

Machine Learning for all

Session 1. Introduction to Machine Learning

César E. Montiel-Olea (SPD)

January 25th, 2019



Topics for today

- What is Machine Learning?
- Concept of “Learning” Task
- Supervised Machine Learning: Classification and Prediction
- Unsupervised Machine Learning: Dimension Reduction

Machine Learning is not a new concept

- First ML algorithm was developed in 1957 and was created to solve “classification” problems



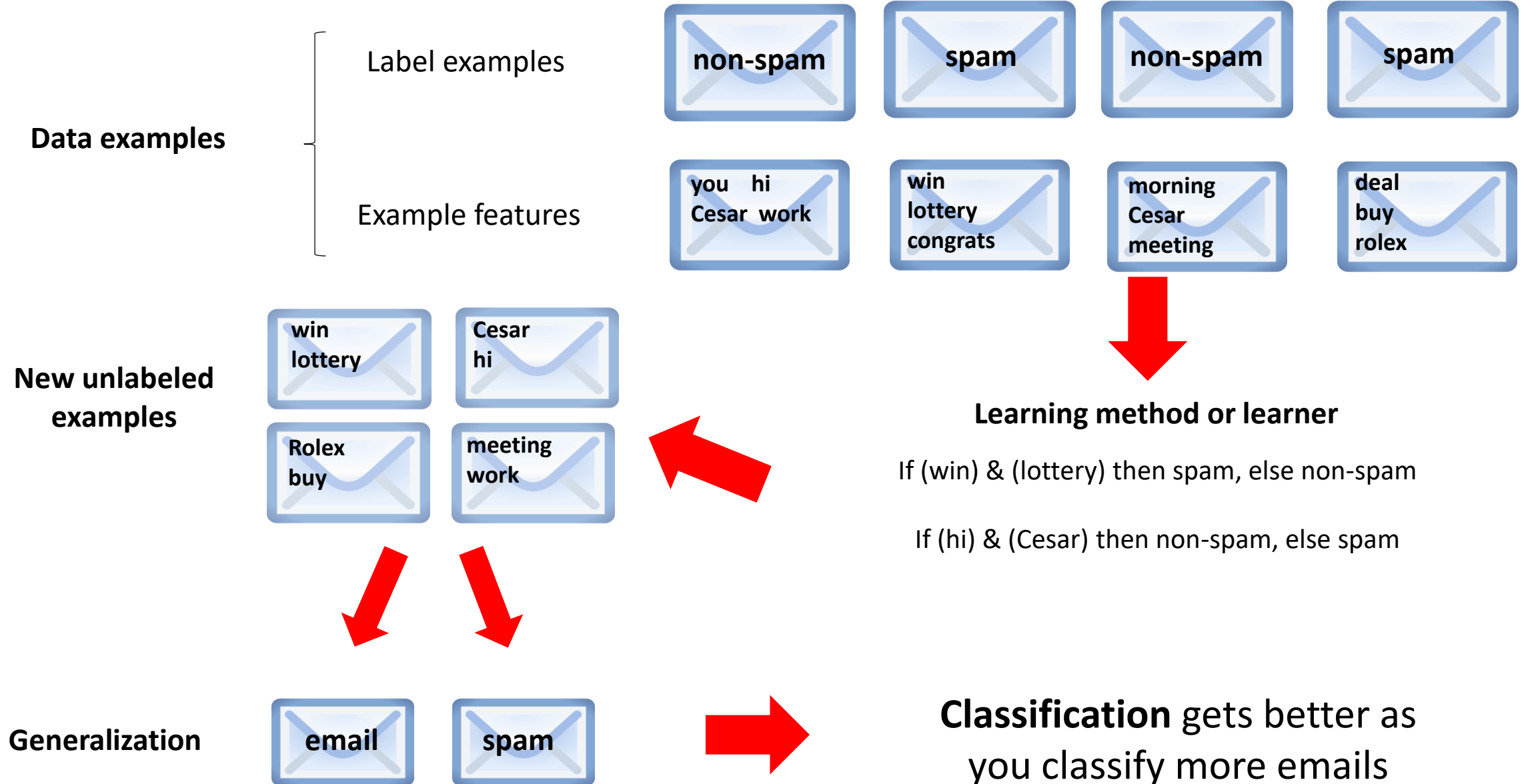
- During the past decades, ML has evolved due to the development of two things: more computational capacity and more availability of data.

What is Machine Learning?

According to Schapire & Freund (2012):

- Machine learning studies computer methods for learning to do tasks; the learning that is being done is always based on data.
- **Definition:** Machine learning is about learning to do better in the future based on what was experienced in the past.
- **Learning Task:** Spam filtering, face detection, medical diagnosis, fraud detection.

“Learning” task: How can a machine filter spam?



What is Machine Learning?

A computer program is said to learn experience **E** with respect to some task **T** and some performance measure **P**, if its performance on T, as measured by P, improves with experience E (Mitchell, 1998)

T: Classifying email as non-spam or spam

E: Watching humans labeling emails as non-spam or spam

P: Fraction of emails correctly classified

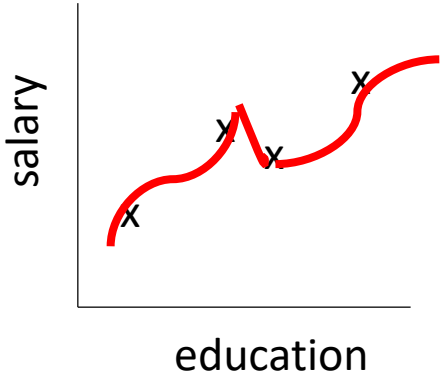
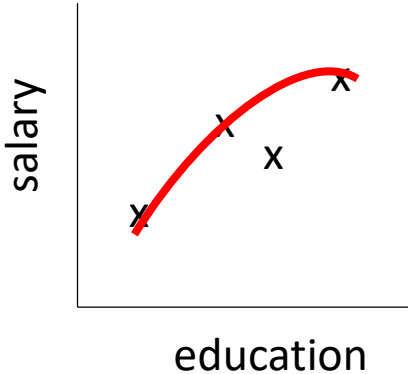
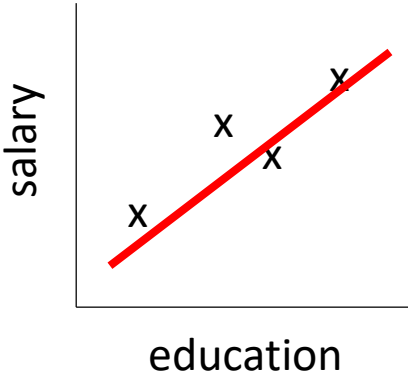
“Learning” Task: How can we predict salaries?

Data examples

Salary (K)	Years of education	Years of experience	Location	Grad school	Gender
\$90	10	2	DC	Yes	Male
\$75	4	3	Maryland	Yes	Female
\$115	6	5	DC	No	Female
\$220	14	3	NY	No	Male

p = 5
n = 4

Learning method



Generalization

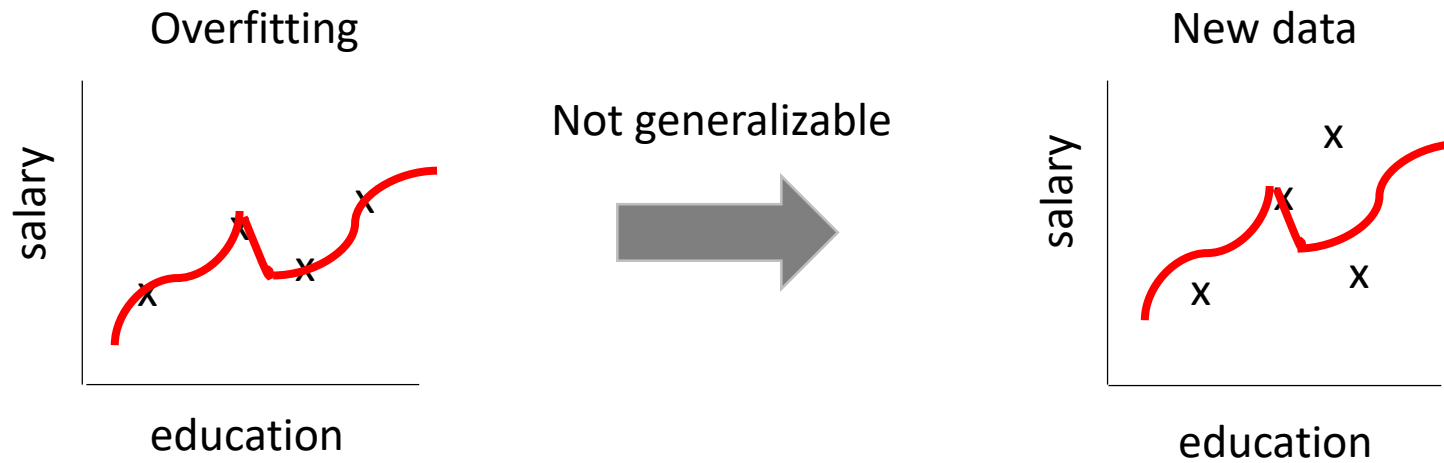


Predicting salaries for a set of new people given their characteristics

Popular set-ups for Machine Learning

Prediction with more covariates than observations

Linear model (OLS) is not suitable \rightarrow unstable out-of-sample predictions \rightarrow Overfitting

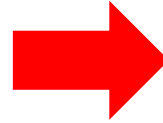


Overfitting: Our learning method fits our data very well but fails to generalize to new data

Popular set-ups for Machine Learning

How do we address overfitting and predict an outcome better?

Choose variables manually



By hand?!

