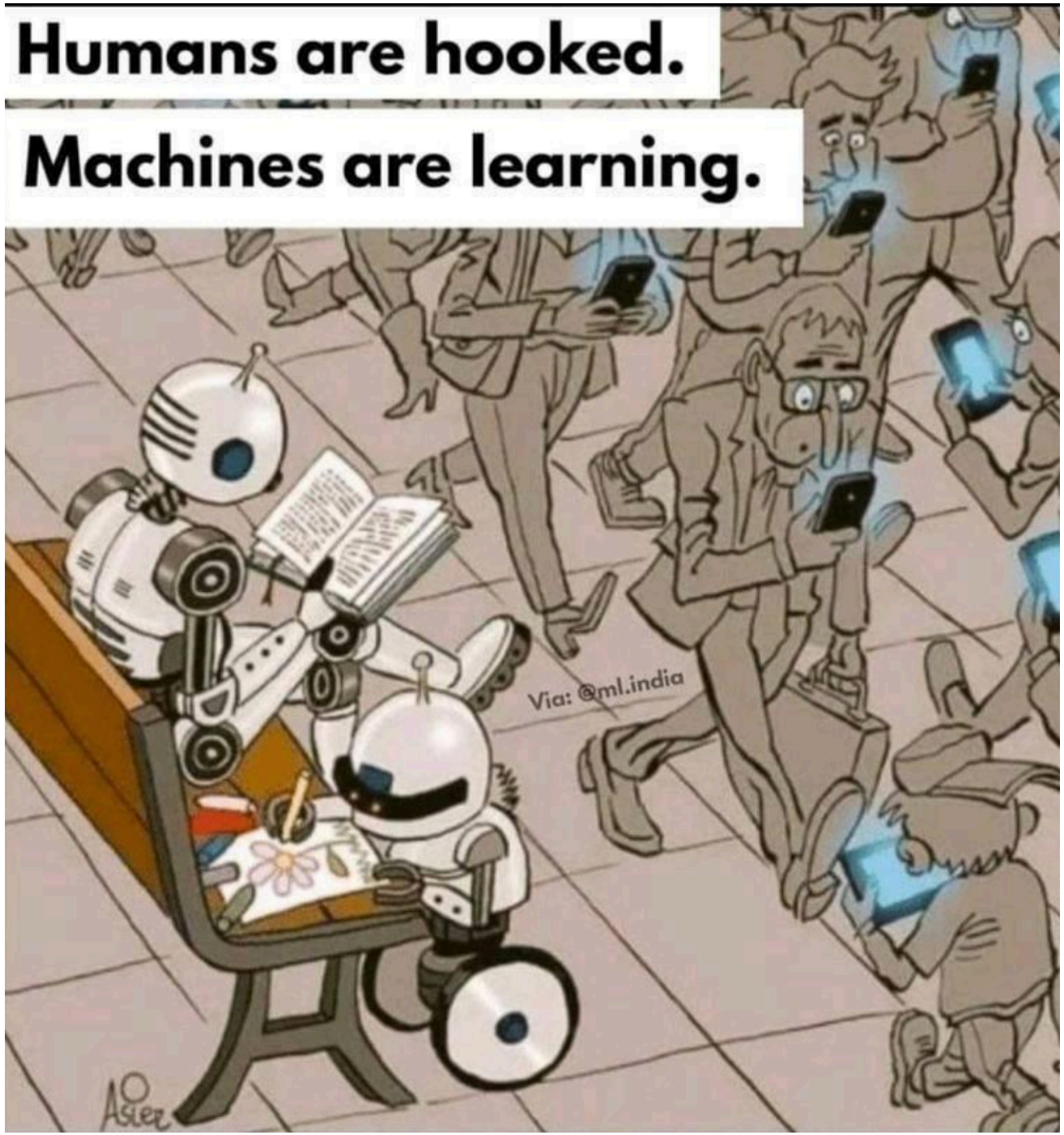
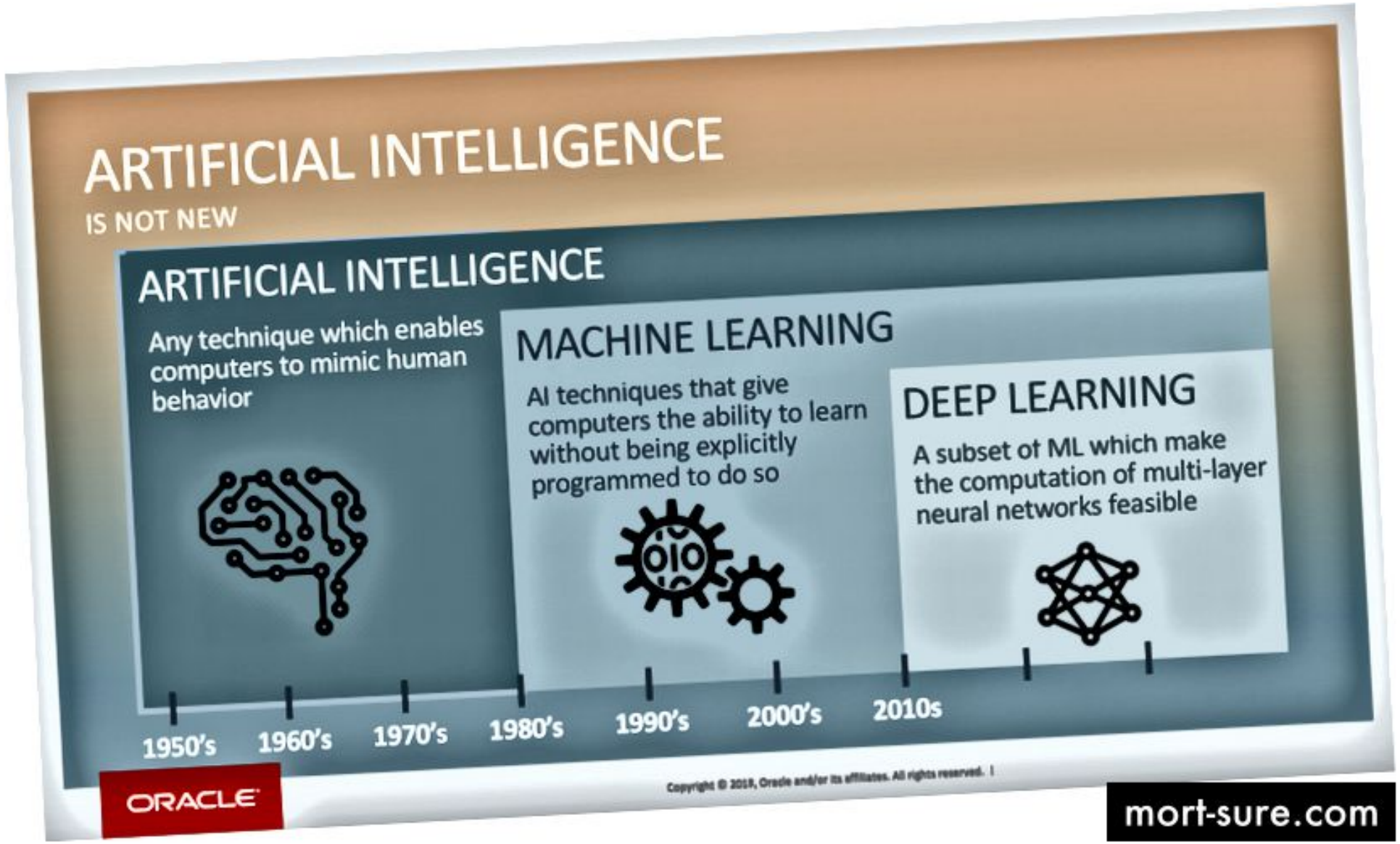


Machine Learning

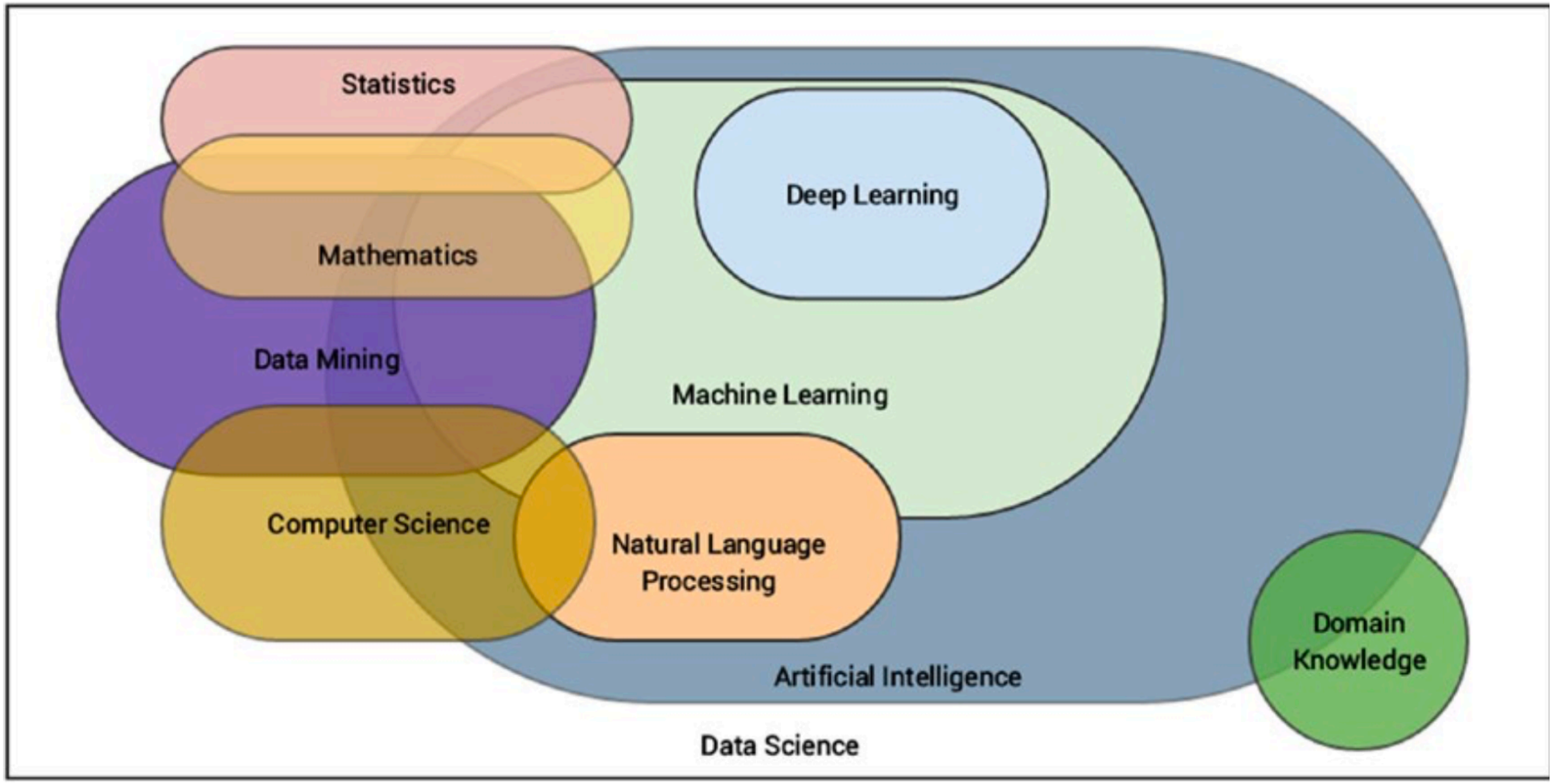
Gerda Jankevičiūtė
MMK Vilnius TECH

- Artificial Intelligence and ML
- Traditional Programming and ML
- ML Online Courses
- Main algorithms
  - Regression
  - Gradient Descent
  - Underfitting, Overfitting
  - Regularization
  - Decision tree
  - Support vector machine
  - PCA, K-mean

