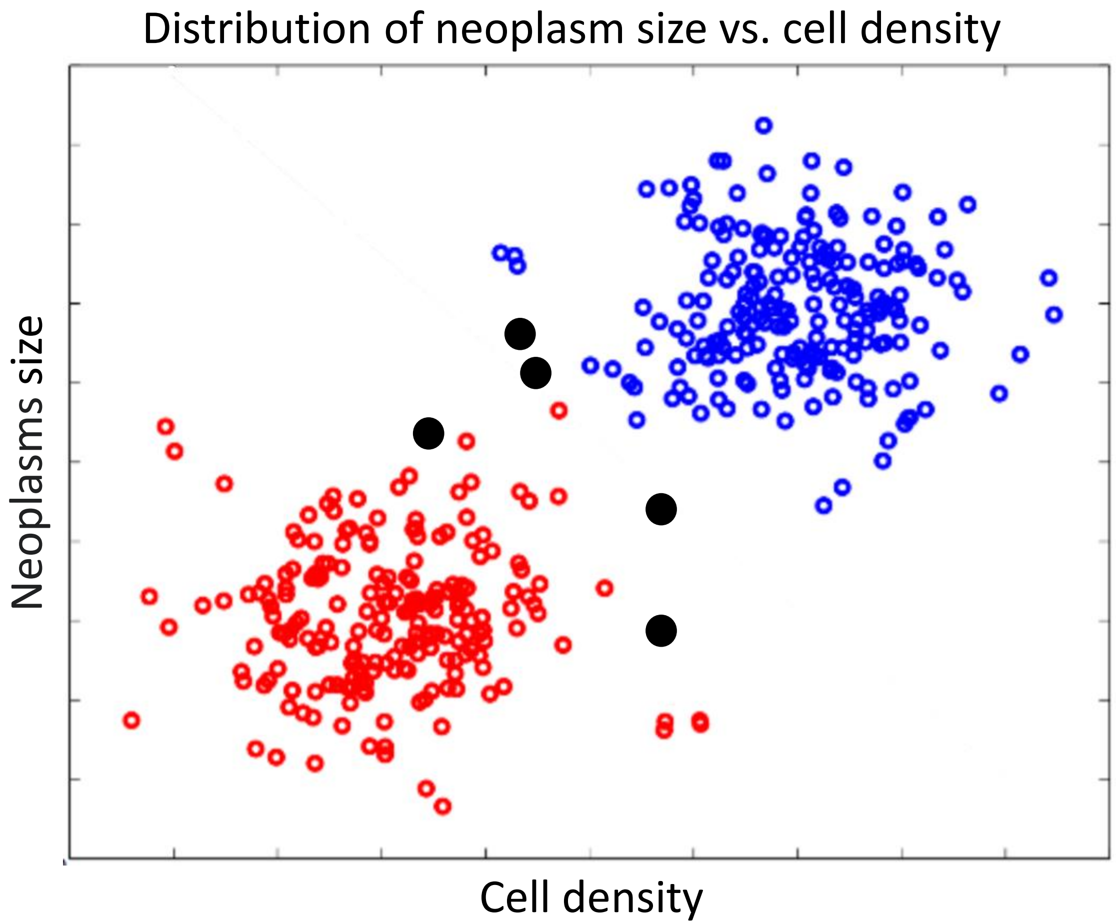


Know : There are two types of neoplasms (Benign and Malignant)