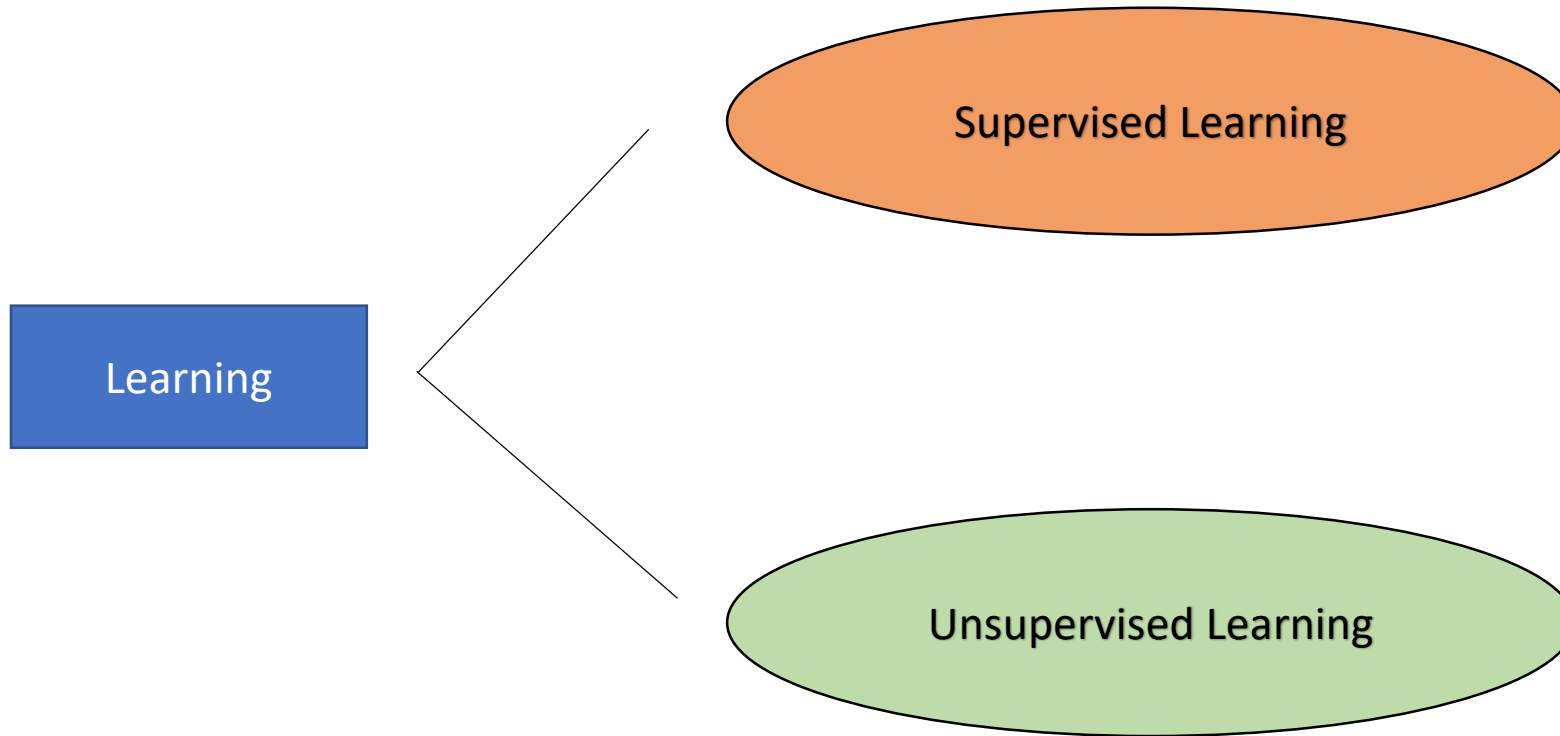


Good luck!



Or we can use ML!

Classes of learning algorithms

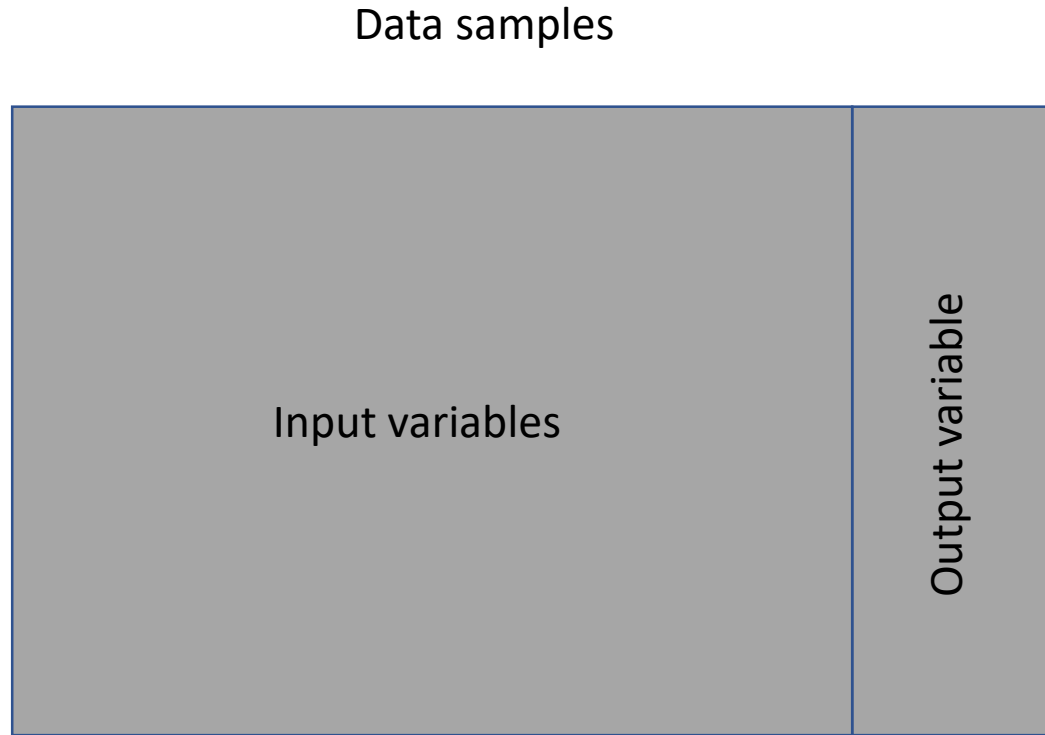


A few precisions on Supervised ML

Following Mullainathan and Spiess (2017):

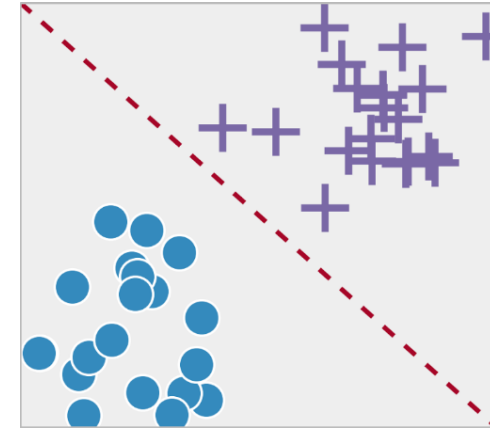
	Econometrics	Supervised ML
Goal	Parameter estimation	Produce out-of-sample predictions
Appeal	Understanding the relationship between y and x 's	Discovering complex structures in the data that can be generalized
Techniques	OLS, instrumental variables, sieves, kernels	LASSO, Ridge, Supporting Vector Machines, Random Forests, Neural Networks

Supervised ML: Two Learning Problems



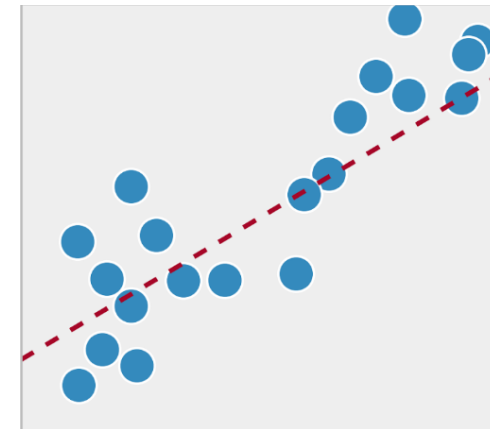
If qualitative → **classification** problem

Spam
problem



If quantitative → **regression** problem

Predict
salaries



Supervised Learning workflow



Randomly split the data

Train a ML algorithm (fit the model)

Do not touch



If there are tuning parameters, evaluate different algorithms

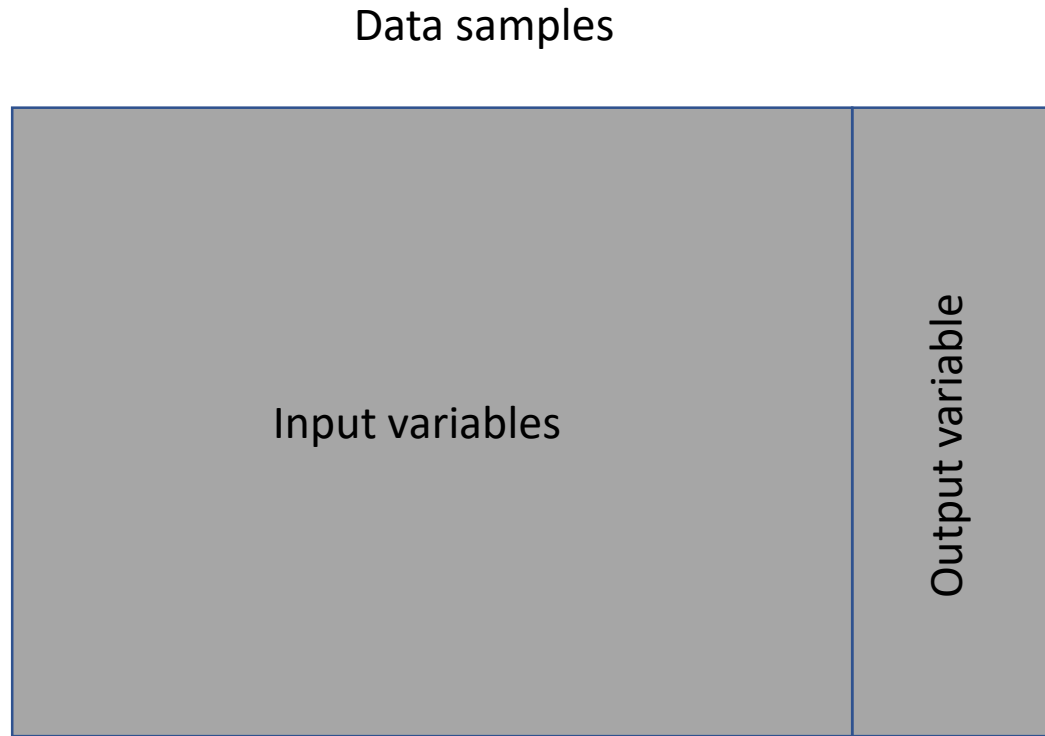
Tune, select



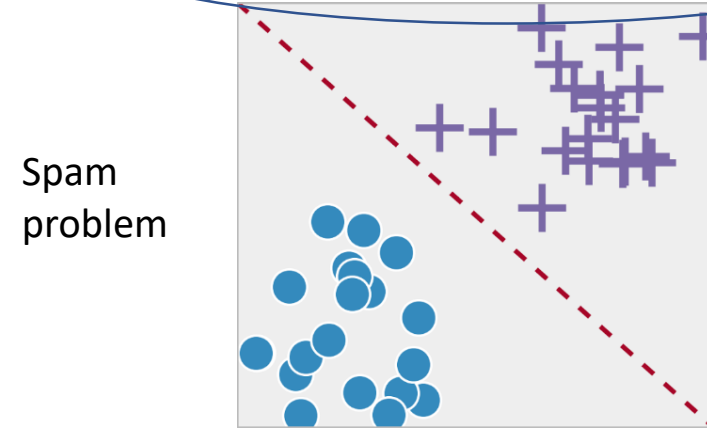
Evaluate performance using model obtained from training data

