



COMP3204/COMP6223: Computer Vision

# Machine learning for pattern recognition

Jonathon Hare  
[jsh2@ecs.soton.ac.uk](mailto:jsh2@ecs.soton.ac.uk)

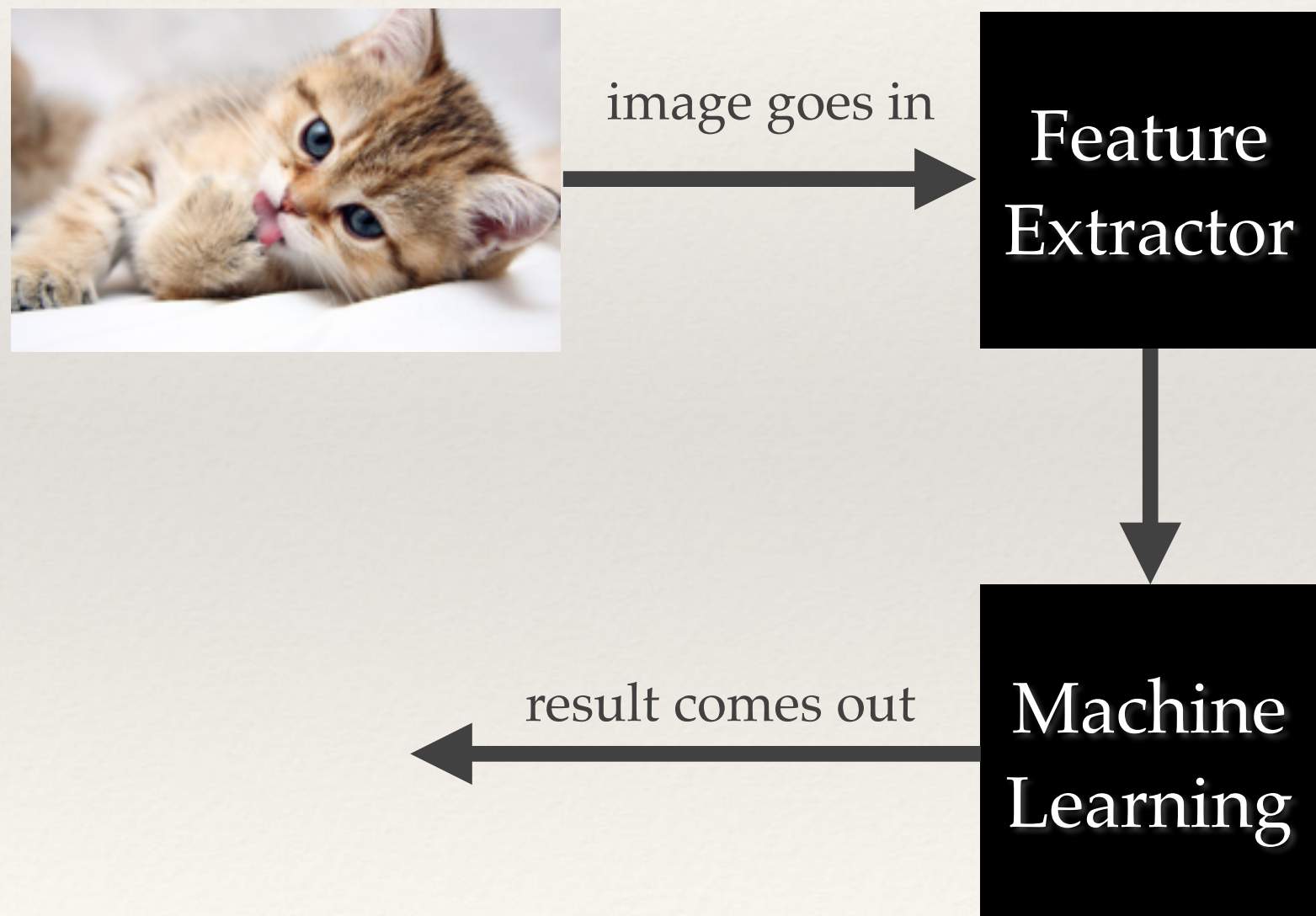
RETURN TO D-STATION.



- ❖ Recognising patterns is a large part of computer vision
  - ❖ i.e. recognising text, people, objects, ...
- ❖ Obviously there's a lot of overlap with intelligent algorithms, machine learning and AI.
- ❖ This lecture will cover (recap?) some of the fundamentals of machine learning and introduce how you connect arrays of pixels to machine learning algorithms.

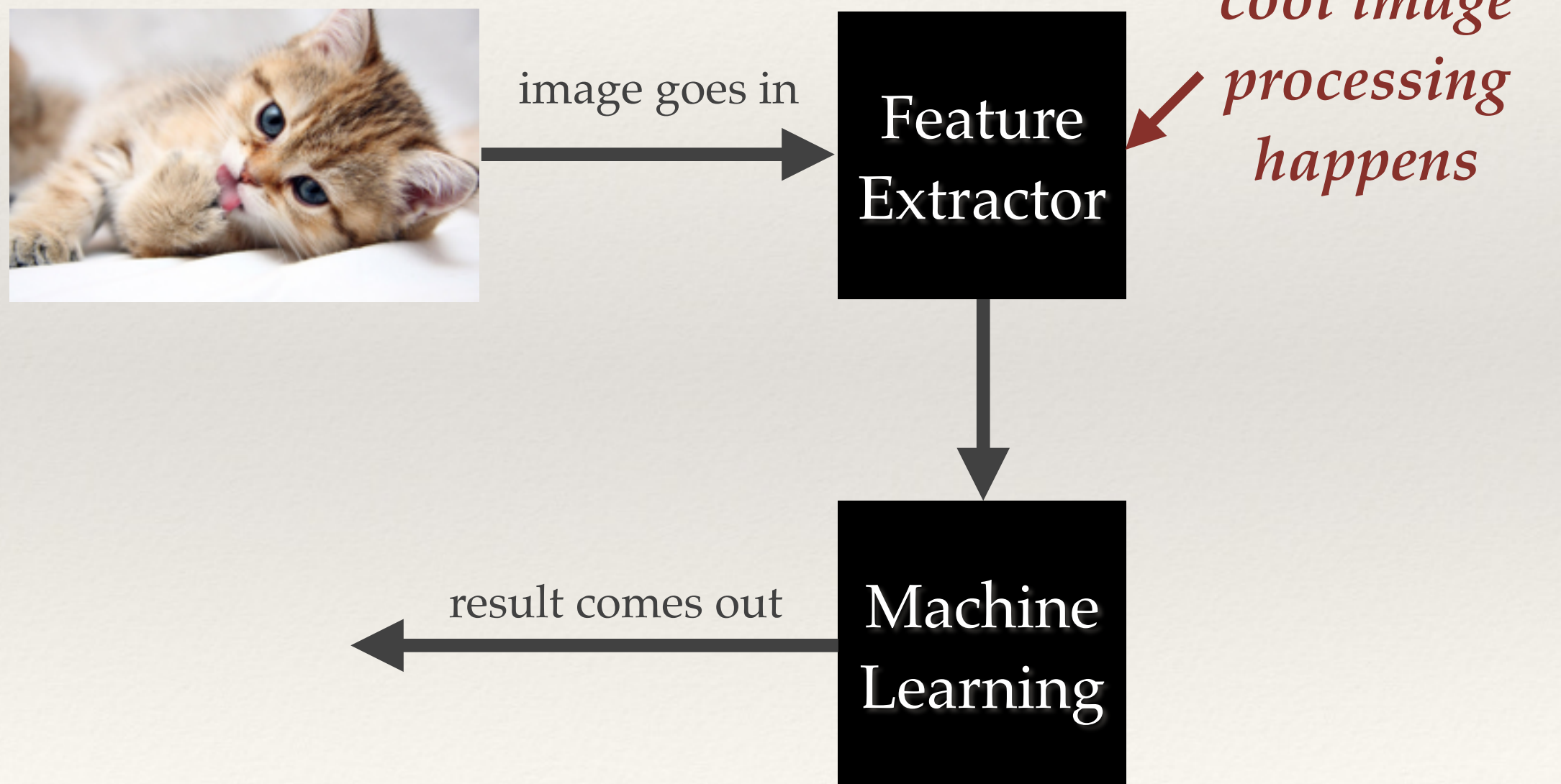
# Feature spaces

Many computer vision applications involving machine learning take the following form:

