# Support Vector Machines

Machine Learning II

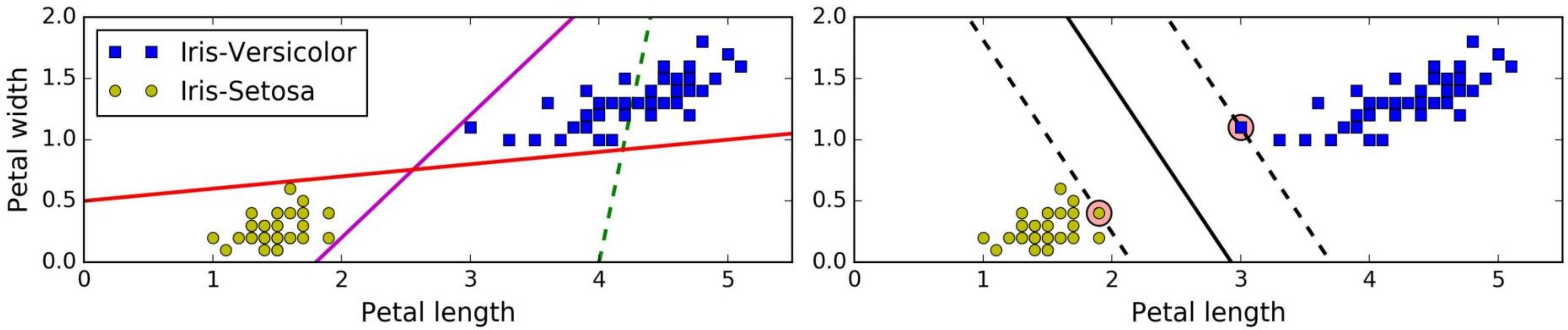
Master in Business Analytics and Big Data

acastellanos@faculty.ie.edu

# Goals

- SVM intuition
  - Large margin classifiers
  - SVM decission boundary
- Sneak preview of kernels
- When is OK to use SVMs
- How to use SVMs
- Practical session on SVMs.

# Large Margin Classifier

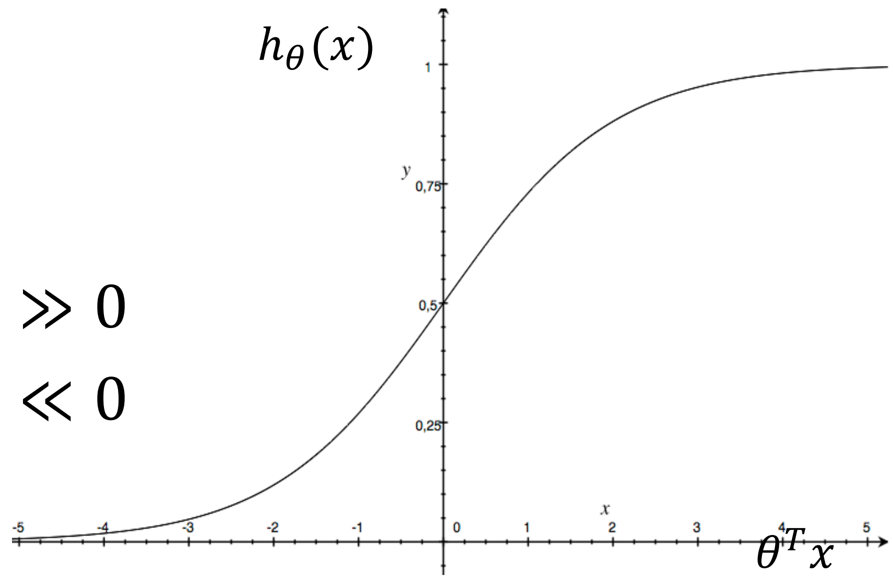


# Logistic Regression

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

If $y = 1$, we want $h_\theta(x) \approx 1,\ \theta^T x \gg 0$

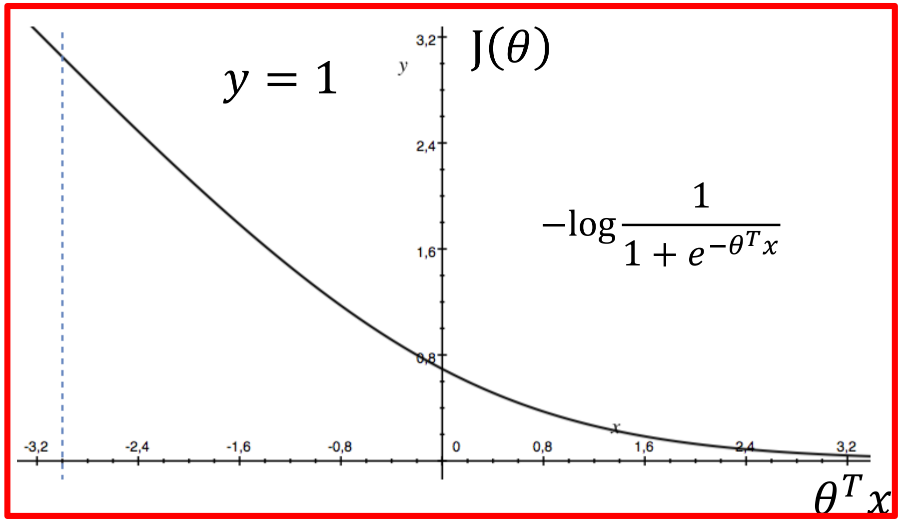
If $y = 0$, we want $h_\theta(x) \approx 0,\ \theta^T x \ll 0$

