

# EECS 395/495 and Data\_Sci 423 Machine Learning

Aggelos K. Katsaggelos

Joseph Cummings Professor  
Northwestern University  
Department of EECS  
Department of Linguistics  
Argonne National Laboratory  
NorthShore University Health System  
Evanston, IL 60208  
<http://ivpl.eecs.northwestern.edu>



# Outline

- Intro to Machine Learning and Numerical Optimization
- Regression
- Classification
- Function Approximation
- Bayesian Regression and Classification
- Matrix Factorization (K-Means, PCA, Recommender Systems,...)

# Philosophy

- Depth vs Breadth
- Be able to derive and implement important ML algorithms
- Be able to easily expand your knowledge

# Lecture 1

- What is Machine Learning
- Applications
- The two pillars: feature design and optimization
- Elements of Optimization

Providing a unique approach to machine learning, this text contains fresh and intuitive, yet rigorous, descriptions of all fundamental concepts necessary to conduct research, build products, tinker, and play. By prioritizing geometric intuition, algorithmic thinking, and practical real-world applications in disciplines including computer vision, natural language processing, economics, neuroscience, recommender systems, physics, and biology, this text provides readers with both a lucid understanding of foundational material as well as the practical tools needed to solve real-world problems. With in-depth Python and MATLAB/OCTAVE-based computational exercises and a complete treatment of cutting edge numerical optimization techniques, this is an essential resource for students and an ideal reference for researchers and practitioners working in machine learning, computer science, electrical engineering, signal processing, and numerical optimization.

#### KEY FEATURES

- A presentation built on lucid geometric intuition
- A unique treatment of state-of-the-art numerical optimization techniques
- A fused introduction to logistic regression and support vector machines
- Inclusion of feature design and learning as major topics
- An unparalleled presentation of advanced topics through the lens of function approximation
- A refined description of deep neural networks and kernel methods

Jeremy Watt received his PhD in Computer Science and Electrical Engineering from Northwestern University. His research interests lie in machine learning and computer vision, as well as numerical optimization.

Reza Borhani received his PhD in Computer Science and Electrical Engineering from Northwestern University. His research interests lie in the design and analysis of algorithms for problems in machine learning and computer vision.

Aggelos K. Katsaggelos is a professor and holder of the Joseph Cummings chair in the Department of Electrical Engineering and Computer Science at Northwestern University, where he also heads the Image and Video Processing Laboratory.



Cover designed by HartMcLeod Ltd

CAMBRIDGE  
UNIVERSITY PRESS  
www.cambridge.org

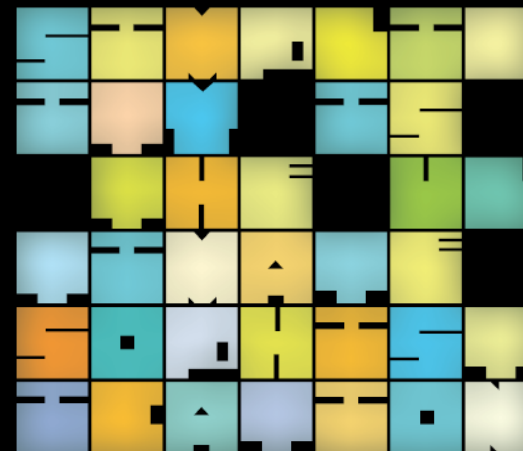


Watt, Borhani,  
and Katsaggelos MACHINE LEARNING REFINED

Jeremy Watt • Reza Borhani • Aggelos K. Katsaggelos

# MACHINE LEARNING REFINED

Foundations, Algorithms, and Applications



Download code + sample chapters  
at [www.mlrefined.com](http://www.mlrefined.com)

# What is Machine Learning

- A machine learning algorithm is an algorithm that is able to learn from data
- But what do we mean by learning?
- “A computer program is said to learn from **experience  $E$**  with respect to some class of **tasks  $T$**  and **performance measure  $P$** , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .” (Mitchell 1997)