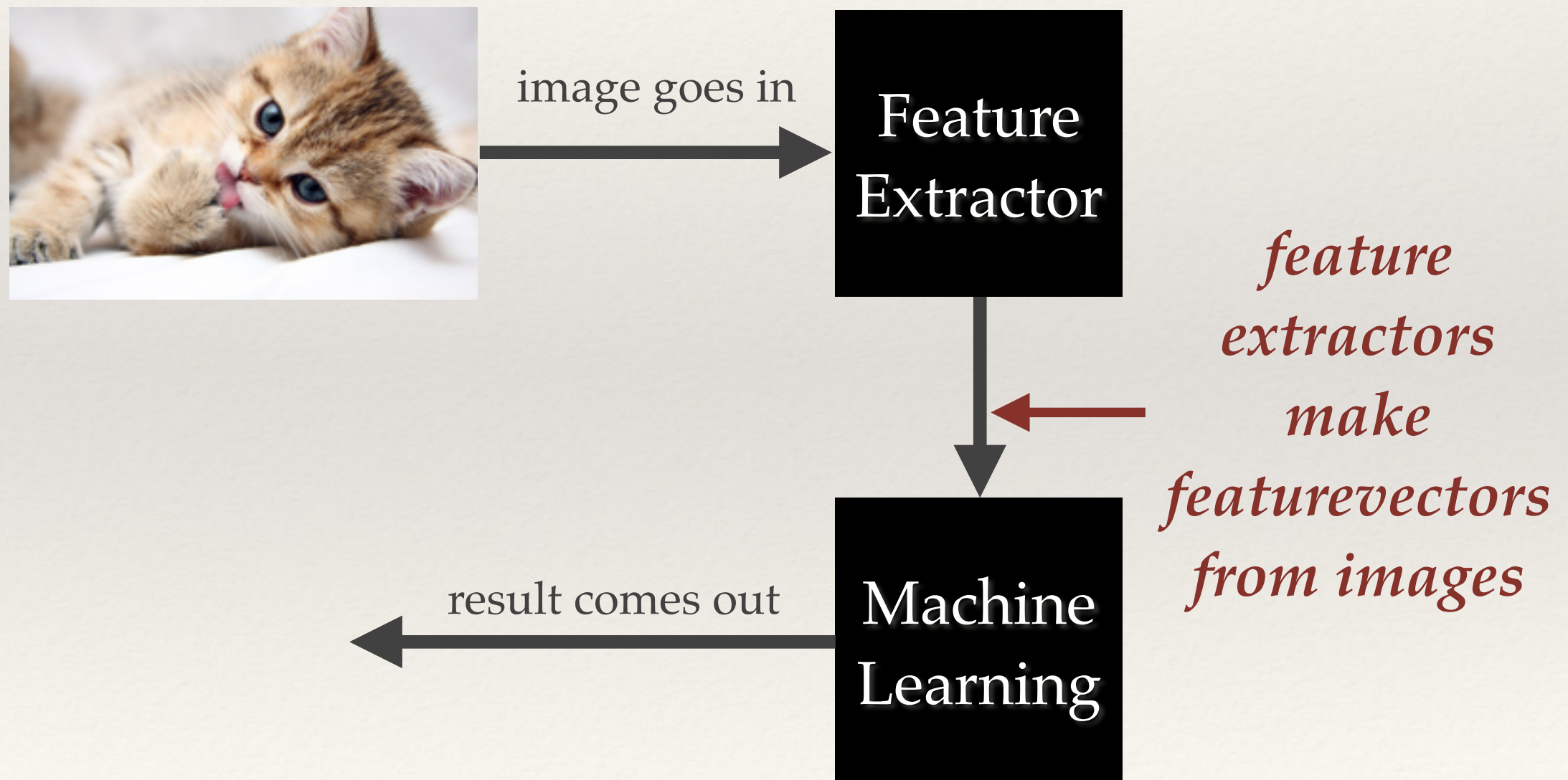




Many computer vision applications involving machine learning take the following form:

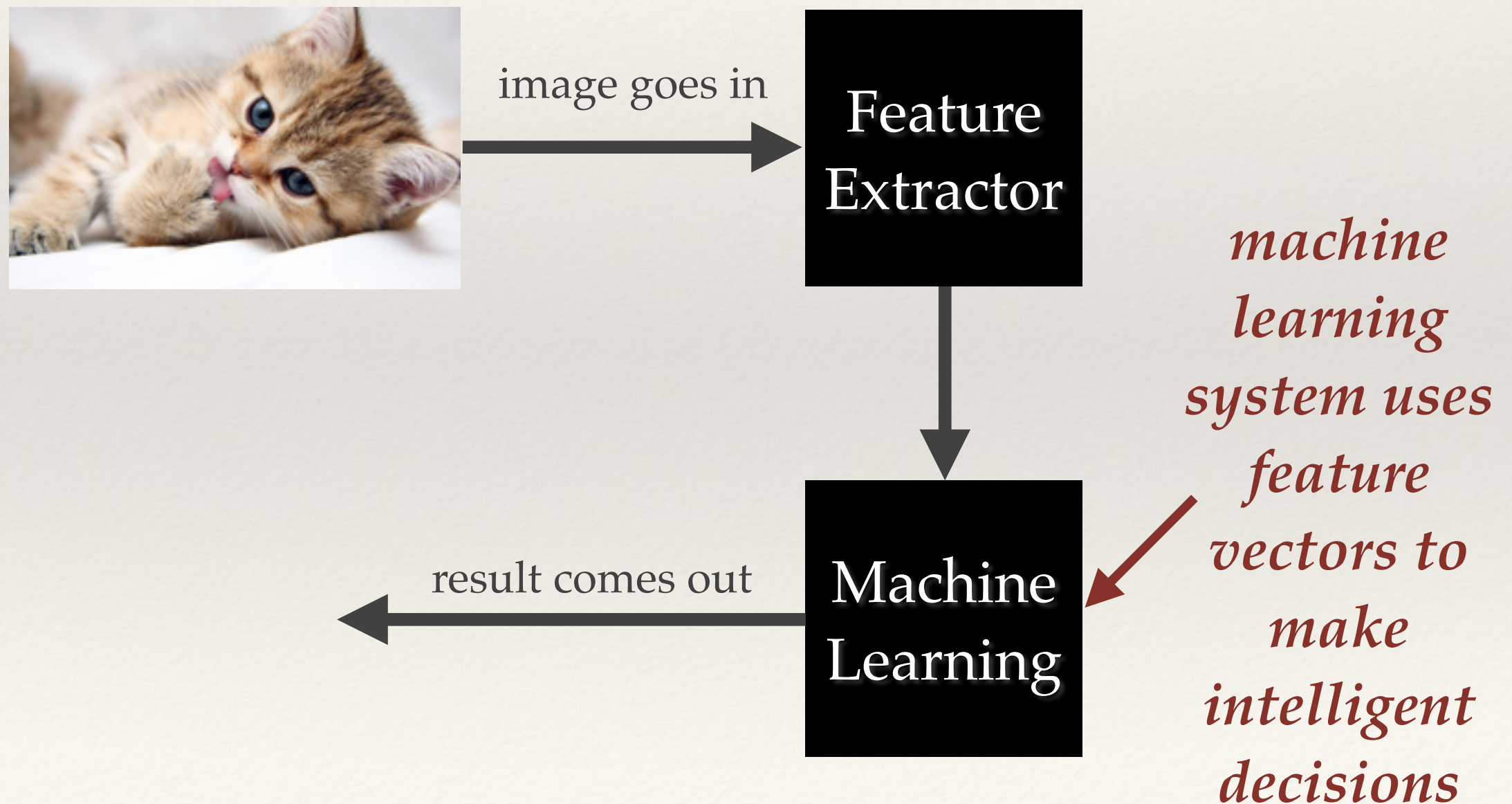


Many computer vision applications involving machine learning take the following form:





Many computer vision applications involving machine learning take the following form:





---

# Key terminology

---

- ❖ **featurevector**: a mathematical vector
  - ❖ just a list of (usually Real) numbers
  - ❖ has a fixed number of **elements** in it
    - ❖ The number of elements is the **dimensionality** of the vector
- ❖ represents a **point** in a **featurespace** or equally a **direction** in the featurespace
- ❖ the **dimensionality of a featurespace** is the dimensionality of every vector within it
  - ❖ vectors of differing dimensionality can't exist in the same featurespace

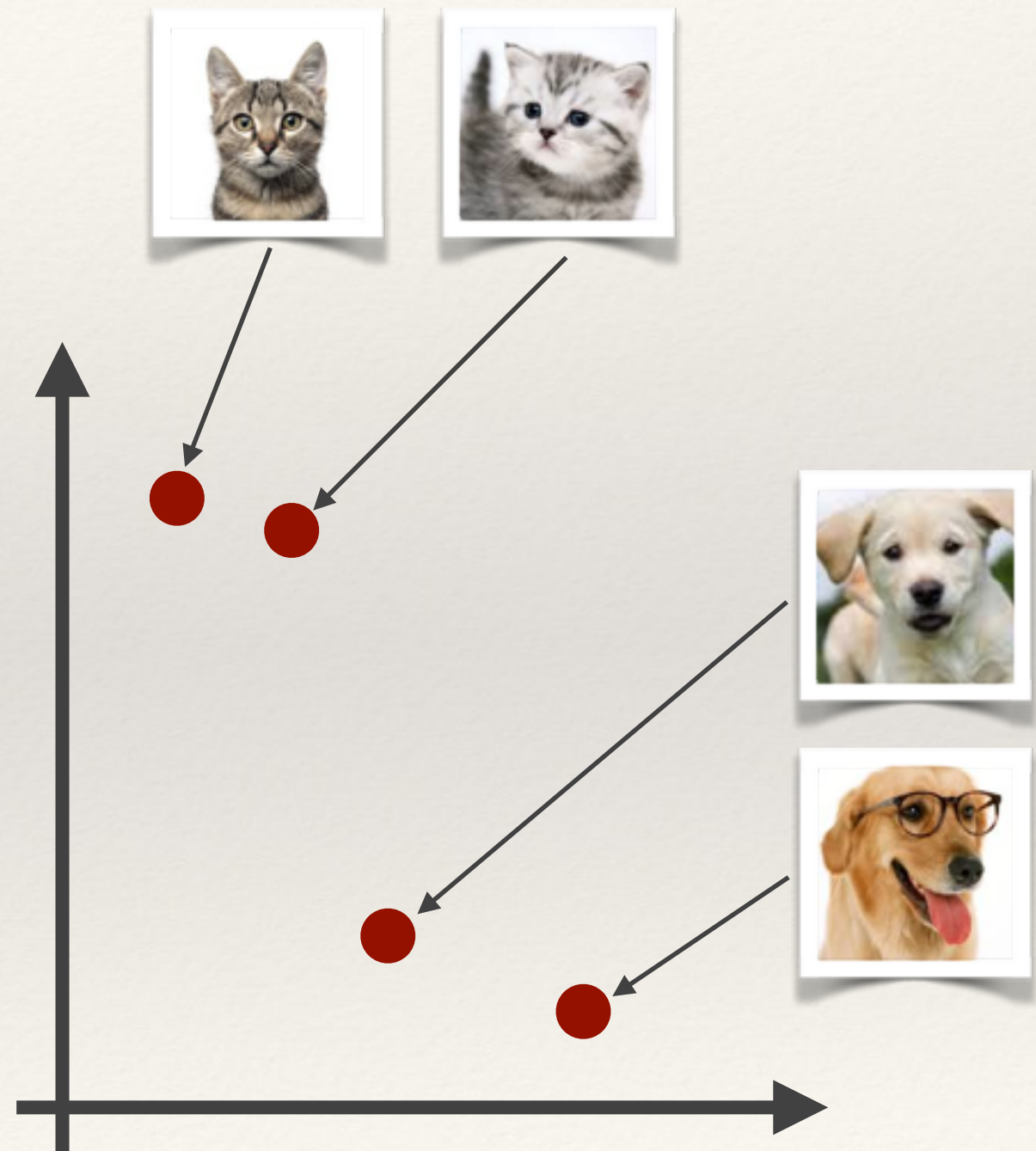
*Demo: a really simple feature  
extractor*



# Distance and similarity

# Distances in featurespace

- ❖ Feature extractors are often defined so that they produce vectors that are *close* together for *similar* inputs
- ❖ Closeness of two vectors can be computed in the feature space by measuring a distance between the vectors.