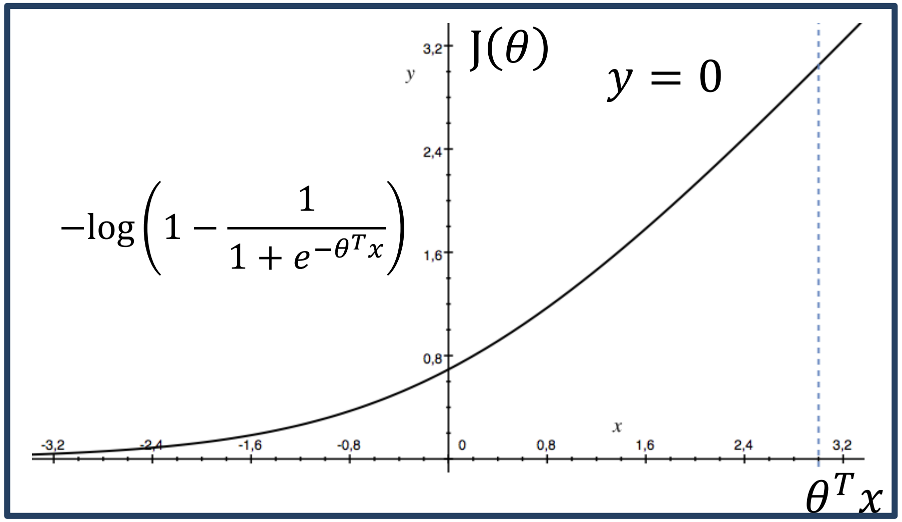


# Cost function in logistic regression

$$J(\theta) = -\left(\mathrm{y} \cdot \log h_\theta(\mathrm{x}) + (1-\mathrm{y}) \cdot \log\left(1 - h_\theta(\mathrm{x})\right)\right)$$

$$= \boxed{-\mathrm{y} \log \frac{1}{1+e^{-\theta^T x}}} - \boxed{(1-\mathrm{y}) \log\left(1 - \frac{1}{1+e^{-\theta^T x}}\right)}$$



# SVM Hypothesis

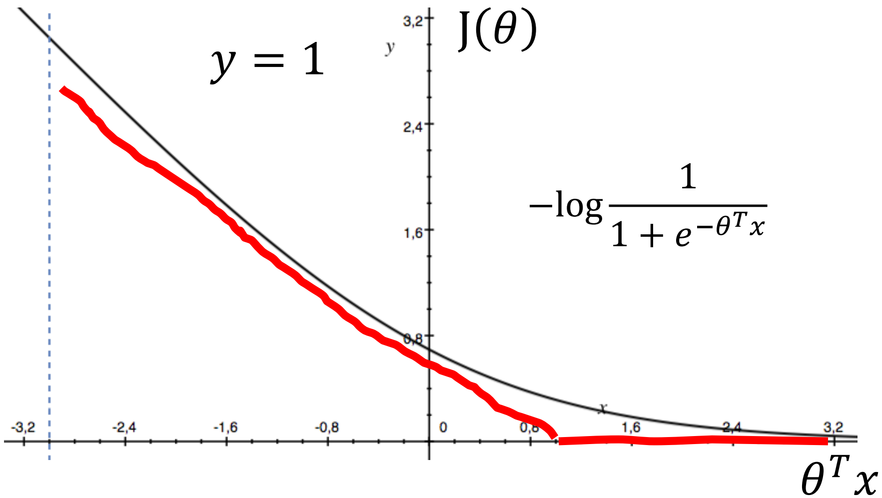
- The hypothesis of the Support Vector Machine is not interpreted as the probability of Y being 1 or 0 (as it is for the hypothesis of logistic regression).

- Instead, it outputs either 1 or 0:

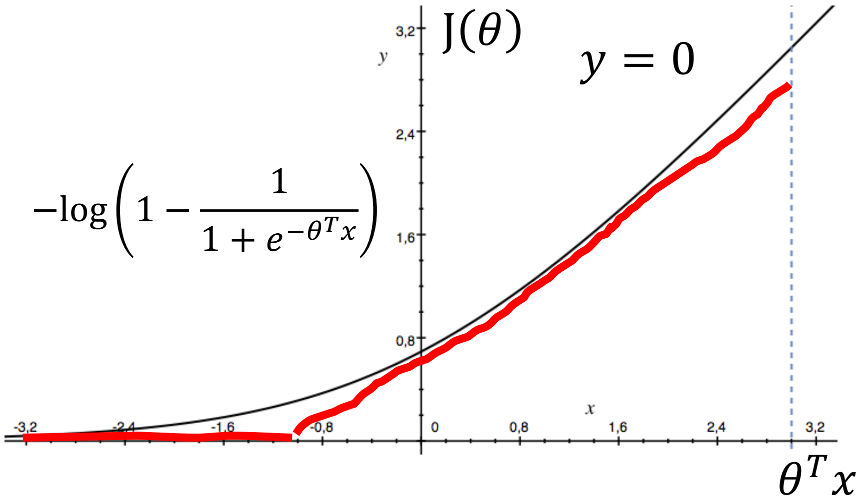
$$h_\theta(x) = \begin{cases} 1 & if\ \theta^T x \geq 0 \\ 0 & otherwise \end{cases}$$

