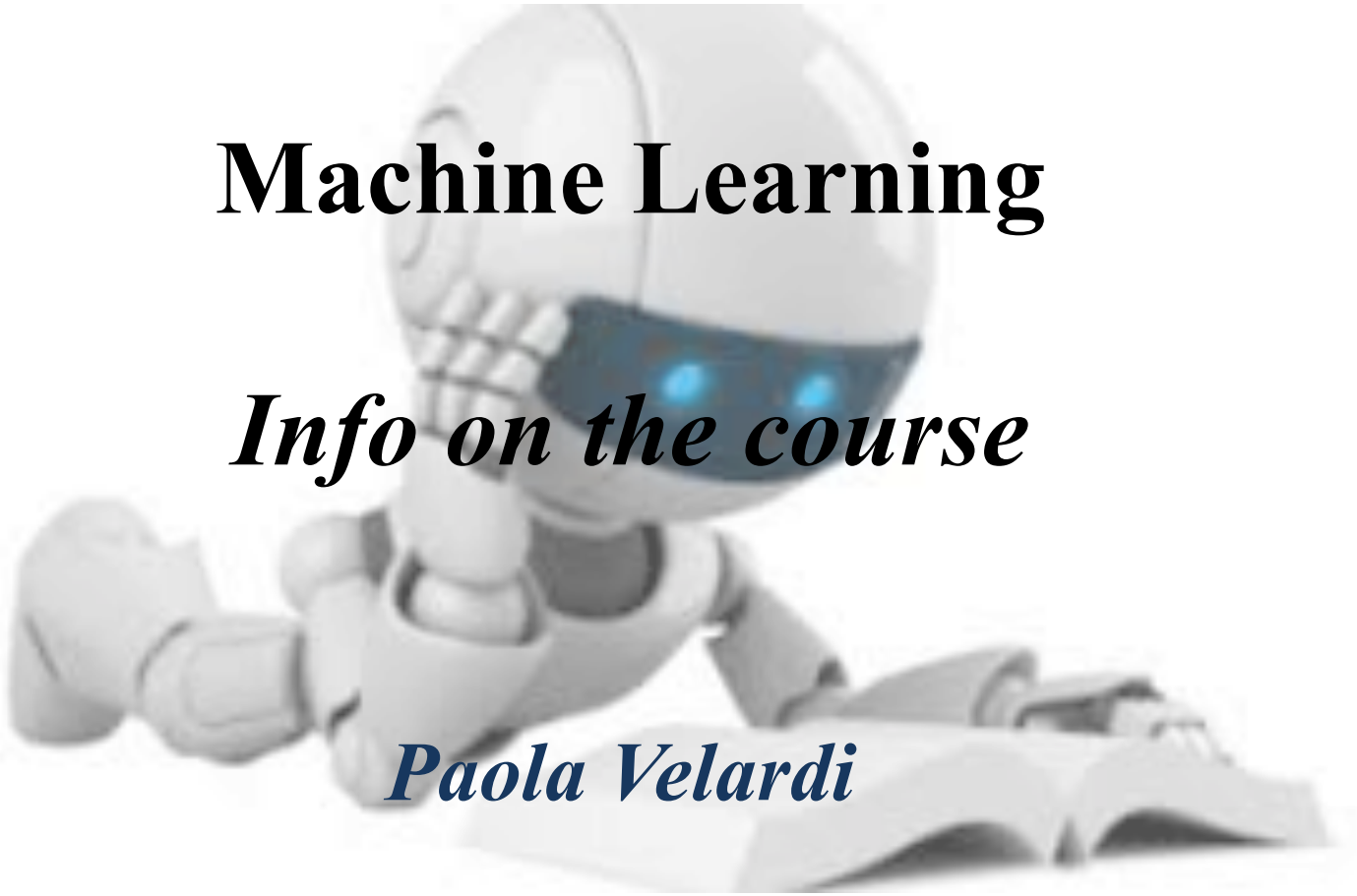
Machine Learning

Info on the course



Paola Velardi

What is machine learning (in a nutshell)

- A set of methodologies to find regularities in data
- These findings are use to predict future outcomes and/or to classify unseen data

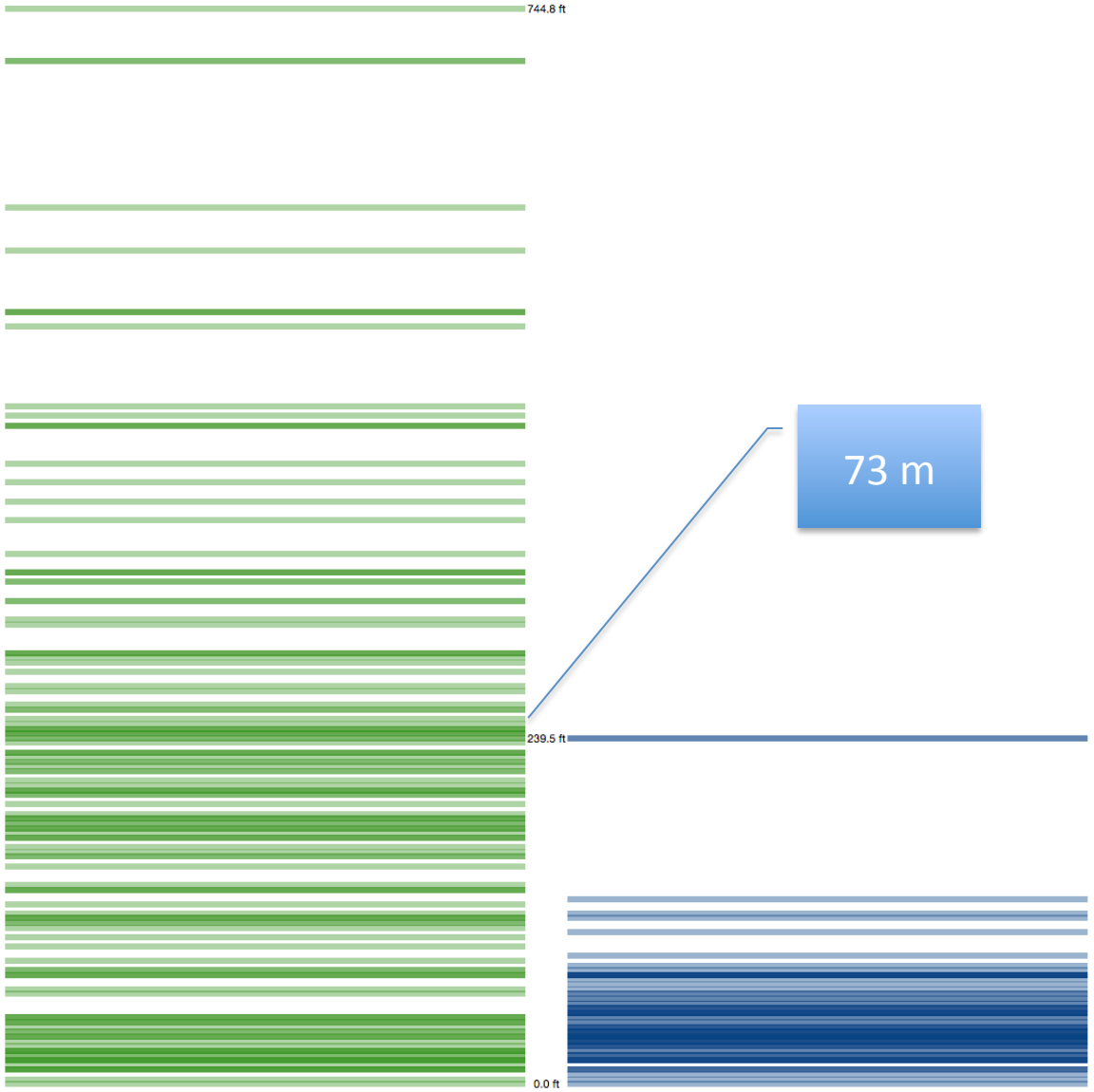
A (VERY) simple example

- Using a data set about homes, we wish to create a machine learning model to distinguish homes in New York from homes in San Francisco.
- Let's say you had to determine whether a home is in **San Francisco** or in **New York**. In machine learning terms, categorizing data points is a **classification** task.

To begin, we consider building elevation

- Since San Francisco is relatively hilly, the elevation of a home may be a good way to distinguish the two cities.
- Based on the home-elevation data to the right, you could argue that a home above 73 meters should be **classified** as one in San Francisco.
- So we can infer the following rule:

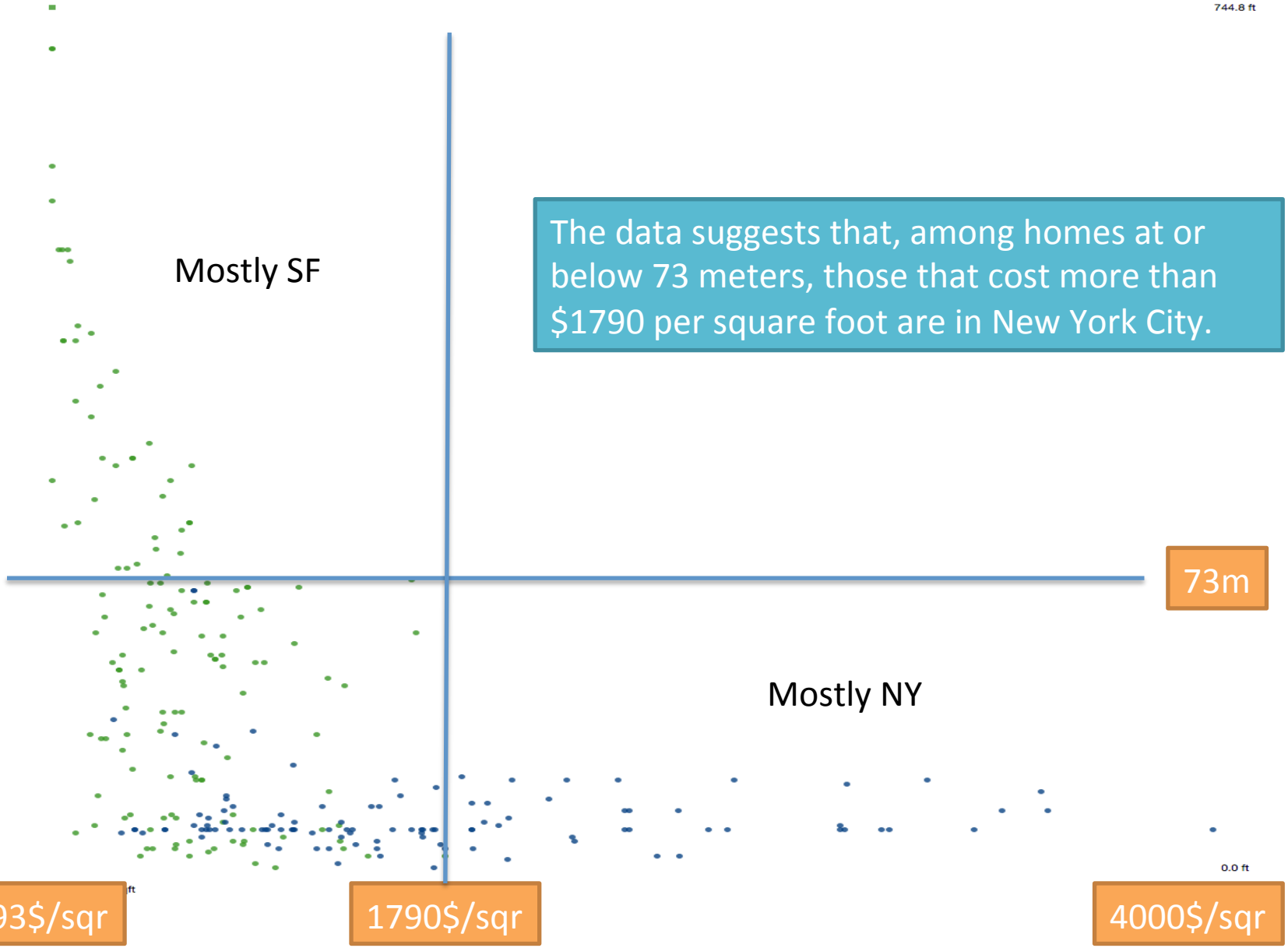
IF elevation>73m THEN city=SF

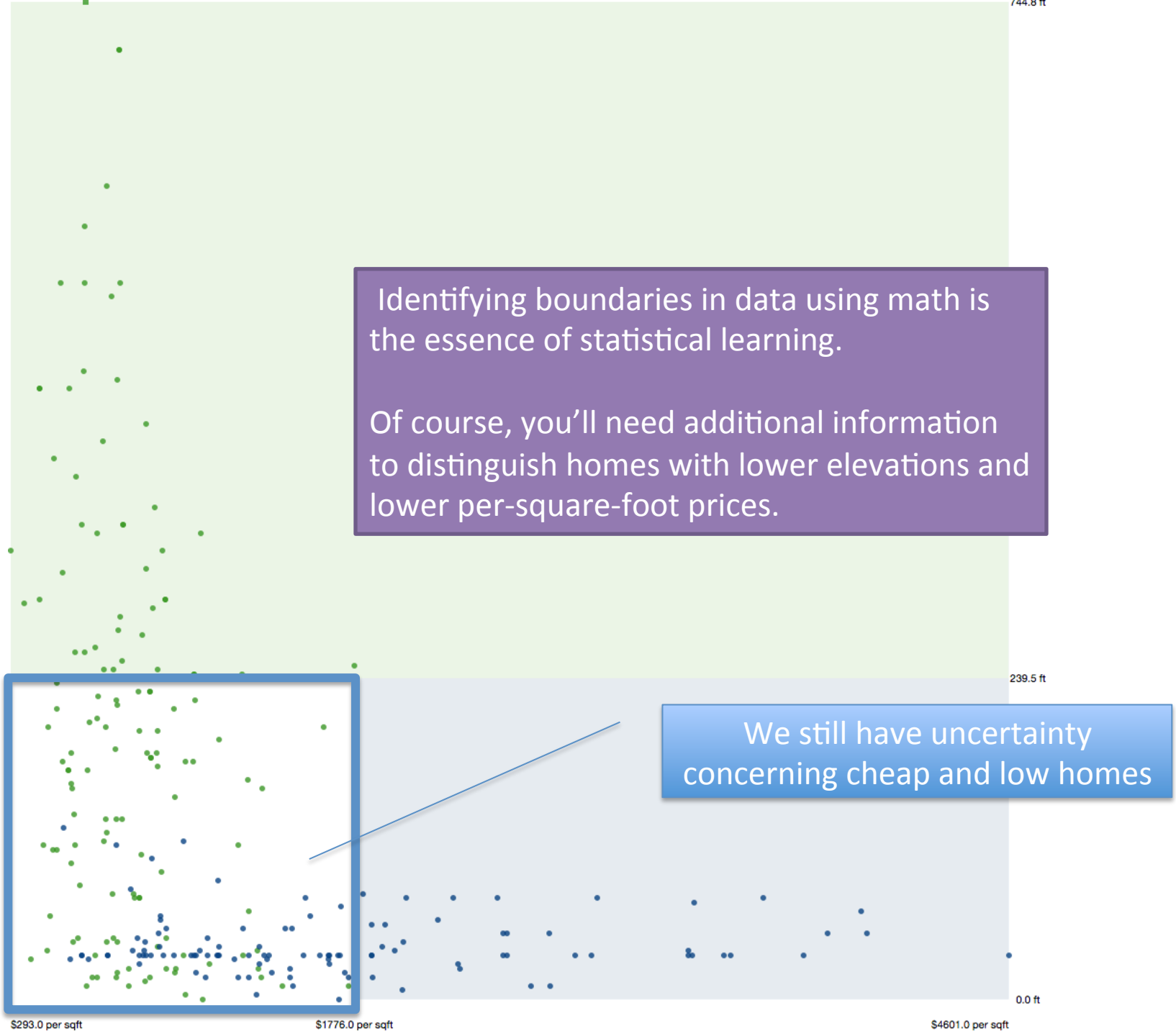


Is this enough?

- What if elevation <73? We cannot tell..
- Adding another **dimension** allows for more nuance. For example, New York apartments can be extremely expensive per square foot.
- Dimensions (height, price) are denoted as FEATURES in machine learning (= what may characterize the objects we wish to classify)

Elevation *and* price per square foot **scatterplot**

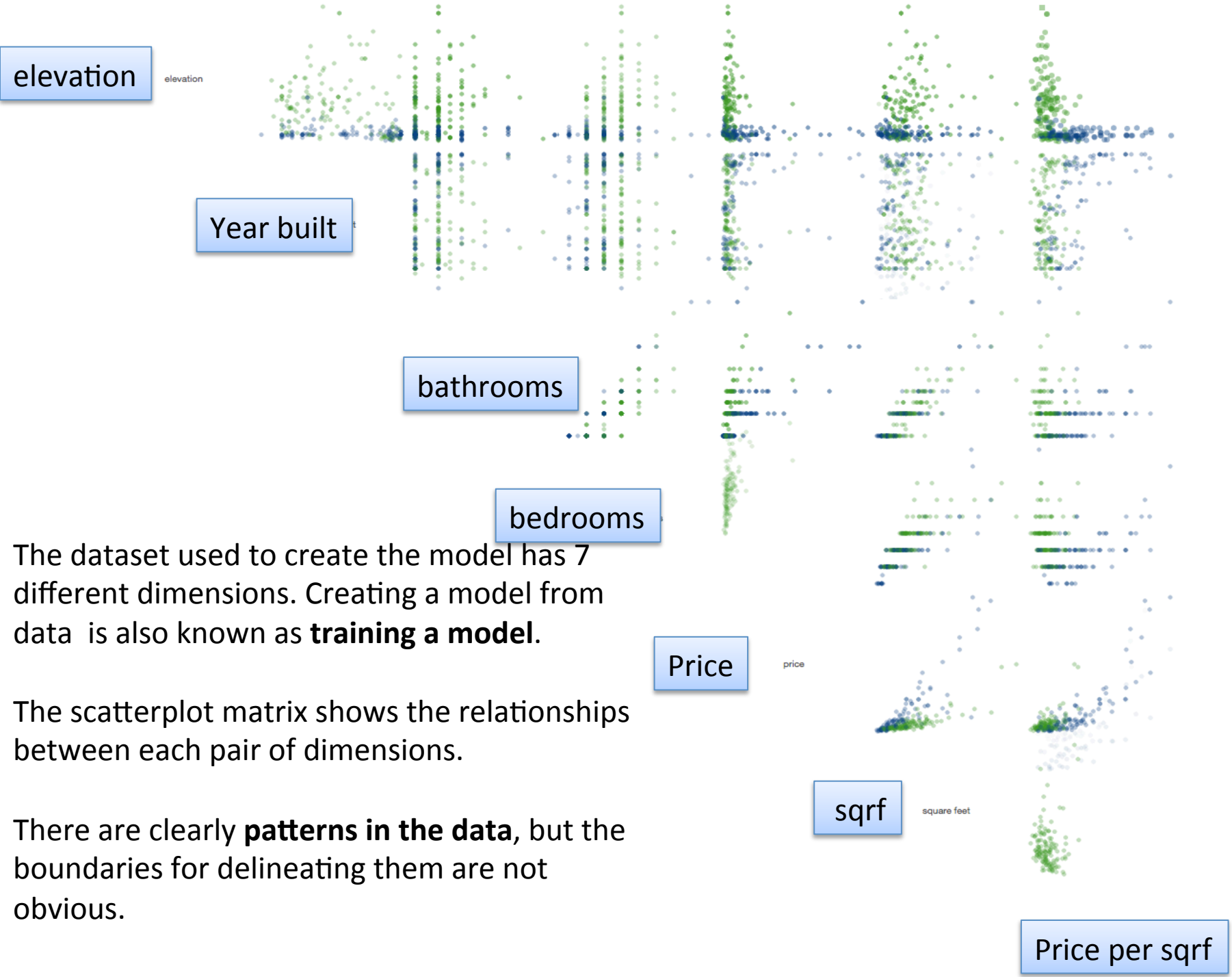




Identifying boundaries in data using math is the essence of statistical learning.