Evaluate

Supervised ML: Two Learning Problems







If qualitative → **classification** problem



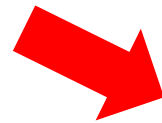
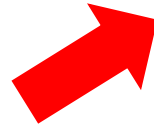
If quantitative → **regression** problem





Supervised Learning: Classification example

Data		
Class	Height	Weight
	1.30m	20 kg
	1.10m	5 kg
	.70m	4kg
.	.	.
.	.	.
.	.	.
	1.15m	12kg



Split data



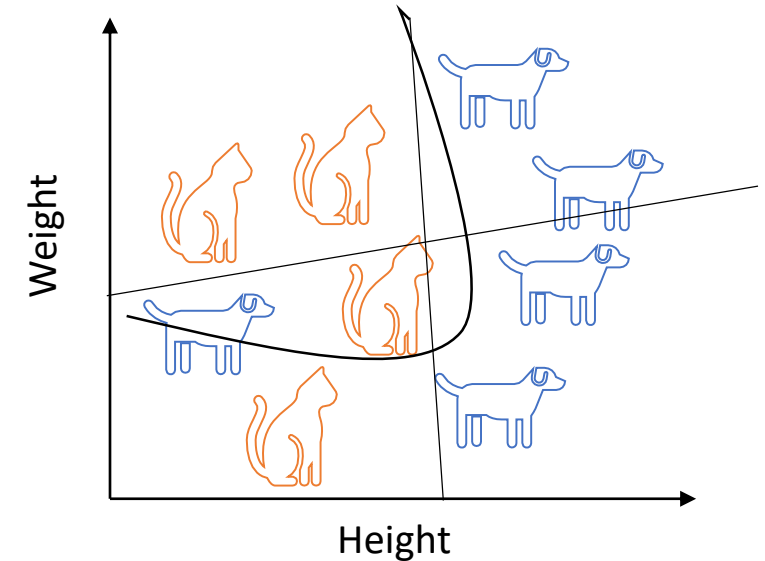
Training Data

Class	Height	Weight
	1.30m	20 kg
	1.10m	5 kg
.	.	.
.	.	.
.	.	.

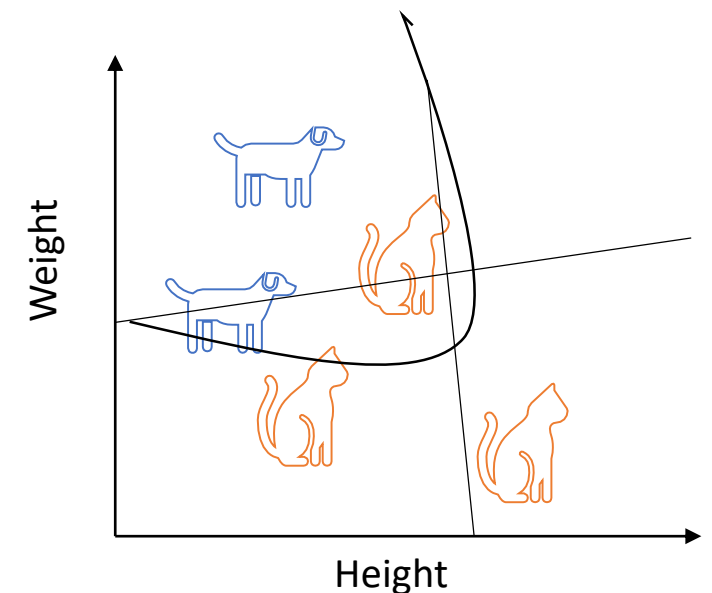
Testing Data

Class	Height	Weight
	1.0 m	8 kg
	1.14m	2 kg
.	.	.
.	.	.
.	.	.

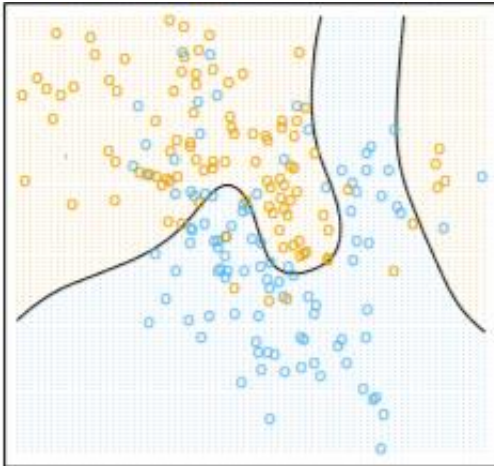
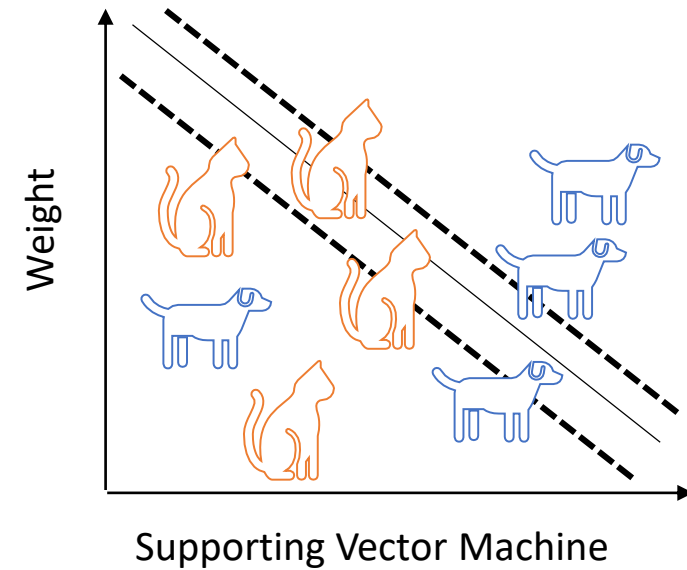
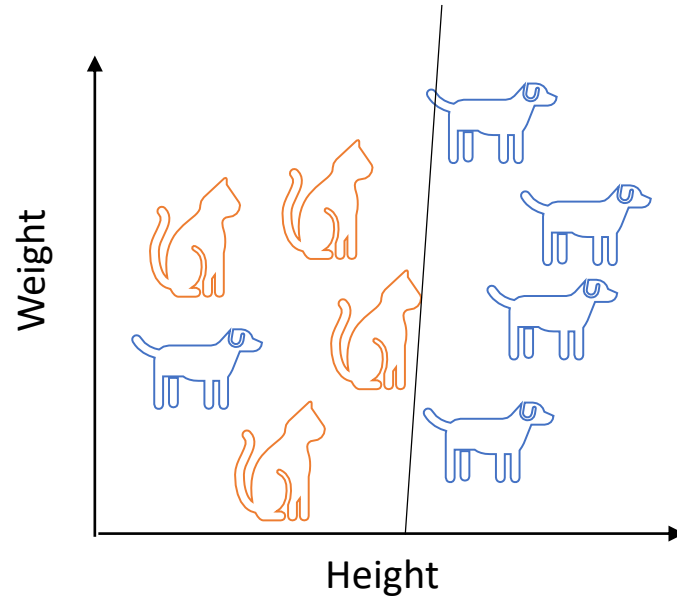
Find a method
to classify



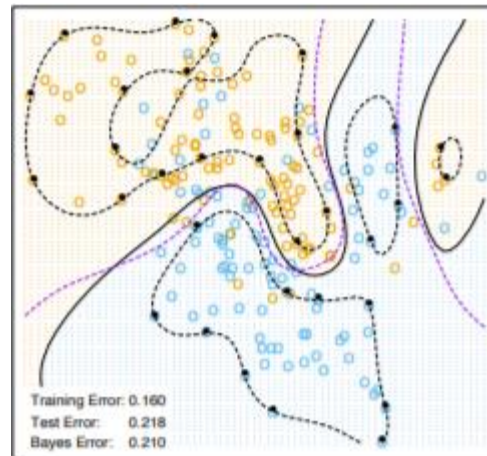
Now tell good
from bad



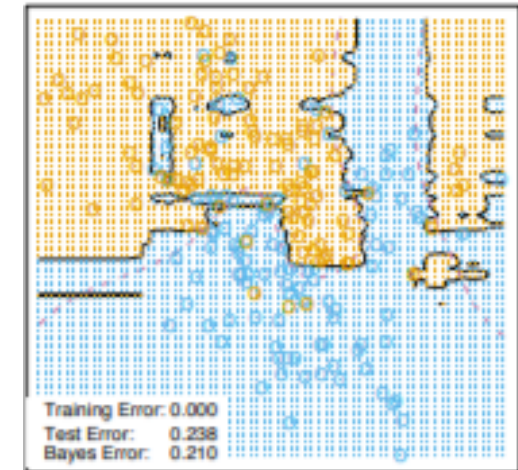
We can try different learning algorithms to classify



Naïve Bayes



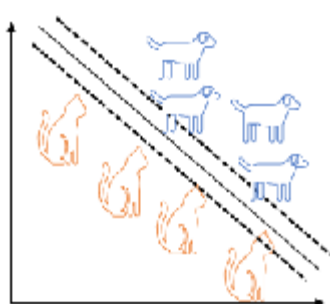
Kernel SVM



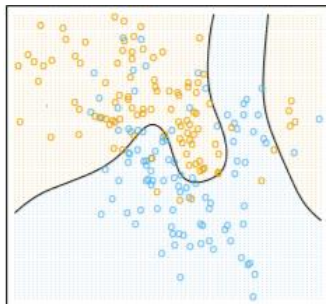
Random Forest (Decision Tree)

Supervised Learning: Classification example

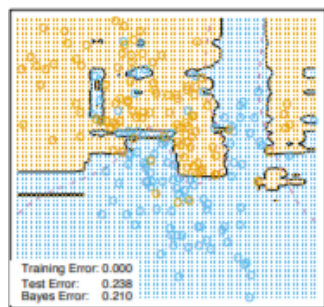
We train different ML algorithms in the training data



Supporting Vector Machine











Naïve Bayes



Random Forest

We choose the one that classifies better in the testing data

True	Predicted
	
	
	
	

1. We use the height and size of each element to predict a class label (dog or cat)
2. We compare the true class with the one that we predicted, and based on that we choose our ML algorithm.