# Cost function in SVM

**Cost1(z)**




**Cost0(z)**



# Support Vector Machines

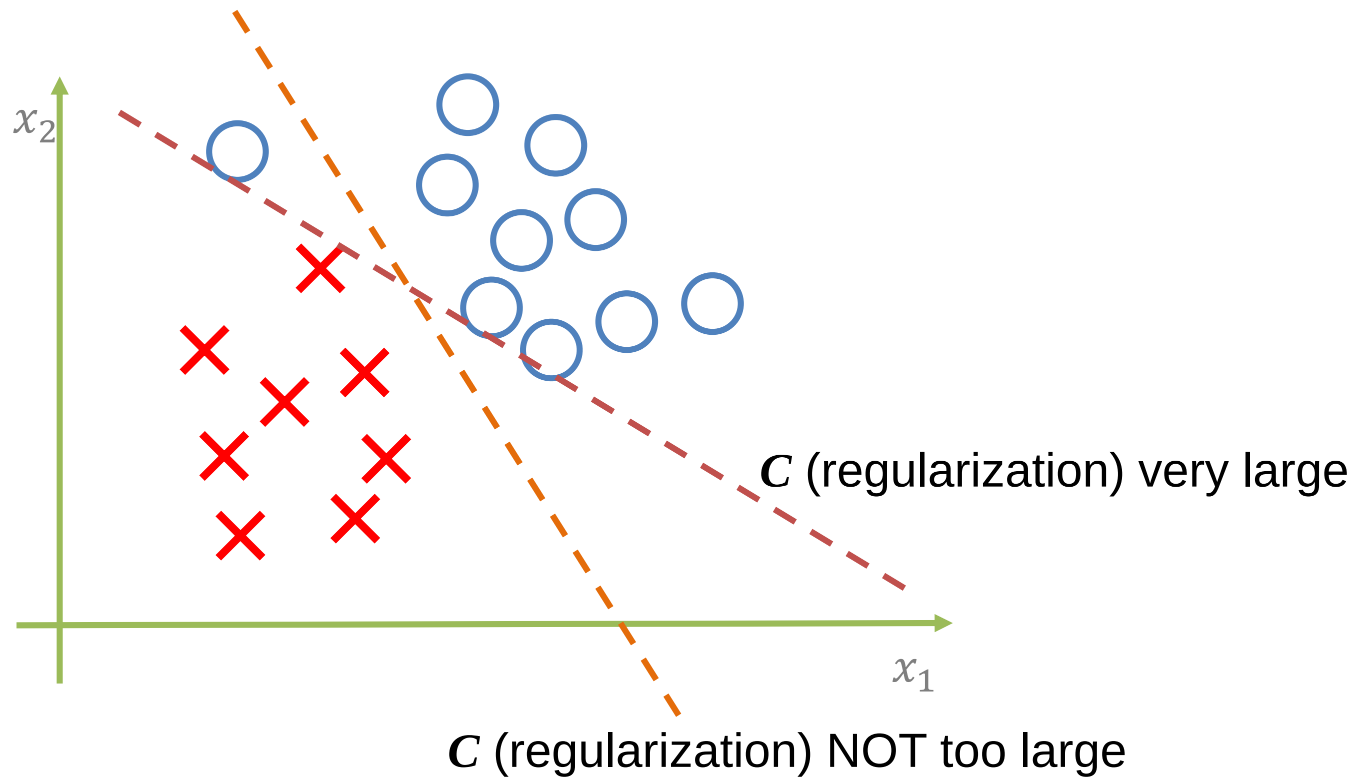
## Logistic Regression

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

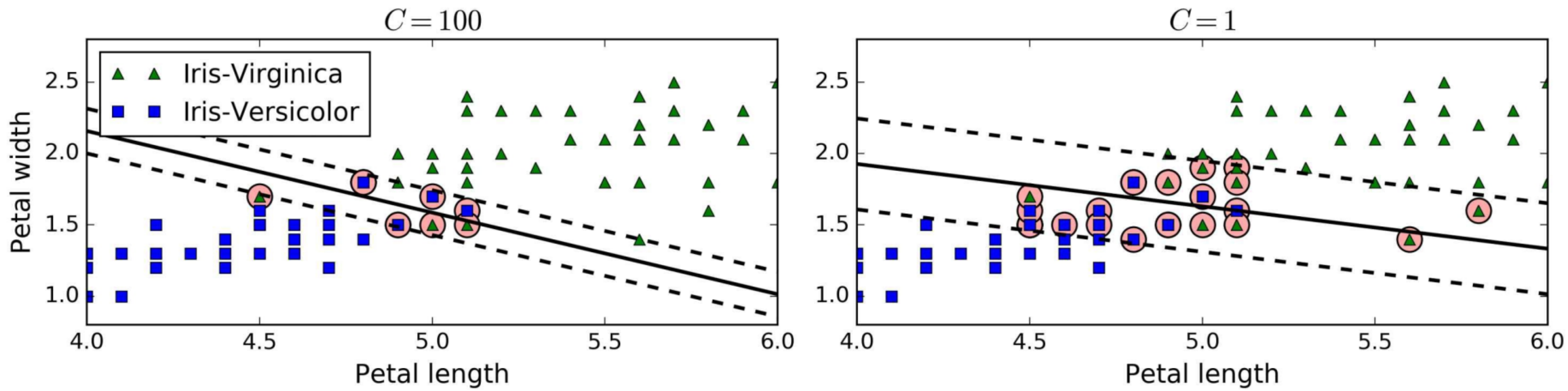
## SVM

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)} cost_1\left(\theta^T x^{(i)}\right) + \left(1-y^{(i)}\right) cost_0\left(\theta^T x^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