Of course, you'll need additional information to distinguish homes with lower elevations and lower per-square-foot prices.

We still have uncertainty concerning cheap and low homes



The dataset used to create the model has 7 different dimensions. Creating a model from data is also known as **training a model**.

The scatterplot matrix shows the relationships between each pair of dimensions.

There are clearly **patterns in the data**, but the boundaries for delineating them are not obvious.

So, machine learning is…

- Finding patterns in data
- Machine learning methods use learning algorithms to identify regularities and boundaries
- “Boundaries” may come in several forms, they can be probabilities, separating lines, or rules (if height >73 then SF)

Course material

- Slides (partly) from: http://www.cs.utexas.edu/users/mooney/cs391L/
- Textbook: Tom Mitchell, Machine Learning, McGraw Hill, 1997.
- Introduction to machine learning ETHEM ALPAYDIN (on line book)
- Course twiki http://twiki.di.uniroma1.it/twiki/view/ApprAuto

Course Syllabus

1. Concept Learning and the General-to-Specific Ordering
2. Decision Tree Learning
3. Ensamble Methods
4. Artificial Neural Networks and Deep learning (convolutional and denoising)
5. Support Vector Machines
6. Probabilistic Learning: Naive Bayes
7. Evaluation methods: experimental and theoretical methods
8. Unsupervised learning:
    1. Data mining (Apriori, FP-Growth)
    2. Clustering methods
9. Reinforcement learning: Q-learning and Genetic Algorithms

Algorithms experimented on Weka toolkit
http://www.cs.waikato.ac.nz/ml/weka/ AND (from november) on
http://scikit-learn.org/stable/ (a python platform)

Bring your PC with Weka package installed, later with sckit (will start sckit on November)

Exam

- Written exam on course material (60%)
- + Course Project: to be defined (see on the course web page examples of previous years and evaluation criteria) (40%)
- Projects can be carried on by teams of 2 students
- A Business Intelligence project with students of the Master in Management can be defined for those who are interested

Issues in Machine Learning

So, what is Machine Learning?

What is Machine Learning?

- Machine Learning is a scientific discipline that addresses the following question: '*How can we program systems to automatically learn and to improve with experience*? '
- **What** is learning?
- **How** do we learn?
- **What** can we learn?
- **How** can we "improve", and over what??
- **What** is "experience"??

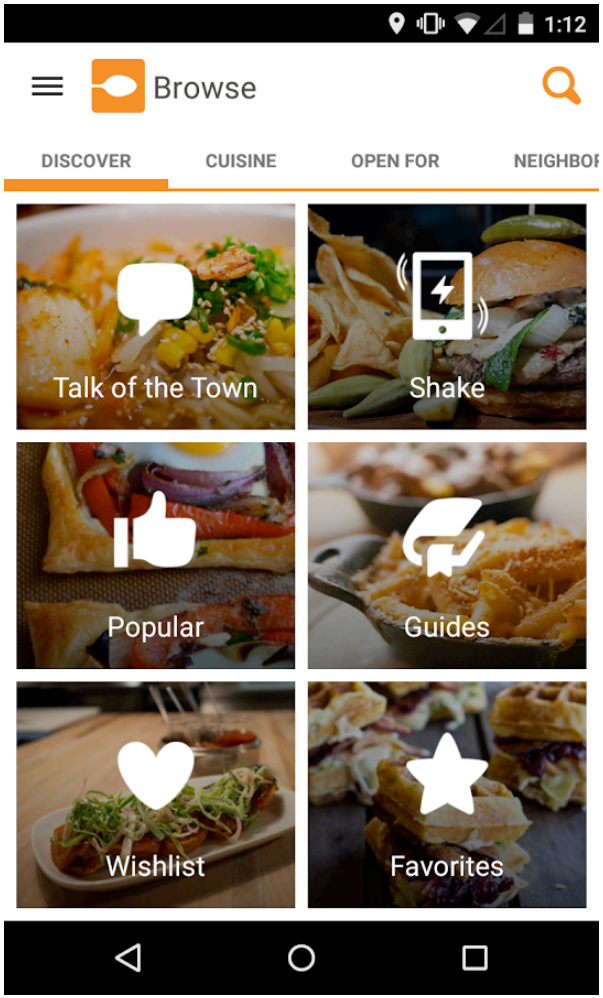
What is learning??

Fire burns!

But we learned using it

you can study (learn) Machine learning

An then build an app to learn your favorite food and advise you on restaurants



What is learning?

- **Make sense** of a subject, event or feeling by interpreting it into our own words or actions.
- **Use** our newly acquired ability or knowledge in conjunction with skills and understanding we already possess **to do something** with the new knowledge or skill and take ownership of it.

- http://www.skillsyouneed.com/general/learning.html#ixzz3QtQQTKvg

What is learning?

UNDERSTAND + GAIN KNOWLEDGE + USE NEW KNOWLEDGE TO DO SOMETHING

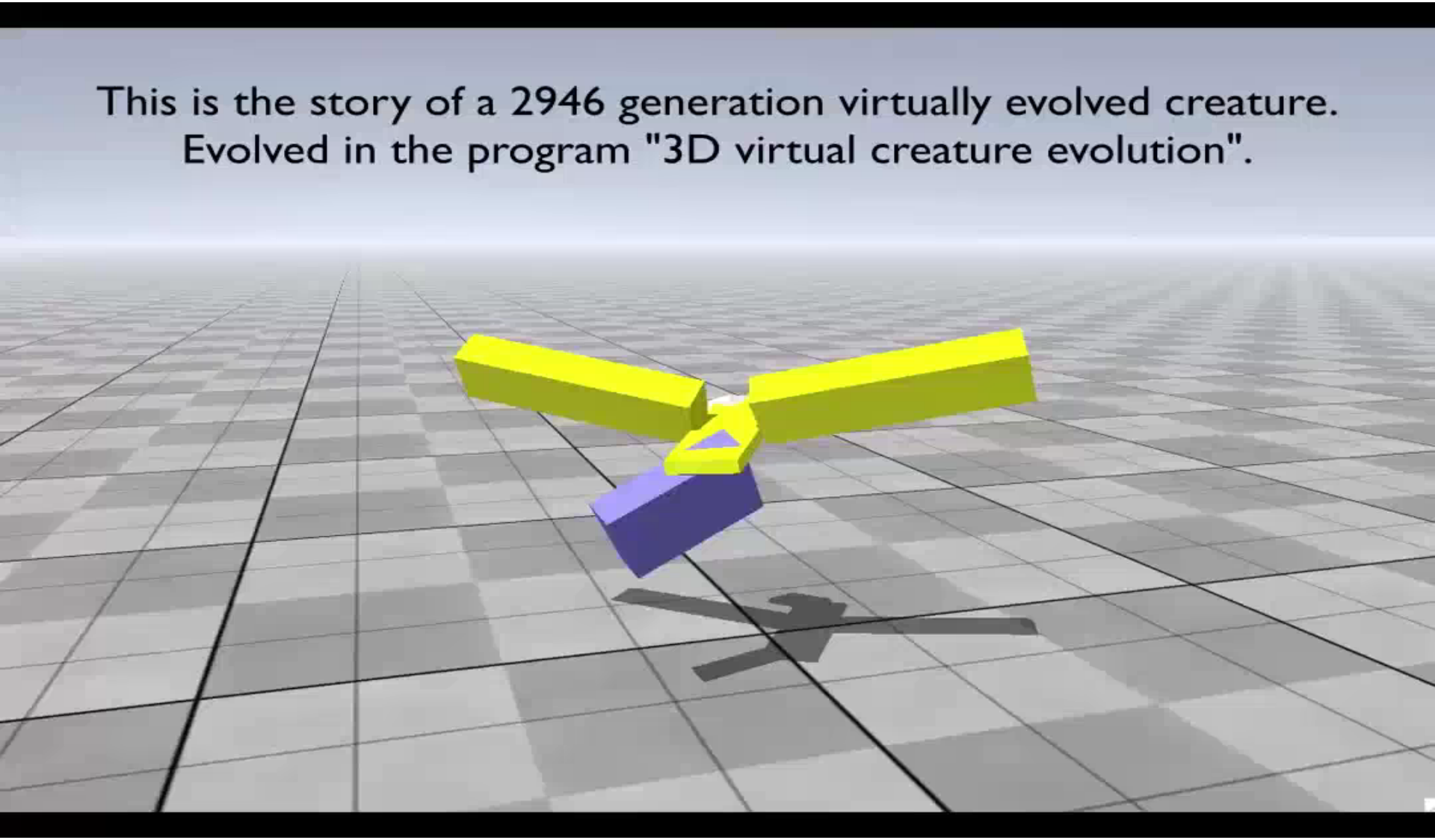
How do we learn??

How do we learn?