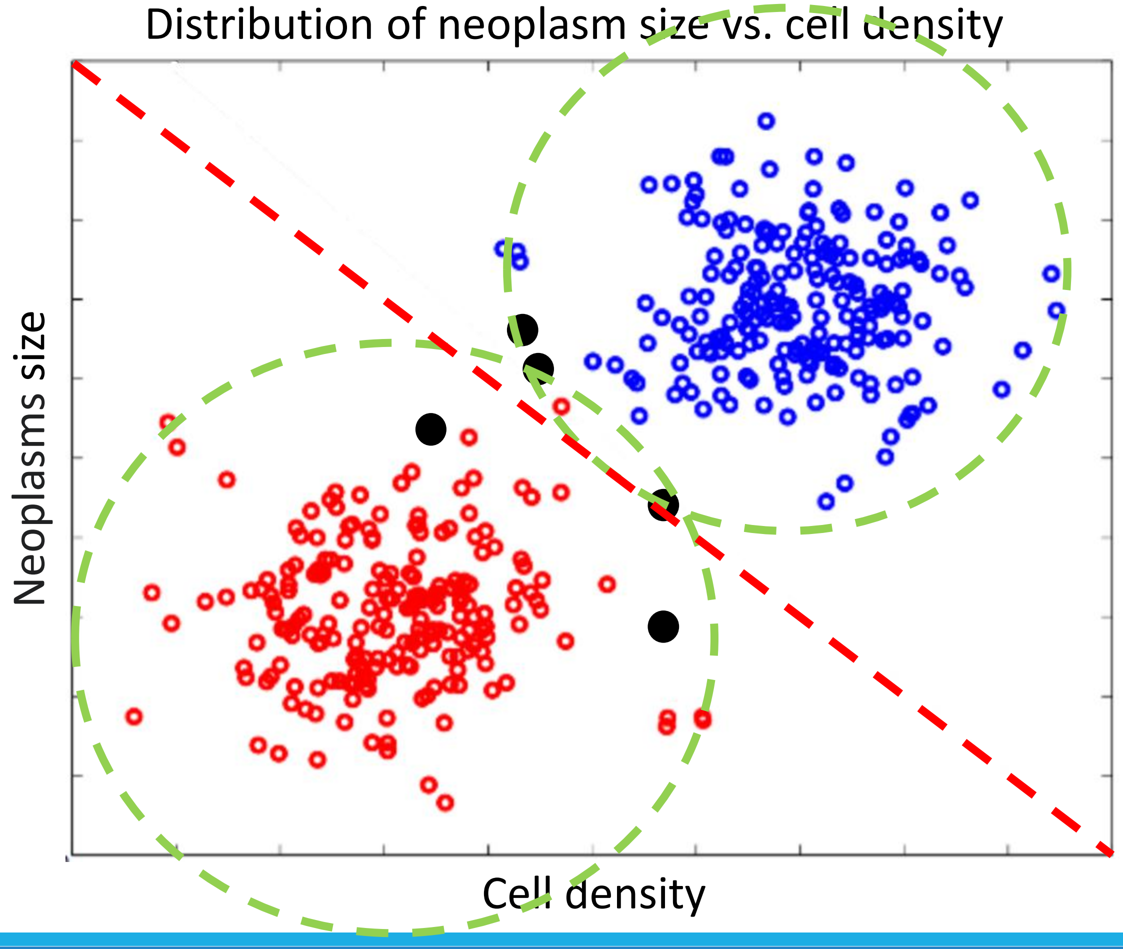
Obvious separator of two groups

Adding some new data

- Suppose we have learned to separate the Benign and Malignant neoplasms
- Now we are given some new data points and want to use model to decide : **what type do they belong to?**