# SVM Decision boundary

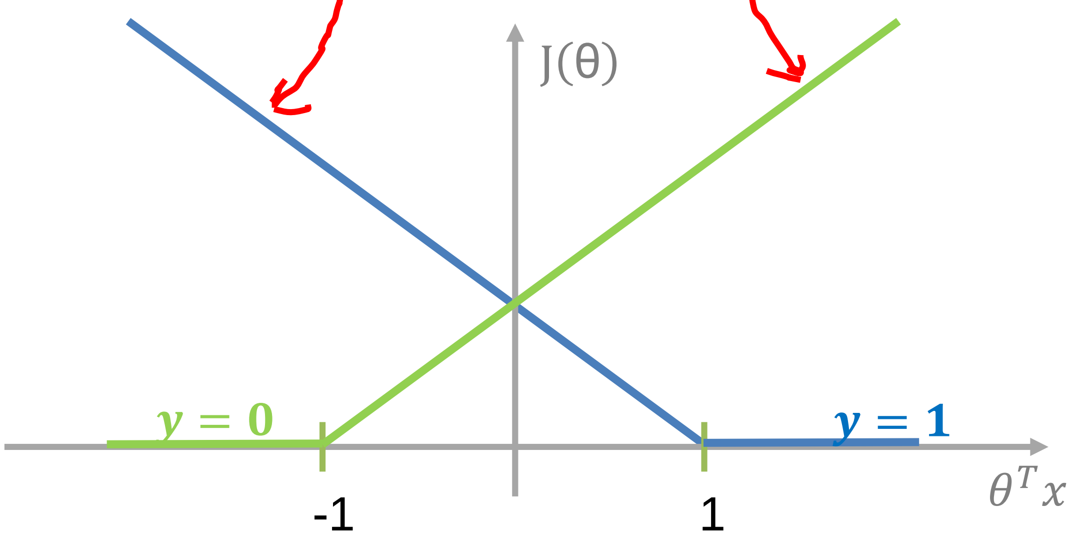


# SVM Decision boundary



# Large margin classifier

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

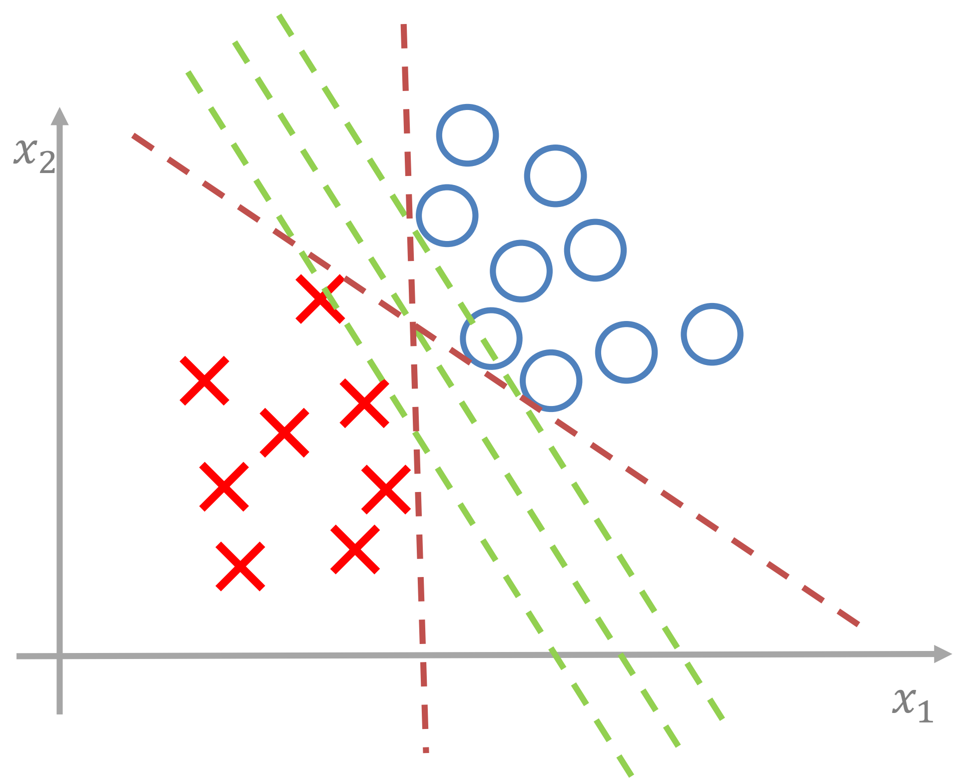



$$\min \frac{1}{2}\sum_{i=1}^{n} \theta_j^2$$

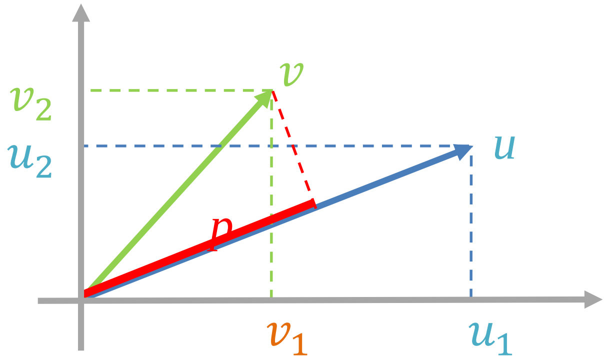
$$\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$

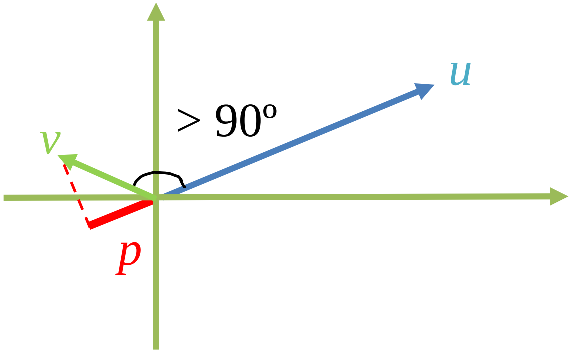
$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

# SVM Decision boundary



# Math behind large margin (1)




$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^T v = ?$$

$\|u\| =$ length of vector $u = \sqrt{u_1^2 + u_2^2}$

$p =$ signed length of projection of $v$ onto $u$.

$$u^T v = v^T u = p \cdot \|u\| = u_1 v_1 + u_2 v_2$$

# Math behind large margin (2)

$$\min_{\theta} C \boxed{\left[\sum_{i=1}^{m} y^{(i)} cost_1\left(\theta^T x^{(i)}\right) + \left(1 - y^{(i)}\right) cost_0\left(\theta^T x^{(i)}\right)\right]} + \boxed{\frac{1}{2} \sum_{j=1}^{n} \theta_j^2}$$

s.t.