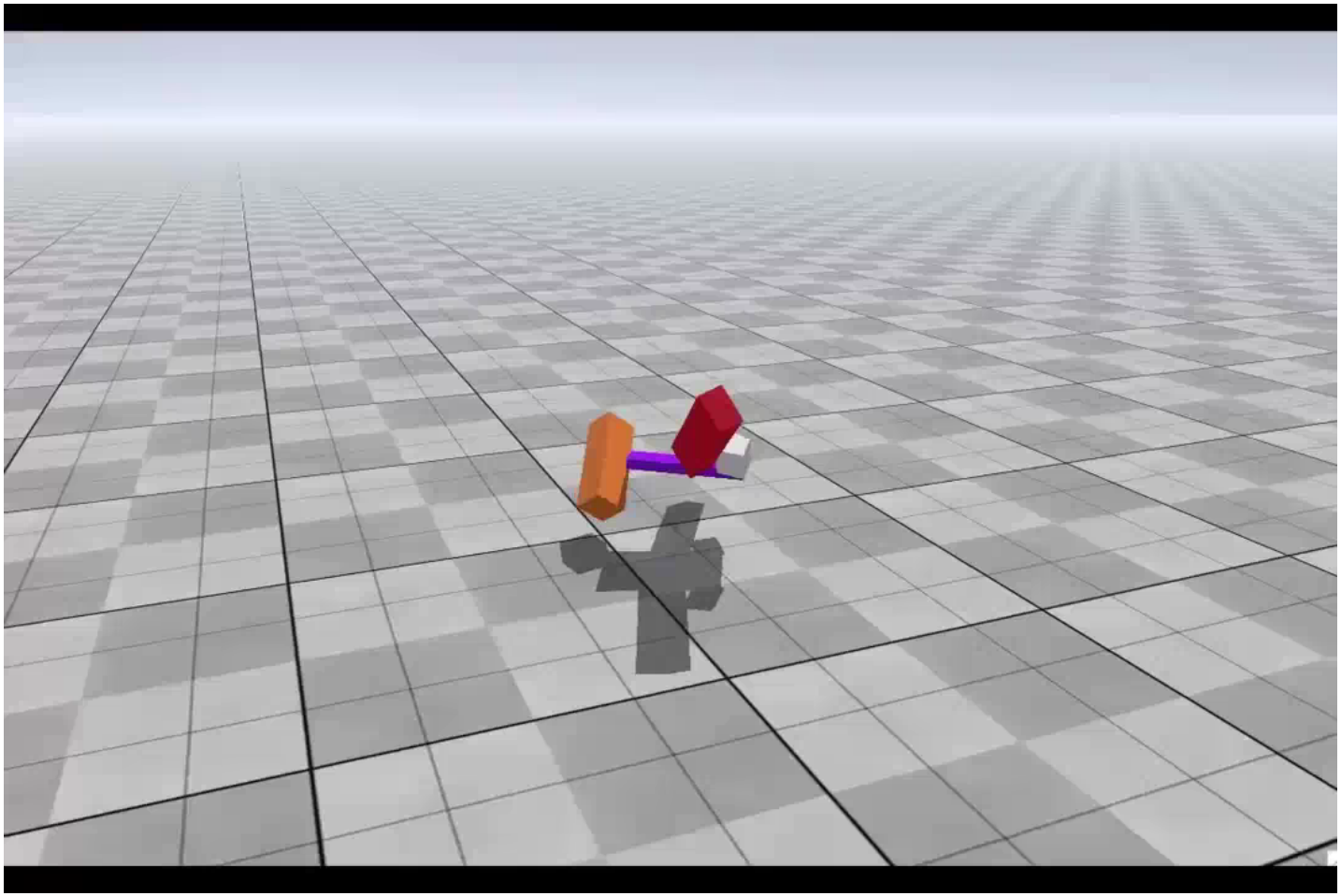
- Someone tell us (teacher, or whatching others) ..as in the ML example
- Try and test (learning by doing) .. As in the fire example

Is there something we cannot learn??

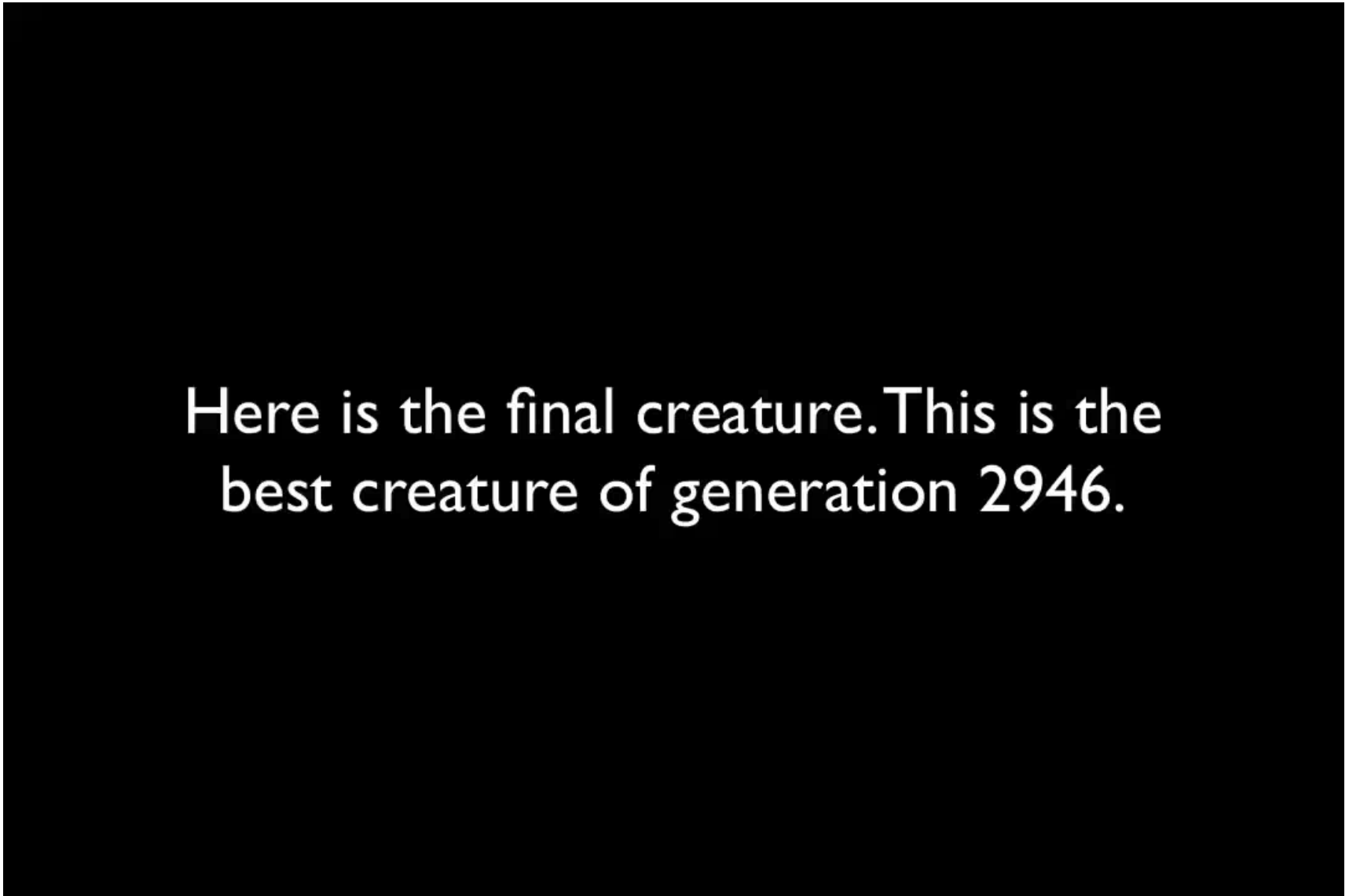
Machines that learn to fly



.. After some time..



After some more time..



Besides things that we cannot learn, there are others that are either..

- Difficult to learn
- Difficult to teach

When is it difficult for humans to learn?

- 1. If there are **too many data**, humans cannot easily make sense of them (e.g. finding regularities in human genoma, learning to recognize one among millions of objects, market analysis and forecasts)

When is it difficult for humans to learn?

- 2. If data **change too frequently**, humans might be unable to continuously adapt their knowledge (e.g. personalized recommendations, market analysis forecast)

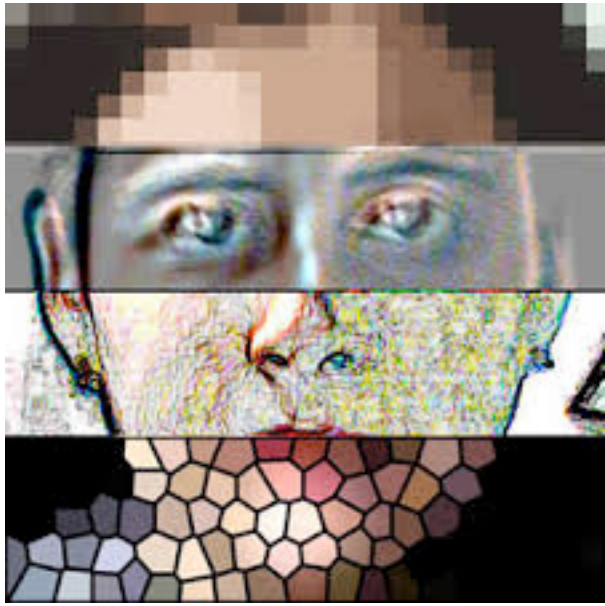
Stock market values
And quotes

When is it difficult for humans to learn?

- 3. If the environment is dangerous, “learning by doing” cannot be applied (e.g. rescue systems)

When is it difficult for humans to teach?

- 4. If there is not enough information or previous expertise to “understand and gain knowledge” (we actually do not understand the image and speech recognition process by humans – it is not “teachable”)



So when is it advisable to use machine LEARNING?

- ML is used when:
  - Human expertise does not exist (navigating on Mars),
  - Humans are unable to explain their expertise (speech/image recognition)
  - Solution changes in time (market data for market forecast)
  - Solution needs to be adapted to particular cases (personalized systems for recommendation, diagnosis, etc.)

So when is it advisable to use Machine Learning?