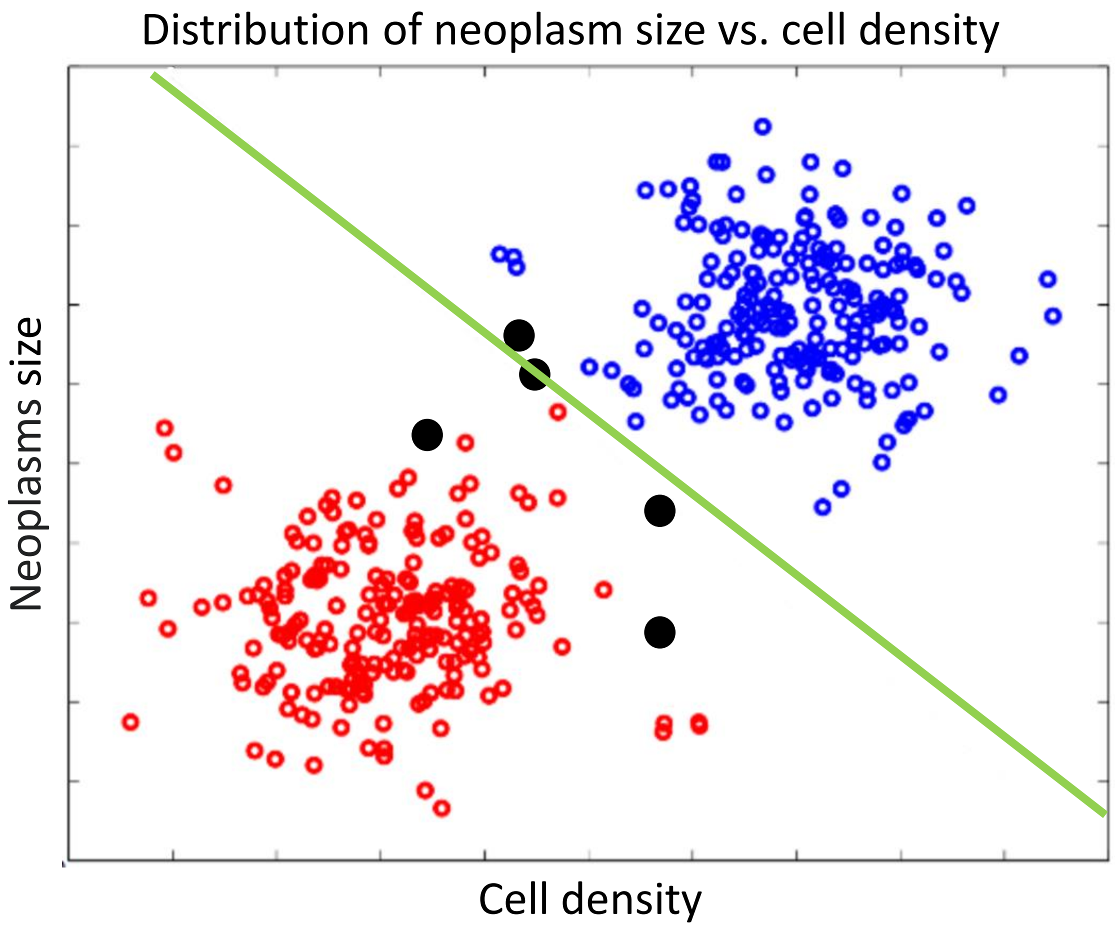


Clustering using unlabeled data



Know : There are two types of neoplasms (Benign and Malignant)

Classified using labeled data



Know : There are two types of neoplasms (Benign and Malignant)

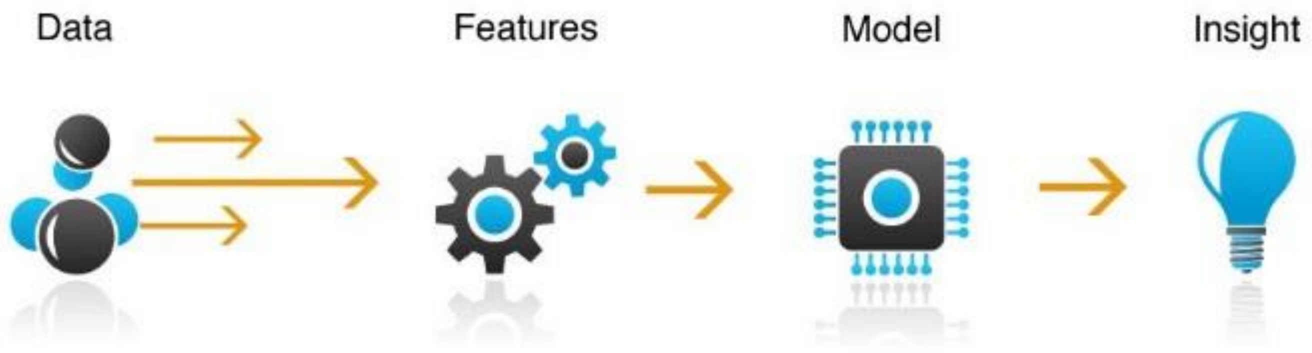
All ML methods require :

- Choosing training data and evaluation method
- Representation of the features
- Distance metric for feature vectors
- Objective function and constraints
- Optimization method for learning the model

Feature Representation

Feature engineering

- Represent examples by feature vectors that will facilitate generalization
- Choose wisely the useful features to avoid an overfitting
- Maximize ratio of useful input to irrelevant input



Reptile classification example

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
	Features					Label
Cobra	True	True	True	True	0	Yes

Initial model :

- Not enough information to generalize

Reptile classification example

	Features					Label
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes

Initial model :