# Task

- ML allows us to tackle tasks that are too difficult to solve with fixed programs written and designed by human beings
  - From a scientific and philosophical point of view, ML is interesting because developing our understanding of ML entails developing our understanding of the principles that underlie intelligence
- ML tasks are usually described in terms of how the machine learning system should process an example
  - An example is a collection of features that have been quantitatively measured from some object or event that we want the ML system to process

# Common ML Tasks

- Classification  $f : R^n \rightarrow \{1, \dots, k\}$
- Classification with missing inputs (learn either all possible mappings or the joint distribution of all inputs which can be then marginalized over missing inputs)
- Regression  $f : R^n \rightarrow R$
- Transcription (optical character recognition, speech processing)
- Structured outputs (any task where the output exhibits important relationships between the different elements, e.g., parsing a natural language segment, image segmentation, image captioning)



## Common ML Tasks

- Anomaly detection (fraud detection; profile of user is build and used)
- Synthesis and Sampling (text to speech, video games: automatically generate textures for large objects)
- Imputation of missing values
- Denoising
- Density (or prob mass function) estimation

# The Performance Measure

- Usually specific to the task  $T$
- E.g. Classification
  - Accuracy (proportion of correct output)
  - Similarly: error rate (expected 0-1 loss)
- E.g. Density Estimation
  - Ave log probability the model assigns to some examples
- E.g. Transcription
  - Accuracy at transcribing entire sequences
  - Or more fine grained performance, e.g. partial credit for getting some words right
- E.g. Regression
  - should we penalize the system more if it frequently makes medium-sized mistakes or if it rarely makes very large mistakes?

# The Experience E

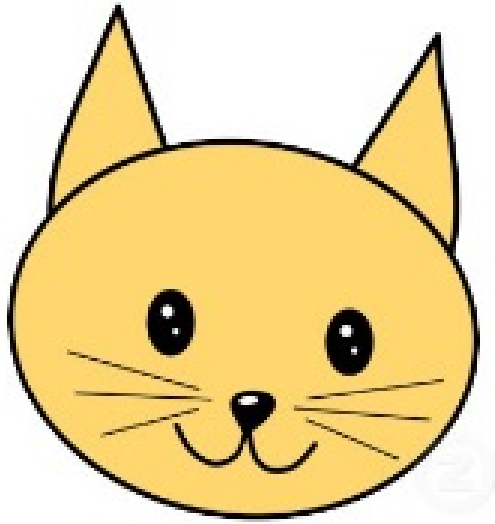
- Machine learning algorithms can be broadly categorized as *unsupervised* or *supervised*
- Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset
- Supervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target

# The Experience E

- In *semi-supervised learning* some examples include a supervision target but others do not
- Some machine learning algorithms do not just experience a fixed dataset
  - For example, *reinforcement learning algorithms* interact with an environment, so there is a feedback loop between the learning system and its experiences

# Classification Pipeline

Is it a cat or a dog?



vs.



# 1. Gather data



## 2. Extract features

(what distinguishes a cat from a dog?)