- Develop systems that are too difficult/expensive to construct manually because they require specific detailed skills or knowledge tuned to a specific task (***knowledge engineering bottleneck***).
  - Expert systems
- Develop systems that can automatically adapt and **customize** themselves to individual users.
  - Personalized news or mail filter
  - Personalized tutoring
  - Recommenders
- Discover new knowledge from large databases (***data mining***).
  - Market basket analysis (e.g. diapers and beer)
  - Medical text mining (e.g. migraines to calcium channel blockers to magnesium)
  - Twitter mining

Related Disciplines

- Artificial Intelligence
- Data Mining
- Probability and Statistics
- Information theory
- Numerical optimization
- Computational complexity theory
- Control theory (adaptive)
- Psychology (developmental, cognitive)
- Neurobiology
- Linguistics
- Philosophy

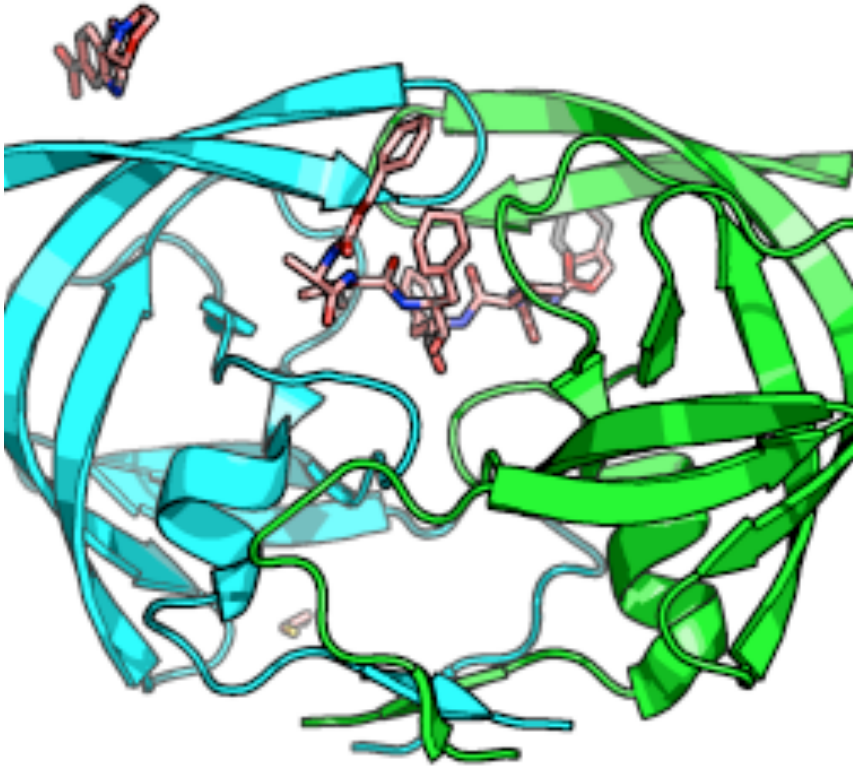
ML is perhaps the most interdisciplinar of CS areas!!

Some “real hot” ML applications

- It is really hard to find a problem where machine learning is not already applied -- machine learning is practically everywhere, in business applications and science!

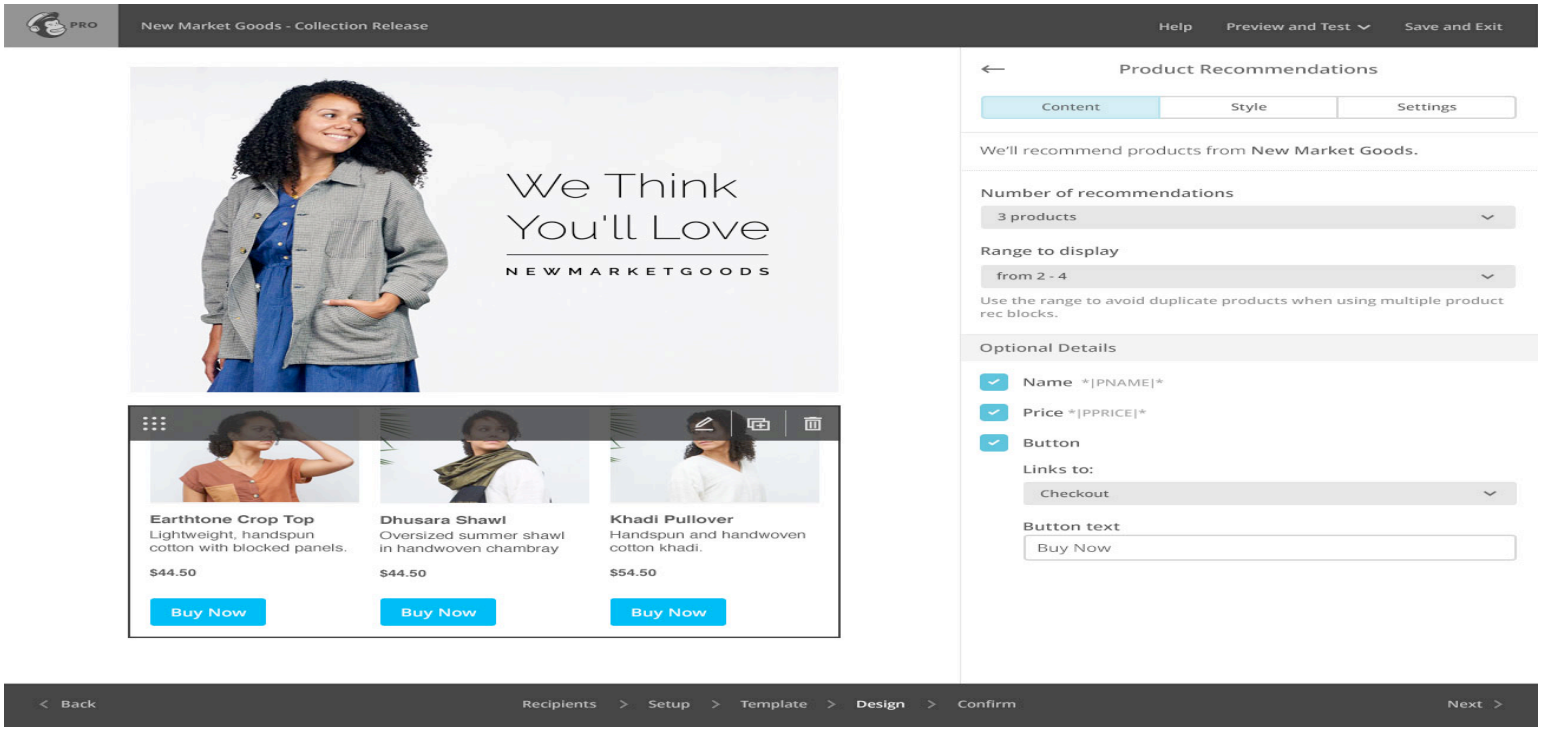
- Here is a list of common applications:

Computational Biology & Drug Discovery/Design



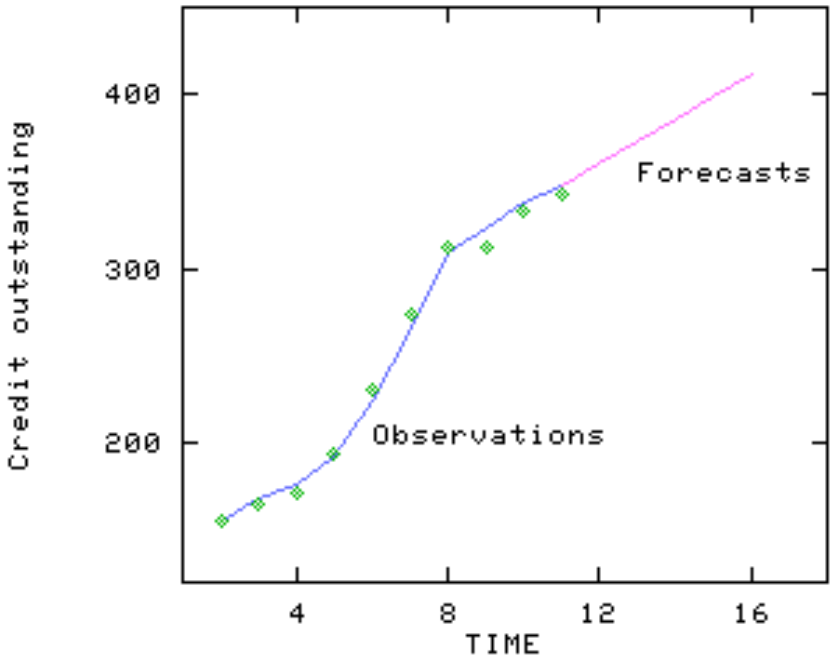
- screening large molecule databases and identify which (drug-like) molecules are likely binding to a particular receptor protein
- predict the potency of a receptor agonist or antagonist
- Tarca, Adi L., et al. "Machine learning and its applications to biology." *PLoS Comput Biol* 3.6 (2007): e116.

Web Search and Recommendation Engines



- find relevant searches, predict which results are most relevant to us, return a ranked output (Google)
- recommend similar products (e.g., Netflix, Amazon, etc.)