- Egg laying
- Has scales
- Is poisonous
- Cold blooded
- No legs

Reptile classification example

	Features					Label
Name	**Egg-laying**	**Scales**	**Poisonous**	**Cold-blooded**	**# legs**	**Reptile**
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes

Current model :

- Has scales
- Cold blooded
- No legs

Boa doesn't fit model, is labeled as reptile
=> Need to refine model

Reptile classification example

Features   Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No

Current model :

- Has scales
- Cold blooded
- No legs

Reptile classification example

		Features				**Label**
Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes

Current model :

- Has scales
- Cold blooded
- Has 0 or 4 legs

Alligator doesn't fit model, but is labeled as reptile => Need to refine model

Reptile classification example

Current model :

- Has scales
- Cold blooded
- Has 0 or 4 legs

Features | Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No

Reptile classification example

Features **Label**

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Current model :

- Has scales
- Cold blooded
- Has 0 or 4 legs

No easy way to add to rule that will correctly classify (since identical feature values)

Reptile classification example

Current model :
- Has scales
- Cold blooded

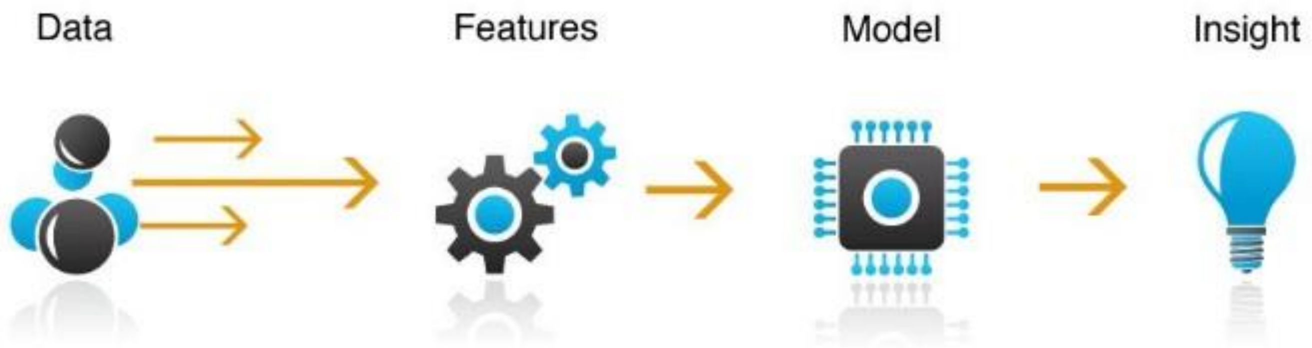
Not perfect, but no false negatives (anything classified as not reptile is correctly labelled); some false positives (may incorrectly label some animals as reptile)

Features | Label

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile
Cobra	True	True	True	True	0	Yes
Rattlesnake	True	True	True	True	0	Yes
Boa constrictor	False	True	False	True	0	Yes
Chicken	True	True	False	False	2	No
Alligator	True	True	False	True	4	Yes
Dart frog	True	False	True	False	4	No
Salmon	True	True	False	True	0	No
Python	True	True	False	True	0	Yes

Feature engineering

- Deciding which features to include and which are merely adding noise to classifier
- Defining how to **measure distances between training examples** (and ultimately between classifiers and new instances)
- Deciding how to weight relative importance of different dimensions of feature vector, which impacts definition of distance



Measuring distance between animals

Name	Egg-laying	Scales	Poisonous	Cold-blooded	# legs	Reptile

Binary features Integer feature

- One way to learn to separate reptiles from non-reptiles is to measure the distance between pairs of examples, and use that :
  - To cluster nearby examples into a common class (unlabeled data), or
  - To find a classifier surface in space of examples that optimally separates different (labelled) collections of examples from other collections

Rattlesnake = (1,1,1,1,0)
Boa constrictor = (0,1,0,1,0)
Dart frog = (1,0,1,0,4)

Can convert examples into feature vectors

Minkowski metric

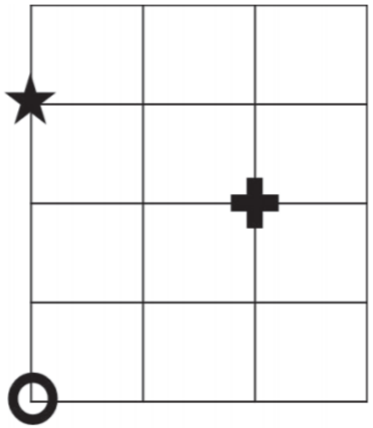
$$dist(X1, X2, p) = \left(\sum_{k=1}^{len} abs(X1_k - X2_k)^p\right)^{1/p}$$

Need to measure distances between feature vectors

p = 1: Manhattan Distance
p = 2: Euclidean Distance

Is circle closer to star or cross ?