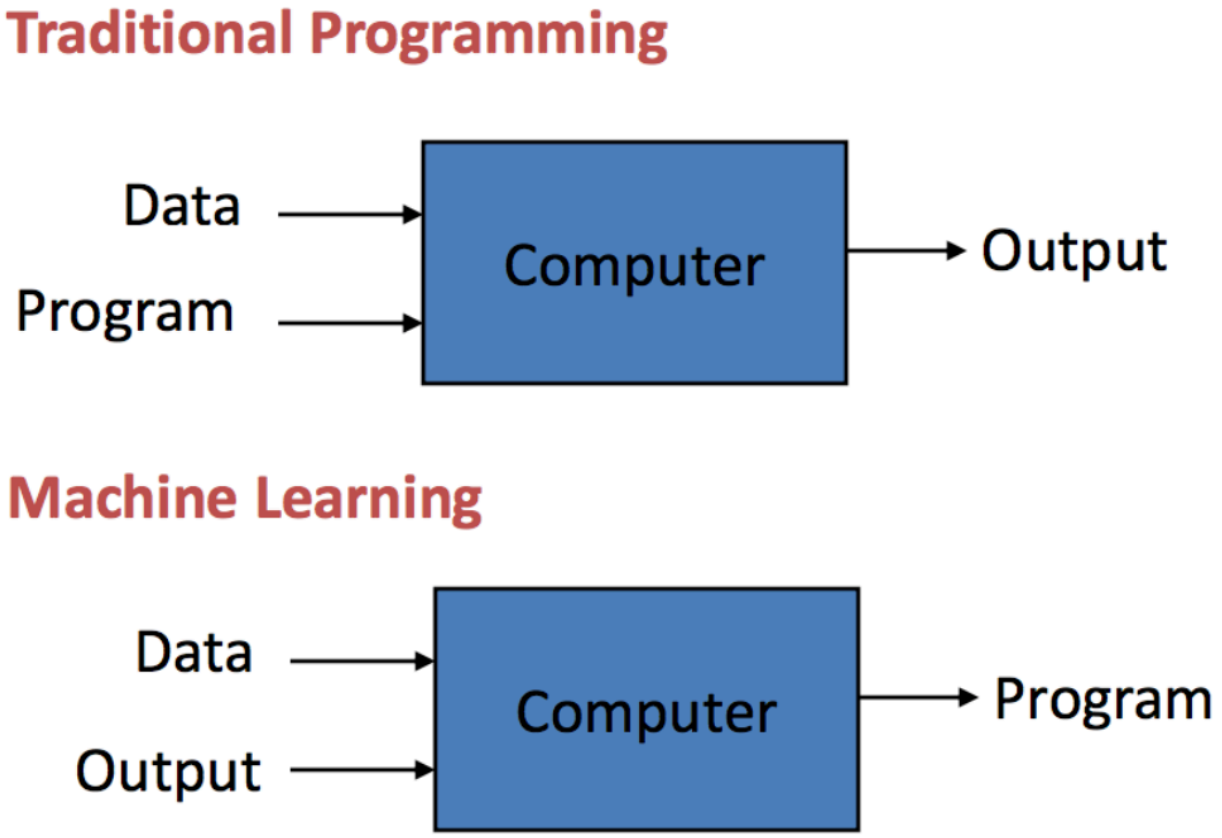


mort-sure.com

Machine Learning: multi-disciplinary field



Traditional Programming and ML



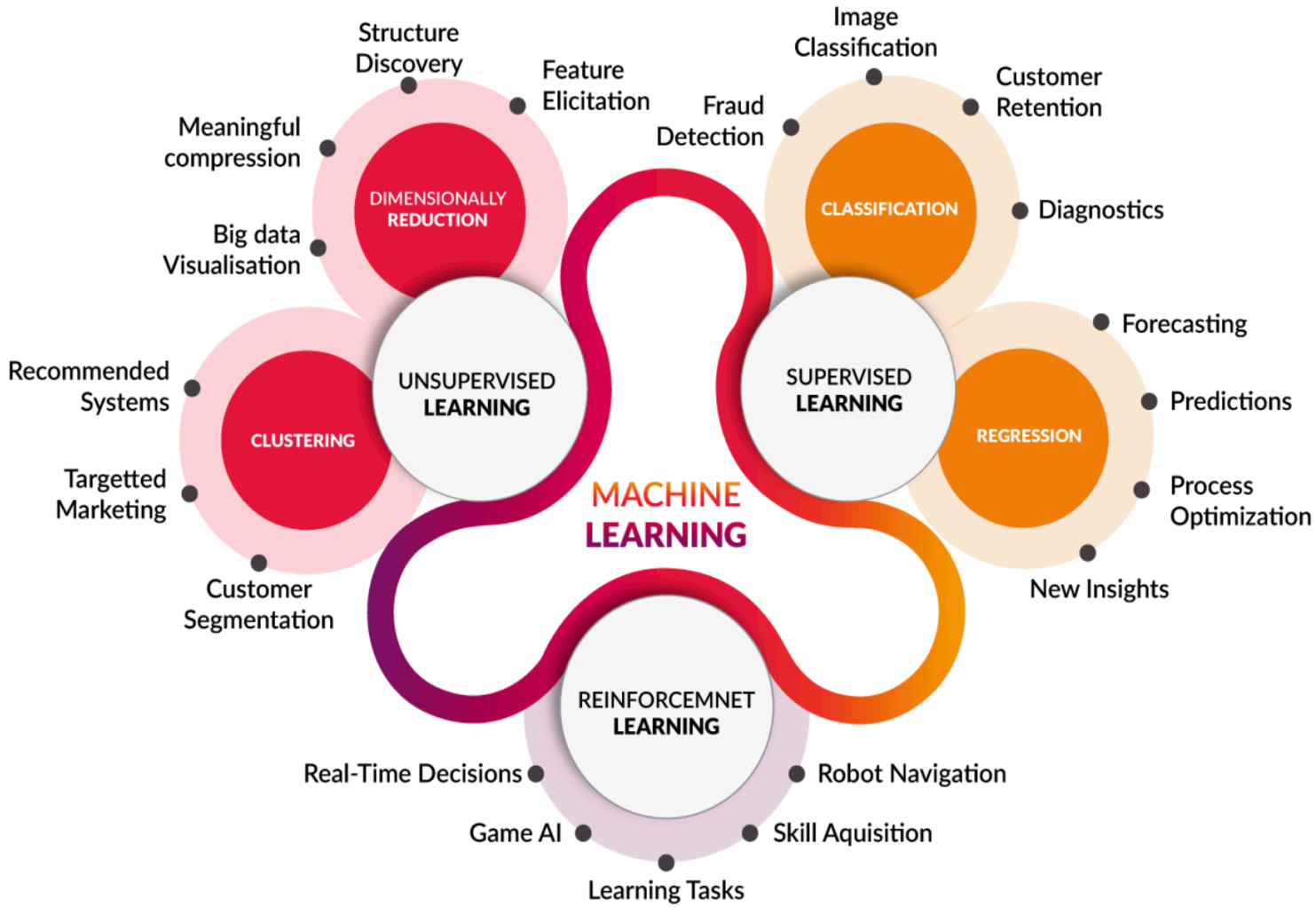


Image source: http://www.cognub.com/index.php/cognitive-platform/

Supervised Learning

Regression

Decision trees

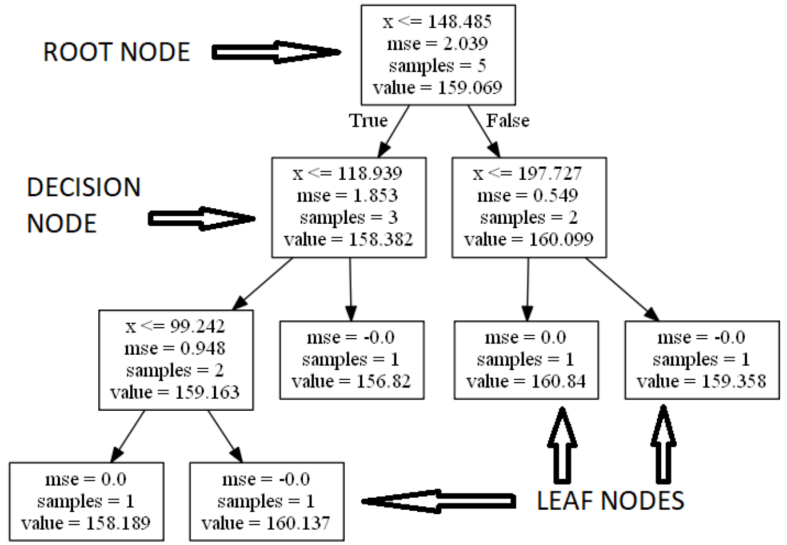
Support Vector Machine

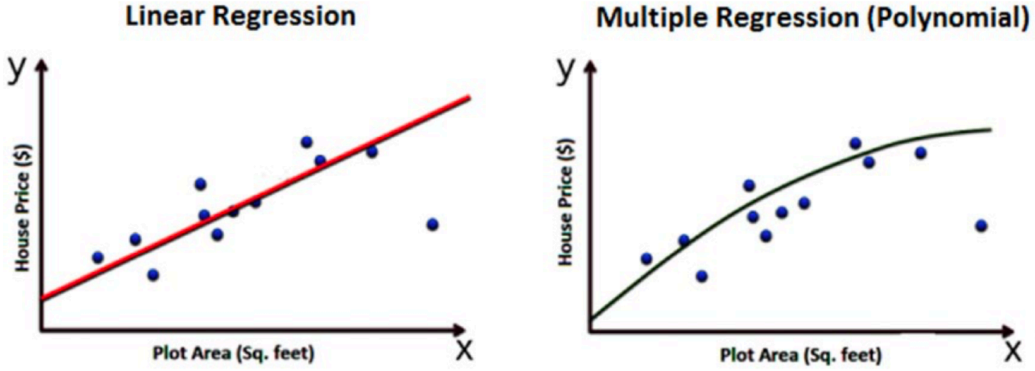
Neural Networks

Ensembles

Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$m = 47$





Supervised learning: regression models for house price prediction

Supervised Learning

Classification

Logistic Regression

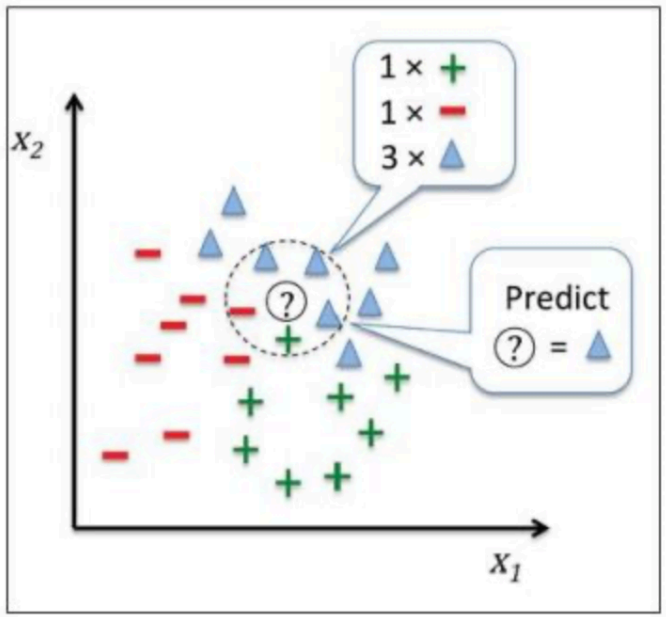
Discriminant Analysis

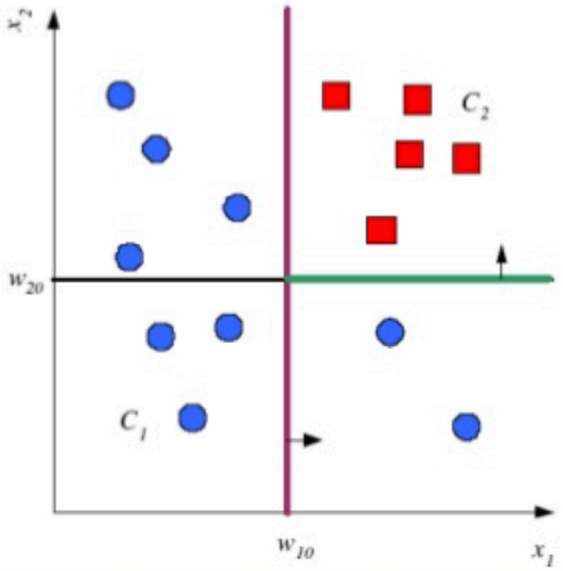
Decision Trees

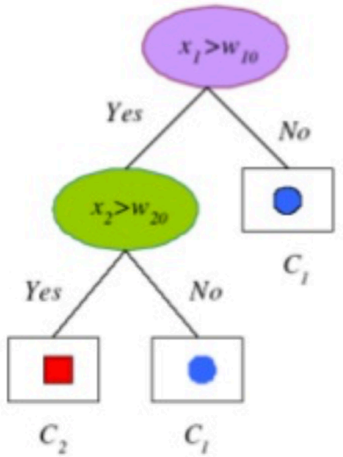
Support Vector Machine

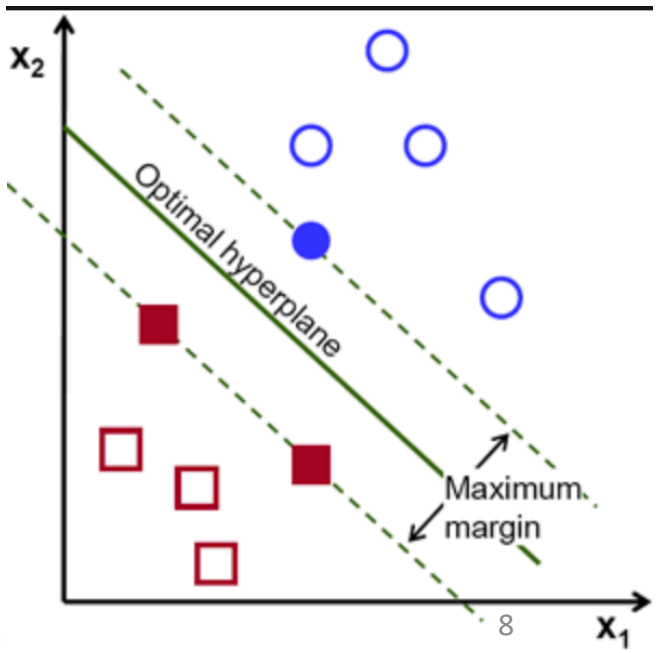
Neural Networks

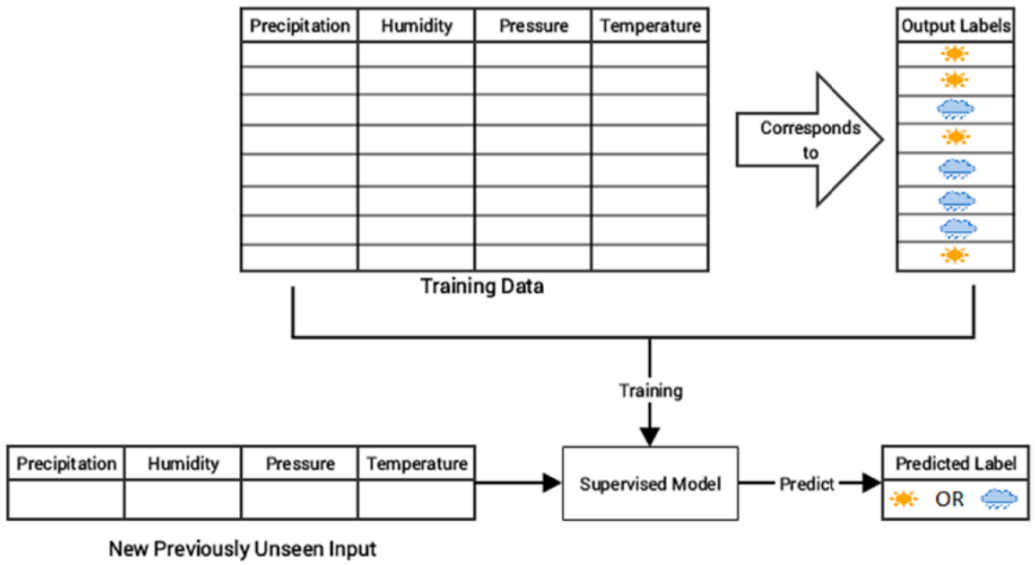
Ensembles







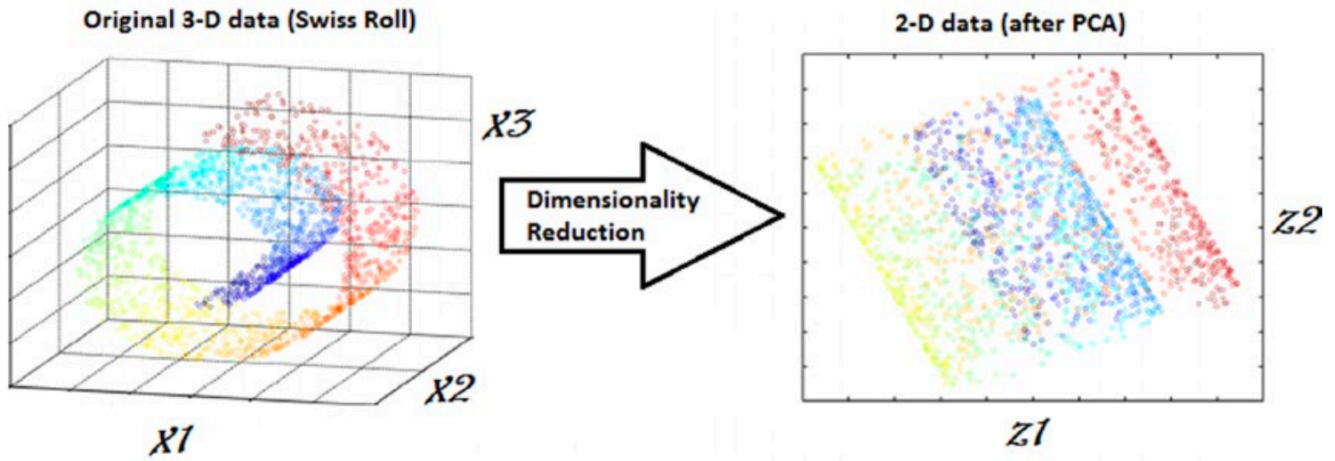




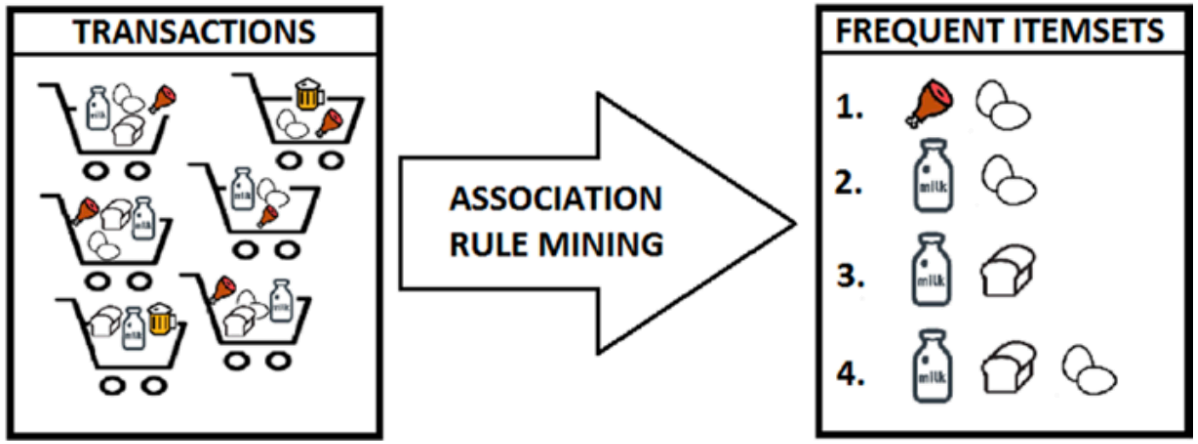
Supervised learning: binary classification for weather prediction

Unsupervised Learning

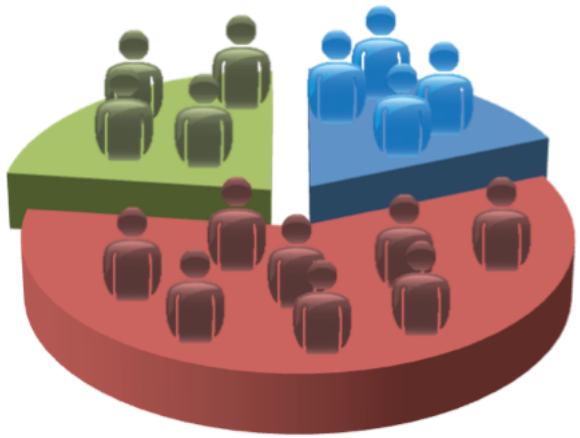
Clustering
Dimensionality reduction



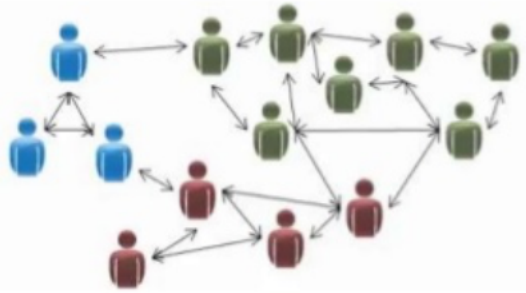
Unsupervised learning: dimensionality reduction



Unsupervised learning: association rule-mining



Market segmentation



Social network analysis

https://www.coursera.org/ (Andrew Ng)

- **Introduction**
- **Linear Regression with One Variable**
- **Gradient Descent** (GD)
- **Multivariate Linear Regression** (Multiple features; GD for Multiple features; Normal Equations)
- **Logistic regression** (Hypothesis Representation; Cost Function; GD; Advanced optimization; Multiclass Classification)
- **Neural Networks** (Non-linear Hypotheses; Neurons and the Brain; Forward and Backward Propagation)
- **Advice for Applying Machine learning** (Train-Test Sets; Regularization; Diagnosing Bias vs Variance; Learning Curves)
- **Support Vector Machines** (Optimization Objective; Large Margin Intuition; Kernels)
- **Unsupervised learning** (K-mean Algorithm)
- **Principal Component Analysis**
- **Anomaly Detection**
- **Large Scale Machine Learning** (Learning with Large Datasets; Stochastic GD; Mini-Batch GD; Map Reduce; Data Parallelism)

KTU