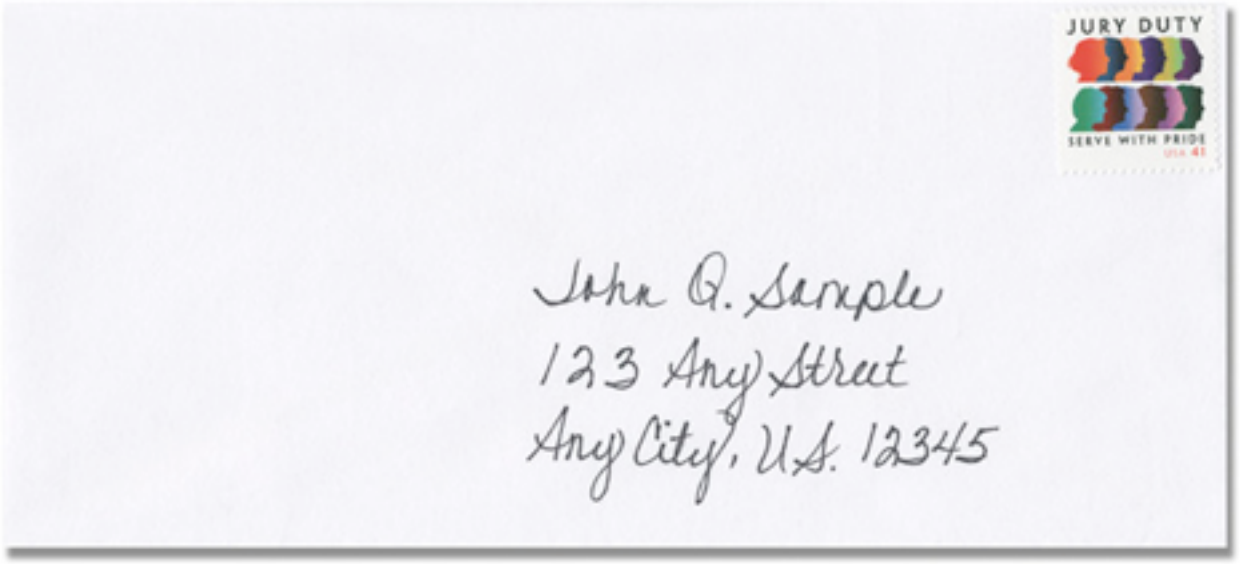
Finance



- predict if an applicant is credit-worthy
- detect credit card fraud
- find promising trends on the stock market

Text and Speech Recognition

- handwritten digit and letter recognition at the post office
- voice assistants (Siri)
- language translation services

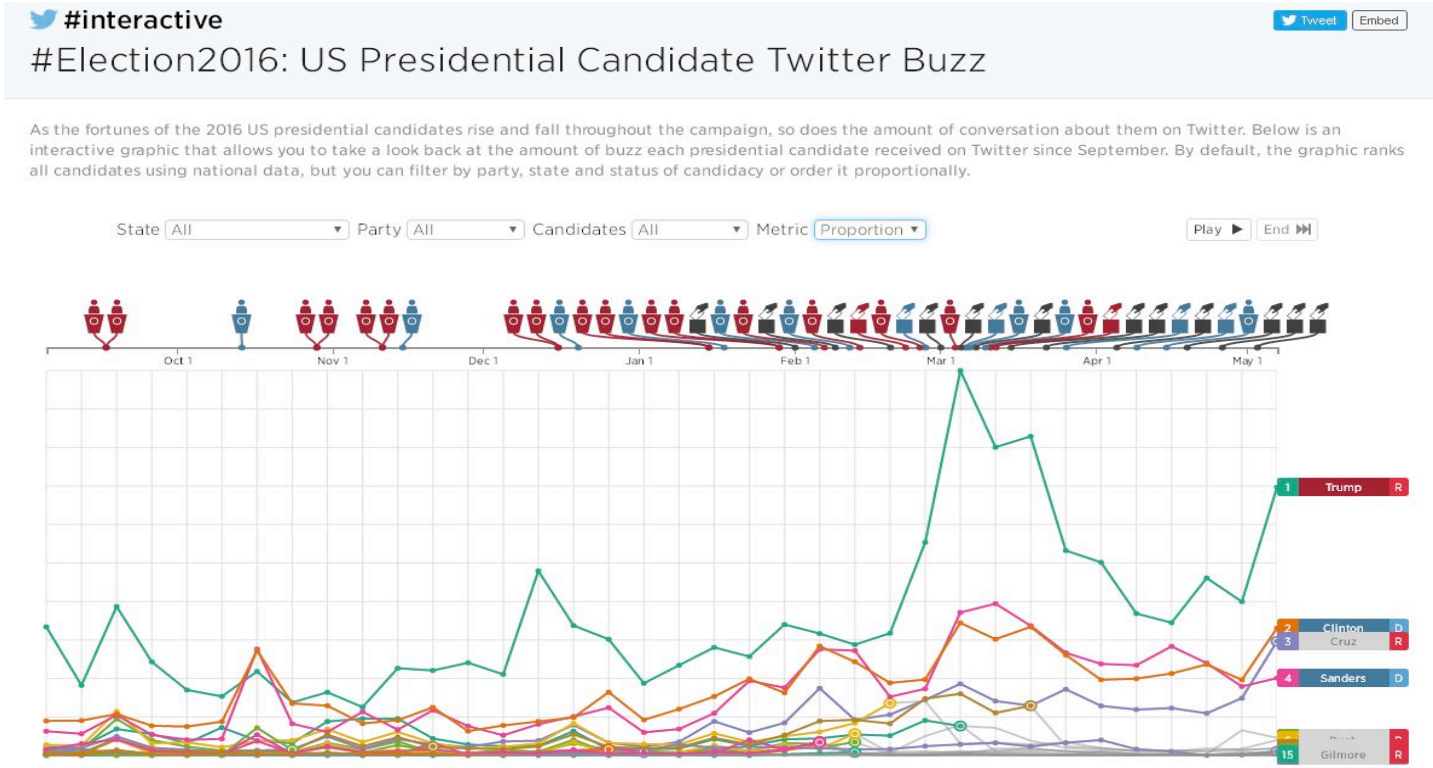


Space, Astronomy, and Robotics

- autonomous Mars robots
- indentifcation of relevant information (objects) in large amounts of Astronomy data

Social Networks and Advertisement

- Social data mining
- data mining of personal information
- selecting relevant ads to show



HOW DO WE BUILD A ML APPLICATION?

An example (simpler!)

The problem: **Crops Contamination by Grass Grubs in New Zeland**

A ML-2014 student project by Sara De Cristofano and Emanuele Giarlini

Problem description

- The grass grubs are one of the most common and voracious species of insects of New Zealand, causing substantial crop damage and significant economic losses for farmers.
- Can we design a program that helps preventing the development of grass grubs? (similar task: can we predict the diffusion of xylella in olive trees in Puglia?)

Why is this “machine learning?”

- The basic answer is: **We don’t know the solution to this problem**;
- Specifically, the problem (**growth of grass grubs**) depends upon a number of possible influencing factors, both due to the characteristics of these insects (e.g. **what they eat**) and to external variable conditions (e.g. **climatic conditions**), but we don’t know which factors are truly relevant and what is their mutual relation
- Therefore, a second answer is: **there is NO standard solution (standard= something that applies in all conditions, like: “if winter average temperature goes below x, then you get grass grubs”)**

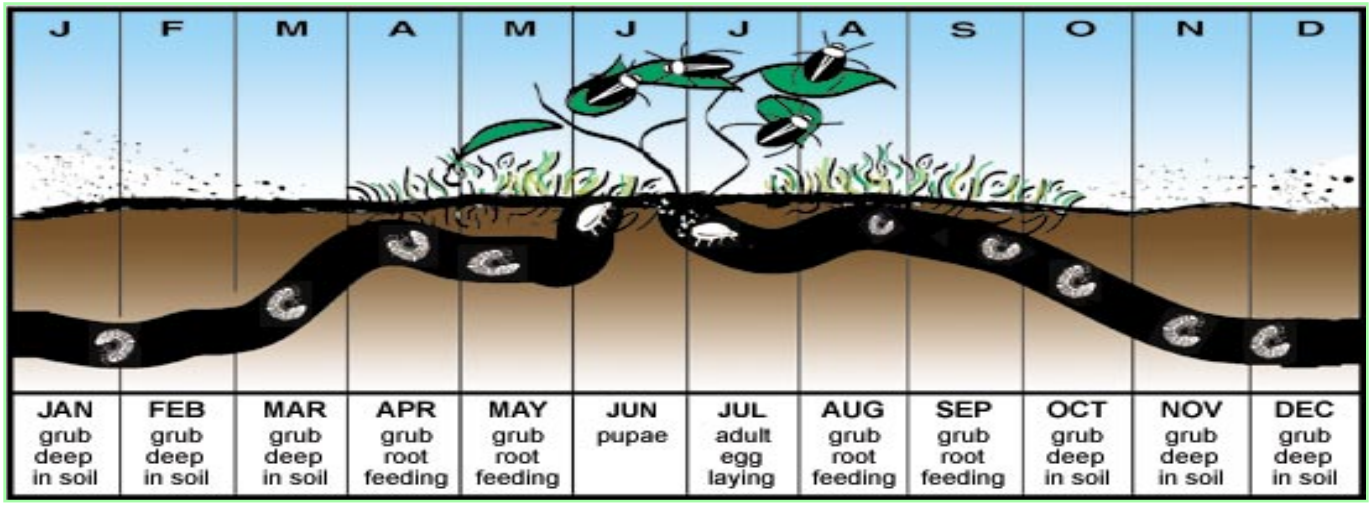
So when are ML algorithms needed?