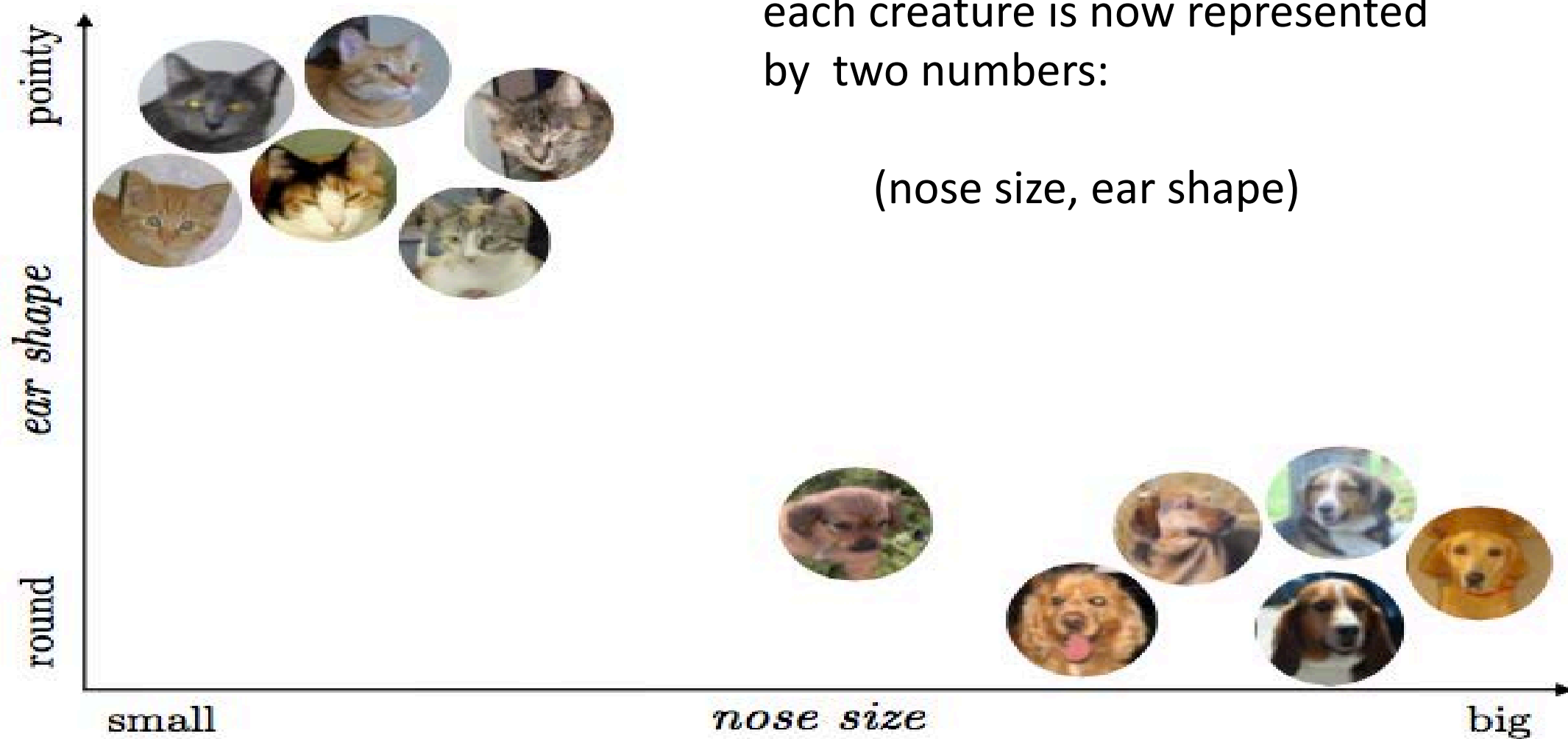
- cats have **small** noses and **pointy** ears
- dogs have **big** noses and **round** ears