- **Introduction**
- **Linear and Logistic Regression**
- **Bias and variance tradeoff**
- **Linear and quadratic discriminant analysis**
- **Decision Trees**
- **Support Vector Methods**
- **Artificial Neural Networks**
- **Ensemble models**
- **Unsupervised learning**
- **Principal Component Analysis**

https://www.udemy.com

- **Python for Data Science and Machine Learning Bootcamp**

(Learn how to use NumPy, Pandas, Seaborn , Matplotlib , Plotly , Scikit-Learn , Machine Learning, Tensorflow , and more!)

- **Machine Learning A-Z™: Hands-On Python & R In Data Science**

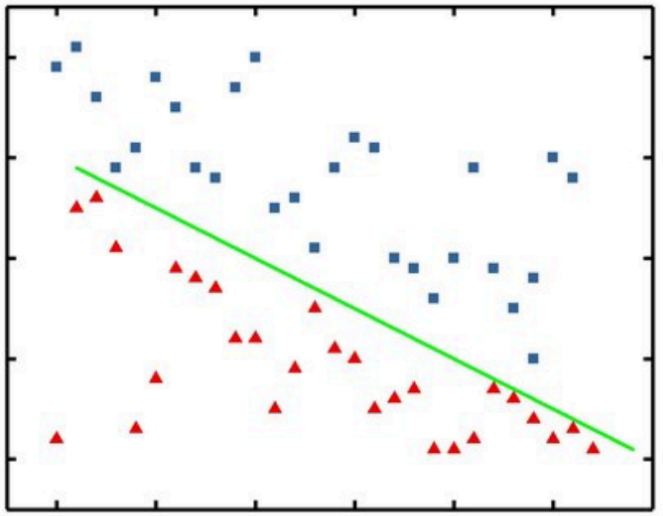
Learn to create Machine Learning Algorithms in Python and R from two Data Science experts. Code templates included.

- **Machine Learning Data Science and Deep Learning with Python**

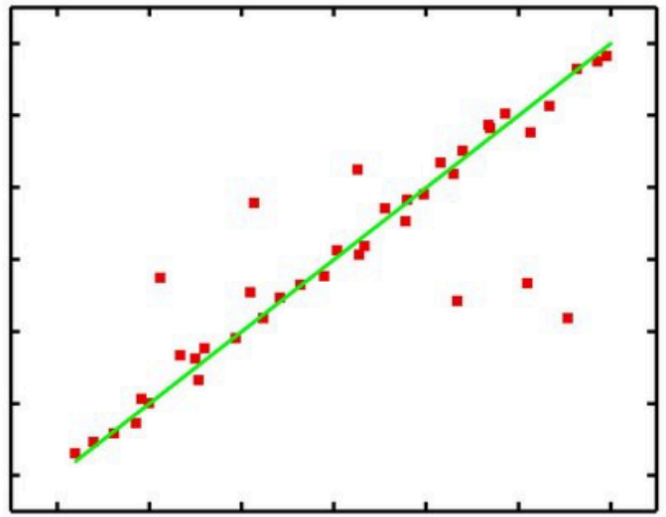
Complete hands-on machine learning tutorial with data science, Tensorflow, artificial intelligence, and neural networks

Supervised Learning

Regression
Gradient Descent
Train - test – validate
Cross - validation
Underfitting
Overfitting
Regularization

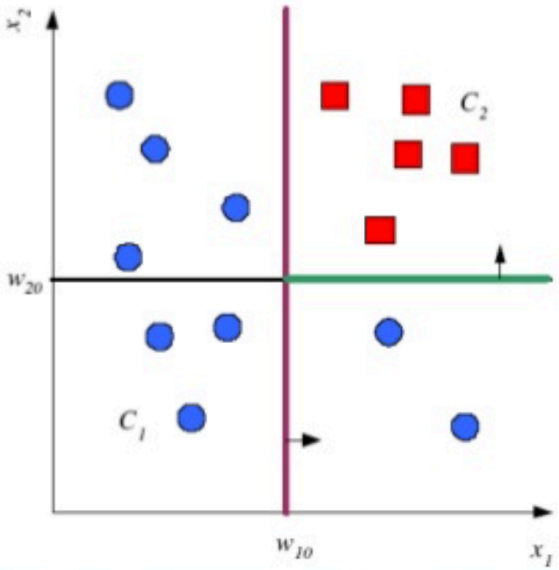


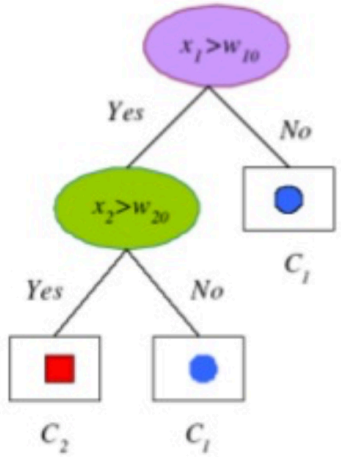
Logistic Regression

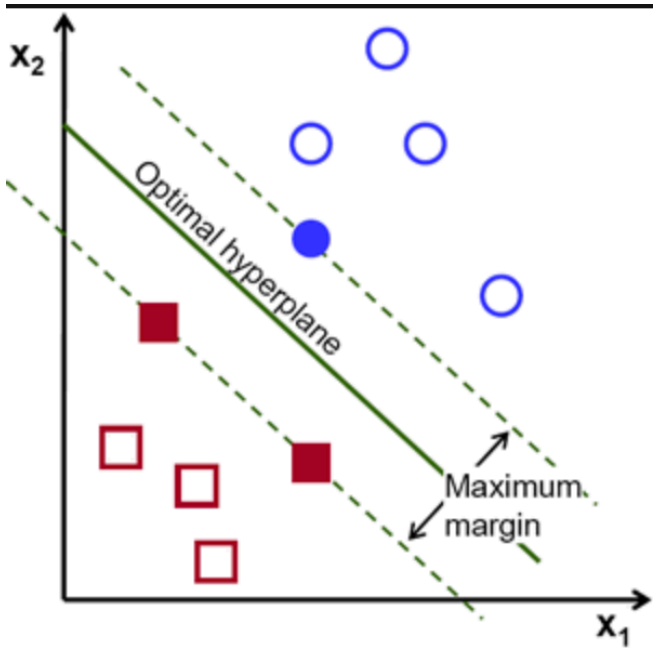


Linear Regression

Decision Tree
Support vector machines







Regression

Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$m = 47$

Hypothesis: $\boldsymbol{h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n}$

n – number of features;

$x^{(i)}$ – input (features) of ith **training example;**

$x_j^{(i)}$ - value of feature j in ith **training example;**

Cost function: $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

Gradient Descent:

α – learning rate

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

Logistic Regression

Classification $y \in \{0, 1\}$

- Email: Spam / Not Spam?
- Online Transaction: Fraudulent (Yes / No)
- Tumor: Malignant/Benign

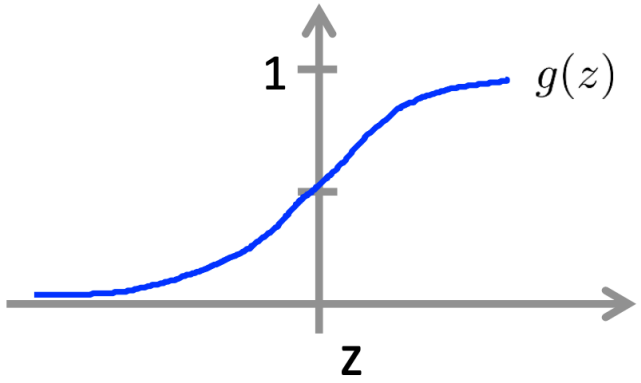
Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1+e^{-z}}$$



Gradient Descent

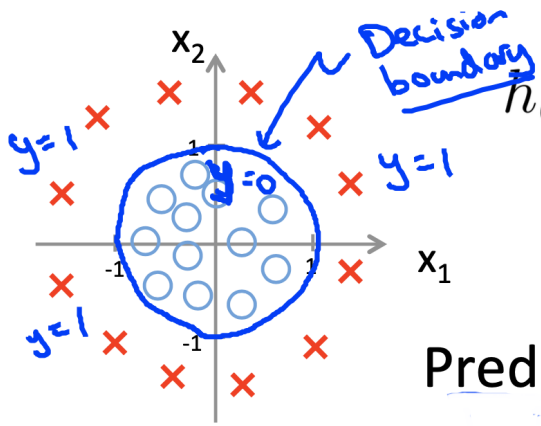
$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$:

Repeat $\{$

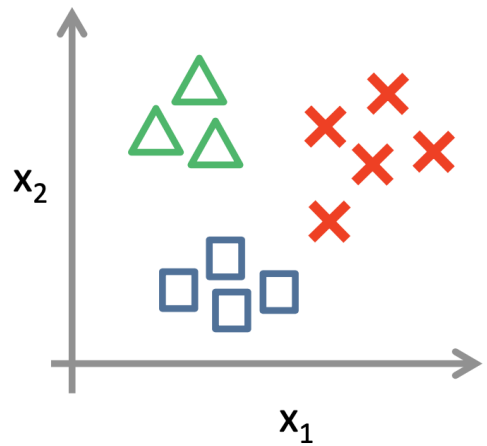
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$\}$