$\theta^T x \geq 1$ If $y = 1$

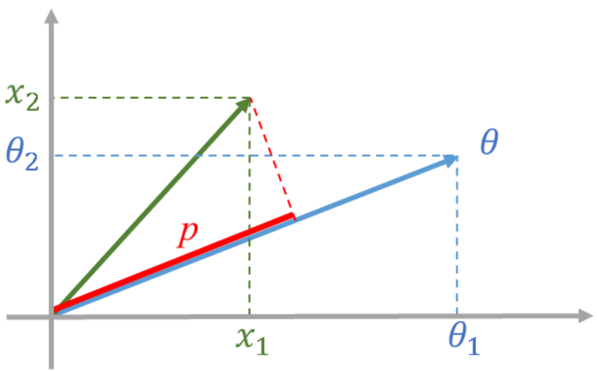
$\theta^T x \leq -1$ If $y = 0$

$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}(\theta_1^2 + \theta_2^2)$$

$$= \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2$$

$$= \frac{1}{2} \|\theta\|^2$$

# Math behind large margin (3)




s.t.

$\theta^T x \geq 1$ If $y = 1$
$\theta^T x \leq$-1 If $y = 0$

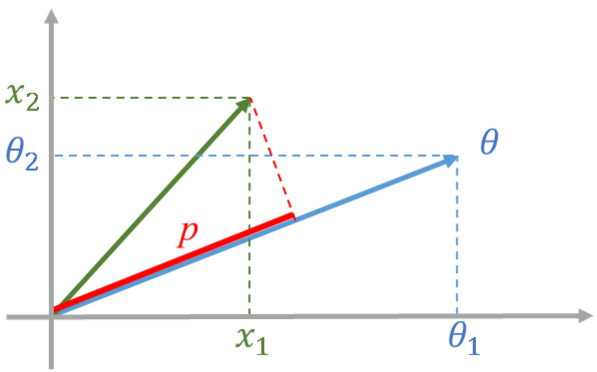
$\theta^T x = p \cdot \|\theta\| = \theta_1 x_1 + \theta_2 x_2$

*We want to maximize the margin ($\gamma$), so we need to maximize this*

s.t.

$p \cdot \|\theta\| \geq 1$ If $y = 1$
$p \cdot \|\theta\| \leq -1$ If $y = 0$

# Math behind large margin (3)




s.t.

$\theta^T x \geq 1$ If $y = 1$
$\theta^T x \leq$-1 If $y = 0$

$$\theta^T x = p \cdot \|\theta\| = \theta_1 x_1 + \theta_2 x_2$$

s.t.

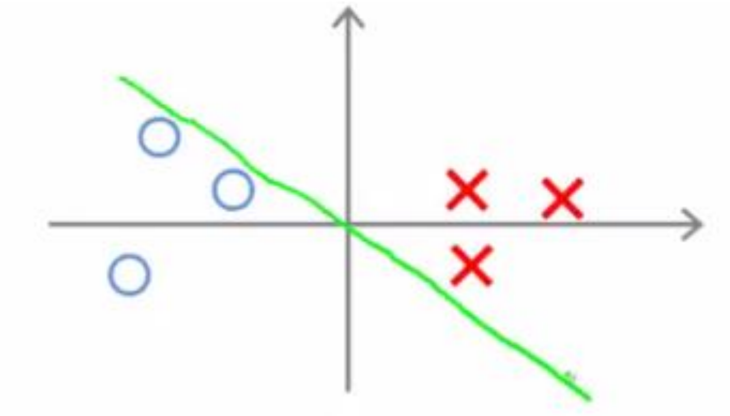
$p \cdot \|\theta\| \geq 1$ If $y = 1$
$p \cdot \|\theta\| \leq -1$ If $y = 0$

*We want to maximize the margin, so we need to minimize this* ***(make θ small)***

$$\min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \min \frac{1}{2} \|\theta\|^2$$

*We want to maximize the margin, so we need to maximize this*
***(make $p \cdot \|\theta\|$ large → make $p$ large)***

# SVM decision boundary

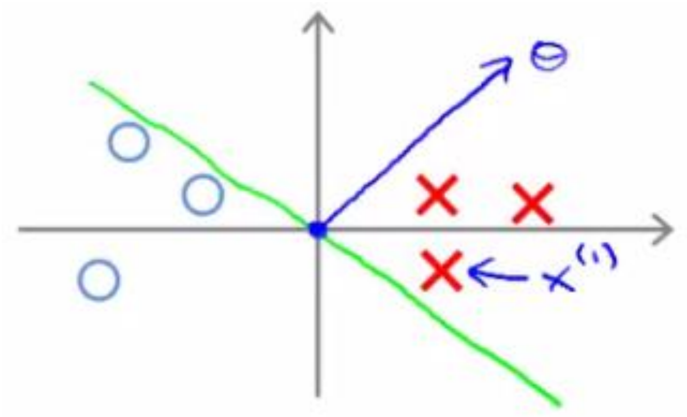


**SVM would not chose this line**

- Decision boundary comes very close to examples
- Lets discuss *why* the SVM would **not** chose this decision boundary