- When the relationships between all system variables is not completely understood!
- This is the case for almost any real system, but often there is one “preferred” solution that works almost ever.
- So, we add some constraints:
  - No domain experts are available, or they are rare and expensive;
  - We have (or we can obtain) sufficiently large data concerning the problem
  - Of course: the problem is either socially or economically RELEVANT

Back to our grass grubs

- Specifically, the problem (growth of grass grubs) depends upon a number of **possible influencing factors**, both due to the characteristics of these insects (e.g. what they eat) and to external variable conditions (e.g. climatic conditions), but we don't know which factors are truly relevant to determine the **observed outcomes**, and **what is their mutual relation**
- In other words, we need to:
  - Identify the influencing factors (input variables) that describe the problem (as we said, we call these **FEATURES**: features are the variables describing the objects/events we deal with)
  - Identify the set of possible effects, or outcomes, of the problem we are observing (we call these **output variables**, or **output categories, classifications** – if discrete-)
  - Learn the relations between input and output variables, e.g. our MODEL ***output=f(input)***

So what we know about grass grubs? (Remember: step 1 is “gaining knowledge”)



- Our students collected quite a bit of info and wrote to specialists in New Zeland to learn that:
  - Damage to crops is concentrated in specific periods
  - They eat specific kinds of leaves and roots
  - If climatic factors are favorable, they stay in the same area for several years

Selected features

- After studying the problem, the following set of input/output discrete features have been selected:

Feature	values
Rainfall	High, medium, low
Type of crop	Fruit, tobacco, cereals, legumes
Soil type	Coast, hills, plains
Preventive phosphorous treatment	Yes, no
Damage (**output**)	High, Medium, Low

Summary so far

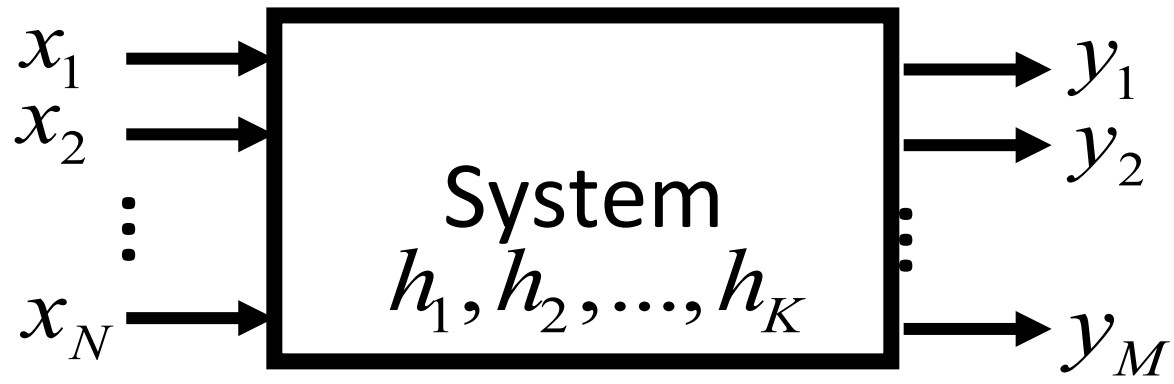
Given a problem that "fits" the area of Machine Learning:

1. **Analyze the problem to identify input and output variables**

PROBLEMS:

- possibly there are hidden variables we are unable to capture, possibly we consider variables that are not relevant
- If variables are real-valued, we might need to discretize, to reduce data variability (and complexity of the learning problem). This is known as FILTERING. Will see this on WEKA.

A Generic ML System



Input Variables: $\mathbf{x} = (x_1, x_2, \ldots, x_N)$

Hidden Variables: $\mathbf{h} = (h_1, h_2, \ldots, h_K)$

Output Variables: $\mathbf{y} = (y_1, y_2, \ldots, y_K)$

Learn: ***y=f(x,h)***

In our example

- For any ground X located in a given area, given known values of rainfall, crop , soil type, and preventive treatment, what is the predicted damage (possibly,with and without preventive treatment)?

$$y = Damage(x) = f(rainfall, crop, soil, treatment)$$

Need to learn f(x)

Our output function is “Damage” (still did not decided the “shape” and co-domain of this function)

Our input data is a quadruple X: (rainfall, crop, soil, treatment). This means that any specific ground X (of which we wish to predict the risk of future damage) is represented by 4 features **values**.

Summary so far

1. Analyze the problem to identify input and output variables

Given domain variables, need to learn *y=f(x)* (y, x are sets of variables, or **vectors**)

2. What kind of classification function for *f*? (e.g. logic function, probabilistic, algebraic..)

Examples:

(*Logic function*)

IF Crop= **tobacco**, Soil= **cost**, Treatment = **NO**, Rainfall = **average** → **damage= HIGH**

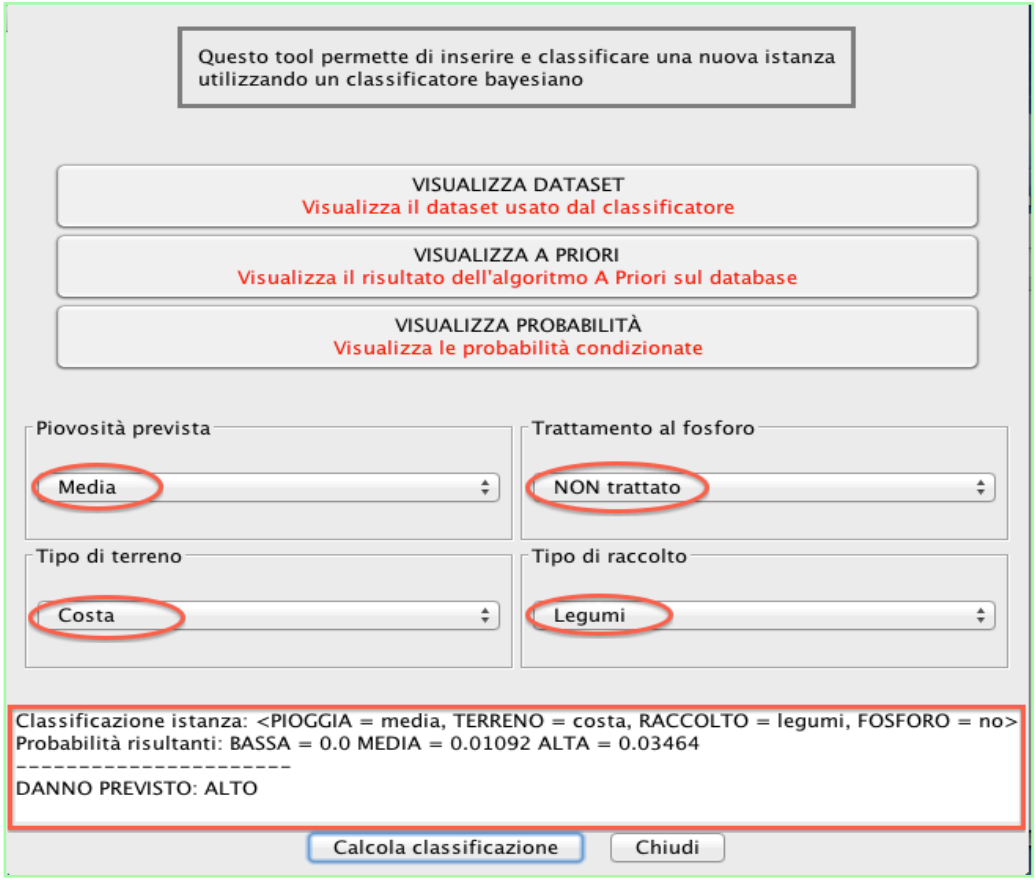
(*Probabilistic*)

Argmax $_{\text{X=(high,medium,low)}}$ (Prob(**damage= x/**Crop= **tobacco**, Soil= **cost**, Treatment = **NO**, Rainfall = **average)**

(*Algebraic*)

$$damage(x) = w1(soil) + w2(crop) + w3(rainfall) - w4(treatment)$$

Our students decided to learn a probability (and discretize output values)



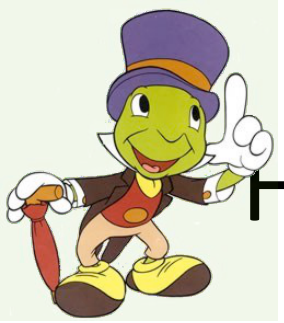
Given input values of RAINFALL, SOIL, CROP and TREATMENT, system forecasts a HIGH/MEDIUM/LOW probability of damage

Learning a function

- Given the "category" of the function to be learned (boolean, conditional probability, polinomial, exponential..) identify a **method** (=a machine learning algorithm) to learn the function y=f(x)
- E.g. if a boolean function, a specific set set of rules, if a linear function, specific values of the coefficients w_i..
- Note that a learned function is an APPROXIMATION (an "hypothesis") of the "real" function –usually impossible to perfectly learn the input/output relation
- This means that, like for humans, f(x) will not be always right in its prediction.. therefore we also have a problem of EVALUATION (*hypothesis testing*)

Learning a function= a class of ML algorithms

- Machine learning algorithms can (also) be classified according to the objective function f(x) to be learned
  - **Boolean/logic** (rule-based) methods: Decision trees, Apriori, etc.
  - **Probabilistic**: Naive Bayes, Bayesian networks,..
  - **Algebraic**: Support Vector Machines, Neural networks..



How do we learn our target function?

Basically, 3 approaches (like for humans!!):

- Supervised Learning (we have **examples** from past history, system learns the function parameters from examples) *Like when you read a handbook or attend lessons from a teacher*
- Reinforcement Learning (there is a teacher able to "**reward**" correct choices/answers and "punish" wrong choices) *Like what you do to train animals..*
- Unsupervised Learning (no examples, no teachers: system only has some "quality" parameter to evaluate the goodness of a solution) *Like when you learn from **experience**: your objective is to maximize your comfort level, or to minimize your risk.*

What about our grass grubs?

- "Available data" means that, for a number of cases x_i, we know: $y_i=f(x_i)$ For example we should have a table like this:

feature

f(x)

SOIL	CROP	RAINFALL	TREATMENT	DAMAGE
HILL	CEREALS	HIGH	YES	MEDIUM
COAST	FRUIT	MEDIUM	NO	LOW
…	…..	…	…	

value

"y=DAMAGE (x)" is an observed output (available from the past), given input values.

damage(x): **SOILxCROPxRAINFALLxDAMAGE➔(HIGH, MEDIUM, LOW)**

..unfortunately

- Our students didn't select an easy problem. No data with "ground truth" available from which to learn!!
- Therefore, they selected an UNSUPERVISED algorithm (Apriori, later in this course) that is able to infer from unclassified data
- But where the data comes from??

No general strategy to create datasets..