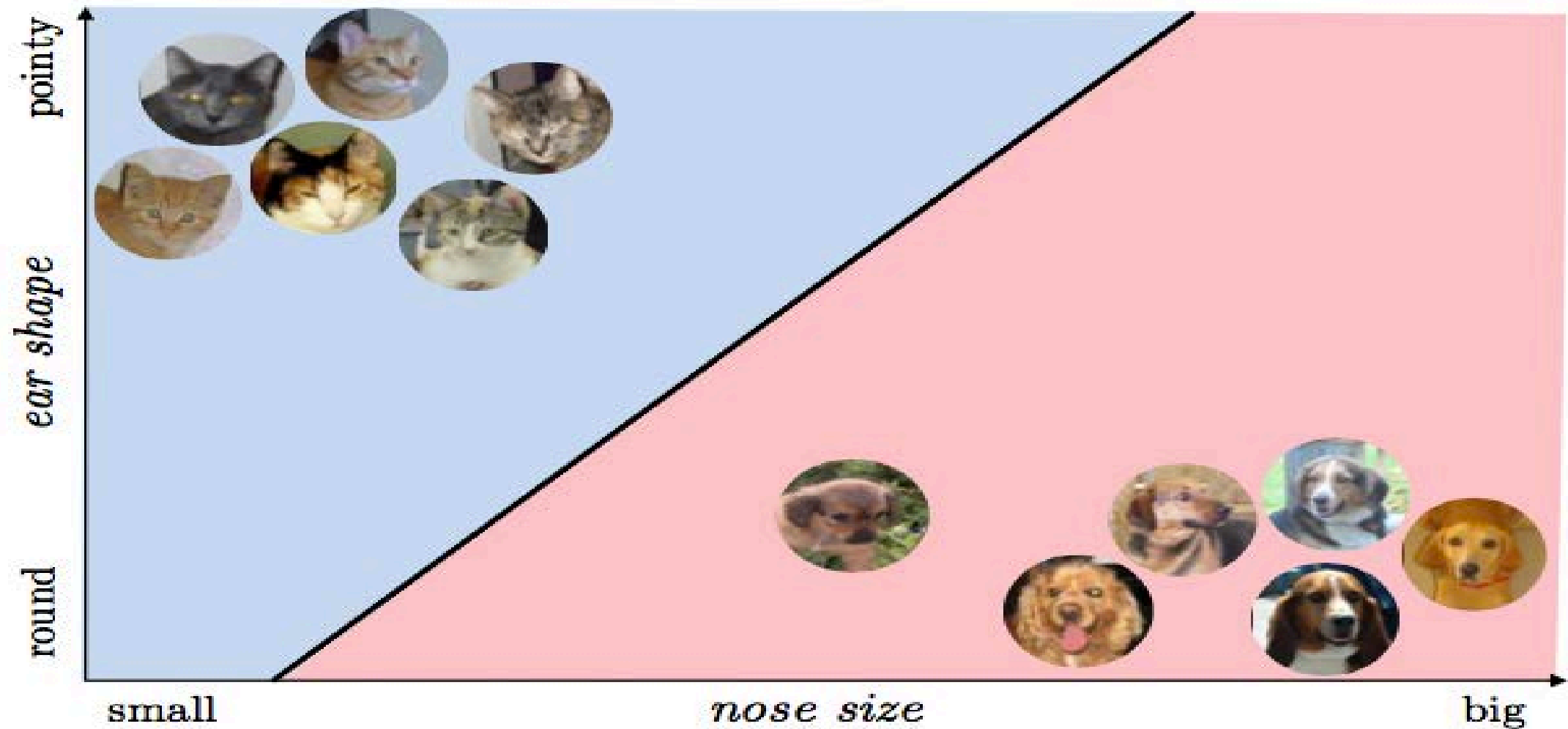
# The *feature space*

each creature is now represented  
by two numbers:

(nose size, ear shape)