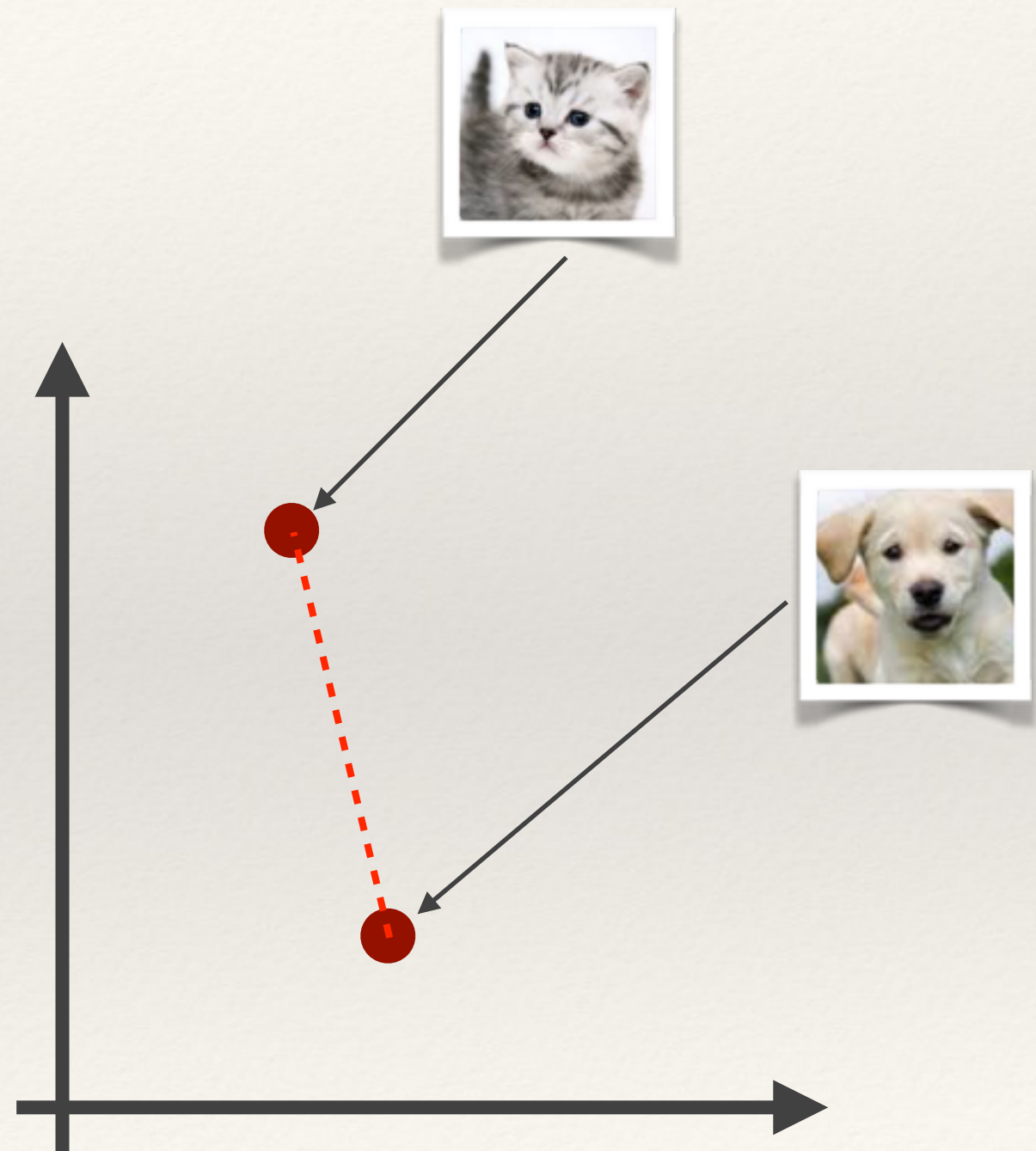




# Euclidean distance (*L2 distance*)

- ❖ L2 distance is the most intuitive distance...
- ❖ The straight-line distance between two points
- ❖ Computed via an extension of Pythagoras theorem to  $n$  dimensions:

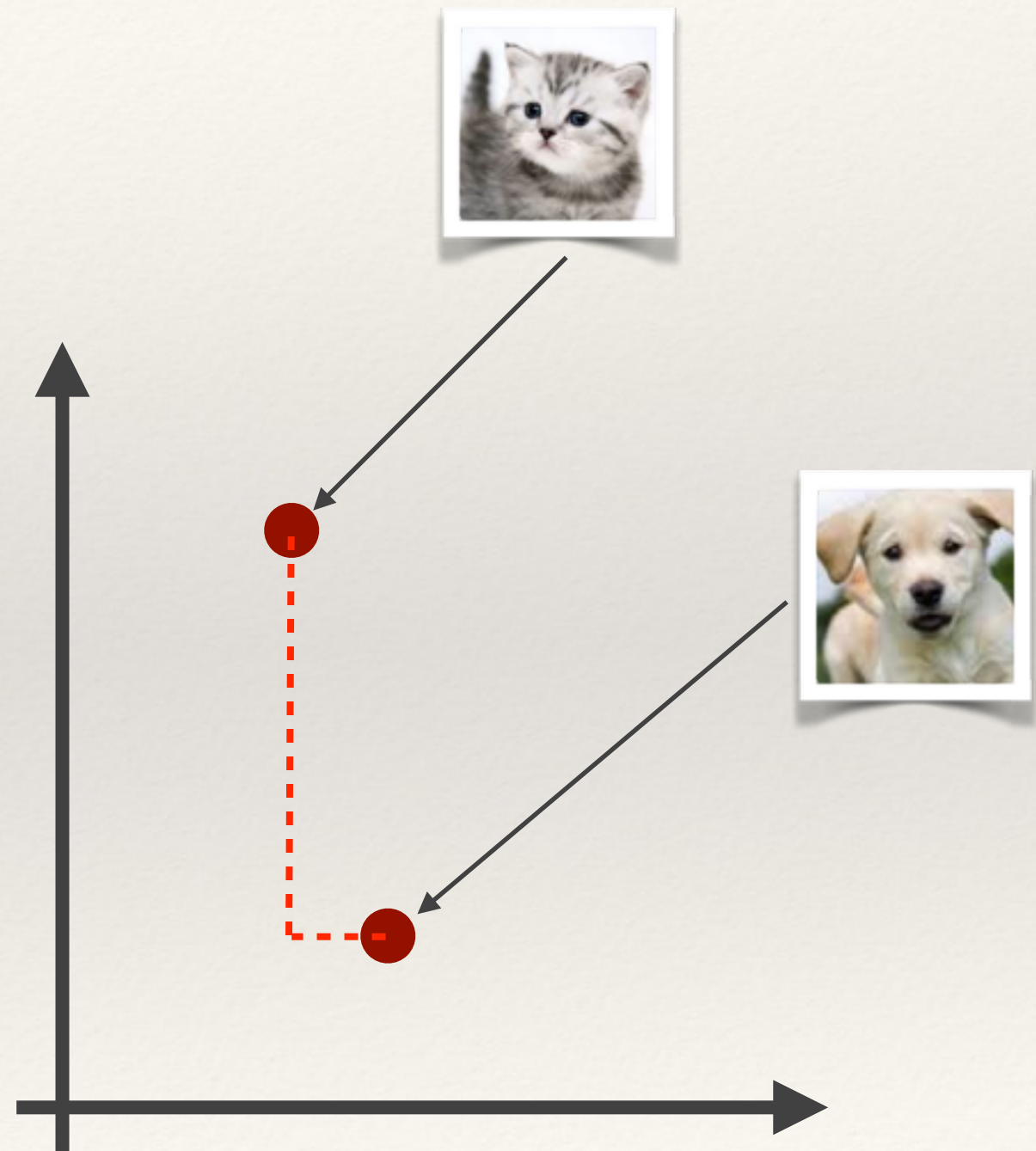
$$D_2(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} = \|p - q\| = \sqrt{(p - q) \cdot (p - q)}$$



# L1 distance (*aka Taxicab/Manhattan*)

- ❖ L1 distance is computed along paths parallel to the axes of the space:

$$D_1(p, q) = \sum_{i=1}^n |p_i - q_i| = \|p - q\|_1$$





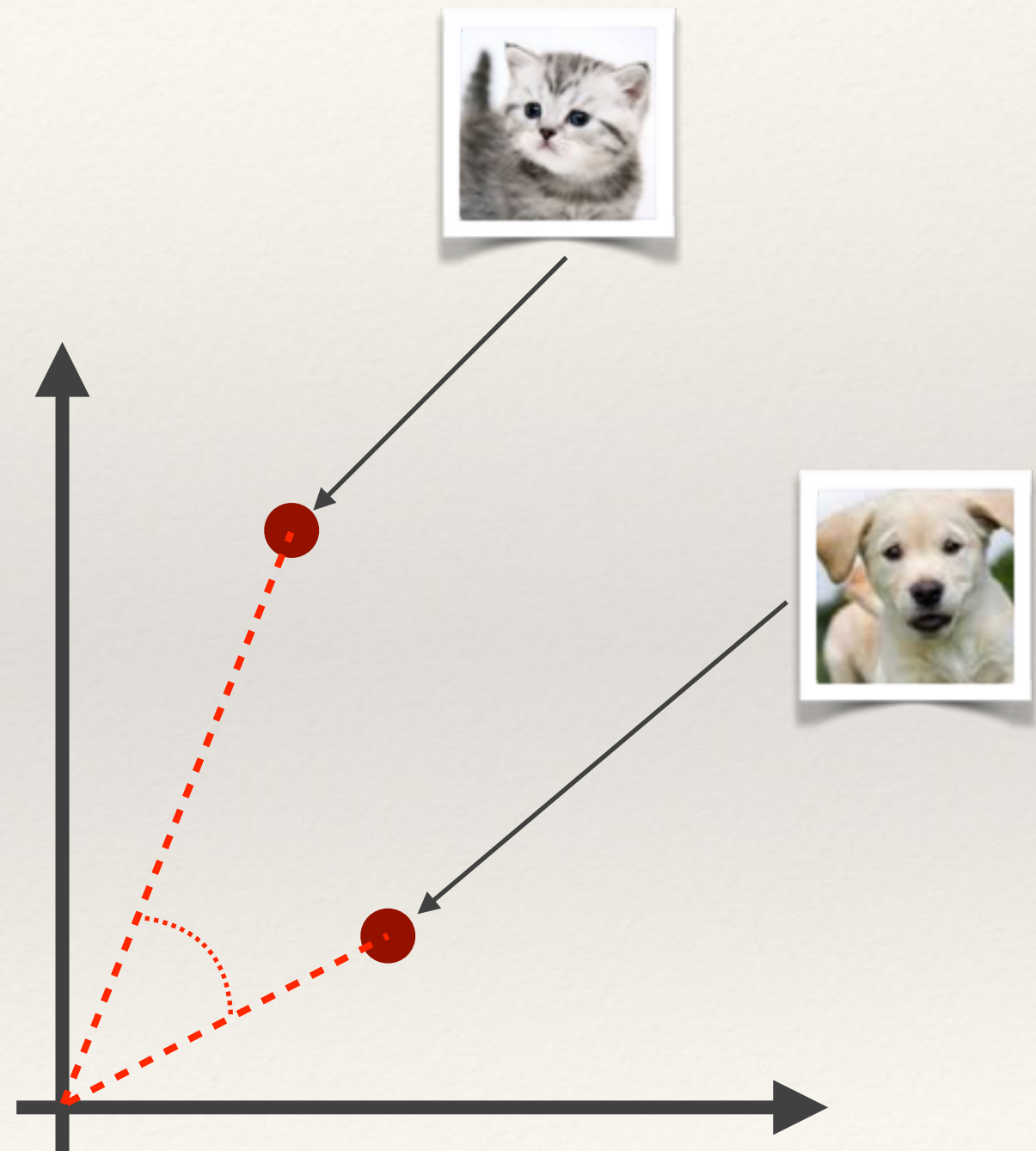
# Cosine Similarity

- ❖ Cosine similarity measures the cosine of the angle between two vectors

- ❖ **It is not a distance!**

$$\cos(\theta) = \frac{p \cdot q}{\|p\| \|q\|} = \frac{\sum_{i=1}^n p_i q_i}{\sqrt{\sum_{i=1}^n p_i^2} \sqrt{\sum_{i=1}^n q_i^2}}$$

- ❖ Useful if you don't care about the relative length of the vectors



Choosing good feature vector  
representations for machine-learning



- ❖ Choose features which allow to distinguish objects or classes of interest
  - ❖ Similar within classes
  - ❖ Different between classes
- ❖ Keep number of features small
  - ❖ Machine-learning can get more difficult as dimensionality of featurespace gets large

# Supervised Machine Learning: *Classification*

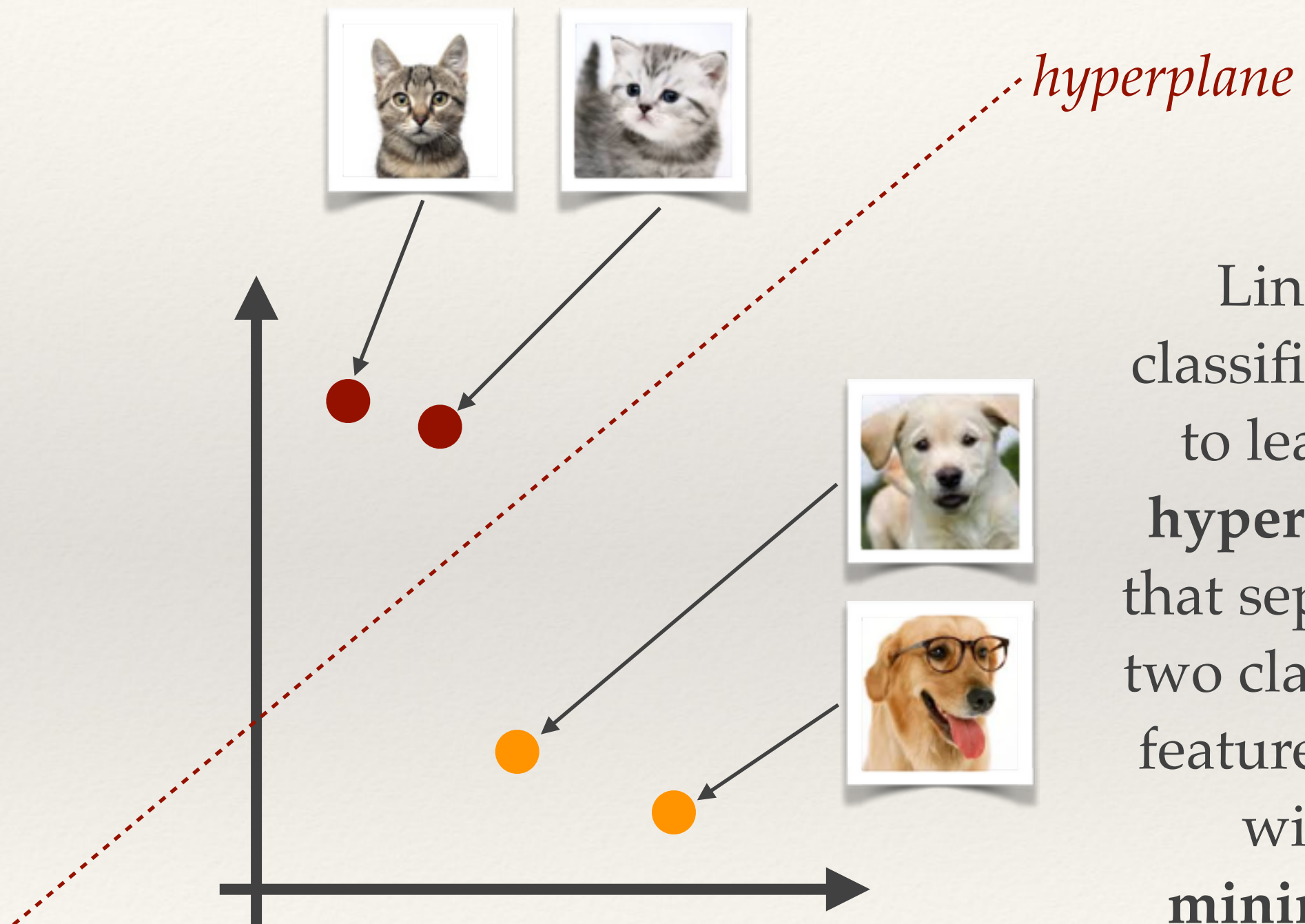


- ❖ **Classification** is the process of assigning a **class label** to an object (typically represented by a vector in a feature space).
- ❖ A **supervised machine-learning algorithm** uses a set of pre-labelled *training data* to learn how to assign class labels to vectors (and the corresponding objects).
- ❖ A **binary** classifier only has two classes
- ❖ A **multiclass** classifier has many classes.



Cat or Dog?