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

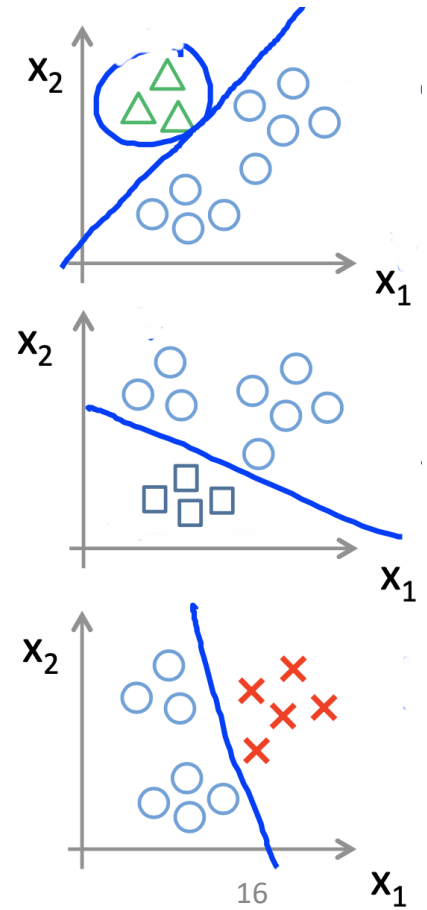
Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$



Class 1: △

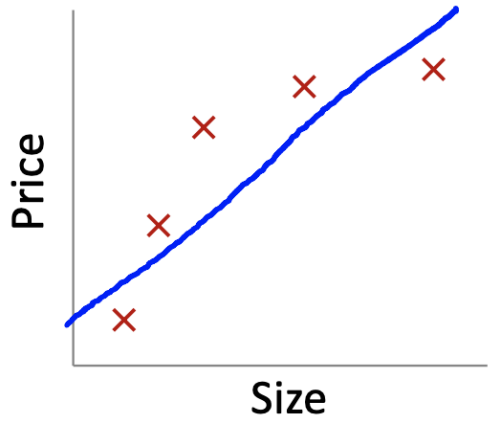
Class 2: □

Class 3: ×



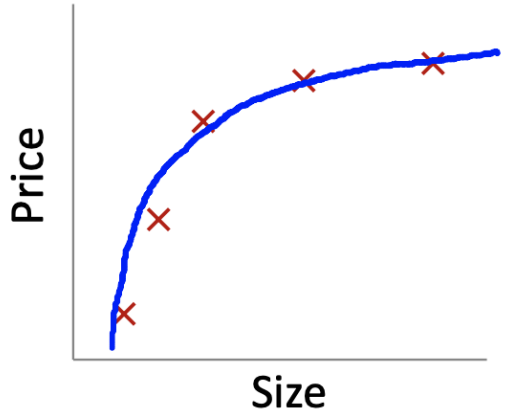
Overfitting

Example: Linear regression (housing prices)



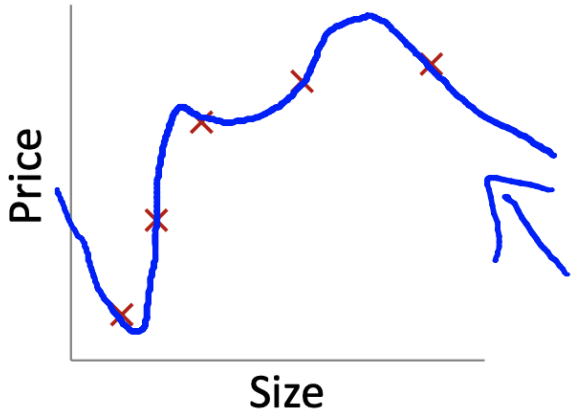
$\rightarrow \theta_0 + \theta_1 x$

"Underfit" "High bias"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$

"Just right"

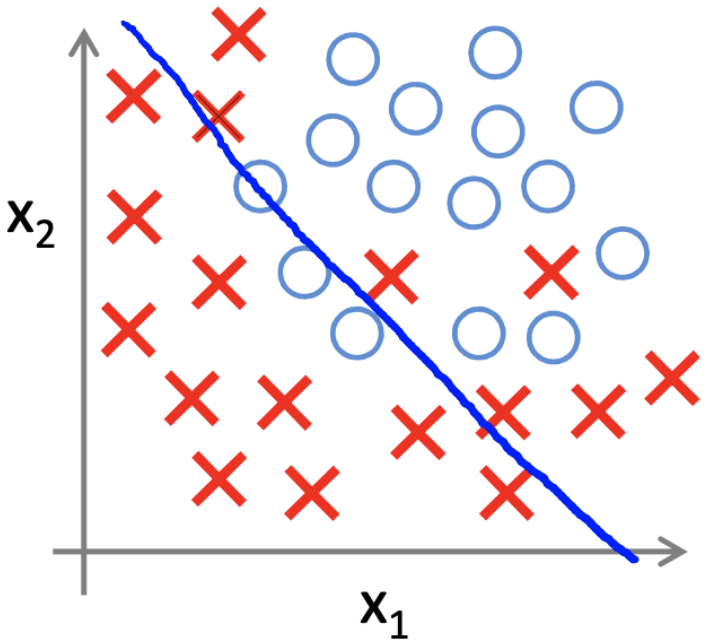


$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

"Overfit" "High variance"

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

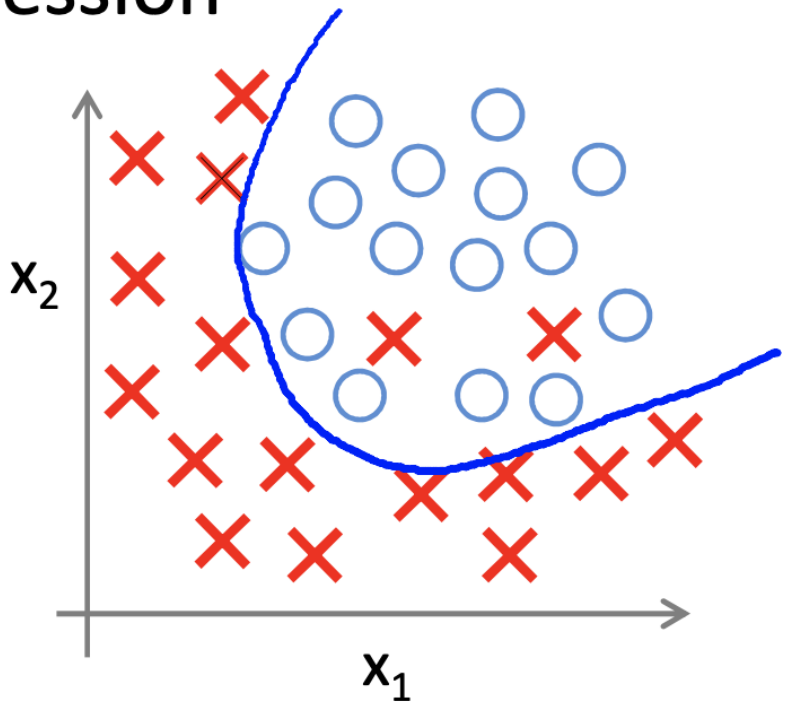
Example: Logistic regression



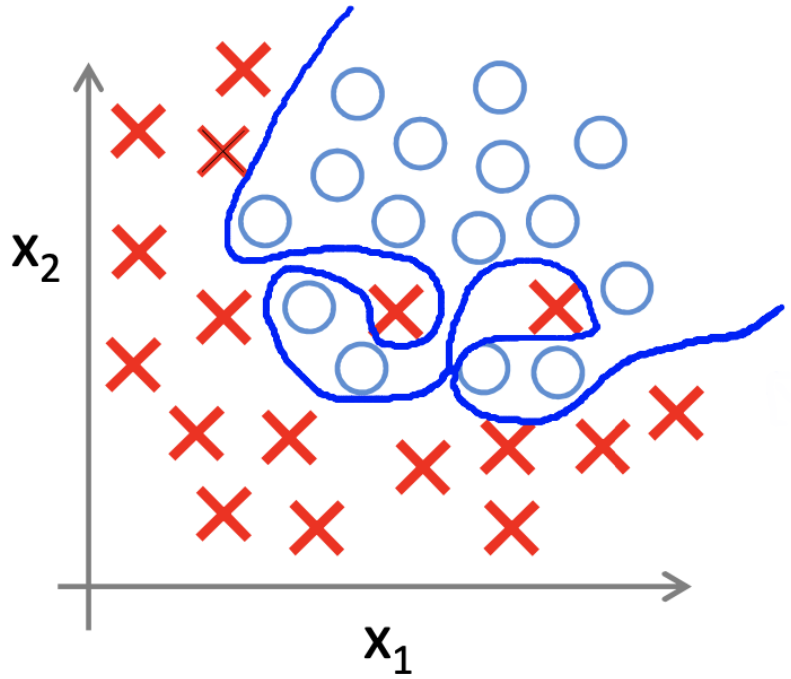
$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

"Underfit"

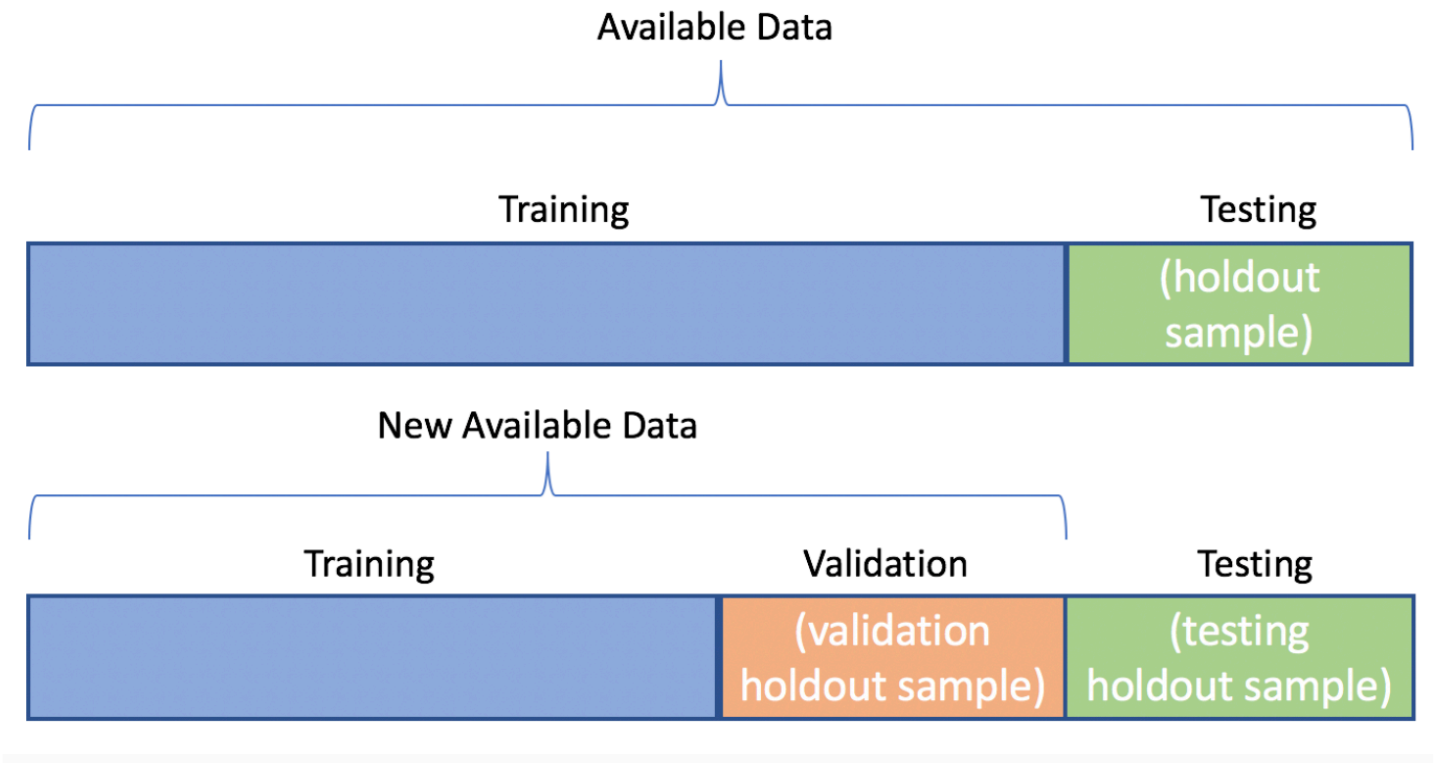


$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$



$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$

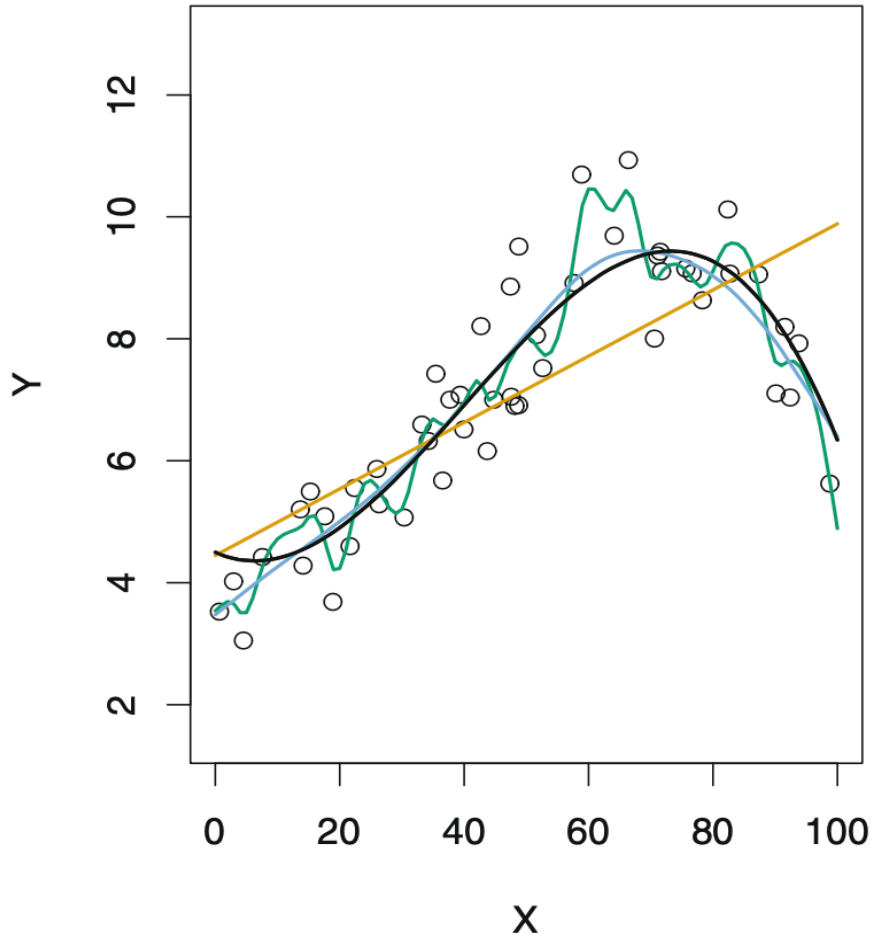
"Overfit"