- This is just a “good” example on how to:

1. How many data are necessary?? This depends on the dimension of the space of feature values: 4 features, with 3,4,3, and 2 values respectively, therefore possible combinations are 3x4x3x2=72; and 3 outcomes (high, medium low damage) – You will see how, but a theorem (computational learning theory) tells us we need at least 794 examples to learn a sufficiently reliable model

.. A good example (cont'd)

- How to generate a "synthetic" dataset??
  - Selected 3 regions in New Zeland (Cromwell, Wellington, Winton)
  - From National Climate database (available on line), collected climatic data for the 3 regions during past 30 years
  - Using retrieved publications, select the most common crops attaked by grass grubs, and information on pesticides
  - Using all these data, they generated a synthetic dataset with combinations of values and they also computed (with Apriori algorithm) the most probable combinations

Example

Dataset

#	Anno	Piovosità	Zona	Raccolto	Fosforo	Danno
0	Anno	Piovosità	Zona	Raccolto	Fosforo	Danno
1	1980	media	costa	cereali	si	medio
2	1980	media	costa	frutta	no	alto
3	1980	media	costa	tabacco	no	alto
4	1980	media	costa	legumi	si	medio
5	1980	media	collina	cereali	si	medio
6	1980	media	collina	frutta	si	medio
7	1980	media	collina	tabacco	no	alto
8	1980	media	collina	legumi	no	alto
9	1980	media	pianura	cereali	si	alto
10	1980	media	pianura	frutta	si	medio
11	1980	media	pianura	tabacco	si	medio
12	1980	media	pianura	legumi	si	alto

Summary so far

- **Analyze the problem to identify input (X) and output (Y) variables**
- **Select a learning function type *Y=f(X)***
- **Select (or invent) an algorithm to learn f(x)**
- **Depending upon the problem being analyzed, find an existing dataset, or create a (realistic) dataset artificially, or establish a strategy to "guide" the algorithm on how to learn f(x)**
- **Now last step: EVALUATION**

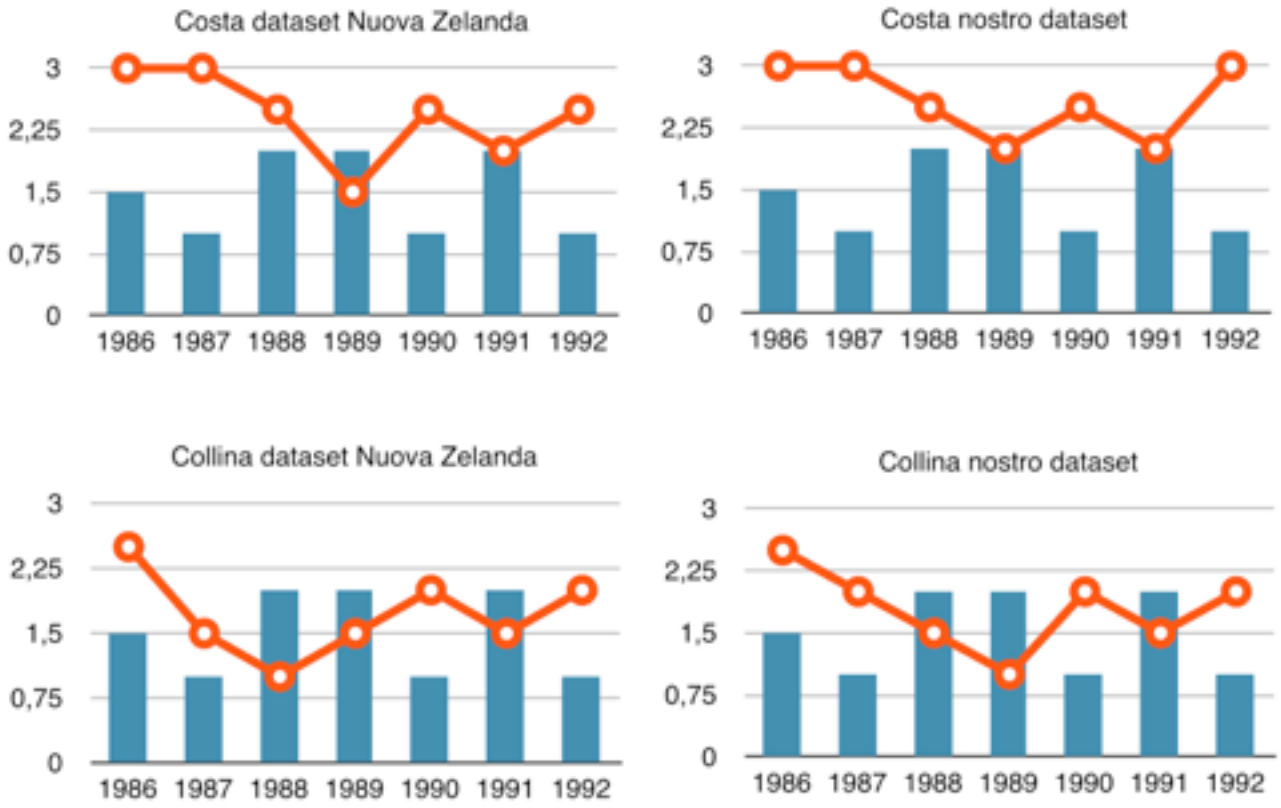
Evaluation

- Basically, we need to verify that the quality of system predictions (in our case, the damage prediction) on UNSEEN data is “good enough”
- What “good enough” means will see during the course!
- In any case, we need to have some reference data to compare with (or someone that can judge)

What our students did?

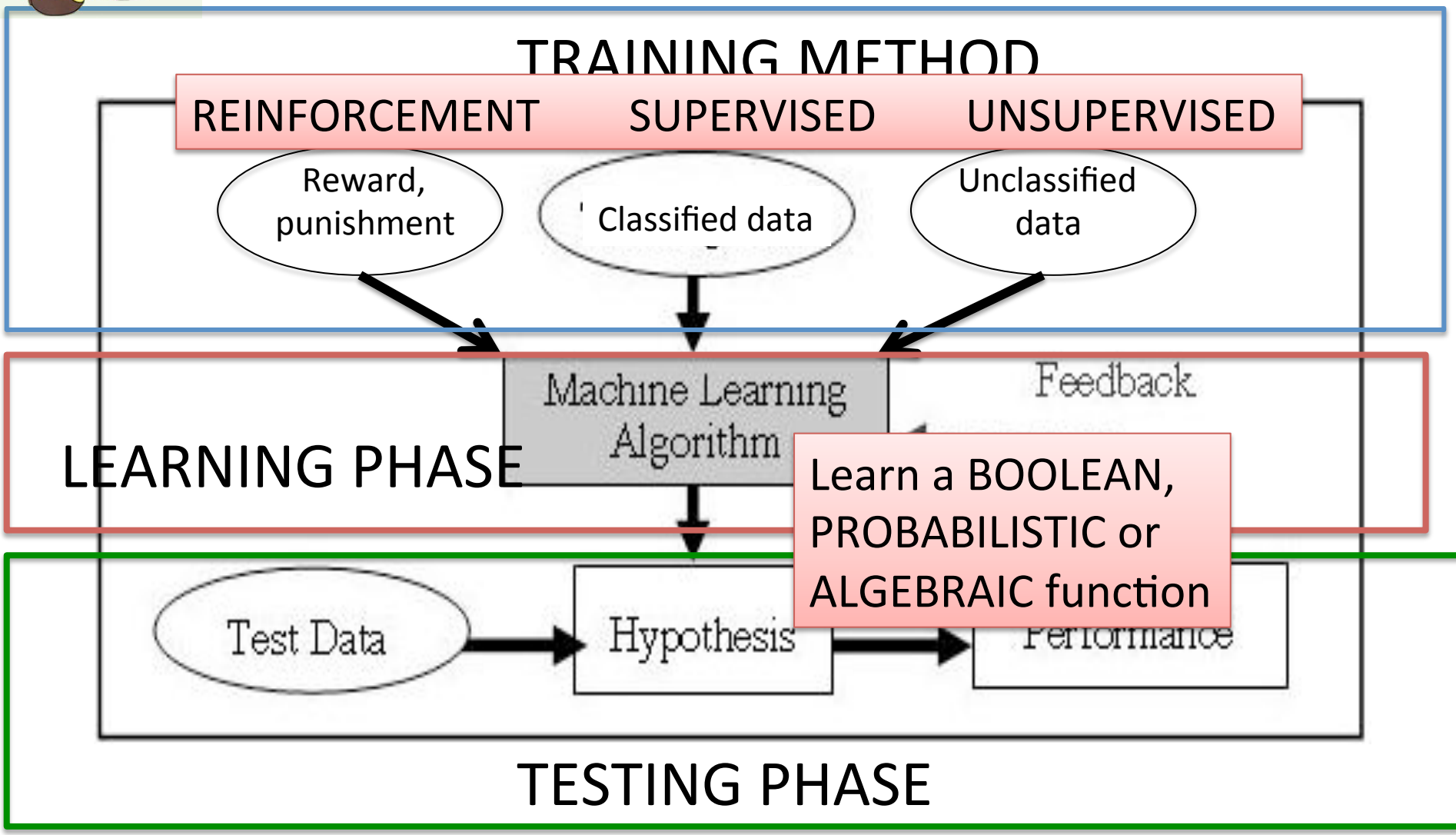
- Not readily available data (e.g. “ground truth”) for evaluation
- However a number of New Zeland universities did some study
- So they sent and email to these universities, received papers and some data (other researchers forecasts)
- One University sent them data on reported damages of grass grubs from 1986 to 1992 (bingo!! ☺)

Results



Bars are reported damages, red line forecasted by student's method (not perfect, but not too bad)

Final Summary



Please read on the web course site examples of “good” student projects, and how they are Evaluated

Much of the difficulty is to analyze the problem, identify good features, create/clean/integrate the dataset, experiment with different algorithms with different parameters, evaluate results.

Most algorithms have good available implementations, so the difficulty is NOT there..