This means that we can bring heights and sizes of different “beings” (no labels) and my classifier will tell me if I have a dog or a cat



Classification: Formalizing the Concept

In the classification problem we have:

- Covariates: $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n$ that can be represented as elements of \mathbb{R}^d
- Class labels: $\tilde{y}_i \in [K] = \{1, 2, \dots, K\}$

And thus we characterize the training data as training data $(\tilde{\mathbf{x}}_1, \tilde{y}_1), \dots, (\tilde{\mathbf{x}}_n, \tilde{y}_n)$

GOAL

1. Using the training data we want to estimate a good classifier ($f : \mathbb{R}^d \rightarrow [K]$)
2. We want the classifier to be able to predict the class of a new measurement in the testing data.

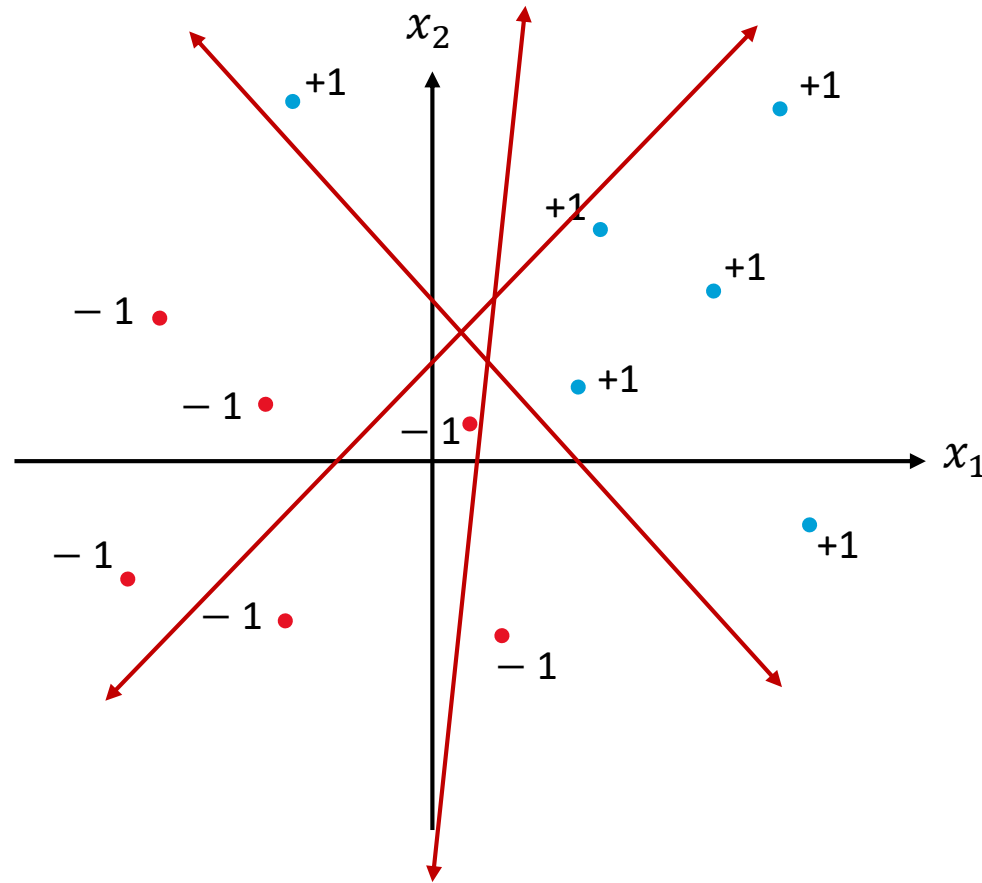
Simplifying assumption

For a binary classifier $K = \{1, 2\}$ we will use the following notation:

$$y \in \{-1, +1\}$$

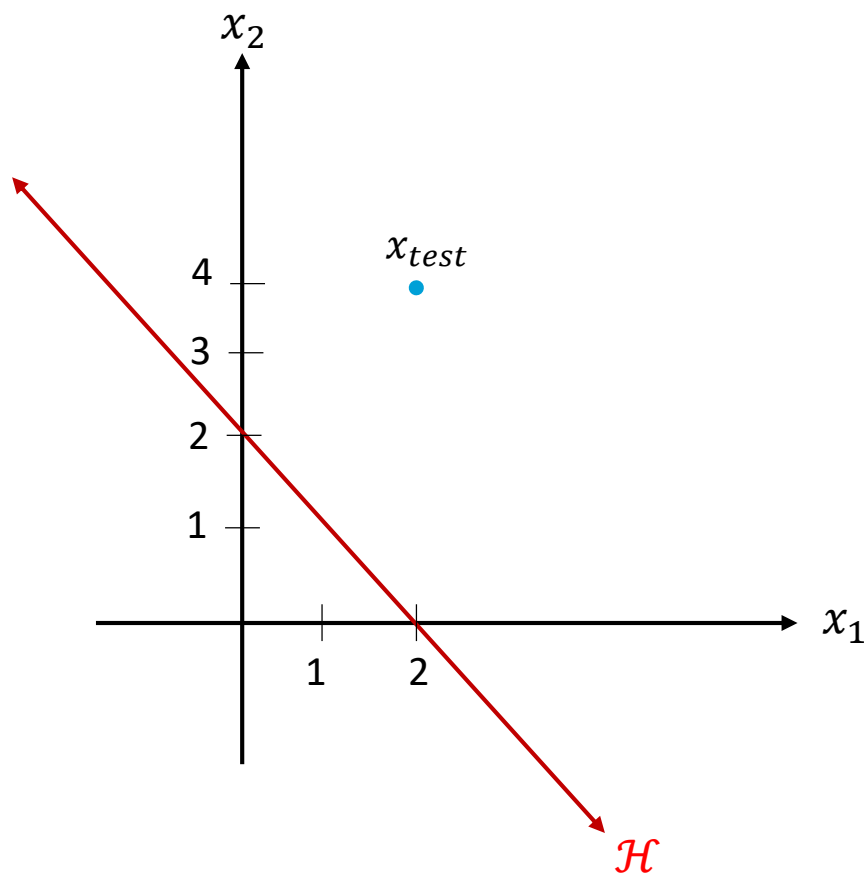
Linear Classification

We can think about linear classification as finding a good decision boundary for our classes:



Linear Classification: A very simple classifier

- The lines that we draw are just affine hyperplanes, that is, planes with a shift from the center.
- Suppose that we want to classify a new point? How do we know if the point is a plus 1 or a negative 1?

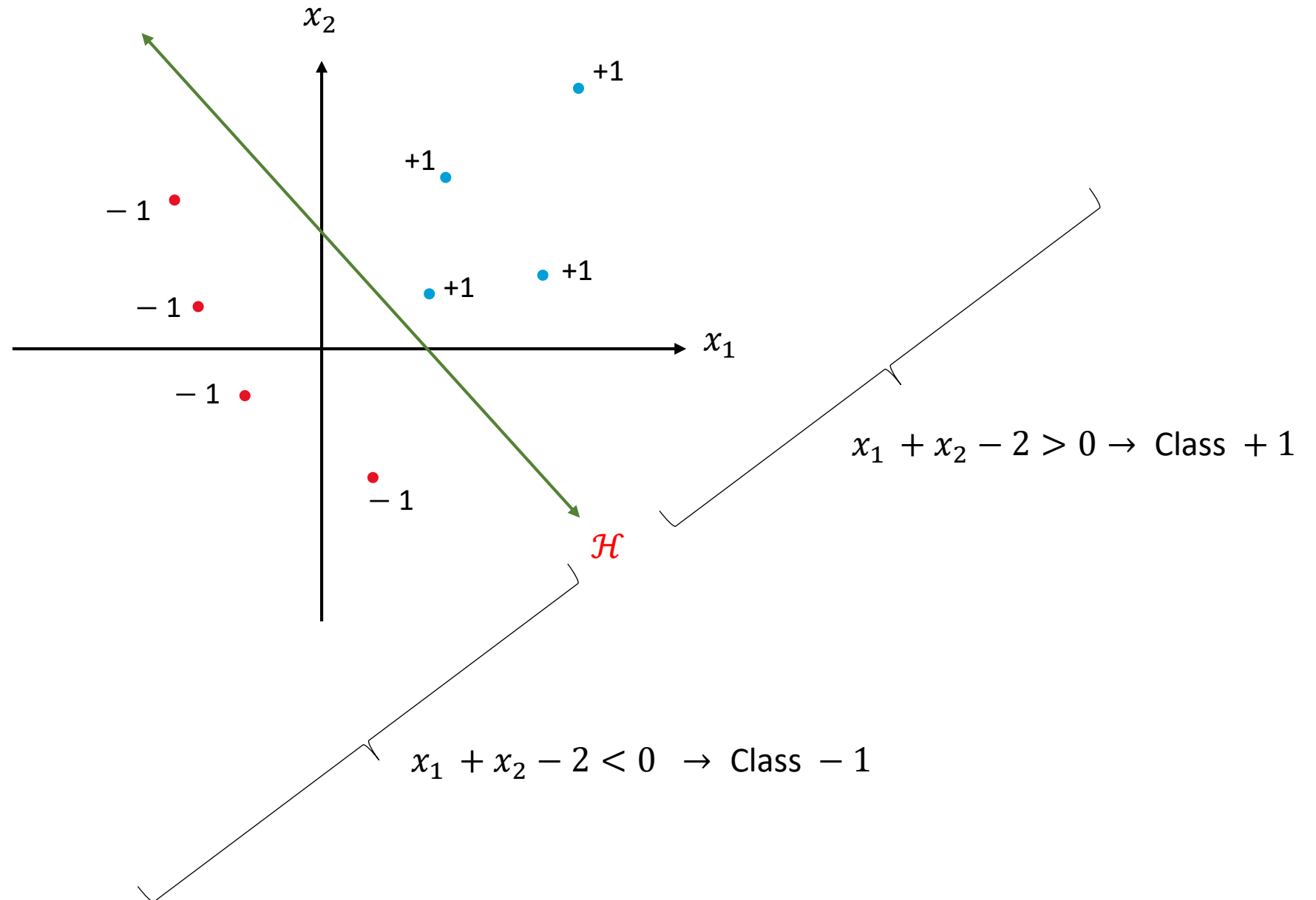


Point $x_{test} = (2, 4)$

$$y = mx + b$$

$$x_2 = -x_1 + 2 \quad \text{or} \quad x_1 + x_2 - 2 = 0$$

To know which side of the plane?