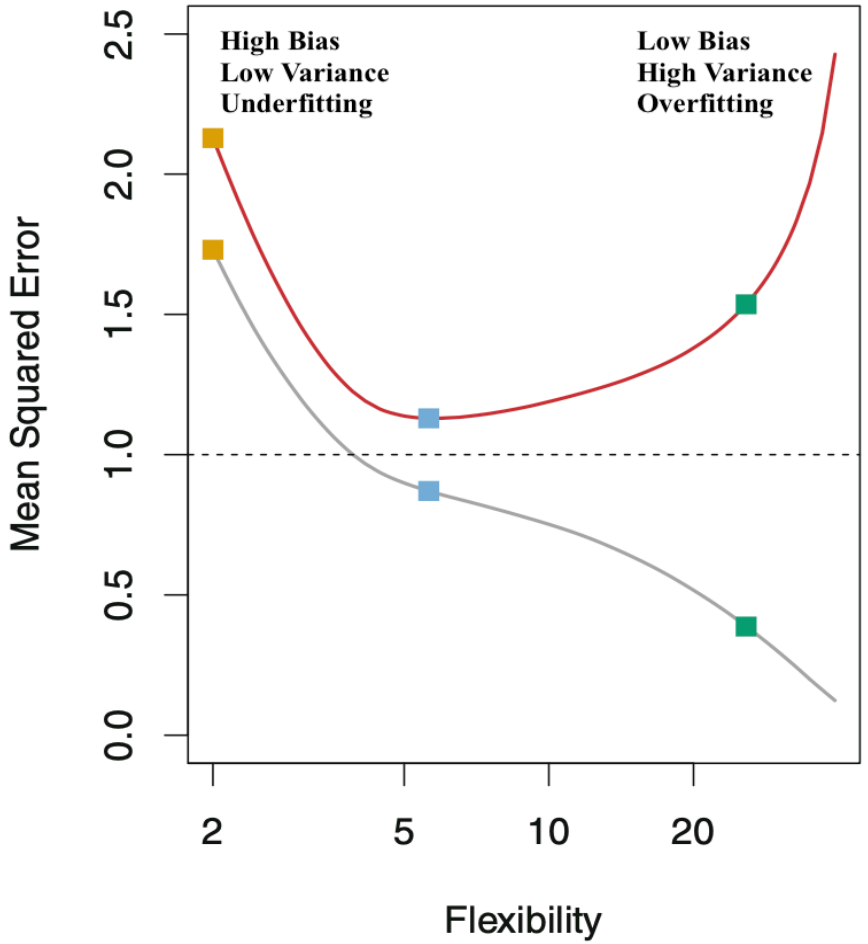


The **training set** is the set of data we analyse (train on) to design the rules in the model.

The **validation set** is a set of data that we did not use when training our model that we use to assess how well these rules perform on new data. It is also a set we **use to tune parameters** and **input features** for our model so that it gives us what we think is the best performance possible for new data.

The **test set** is a set of data we did not use to train our model or use in the validation set to inform our choice of parameters/input features. We will use it as a **final test once we have decided on our final model,** to get the best possible estimate of how successful our model will be when used on entirely new data.



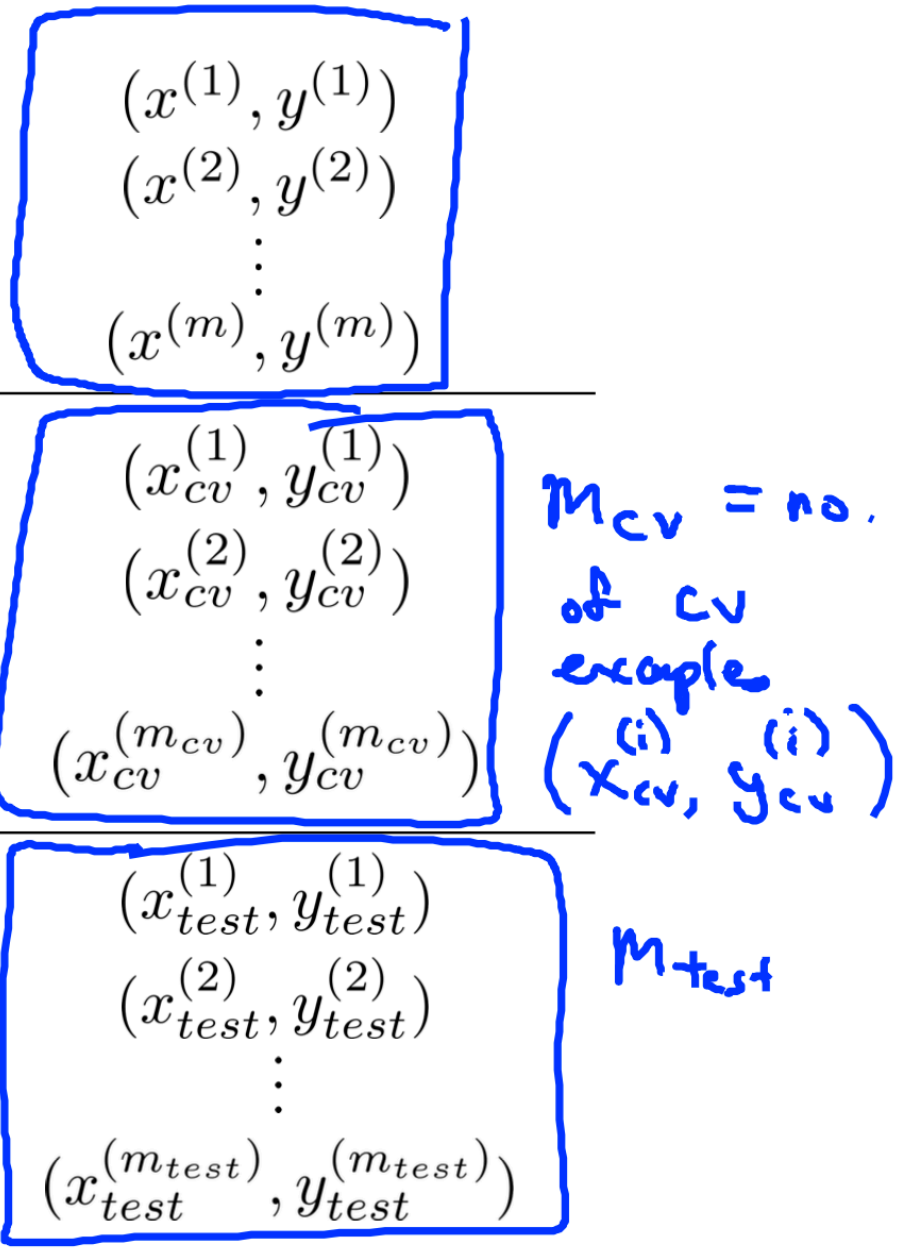


Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Evaluating your hypothesis

Dataset:

Size	Price	
2104	400	Training set (60%)
1600	330	
2400	369	
1416	232	
3000	540	
1985	300	
1534	315	Cross validation set (CV) (20%)
1427	199	
1380	212	test set (20%)
1494	243	



Model selection

1. $h_\theta(x) = \theta_0 + \theta_1 x$ → $\min_\theta J(\theta)$ → $\theta^{(1)}$ → $J_{cv}(\theta^{(1)})$
2. $h_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$ → $\theta^{(2)}$ → $J_{cv}(\theta^{(2)})$
3. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$ → $\theta^{(3)}$ → $J_{cv}(\theta^{(3)})$

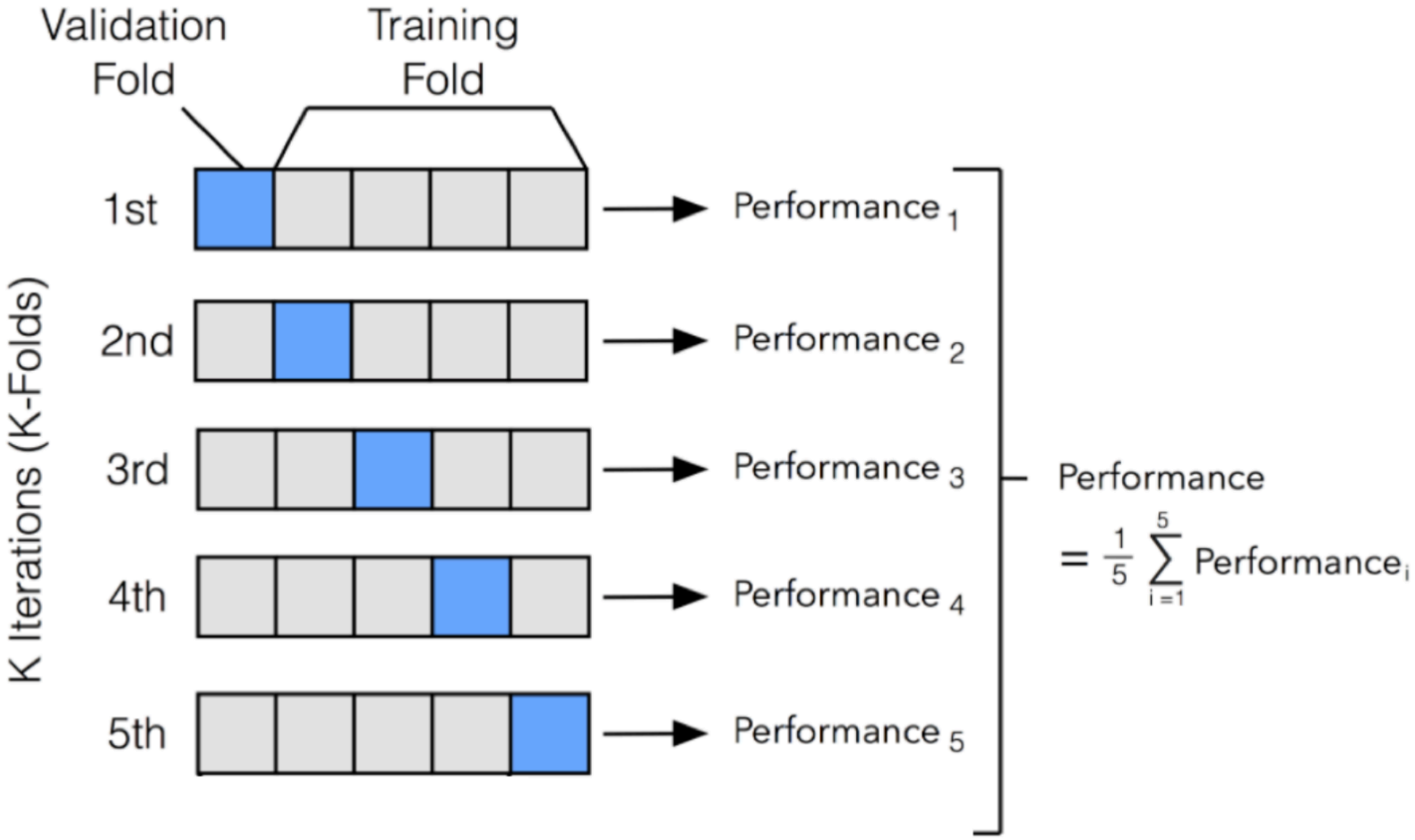
⋮

10. $h_\theta(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$ → $\theta^{(10)}$ → $J_{cv}(\theta^{(10)})$

Pick $\theta_0 + \theta_1 x_1 + \cdots + \theta_4 x^4$

Estimate generalization error for test set $J_{test}(\theta^{(4)})$ ←

K-Fold Cross Validation



Ridge and Lasso Regression

In regularized linear regression, we choose θ to minimize

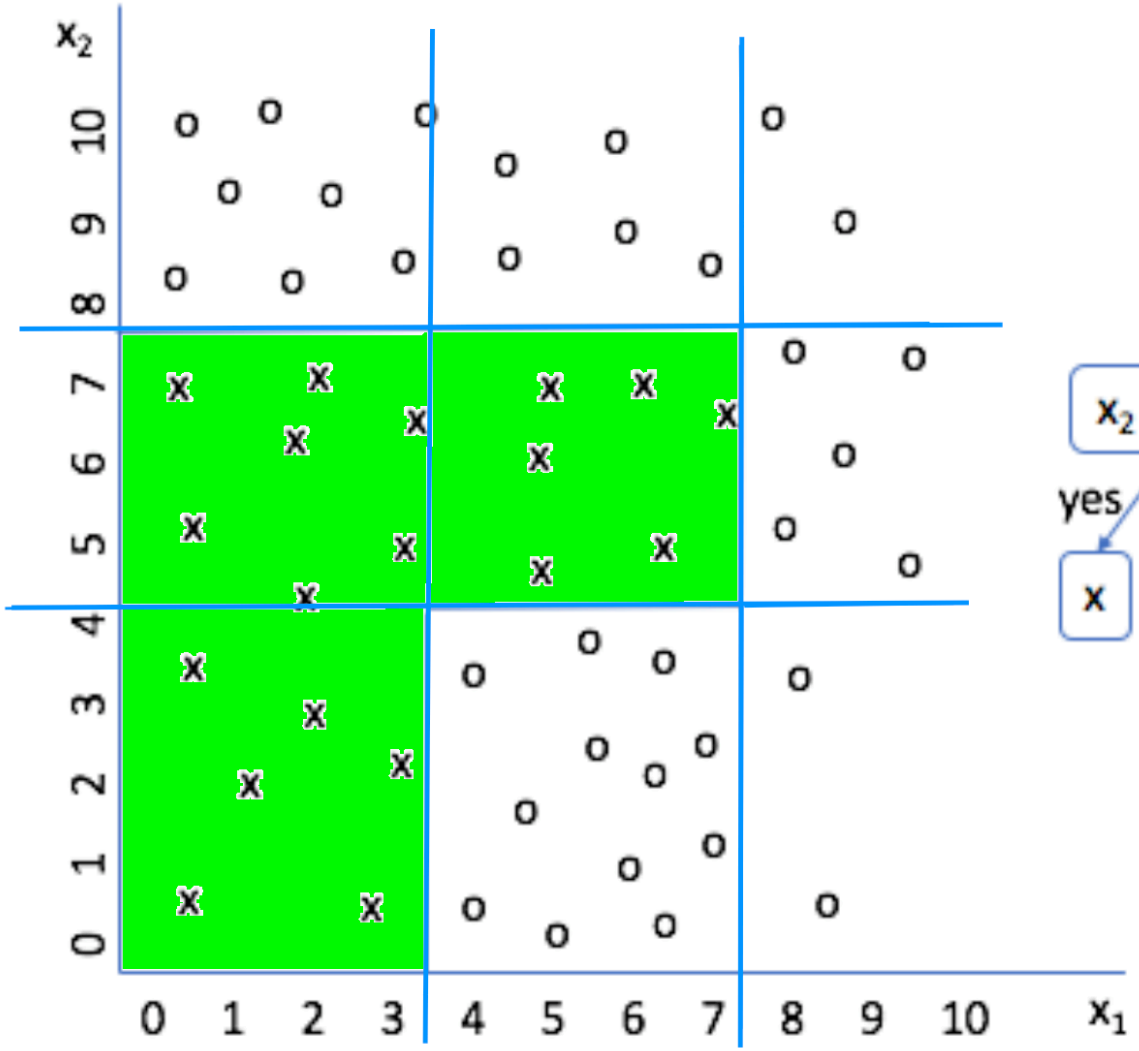
$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}\theta_j^2\right]$$