http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

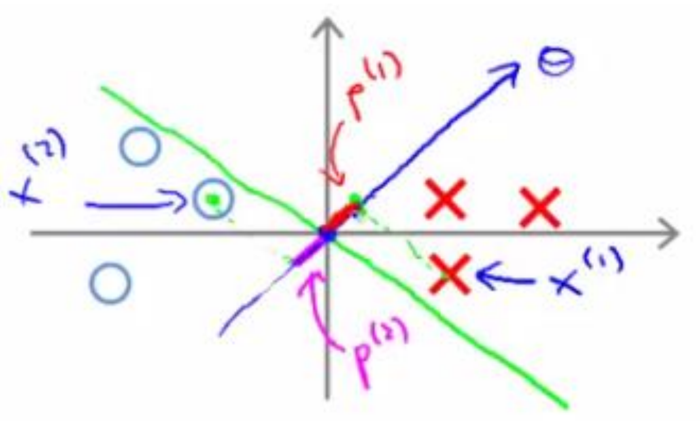
# SVM decision boundary




**Look at first example (x1)**

http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# SVM decision boundary




**Look at first example (x1)**

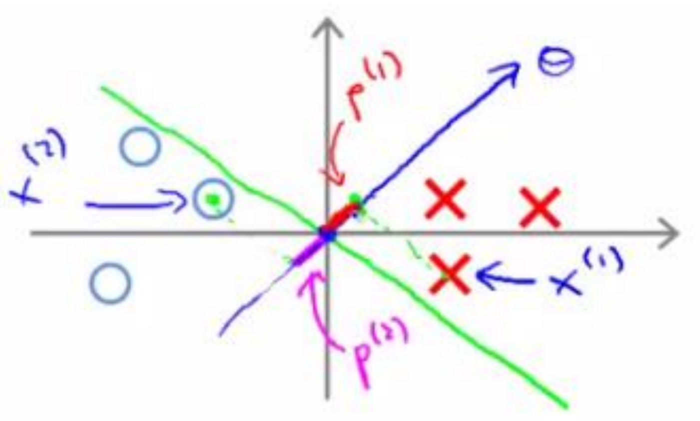
- Project a line from x1 on to to the θ vector (so it hits at 90 degrees)
- The distance between the intersection and the origin is (**p1**)

**Similarly, look at second example (x2)**

- Project a line from x2 into to the θ vector
- This is the magenta line, which will be **negative** (**p2**)

http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# SVM decision boundary




**p values are pretty small**

Look back at our **optimization objective**

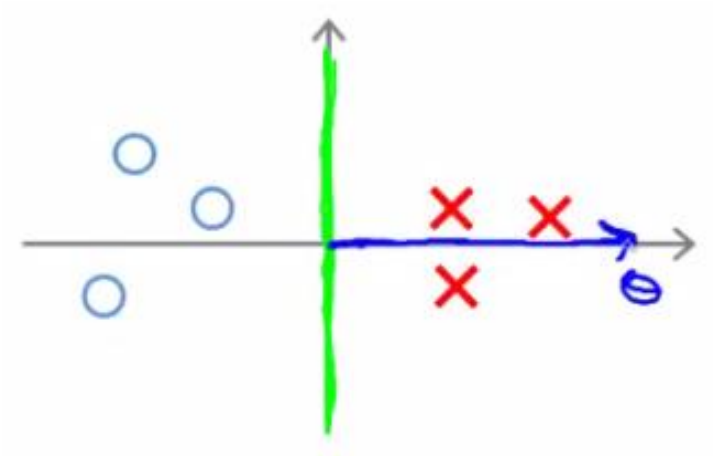
- p1 * ||θ|| needs to be ≥ 1 for positive examples
  - If p is small Means that ||θ|| must be pretty large
- p2 * ||θ|| needs to be ≤ –1 for negative examples
  - p2 is small ||θ|| must be a large number

**Why is this a problem?**

- The **optimization objective** is trying to find a set of parameters where the **norm of theta is small**
  - So this **doesn't seem like a good direction** for the parameter vector (because as p values get smaller ||θ|| must get larger to compensate)
  - We should make p values larger which allows ||θ|| to become smaller

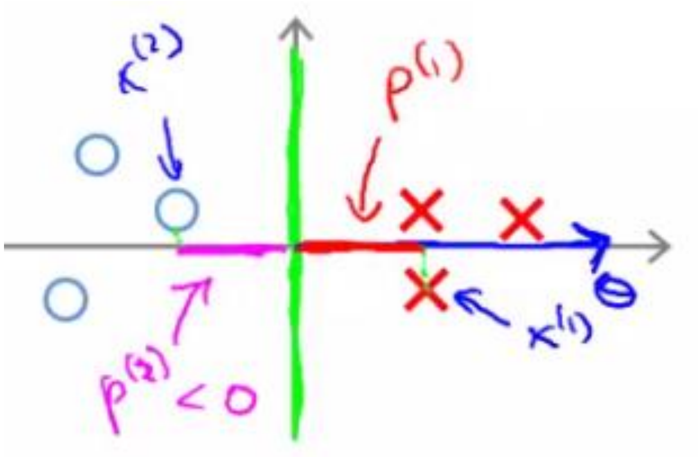
http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# SVM decision boundary