Linear Classification (for two classes)

Therefore, we can simply use:

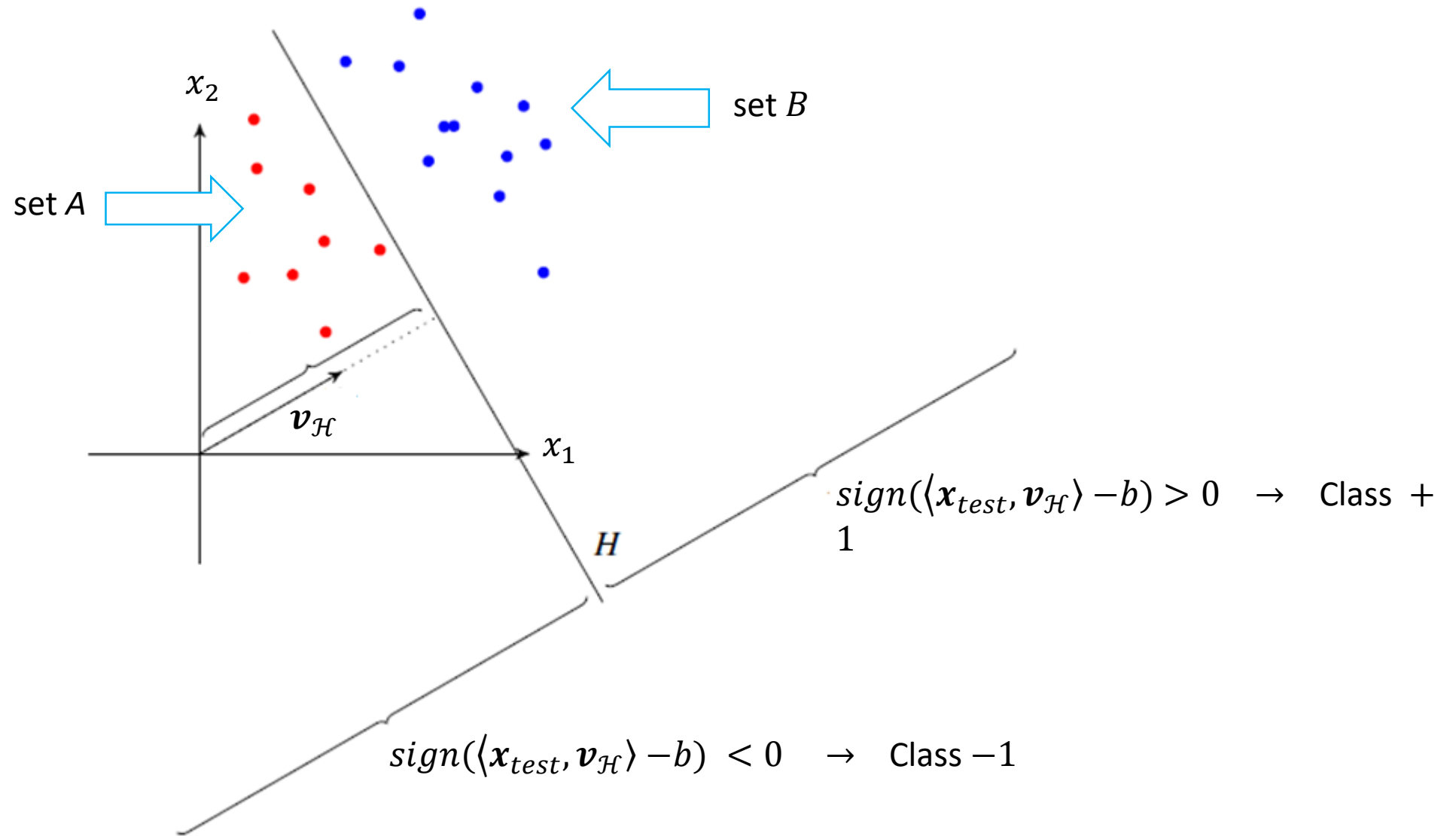
$$\text{sign}(x_1 + x_2 - 2)$$

More generally, this means that a linear classifier (for $y \in \{-1, +1\}$) takes the form:

$$f_n(\mathbf{x}) := \text{sign}(\langle \mathbf{x}_{test}, \mathbf{v}_{\mathcal{H}} \rangle - b)$$

Where $\mathbf{x}_{test} \in \mathbb{R}^d$, $\mathbf{v}_{\mathcal{H}} \in \mathbb{R}^d$ is a normal vector of hyperplane \mathcal{H} and $b \in \mathbb{R}$ is the shift

Therefore, two sets $A, B \in \mathbb{R}^d$ are linearly separable if there is an affine hyperplane \mathcal{H} which separates them, that is, which satisfies:



The Perceptron Algorithm

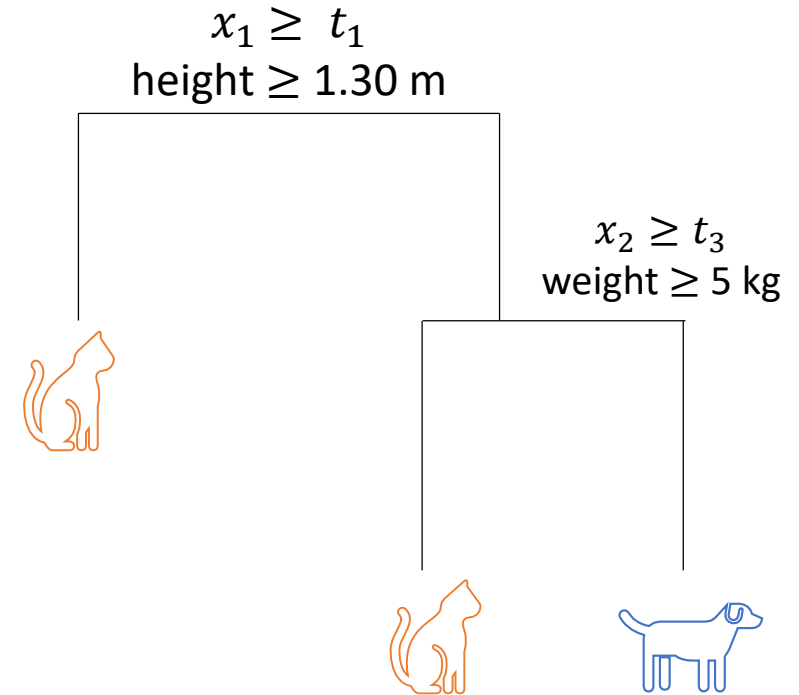
For linearly separable training data



Tree Classifiers

A tree classifier is a binary tree in which:

- Each inner node is rule of the form $x_i > t_i$
- The threshold values t_i will be the parameters which specify the tree
- Each leaf will be a class label K

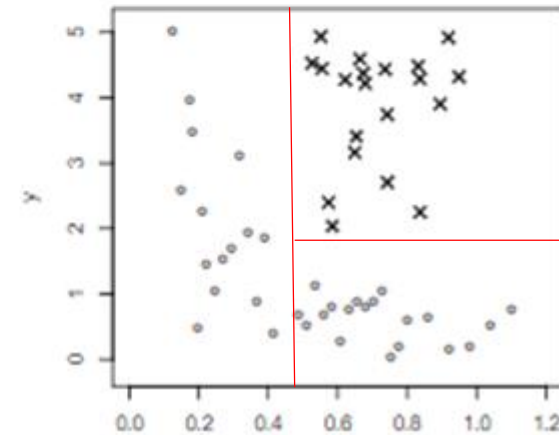
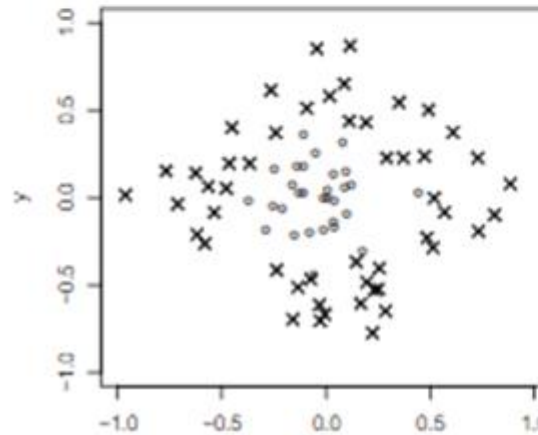
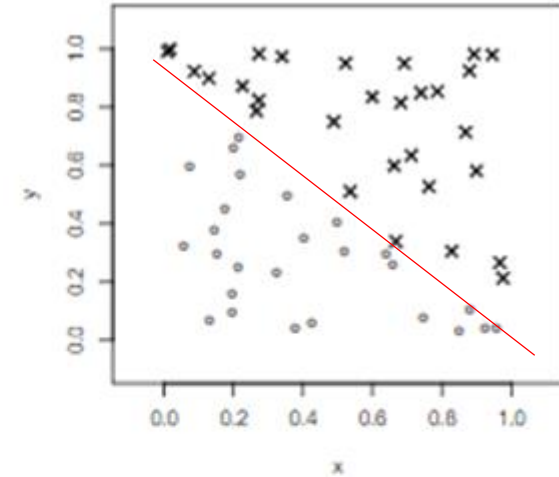
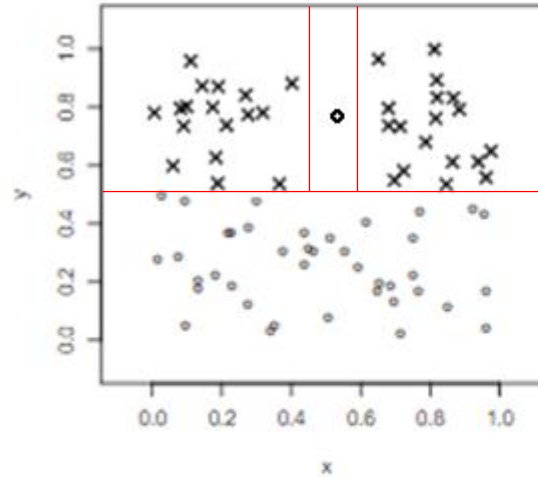