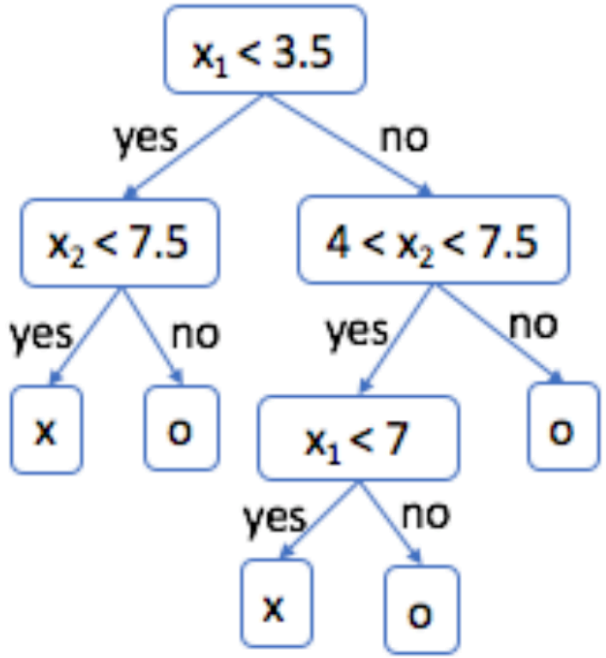
Ridge regression shrinks the coefficients and it helps to reduce the model complexity and multi-collinearity.

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n}|\theta|\right]$$

Lasso regression can lead to zero coefficients i.e. some of the features are completely neglected for the evaluation of output. So Lasso regression not only helps in reducing over-fitting but it can help us in feature selection.

Decision Tree





Decision Tree

Heuristic construction rules were invented to construct decision tree

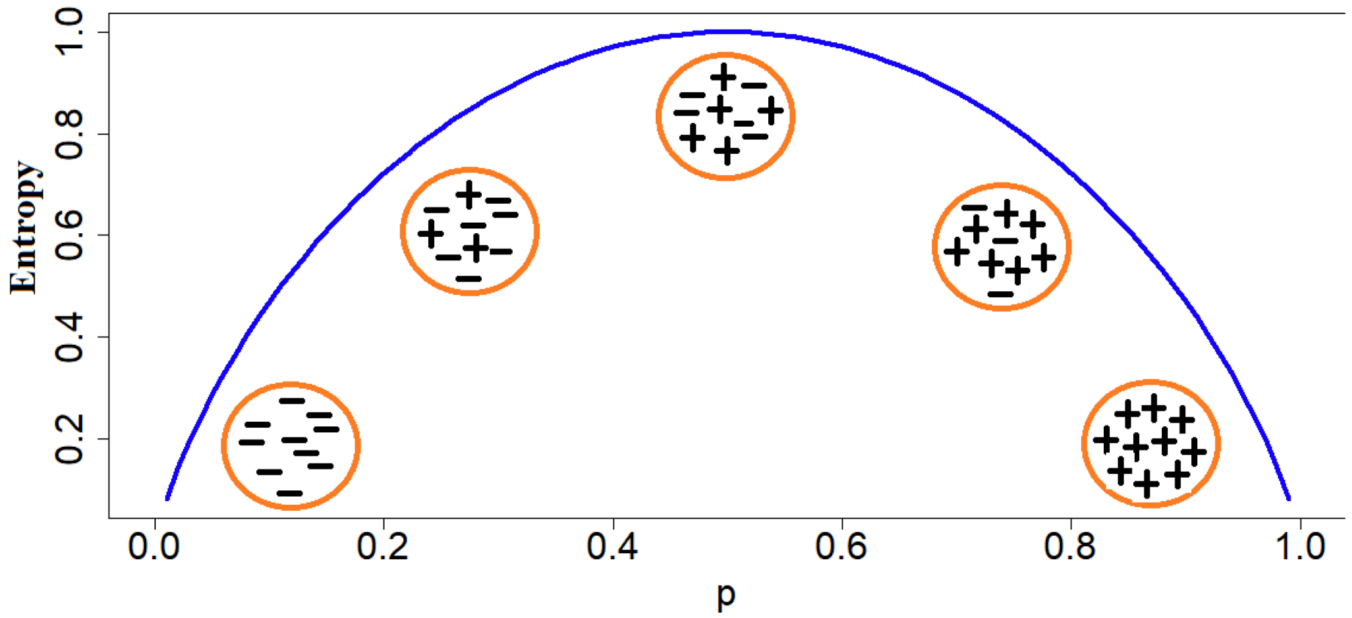
One of them: Select the split with the **lowest entropy (or Gini index)** or **highest information gain**

- Formally, entropy is a number, calculate with the following expression:

$$E(S) = \sum_{i=1}^{K} -p_i \log_2 p_i$$

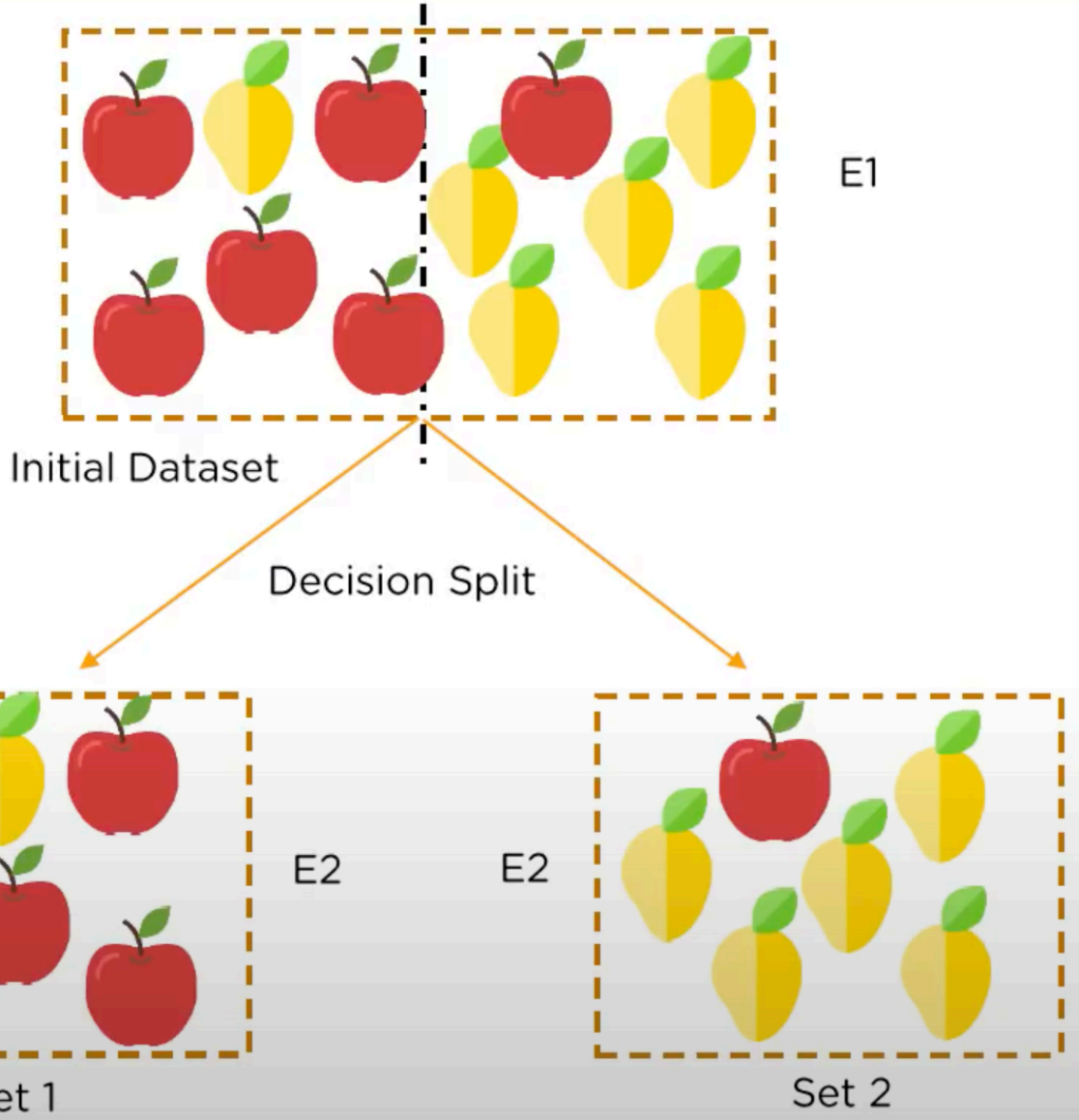
p_i - is ith class proportion in data set S. K – number of different classes.

- The larger the entropy, the bigger the mess (≤ 1)



Information gain

It is the measure of decrease in entropy after the dataset is split



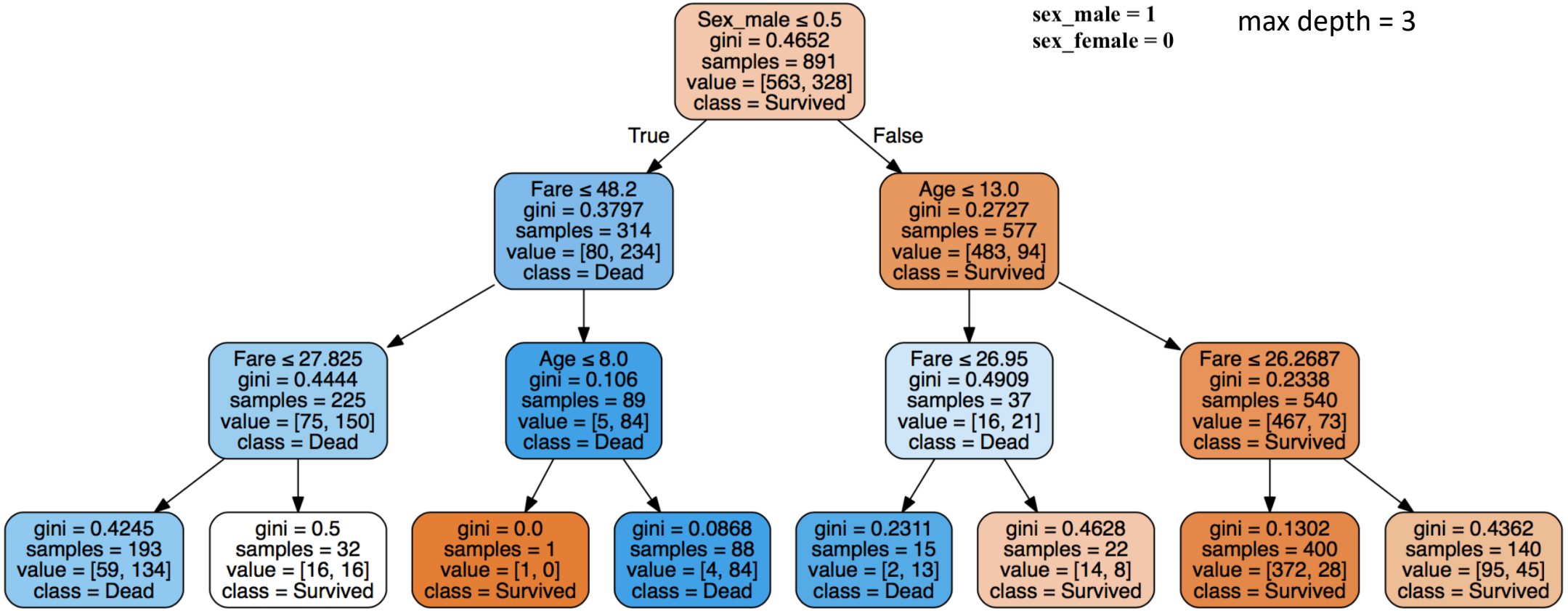
High entropy E1

After splitting

Lower entropy E2

Information gain= E1-E2

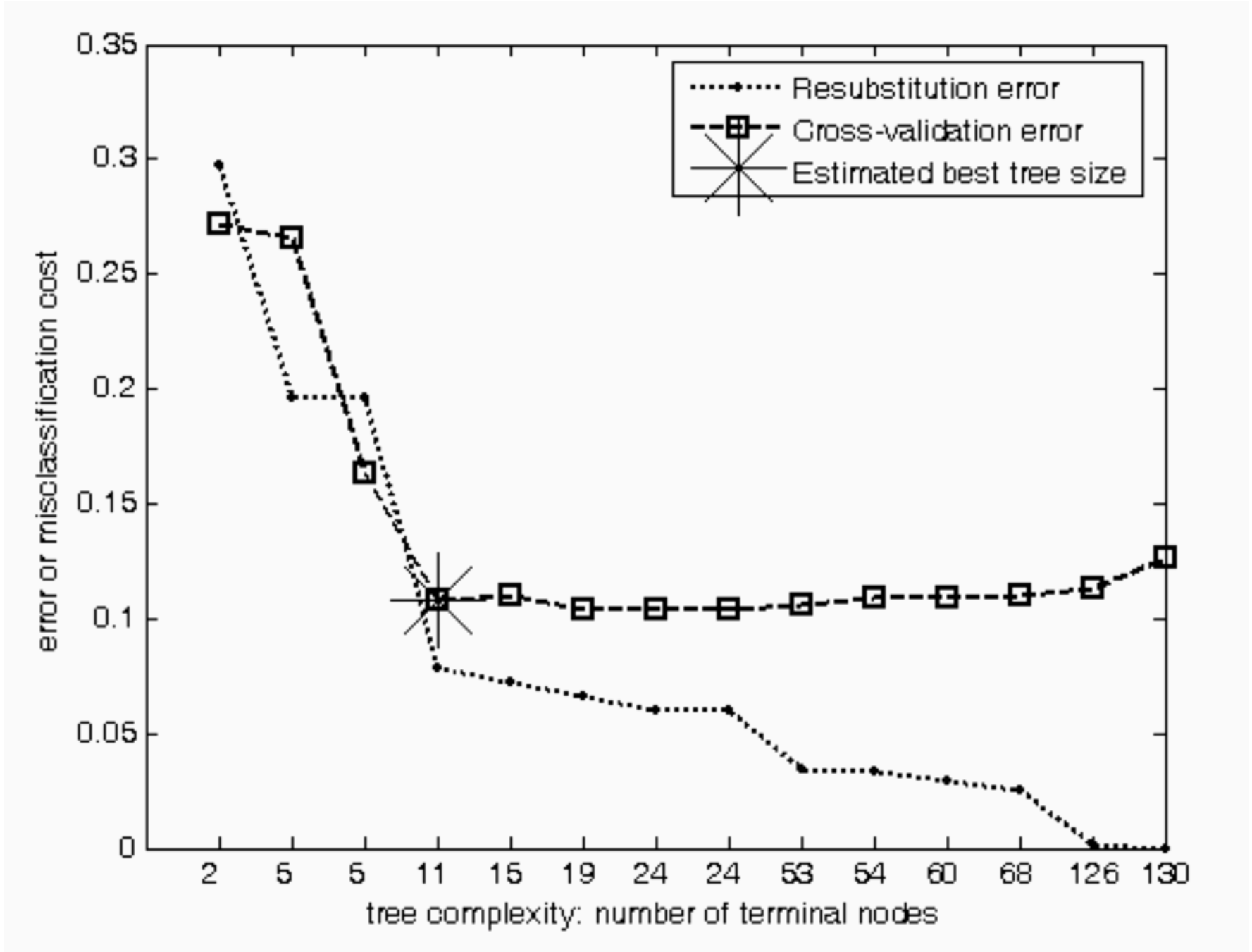
Would You Survive the Titanic?