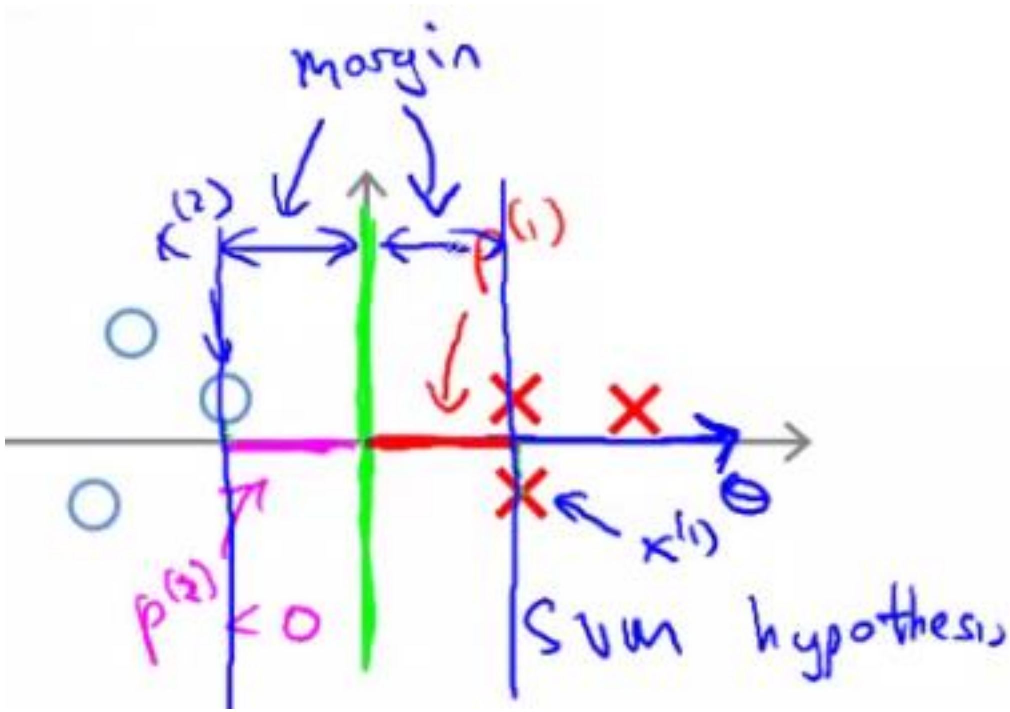
http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# SVM decision boundary