Classification Exercise

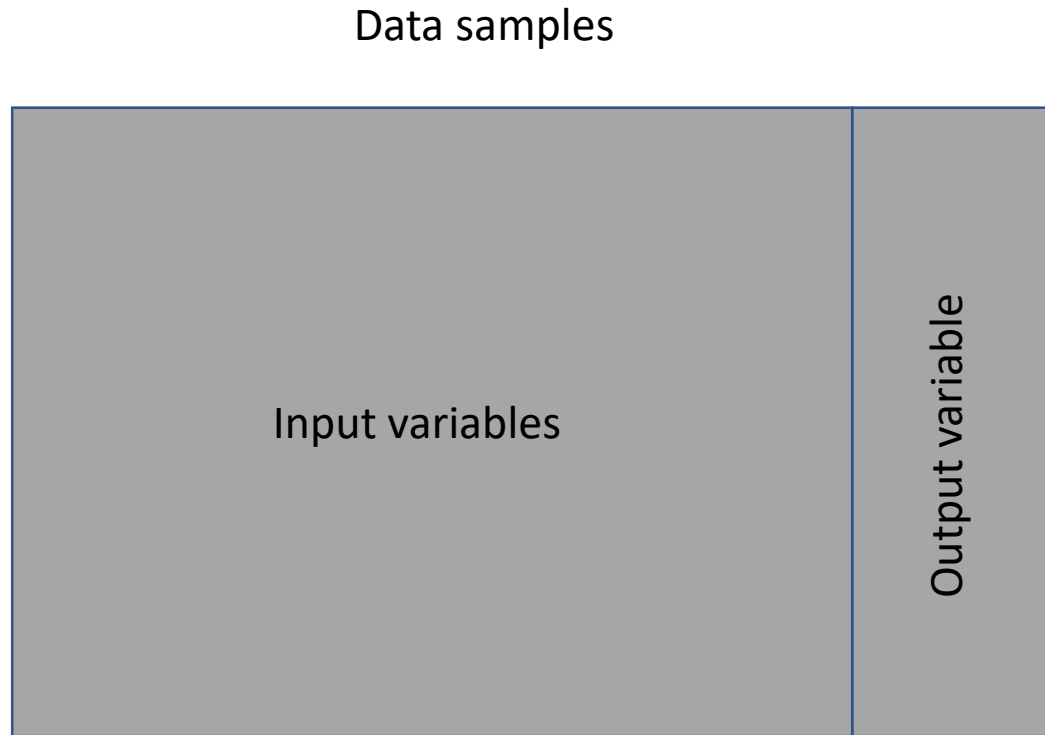
To have a better intuition on classification problems, let's try to decide together which classifier we could use given the following data sets:

When can we use a linear classifier?

When a decision tree?

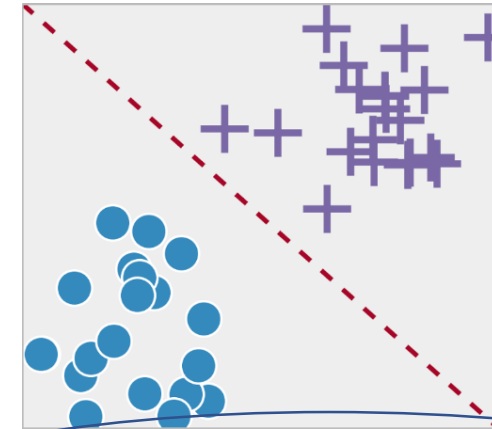


Supervised ML: Two Learning Problems



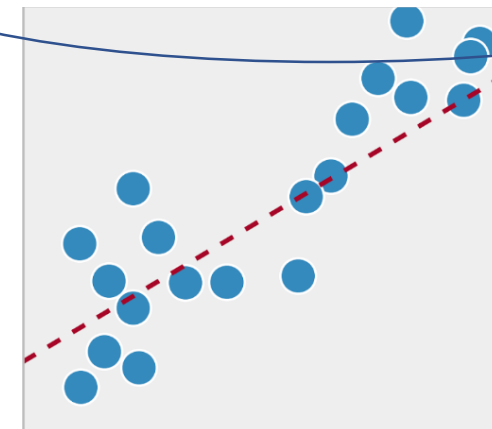
If qualitative → **classification** problem

Spam
problem



If quantitative → **regression** problem

Predict
salaries



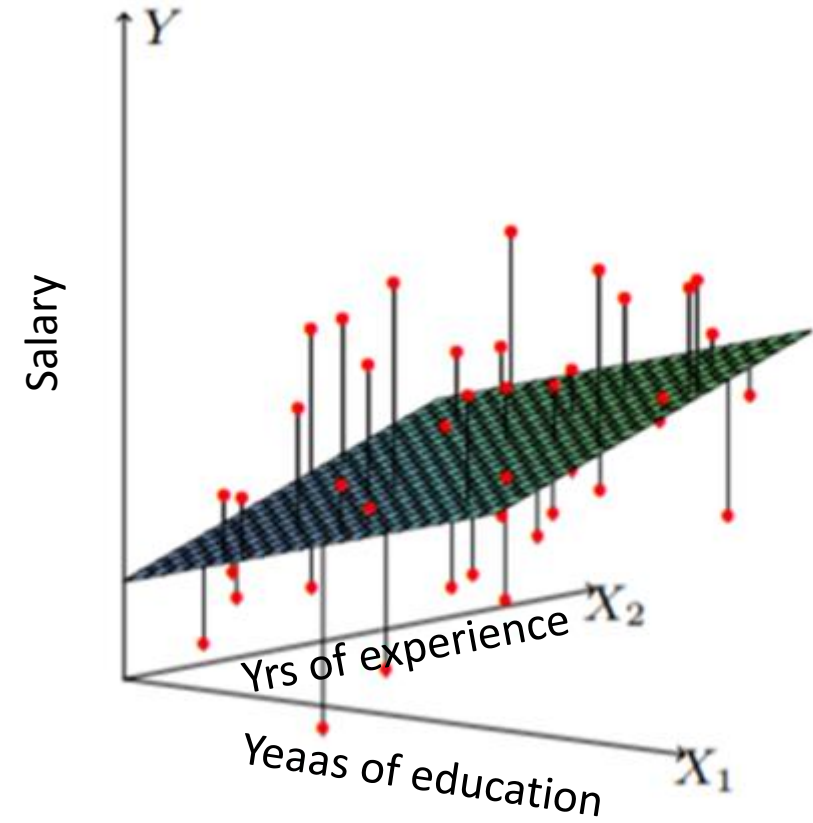
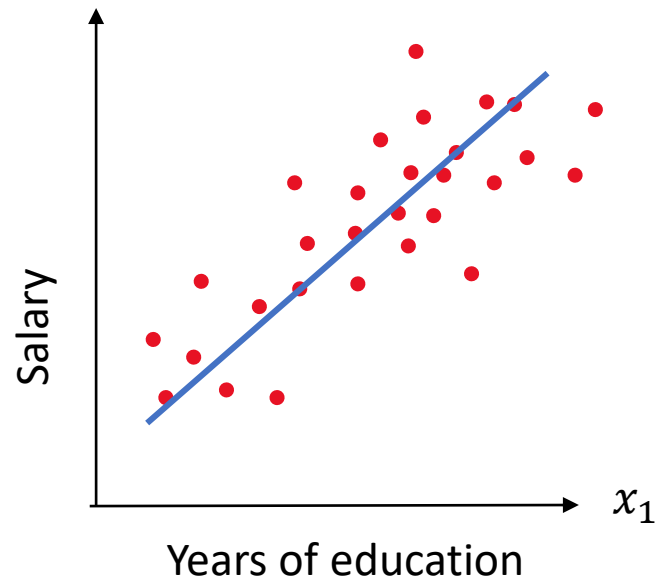
Regression

1. Recall that in classification we want to predict a class or a label
2. With regression we want to predict a quantitative outcome (a quantity)

Examples:

- Predicting salaries given gender, education, background, and experience.
- Predicting house values based on characteristics of the house (# of rooms, bathrooms, light bulbs, etc.).

Regression



Regression: Formalizing the concept

In the regression problem (as in econometrics) we have:

- Covariates or inputs: $\tilde{x}_1, \dots, \tilde{x}_n$ that can be represented as elements of \mathbb{R}^d
- Dependent variable or outcome variable: $y \in \mathbb{R}$ (is real-valued)

GOAL

1. Predicting a quantitative response
2. Using our covariates we want to approximate $f(x) = \hat{y}$ (find a predictor $f: \mathbb{R}^d \rightarrow \mathbb{R}$). This predictor is called a regression function.

Linear Regression

Typically, we can use a standard linear model to predict an outcome:

$$\hat{y}_i(\text{salary}) = \hat{\beta}_0 + \hat{\beta}_1 \text{Education}_i + \dots + \hat{\beta}_p \text{Years of experience}_i$$

In fact, to find $f(x) = y$ we can use least-square regression methods that minimize the Residual Sum of Squares:

$$RSS = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

And this is minimized by the vector $\hat{\beta}$:

Linear Regression

Important to keep in mind:

- In SML we want to make a good prediction for a new set of unseen observations, that is, we want to minimize the Residual Sum of Squares in the testing data, not the training data.
- We want to minimize the testing error not the training error
- More technically, we want to know whether $\hat{f}(x_0)$ is approximately equal to y_0 , where (x_0, y_0) is a previous data example from the testing data not used to train our learning method.