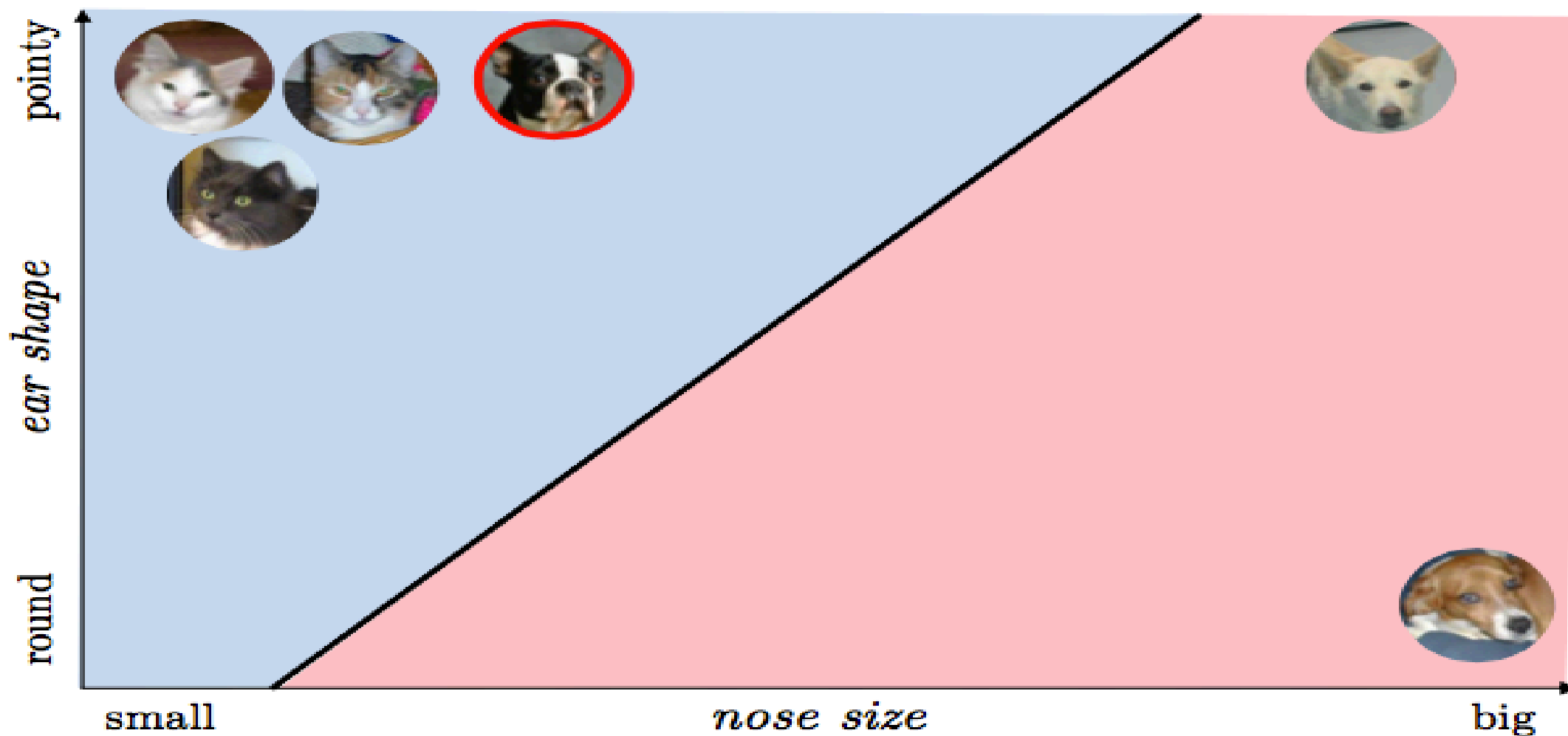
### 3. Train the model (find best parameters via numerical optimization)