sex_male = 1
sex_female = 0

max depth = 3

Resubstitution error: $\mathrm{R}(T) = \frac{(16+0+8+28+45)+(59+4+2)}{891} = \frac{162}{891} = 0.18$


Resubstitution error
Cross-validation error
Estimated best tree size
error or misclassification cost
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
2
5
5
11
15
19
24
24
53
54
60
68
126
130
tree complexity: number of terminal nodes

Support Vector Machine

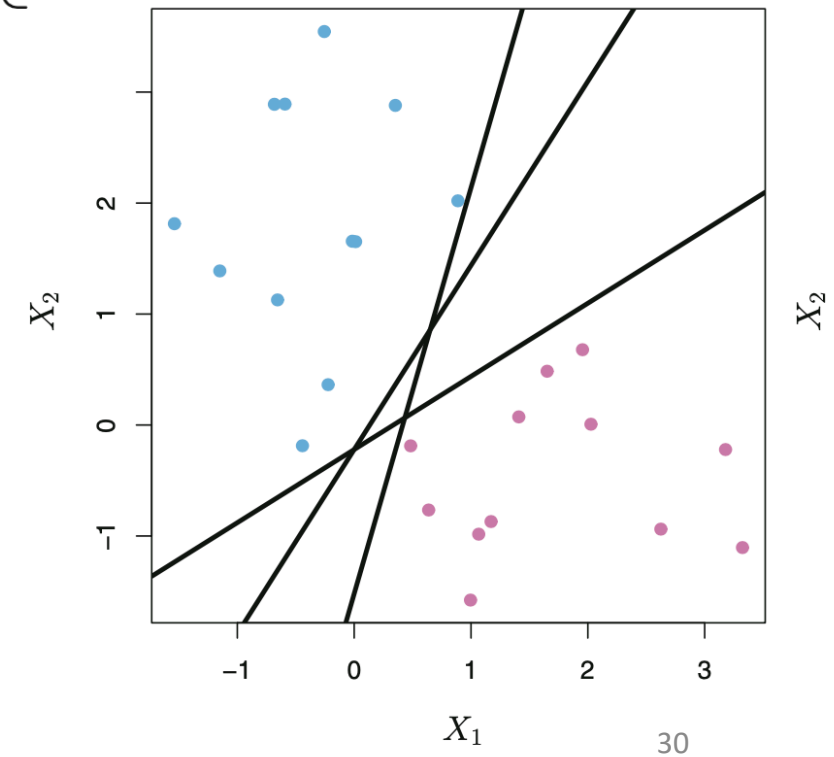
Now suppose that we have a $n \times p$ data matrix $\mathbf{X}$ that consists of n training observations in p-dimensional space,

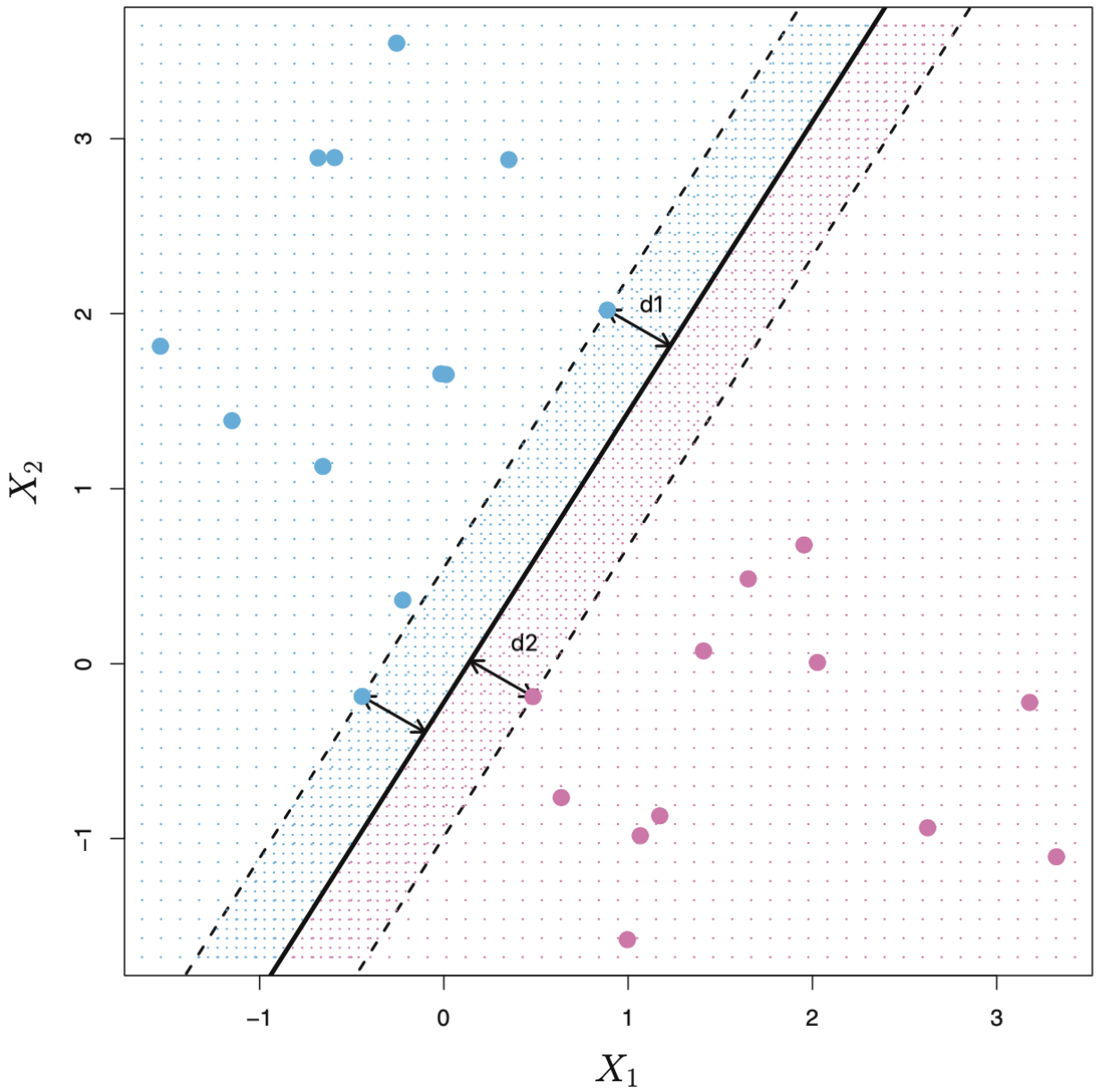
$$x_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1p} \end{pmatrix}, \ldots, x_n = \begin{pmatrix} x_{n1} \\ \vdots \\ x_{np} \end{pmatrix},$$

and that these observations fall into two classes—that is, $y_1, \ldots, y_n \in \{-1, 1\}$ where -1 represents one class and 1 the other class.

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1.$$





$$M = d_1 + d_2$$

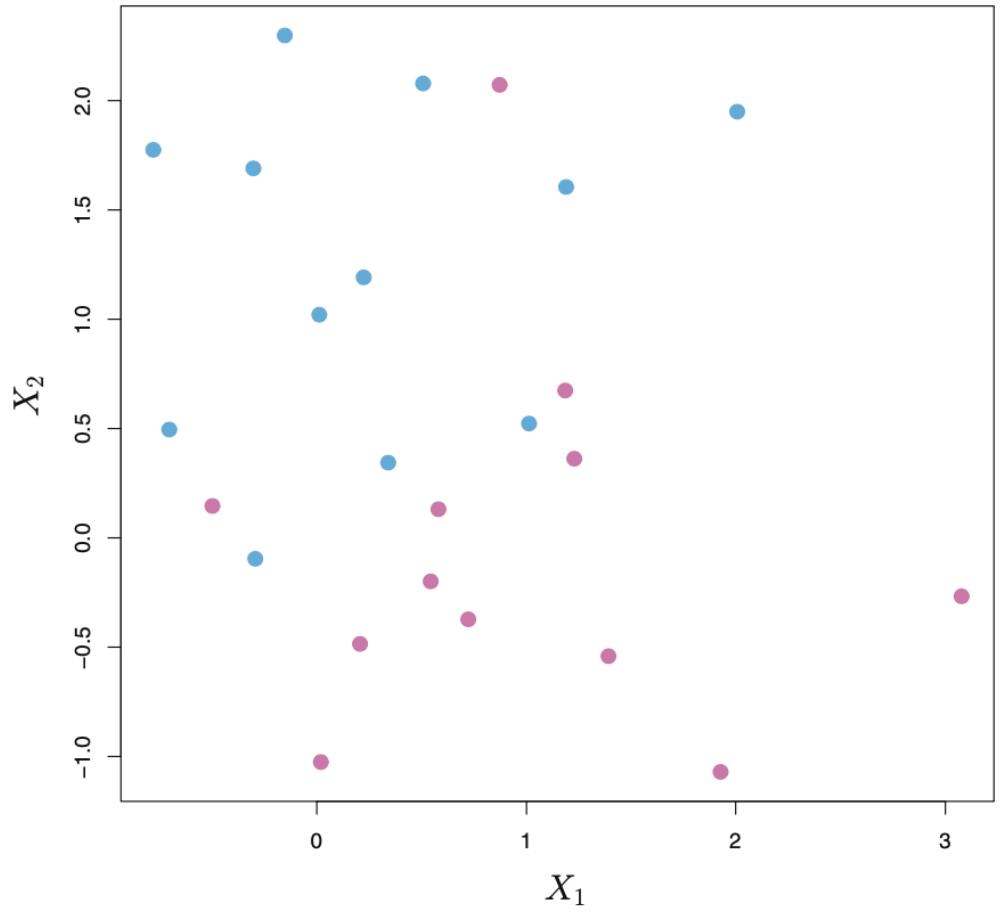
$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \; M$$

$$\text{subject to} \sum_{j=1}^{p} \beta_j^2 = 1,$$

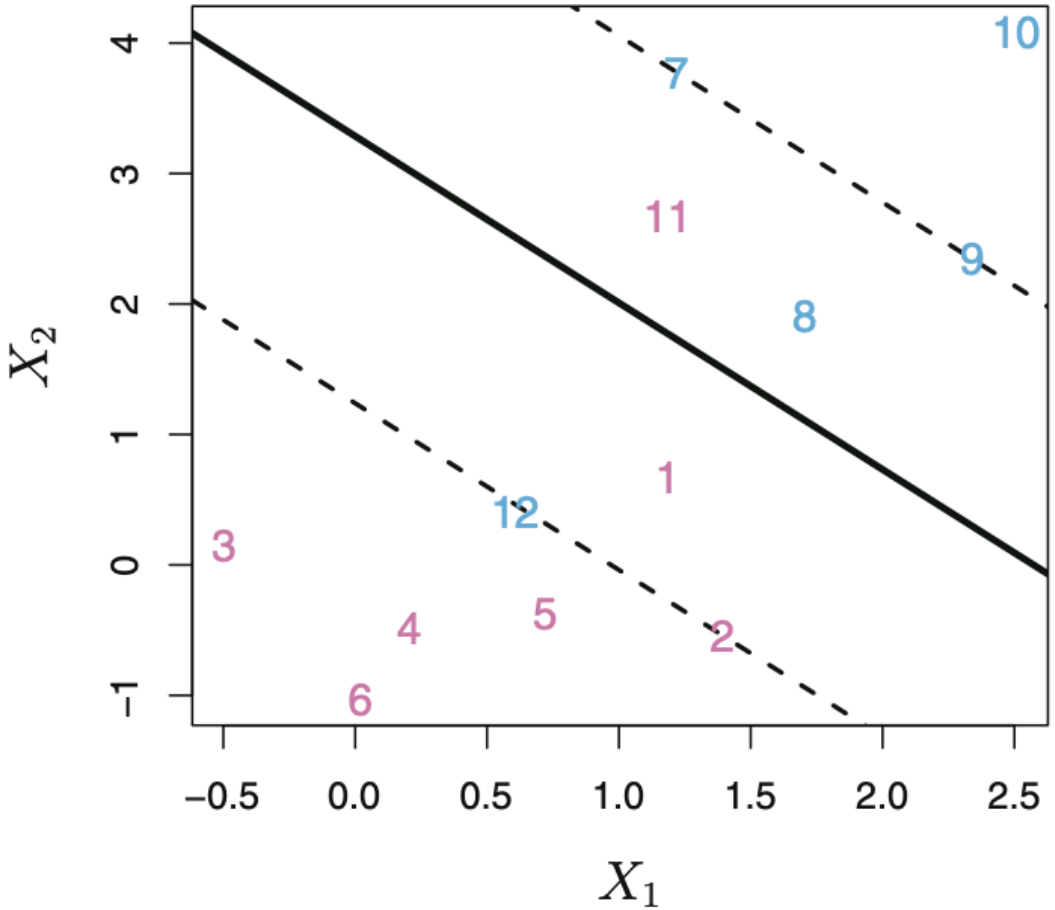
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M$$

$$\forall\, i = 1, \ldots, n.$$

Support Vector Classifiers



There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.