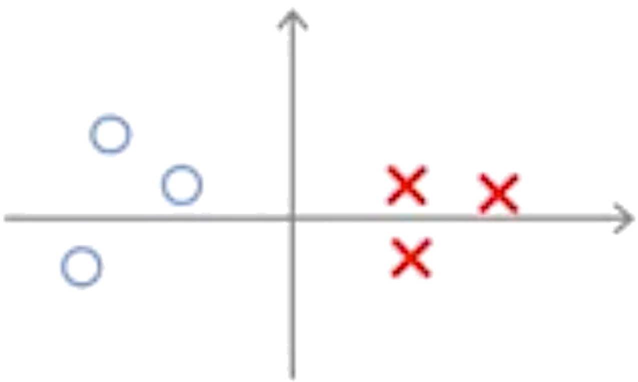

http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

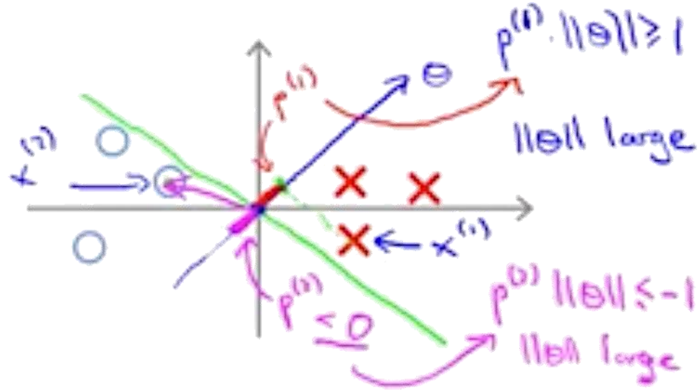
# SVM decision boundary

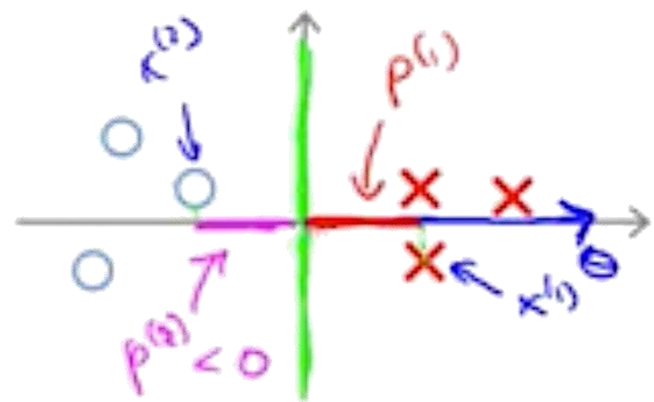



http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# SVM decision boundary

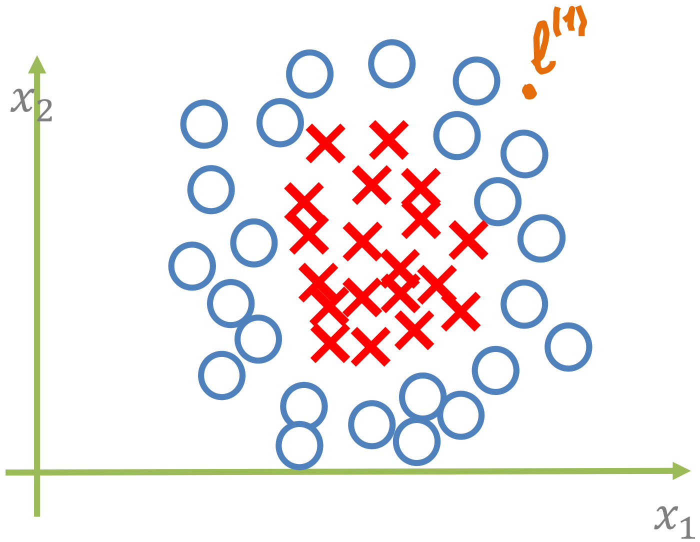




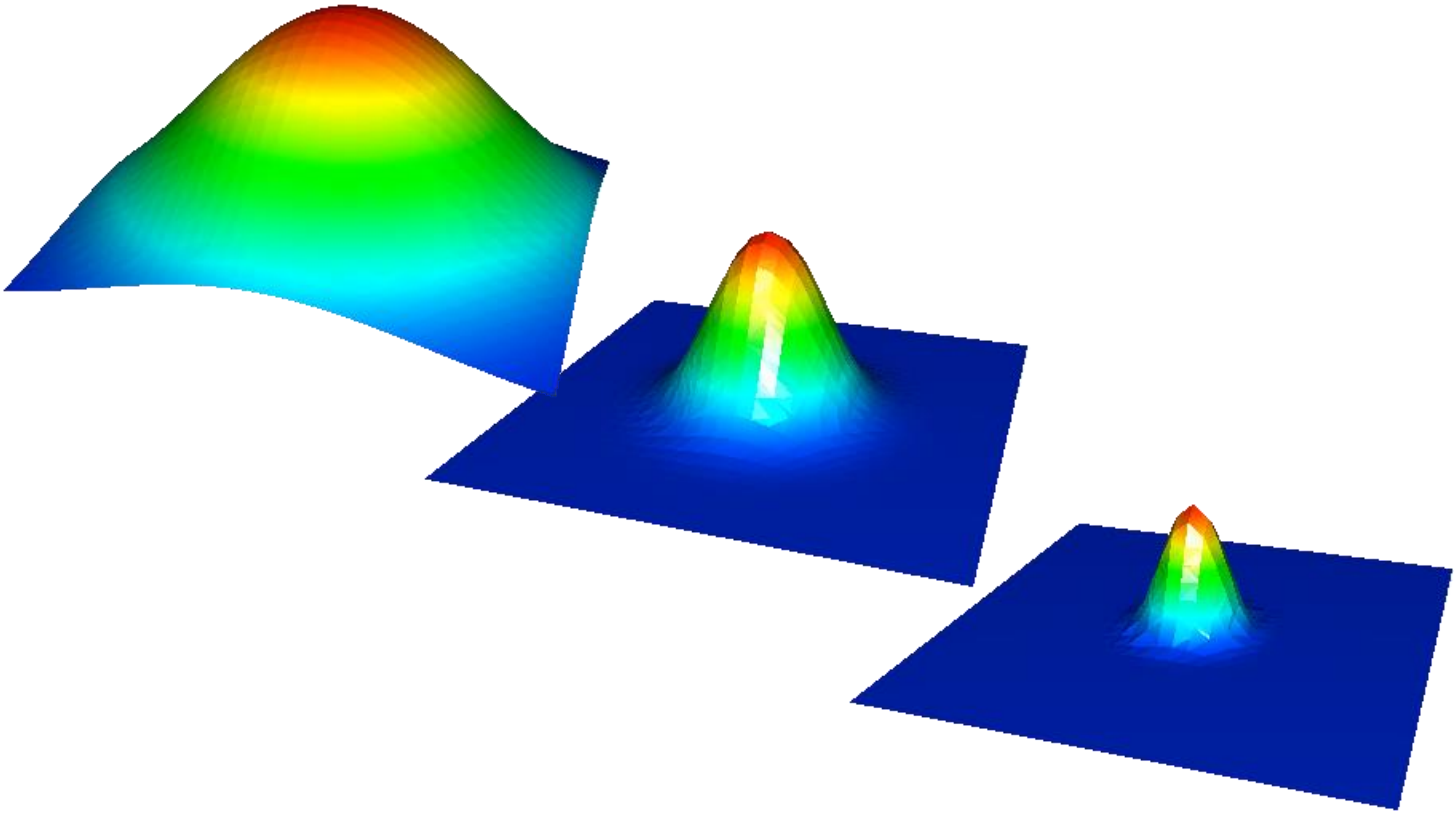




http://www.holehouse.org/mlclass/12_Support_Vector_Machines.html

# Kernels intuition




$$f_1 = similarity(x, l^{(1)}) = \begin{cases} \approx 1, \text{if } x \approx l^{(1)} \\ \approx 0, \text{if } x \not\approx l^{(1)} \end{cases}$$



# Non linear cases (kernels)

- Kernels allow us to use SVM when there is a non-linear separation between classes.
  - The kernel *defines* the inner product or a similarity in a transformed space.

$$h_\theta(x) = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$$

$$\min_\theta C\left[\sum_{i=1}^{m} y^{(i)} cost_1\left(\theta^T f^{(i)}\right) + \left(1 - y^{(i)}\right) cost_0\left(\theta^T f^{(i)}\right)\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$Predict = \begin{cases} 1 & \text{if } h_\theta(x) \geq 0 \\ 0 & \text{if } h_\theta(x) < 0 \end{cases}$$

# Multiclass SVM

- **1 versus the rest (Vapnik): train K separate SVMs**
  - **Problems:**
    - Inconsistent results (input assigned to multiple classes)
    - Imbalanced training sets.
  - **Alternatives:**
    - Define a single objective function for training all K SVMs simultaneously.
    - One-versus-one: Train K(K-1)/2 different 2-class SVM with all possible pairs of classes.
- **Single Class**
  - There are also single-class support vector machines, which solve an unsupervised learning problem related to probability density estimation.

# SVMs Pros and Cons

### SVM

**Pros:**
- Can handle large feature space
- Can handle non-linear feature interactions

**Cons:**
- Not very efficient with large number of observations
- It can be tricky to find appropriate kernel sometimes

### Decision Trees

**Pros:**
- Intuitive Decision Rules
- Can handle non-linear features
- Take into account variable interactions

**Cons:**
- Highly biased to training set [Random Forests solve this]
- No ranking score as direct result

### Logistic Regression

**Pros:**
- Convenient probability scores
- Efficient implementations available
- Wide spread industry comfort for logistic regression solutions

**Cons:**
- Doesn’t perform well when feature space is too large
- Doesn’t handle large number of categorical features/variables well
- Relies on transformations for non-linear features