- Euclidean distance :
  - Cross - 2.8
  - Star – 3
- Manhattan distance :
  - Cross – 4
  - Star - 3



Typically use Euclidean metric; Manhattan may be appropriate if different dimensions are not comparable

Reptile classification example

```
rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,4]
```

Reptile classification example

```
rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,4]
```

	rattlesnake	boa constrictor	dart frog
rattlesnake	--	1.414	4.243
boa constrictor	1.414	--	4.472
dart frog	4.243	4.472	--

Using Euclidian distance, rattlesnake and boa constrictor are much closer to each other, then they are to the dart frog

Add an alligator

```
▪alligator = Animal('alligator', [1,1,0,1,4])
▪animals.append(alligator)
▪compareAnimals(animals, 3)
```

Add an alligator

```
▪alligator = Animal('alligator', [1,1,0,1,4])
▪animals.append(alligator)
▪compareAnimals(animals, 3)
```

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	--	1.414	4.243	4.123
boa constrictor	1.414	--	4.472	4.123
dart frog	4.243	4.472	--	1.732
alligator	4.123	4.123	1.732	--

Alligator is closer to dart frog than to snakes – why ?

- Alligator differs from frog in 3 features, from boa in only 2 features
- But scale on “legs” is from 0 to 4, on other features is 0 to 1
- **“Legs” dimension is disproportionately large**

Using binary features

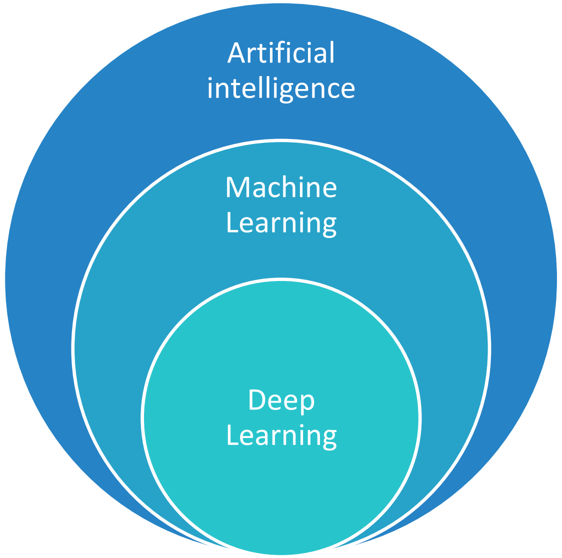
```
rattlesnake = [1,1,1,1,0]
boa constrictor = [0,1,0,1,0]
dartFrog = [1,0,1,0,1]
Alligator = [1,1,0,1,1]
```

	rattlesnake	boa constrictor	dart frog	alligator
rattlesnake	--	1.414	1.732	1.414
boa constrictor	1.414	--	2.236	1.414
dart frog	1.732	2.236	--	1.732
alligator	1.414	1.414	1.732	--

Now alligator is closer to snakes than it is to dart frog => **Feature Engineering Matters**

Summary

- Machine Learning methods provide a way of building models of processes from data sets
  - Supervised learning uses labelled data, and creates classifiers that optimally separate data into known classes
  - Unsupervised learning tries to infer latent variables by clustering training examples into nearby groups
- Choice of features influences results
- Choice of distance measurement between examples influence results
- We will see some examples of clustering methods
- We will see some examples of classifiers
- We will see some advanced techniques such as Deep Learning



Sources

- MIT course “Introduction to Computational Thinking and Data Science” (Prof. Eric Grimson, Prof. John Guttag)
- Open Machine Learning Course (by Yury Kashnitsky, mlcourse.ai)
- YouTube lections “Algorithms and Concepts” (by CodeEmporium)