Therefore, we want to compute:

$$\text{Average } (y_0, \hat{f}(x_0))^2$$

Linear Regression

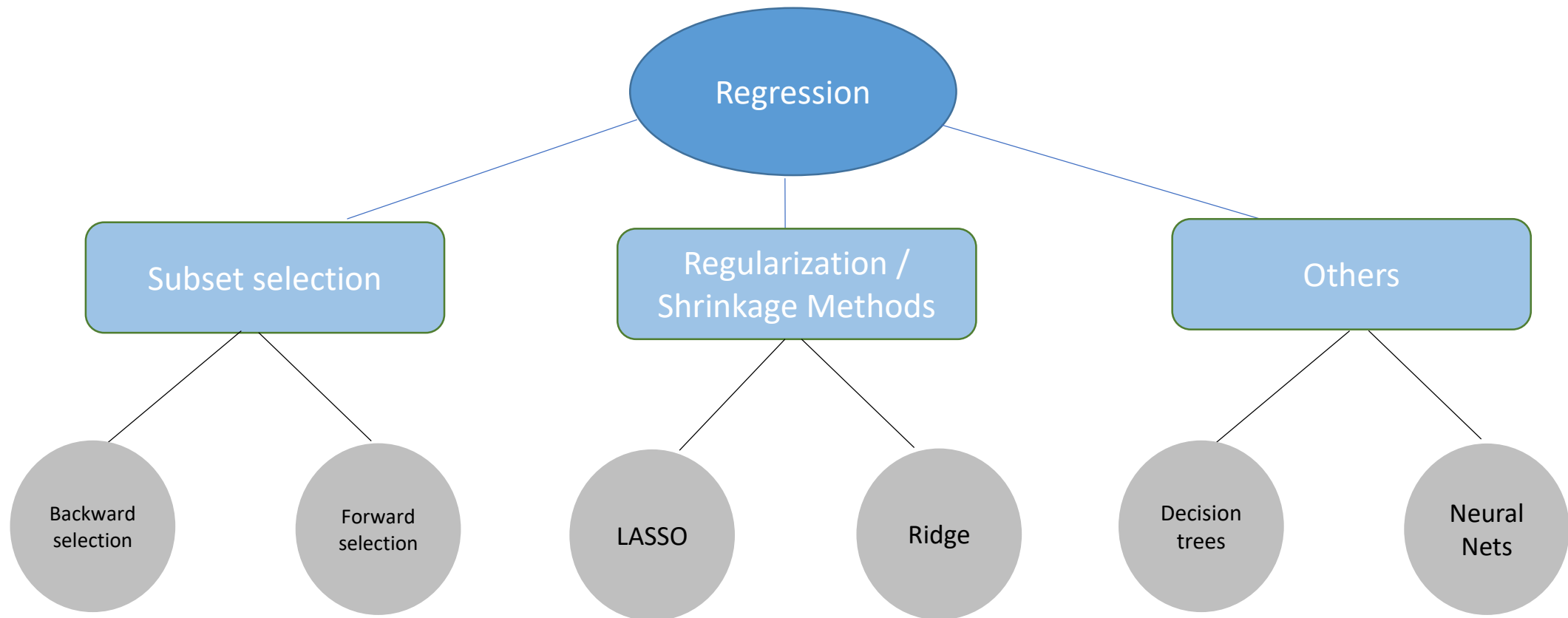
How do we choose the optimal model:

- We could either use the RSS or R-squared from the training data to choose the best model → This would be misleading
- We can include hundreds of variables to decrease the training error (RSS) but that does not guarantee that the testing error will be small → Overfitting very likely

There are two common approaches to address this problem:

1. **Subset selection methods:** Select among a set of different models with a subset of predictors.
2. **Regularization methods:** Using of all our predictors, use a training algorithm that allows us to regularize or constrain the coefficient estimates.

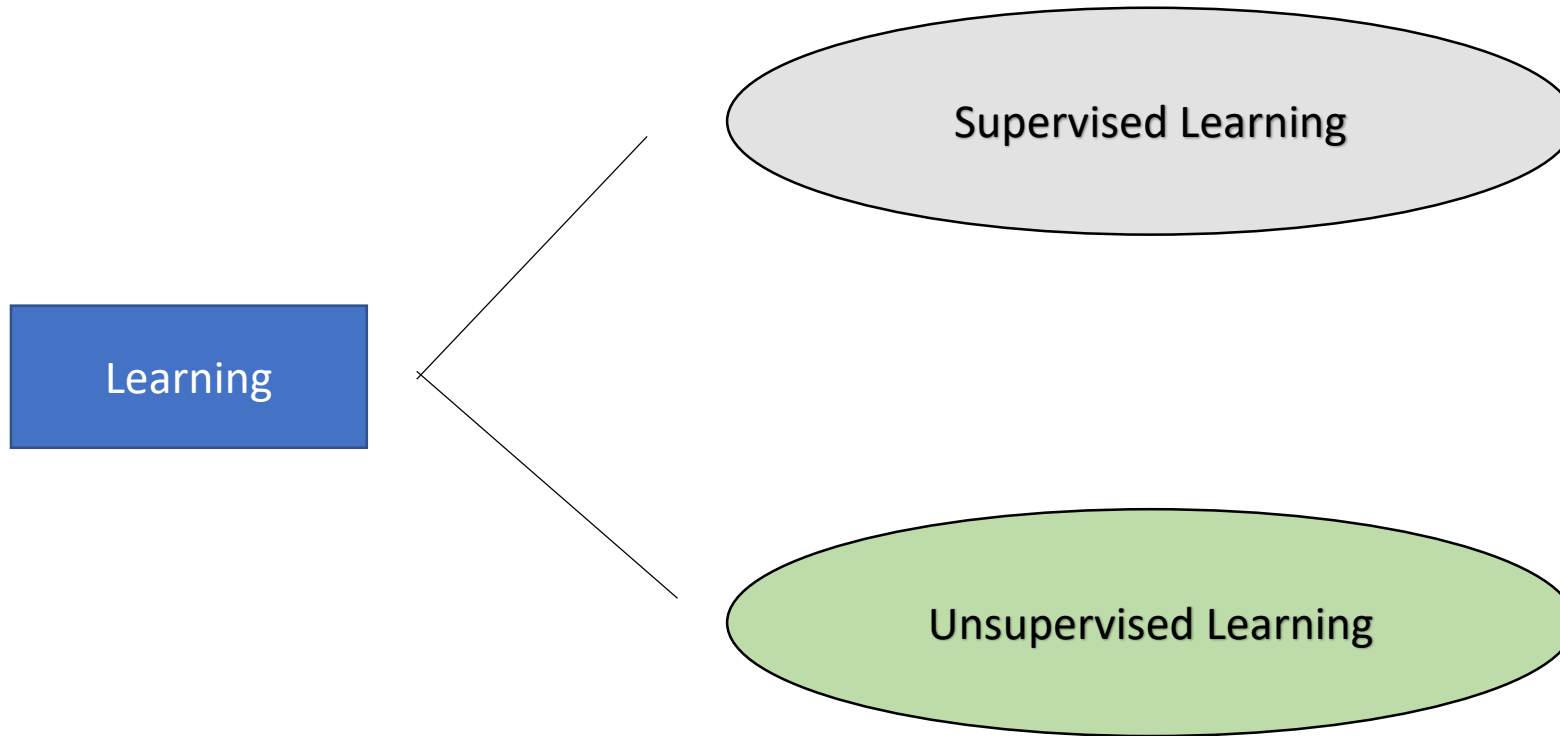
Different regression methods



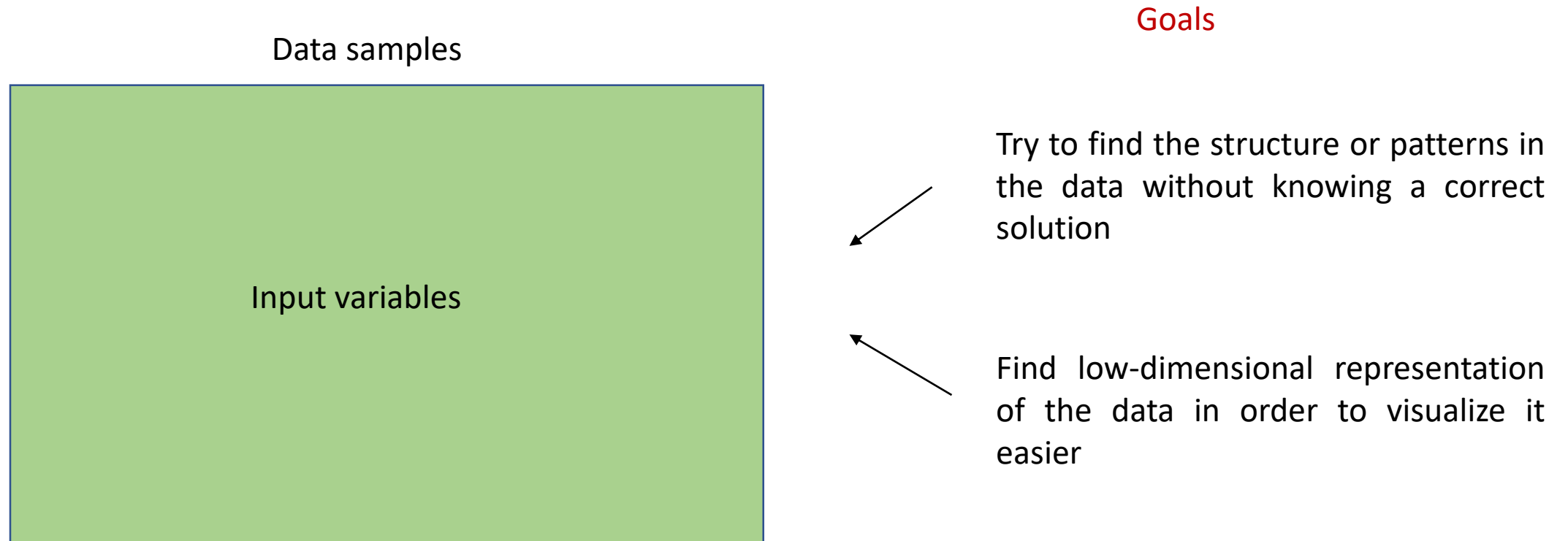
Supervised Learning Applications

- **Energy:** Predict energy savings from energy efficiency upgrades in schools (Burlig 2017)
- **Social policy:** Improve errors of inclusion and exclusion in conditional-cash transfer programs using survey data (Noriega 2018)
- **Program evaluation:** Estimating effects of certain policies (treatments) (Belloni, Chernozhukov and Hansen 2014)

Classes of learning algorithms



Unsupervised Machine Learning



Unsupervised Machine Learning

In short

- No label information is available (not outcome to predict or classify)
- There is no training and testing data
- More technically, our purpose is to fit model when only $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n$ are available