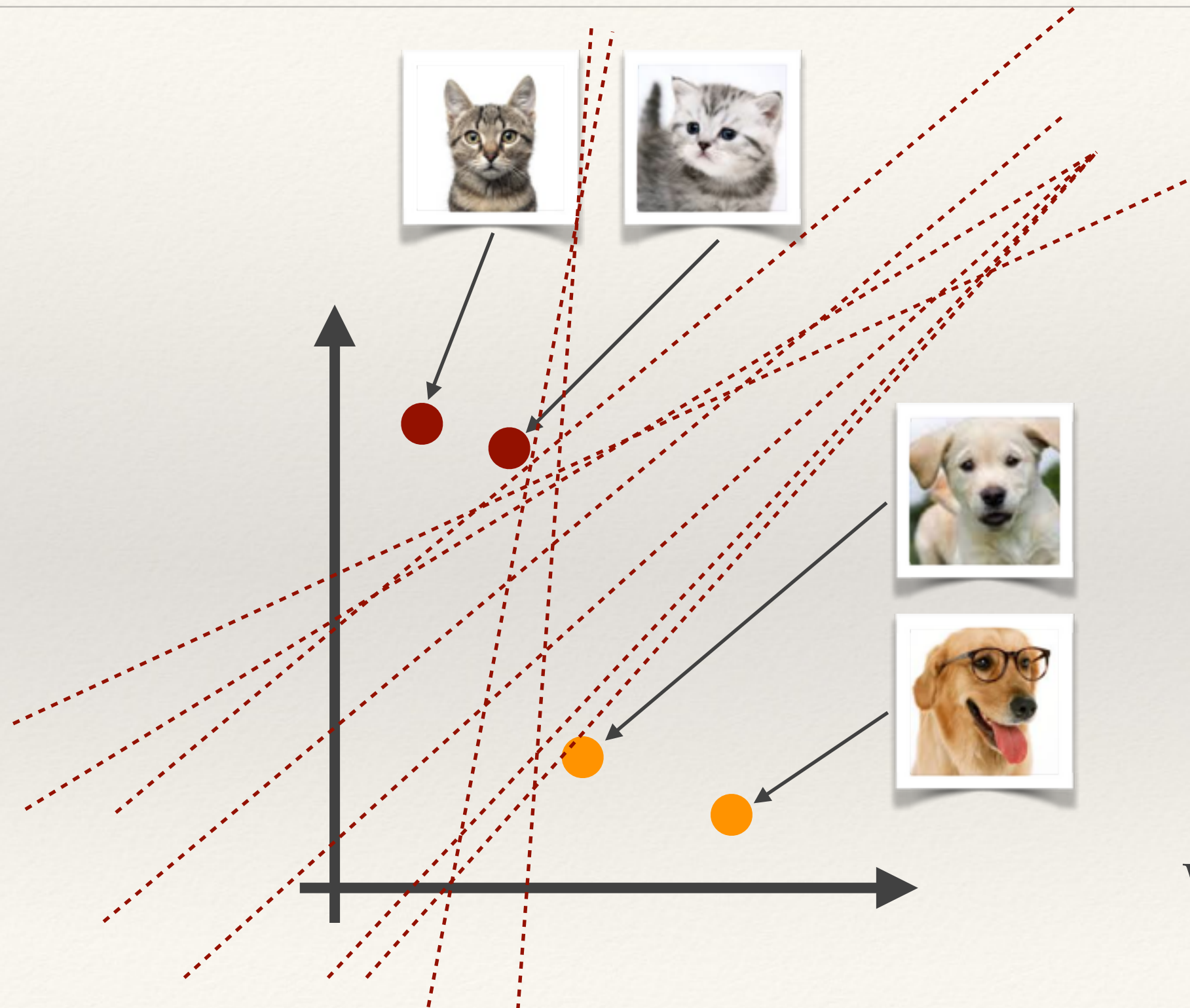
# Linear classifiers



Linear classifiers try to learn a **hyperplane** that separates two classes in featurespace with **minimum error**



# Linear classifiers



Lots of  
hyperplanes  
to choose  
from...  
different  
linear  
classification  
algorithms  
apply  
differing  
constraints  
when learning  
the classifier



# Linear classifiers



*hyperplane*

*Cats*

*Dogs*

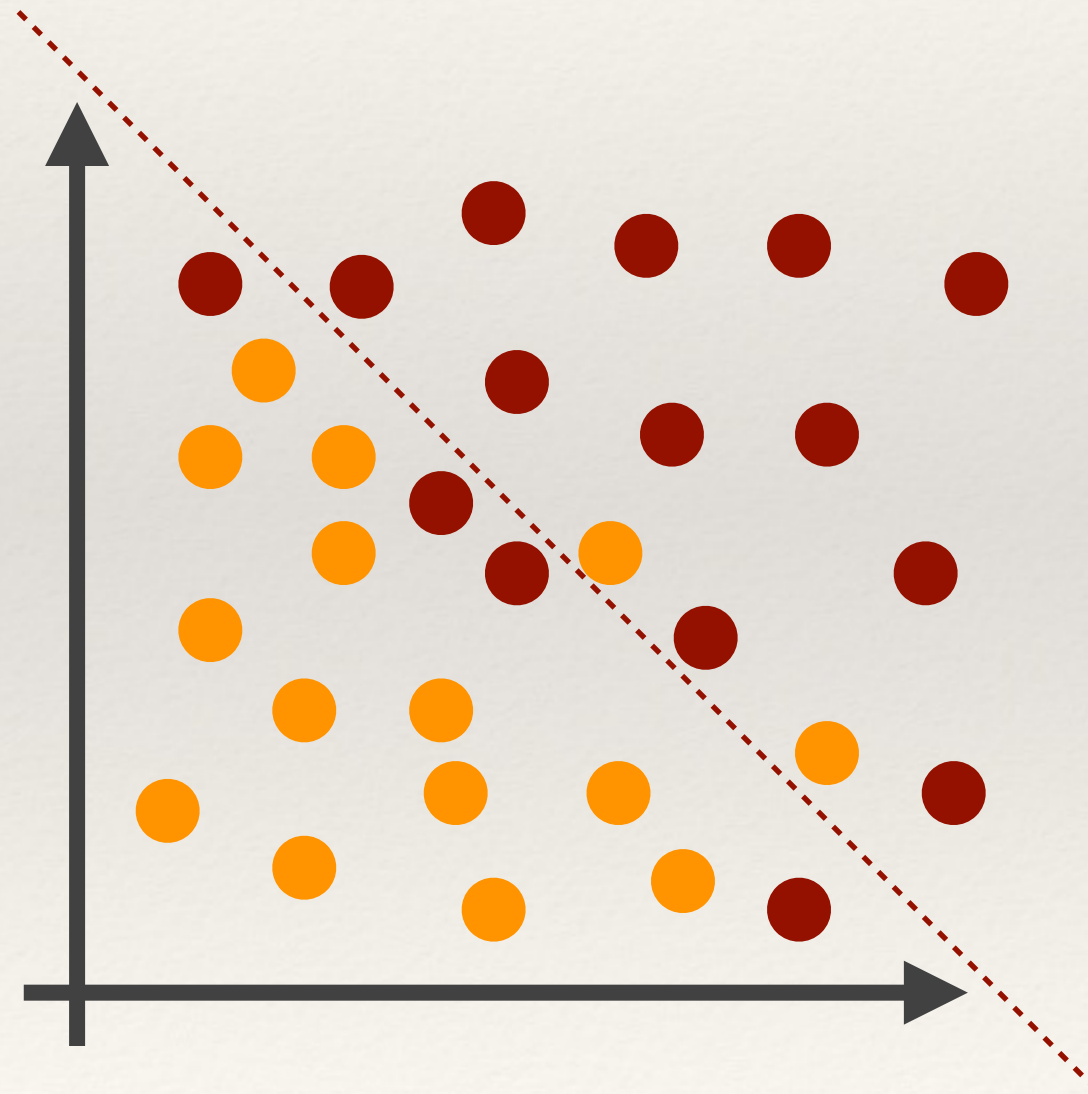
To classify a new image, you just need to check what side of the hyperplane it is on