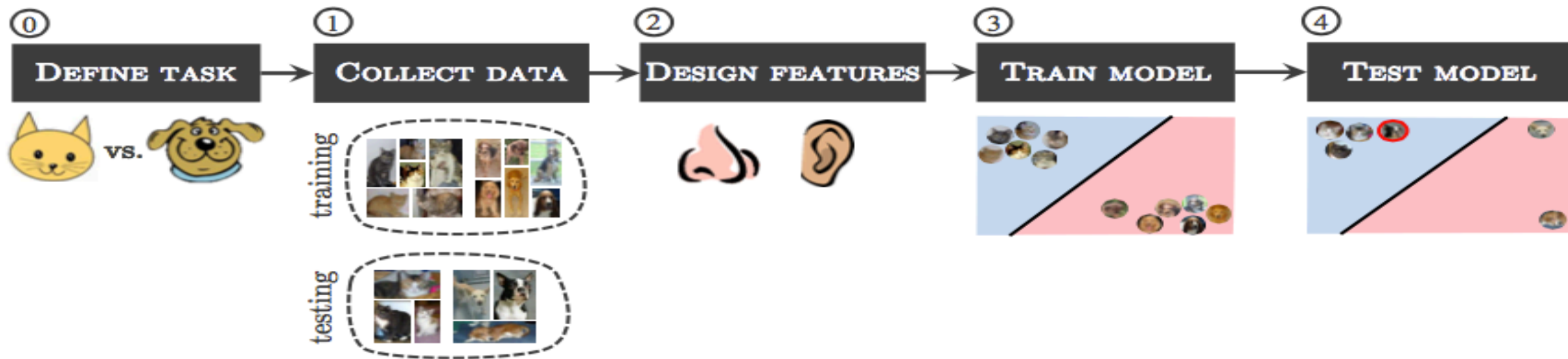


## 5. Test the model (on new data)

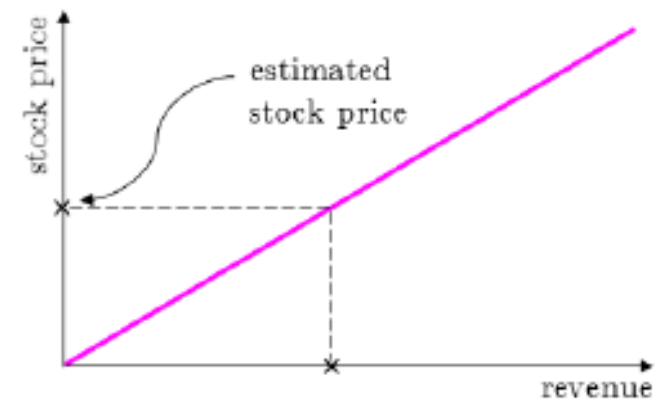
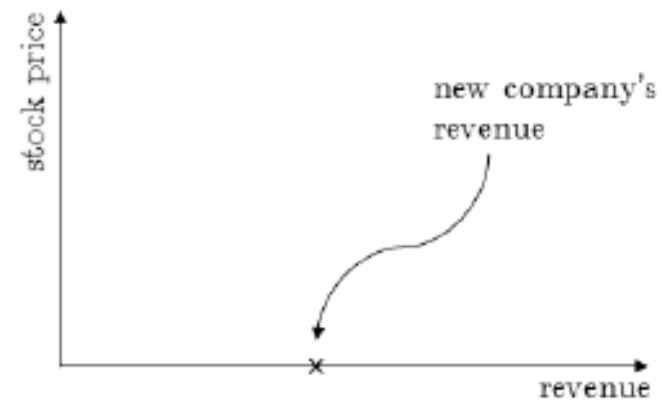
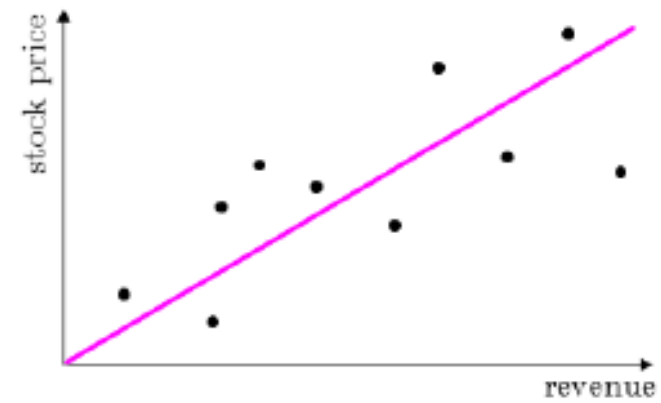
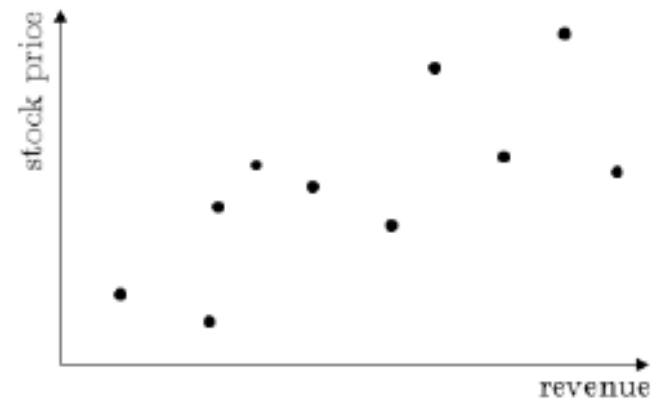


Meanwhile in the *feature space*...





# Regression



# Classification