Unsupervised ML: Popular algorithms

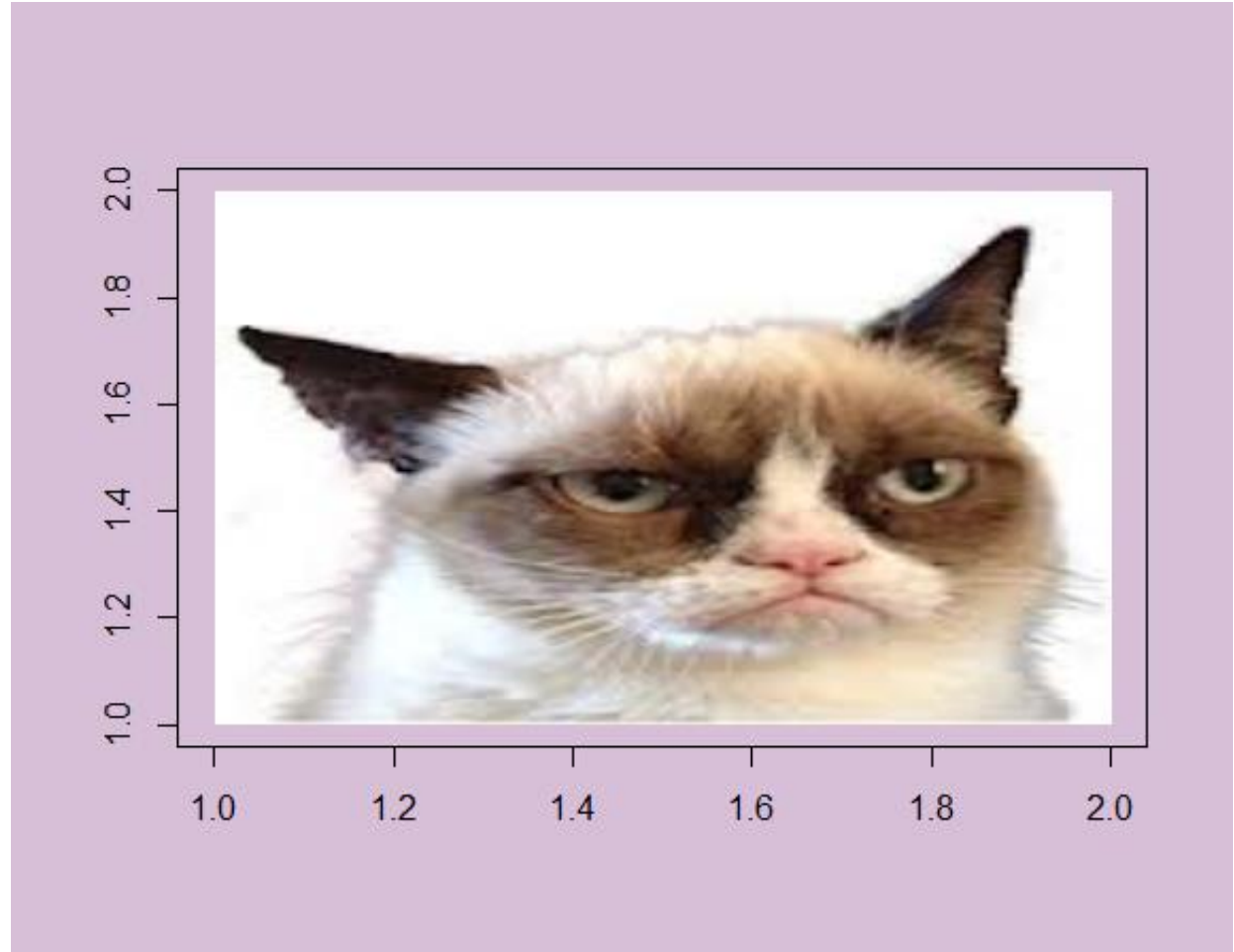
- K-means clustering
- Principal Component Analysis (PCA)
- Hierarchical Clustering

Dimension Reduction: Grumpy Cat

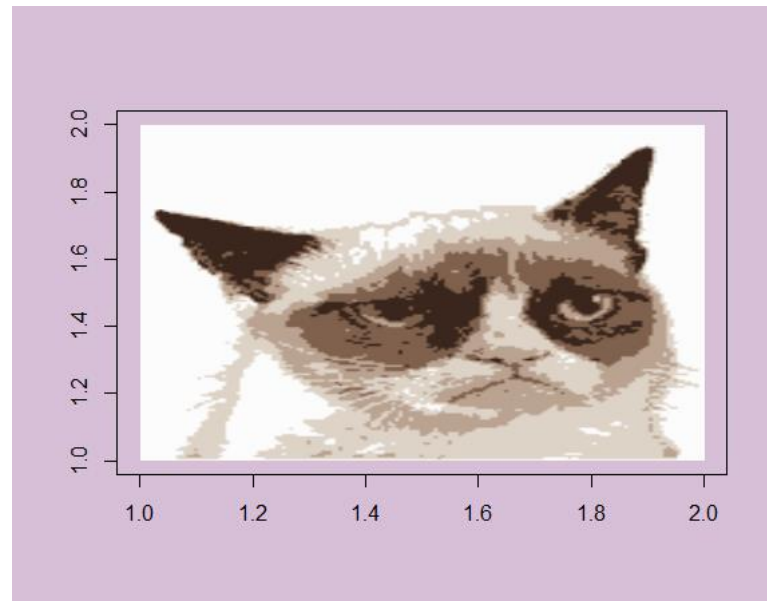
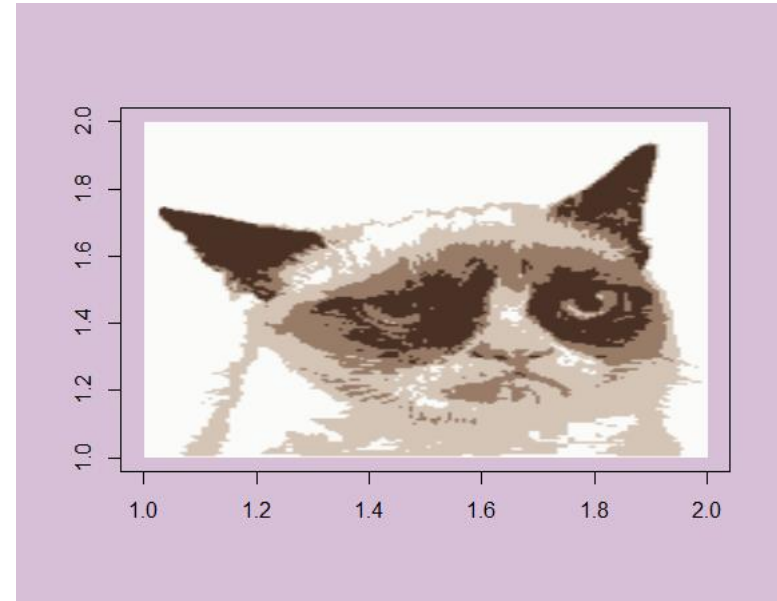
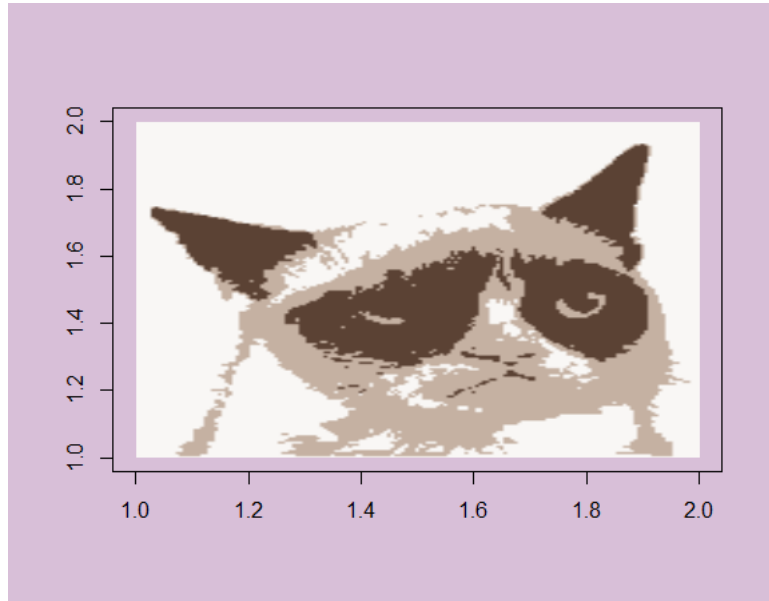
Every image, including Grumpy Cat, consists of different pixels

The size of a color image requires a lot space to be stored

We can reduce the number of pixels (dimension) and therefore the memory requirements



Example of Image Segmentation



Some applications of Unsupervised ML

- **Poverty:** Calculate meaningful measures of poverty or index of wealth based on survey or census data (Azuerro 2014, Corral & Montiel-Olea 2018)
- **Water:** Improve location of water pumps based on geographical clusters in Tanzania(O'Keeffe 2017)
- **Crime:** Identify groups of municipalities of Mexico according to crime activity in order to improve targeting strategies to reduce it (Montiel-Olea 2017).

What we will see in this workshop?

February 1st. Introduction to R (optional)

February 8th. OLS, Logistic Regression, and MLE in R

February 15th. Decision Trees

February 22th. Model Selection and Regularization

March 1st. Neural Nets (if there is time Supporting Vector Machine)

March 8th. Unsupervised Machine Learning

April (tentative). Natural Language Processing

Resources

<https://idbg.sharepoint.com/sites/EVP/MLPA/Pages/Learning.aspx>

Thank you!

References

- Azuero, R. 2014. Wealth and the Construction of Non-cognitive Skills: the case of Colombia. Documentos de trabajo, CEDE.
- Athey, S., & Wager, S., 2017. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. Journal of the American Statistical Association.
- Belloni, A., Chernozhukov, V., Hansen C. 2014. "High-Dimensional Methods and Inference on Structural and Treatment Effects. Journal of Economic Perspectives. Vol 28(2), pp 1 -23.
- Burlig F., Knittel, C., Rapson, D., Reguant, M., & Wolfram. 2018. Machine Learning from Schools about Energy Efficiency. Working Paper. University of Chicago.
- Corral, L., & Montiel-Olea, C. (2018). "What Drives Take-up in Land Regularization: Ecuador's Rural Land Regularization and Administration Program, SIGTierras. Working paper.
- Hastie, T., Tibshirani, R & Friedman, J. 2001. The Elements of Statistical Learning. Data Mining, Inference, and Prediction.
- Mitchel, T. 1997. *Machine Learning*. McGraw Hill.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A., 2012. Foundations of Machine Learning. MIT Press. Cambridge, Massachusetts
- Montiel-Olea, C. 2017. "An Unsupervised Learning Approach to Address Crime in Mexico" Unpublished master's thesis. Columbia University.
- Mullainathan, S". & Spiess, J. 2017. Machine Learning: An Applied Econometric Approach. Journal of Economic Perspectives. Vol 31(2), pp 87-106.
- Noriega, Alejandro. 2018. Fair and Efficient Targeting for Social Prosperity. Presented at the Inter-American Development Bank. November, 30, 2018.
- O'Keefe-O'Donovan, R. 2018. Water, Spillover and Free Riding: Provision of Local Public Goods in Spatial Network. Job Market Paper. University of Pennsylvania.
- Schapire. R., & Freund, Y. 2012 *Boosting: foundations and algorithms*. The MIT Press, Cambridge Massachusetts.