*Demo: perceptron linear classifier*

---

# Non-linear binary classifiers

---



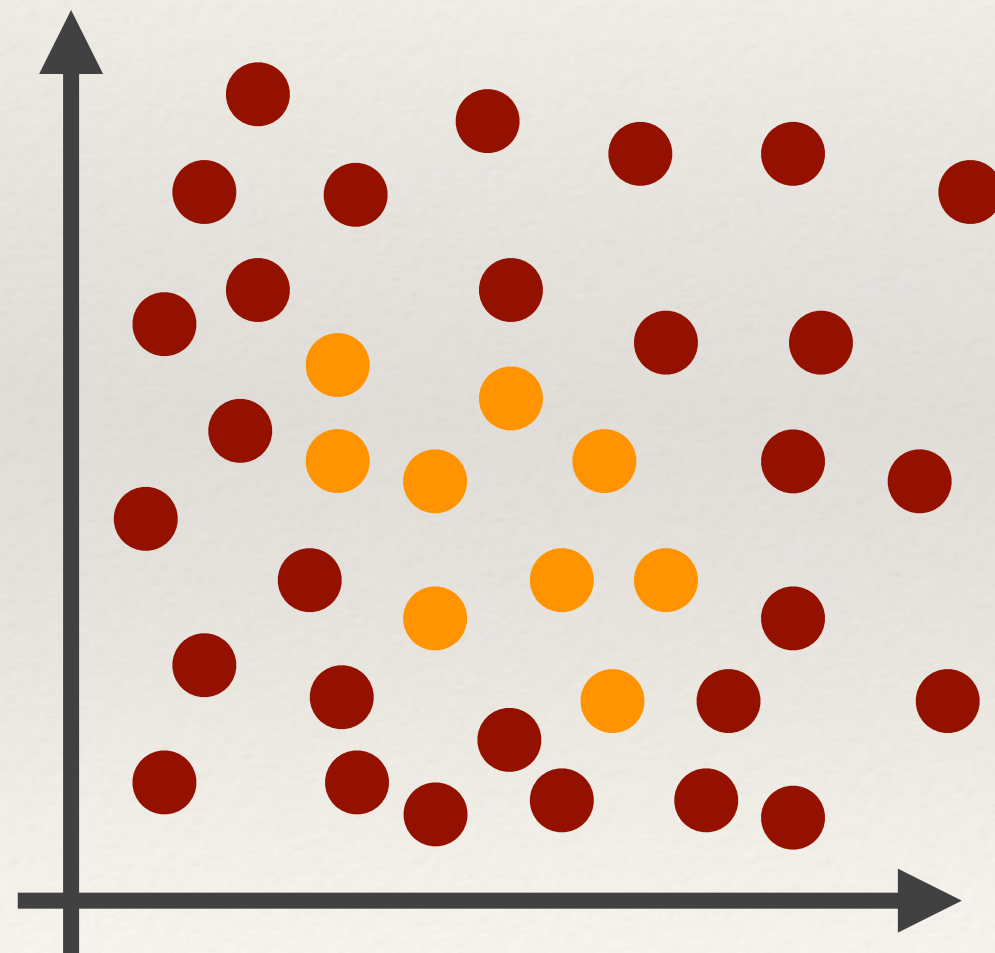
Linear  
classifiers  
work best  
when the data  
is linearly  
separable...



---

# Non-linear binary classifiers

---

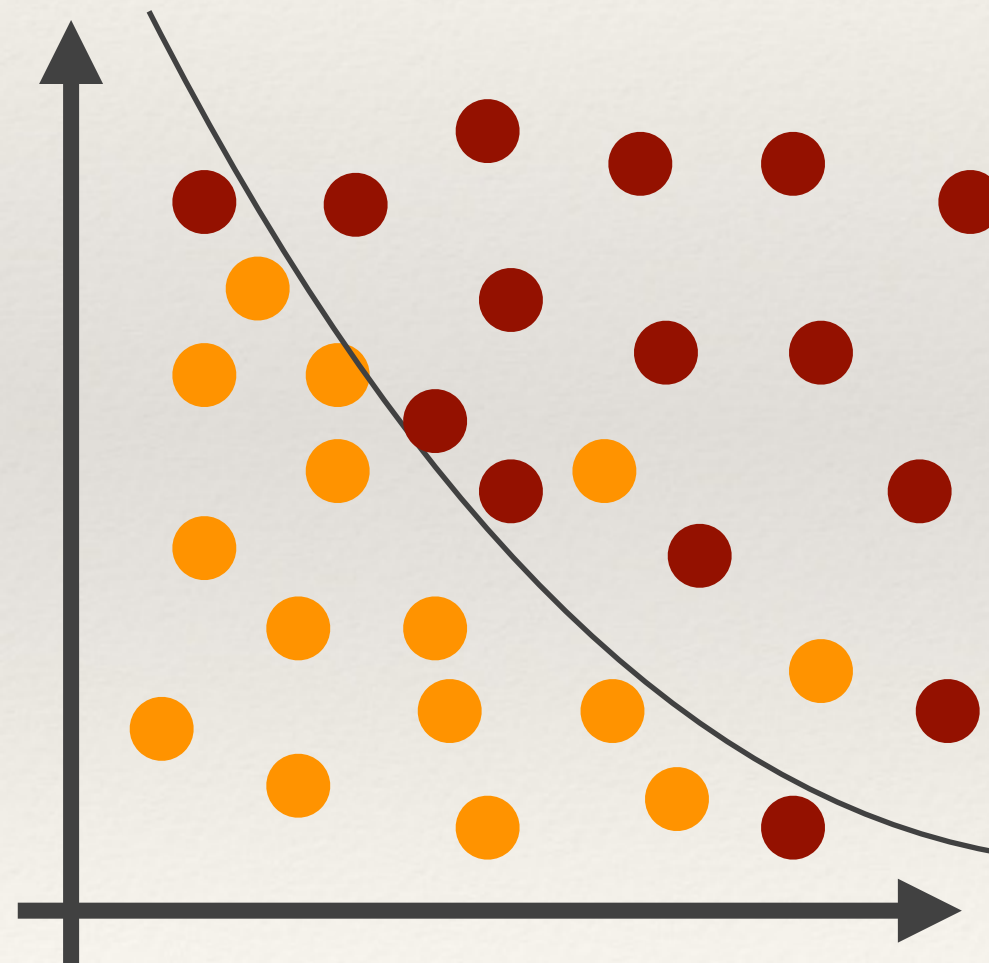


No hope for a  
linear  
classifier!