$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

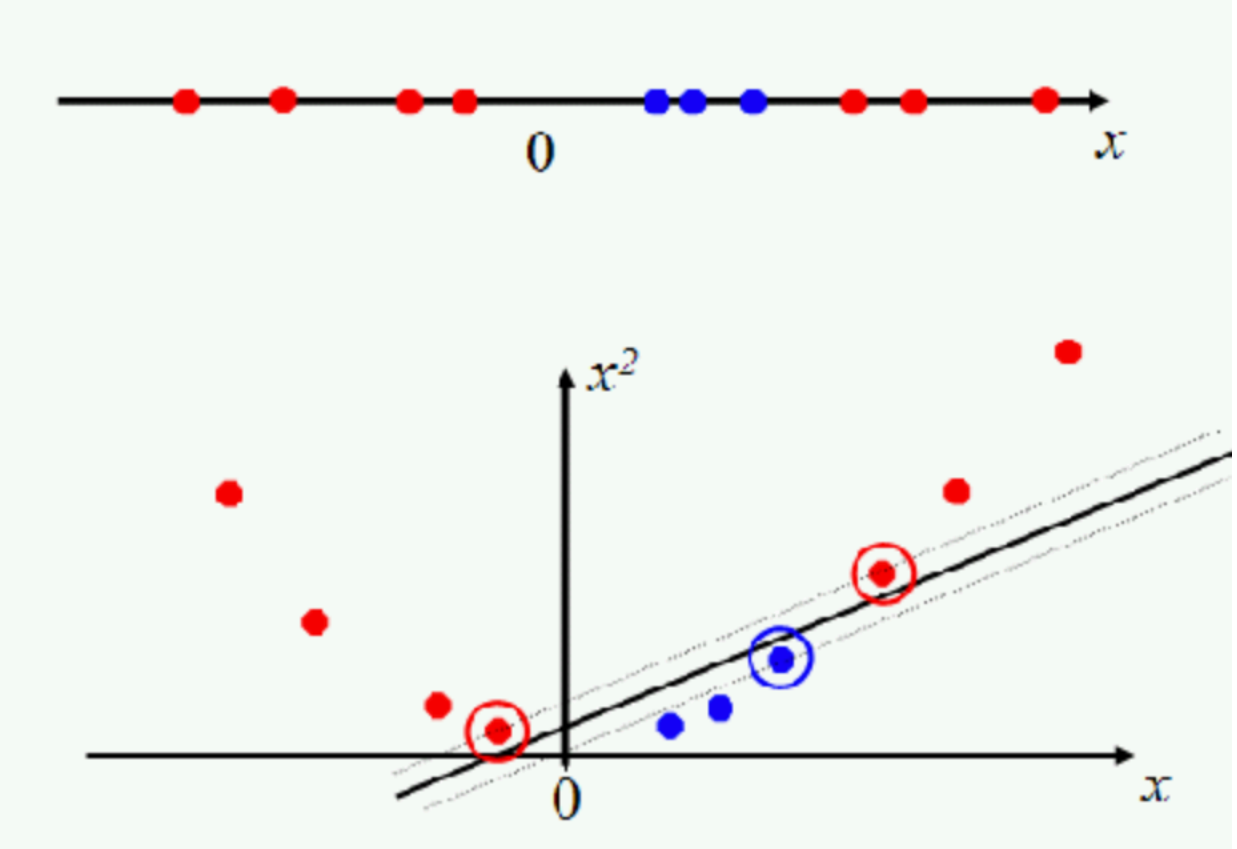
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

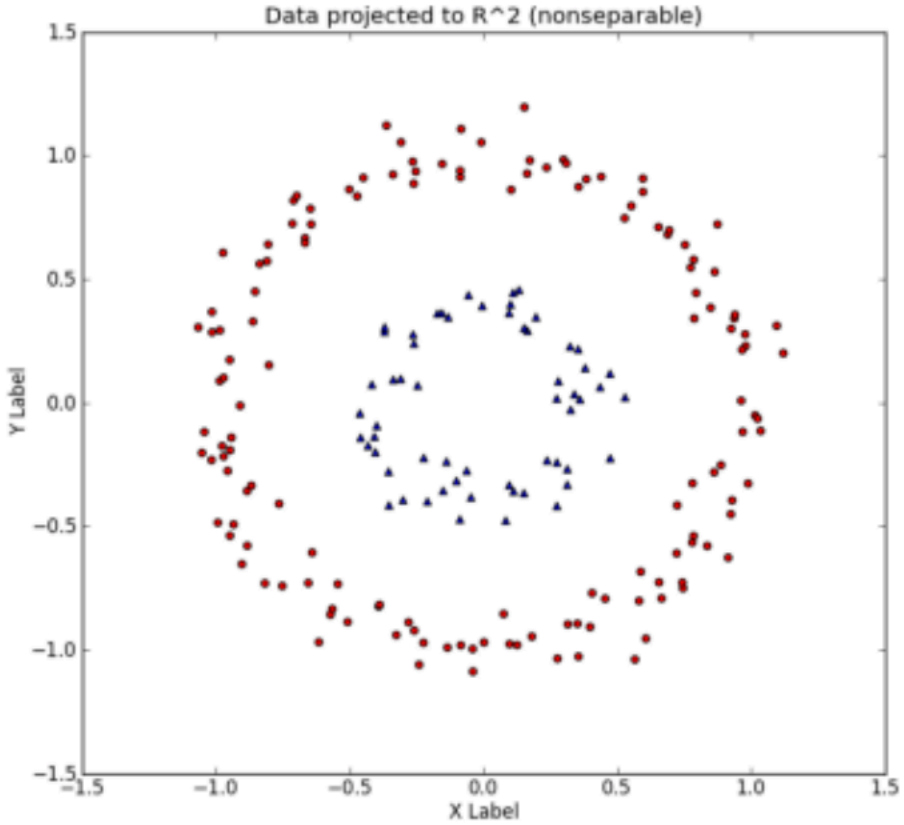
$\epsilon_1, \ldots, \epsilon_n$ are *slack variables* that allow individual observations to be on the wrong side of the margin or the hyperplane;

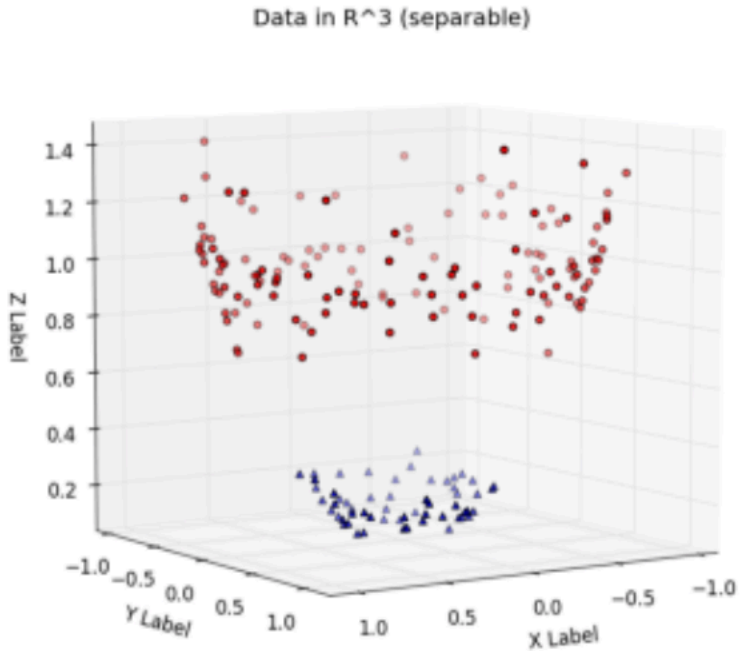
If $\epsilon_i > 0$ then the ith observation is on the wrong side of the margin, and we say that the ith observation has *violated* the margin. If $\epsilon_i > 1$ then it is on the wrong side of the hyperplane.

Classification with Non-linear Decision Boundaries



Classification with Non-linear Decision Boundaries



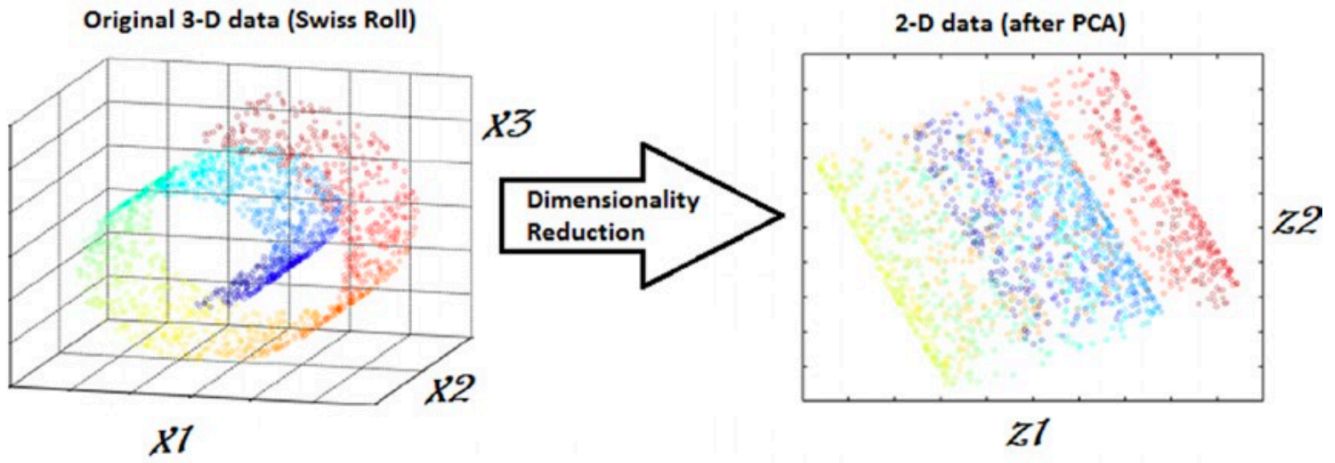


Performing Non-Linear Classification using Kernel Method

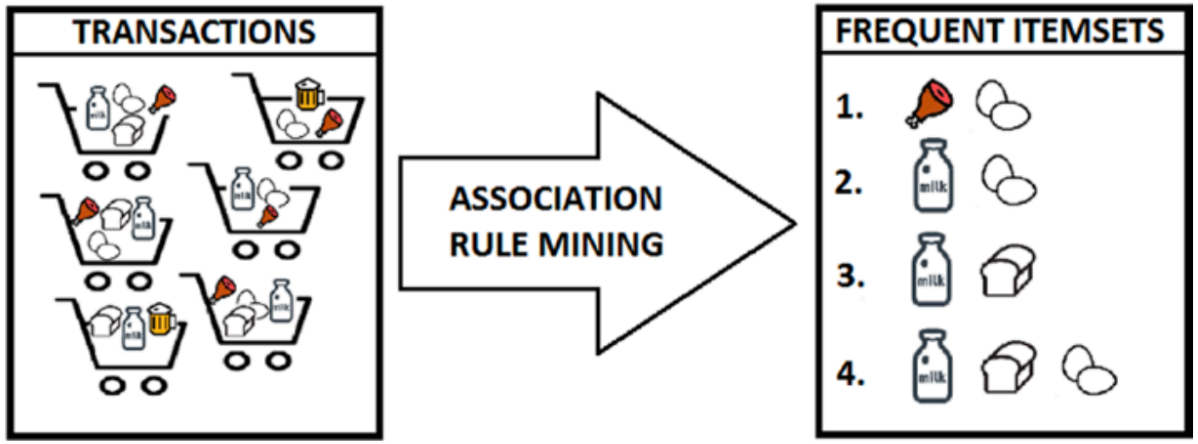
Unsupervised Learning

Clustering

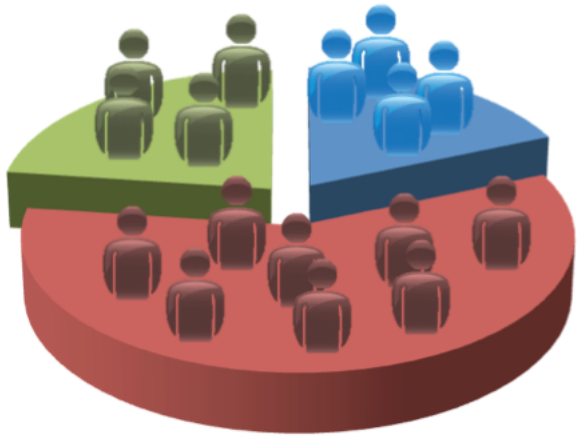
Dimensionality reduction (PCA)



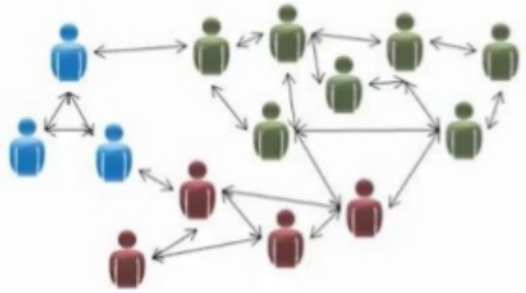
Unsupervised learning: dimensionality reduction



Unsupervised learning: association rule-mining



Market segmentation



Social network analysis

K-Means Clustering

We have a set of features $X_1, X_2, \ldots, X_p$ measured on n observations.

The K-means clustering procedure results from a simple and intuitive mathematical problem. We begin by defining some notation. Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:

1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

For instance, if the ith observation is in the kth cluster, then $i \in C_k$. The idea behind K-means clustering is that a *good* clustering is one for which the *within-cluster variation* is as small as possible. The within-cluster variation for cluster C_k is a measure $W(C_k)$ of the amount by which the observations within a cluster differ from each other. Hence we want to solve the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}.$$

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2,$$

where $|C_k|$ denotes the number of observations in the kth cluster. In other words, the within-cluster variation for the kth cluster is the sum of all of the pairwise squared Euclidean distances between the observations in the kth cluster, divided by the total number of observations in the kth cluster.

K-means clustering,

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}.$$

Principal Component Analysis

Given a $n \times p$ data set $\mathbf{X}$, how do we compute the first principal component? Since we are only interested in variance, we assume that each of the variables in $\mathbf{X}$ has been centered to have mean zero (that is, the column means of $\mathbf{X}$ are zero). We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. In other words, the first principal component loading vector solves the optimization problem

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1.$$