---

# Non-linear binary classifiers

---

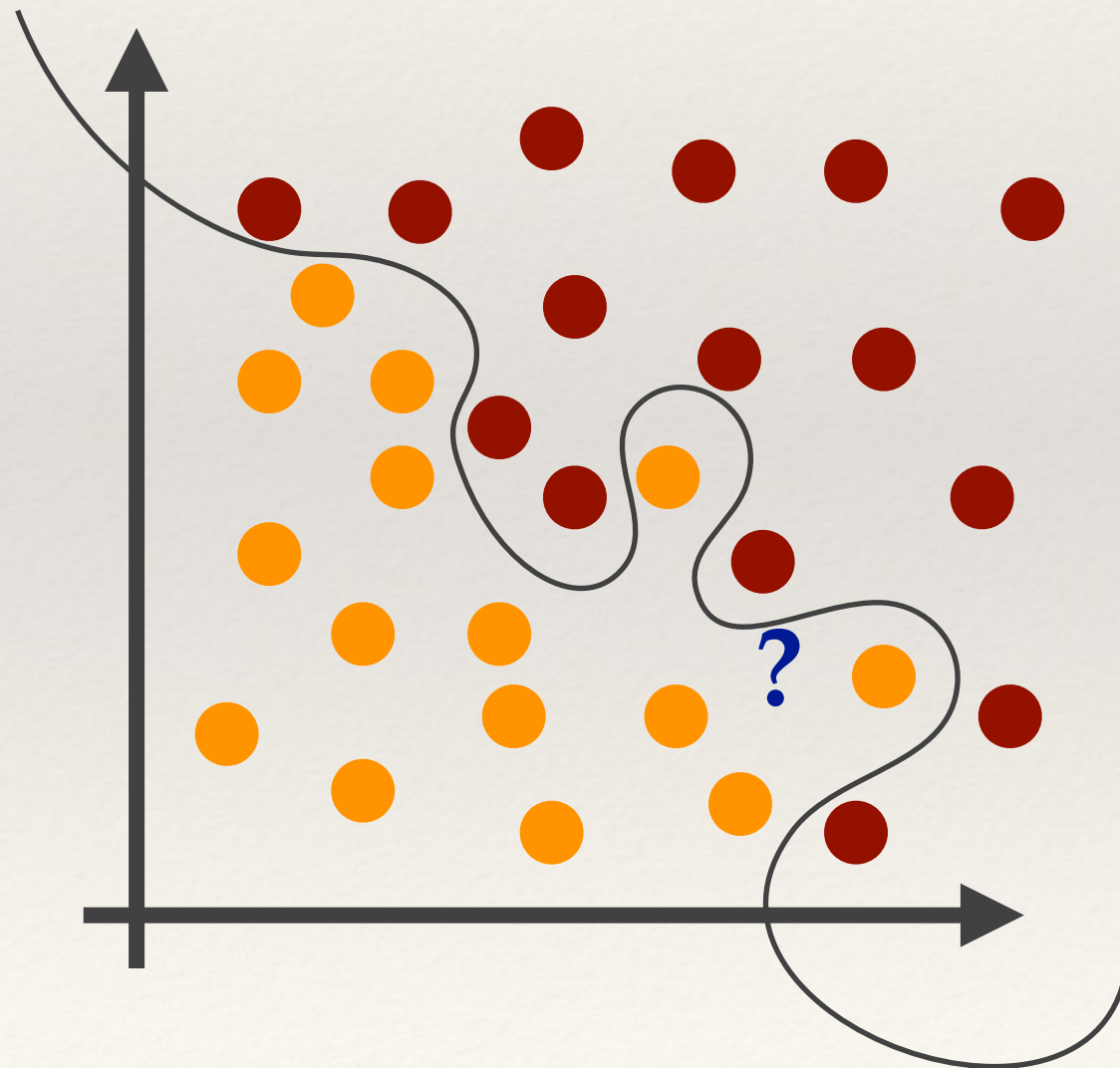


Non-linear  
binary  
classifiers,  
such as  
**Kernel  
Support  
Vector  
Machines**  
learn non-  
linear decision  
boundaries

---

# Non-linear binary classifiers

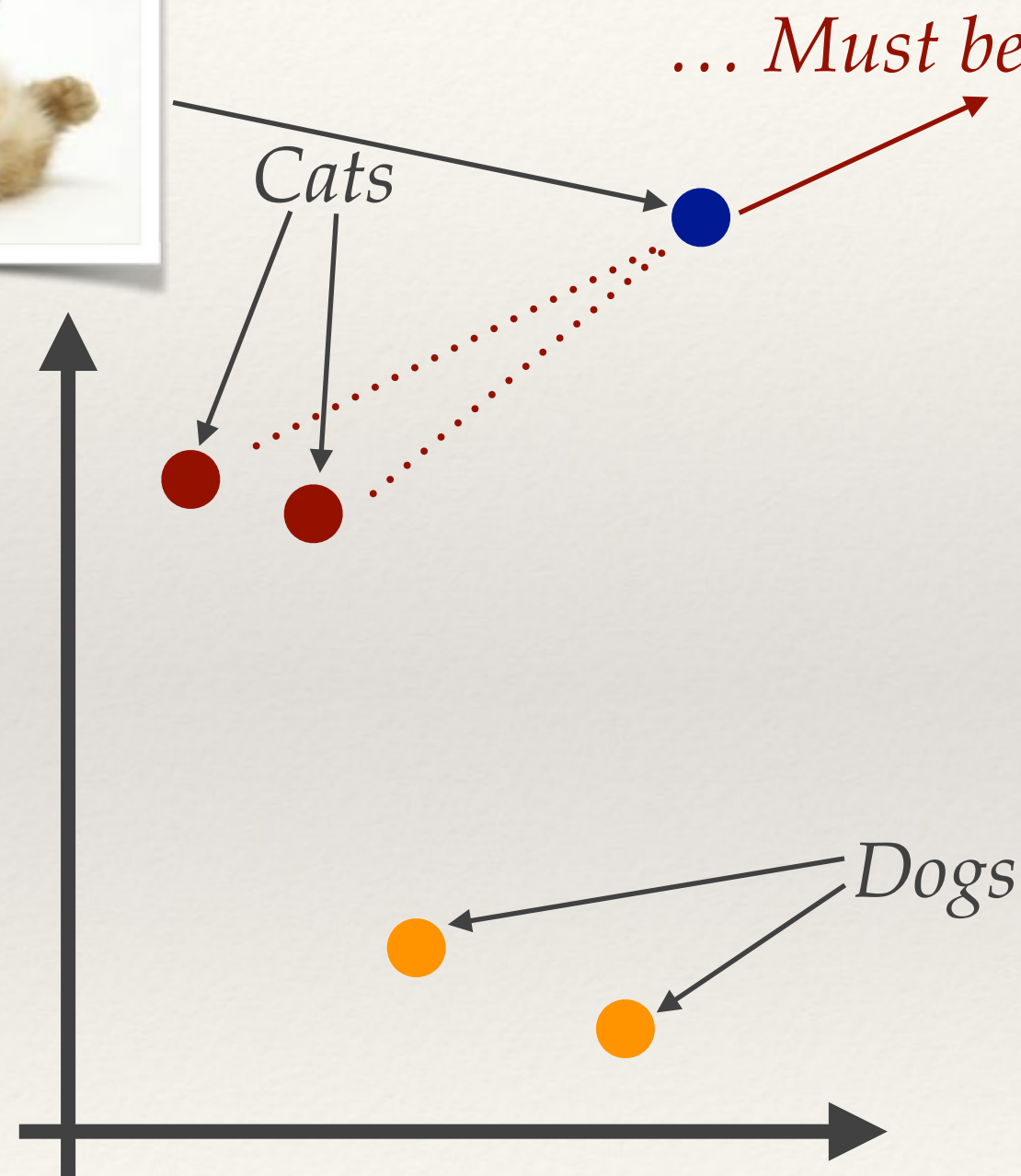
---



Have to be  
careful... you  
might lose  
generality by  
overfitting



# Multiclass classifiers: KNN



Assign class of  
unknown  
point based on  
majority class  
of *closest K*  
neighbours in  
featurespace

# *Demo: KNN Classification*

---

# KNN Problems

---

- ❖ Computationally expensive if there are:
  - ❖ Lots of training examples
  - ❖ Many dimensions



---

# Multiclass linear classifiers

---

- ❖ A linear classifier is by definition binary
- ❖ So, how can we solve multiclass problems with linear classifiers?
  - ❖ One versus All (OvA) / One versus Rest (OvR)
    - ❖ one classifier per class
  - ❖ One versus One (OvO)
    - ❖  $K(K - 1) / 2$  classifiers

# Unsupervised Machine Learning:

## *Clustering*



- ❖ Clustering aims to group data without any prior knowledge of what the groups should look like or contain.
- ❖ In terms of feature vectors, items with similar vectors should be grouped together by a clustering operation.
- ❖ Some clustering operations create overlapping groups; for now we're only interested in disjoint clustering methods that assign an item to a single group.





---

# K-Means Clustering

---

- ❖ K-Means is a classic featurespace clustering algorithm for grouping data into  $K$  groups with each group represented by a *centroid*:
  - ❖ The value of  $K$  is chosen
  - ❖  $K$  initial cluster centres are chosen
  - ❖ Then the following process is performed iteratively until the centroids don't move between iterations:
    - ❖ Each point is assigned to its closest centroid
    - ❖ The centroid is recomputed as the mean of all the points assigned to it. If the centroid has no points assigned it is randomly re-initialised to a new point.
- ❖ The final clusters are created by assigning all points to their nearest centroid.

# *Demo: K-Means Clustering*

---

# Summary

---

- ❖ Extracting features is key part of computer vision
  - ❖ Typically, these are numerical vectors that can be used with machine-learning techniques.
  - ❖ Feature vectors can be compared by measuring distance
- ❖ Classification learns what class to assign a feature to.
- ❖ Clustering groups similar features.