Unsupervised Learning, Recommenders, Reinforcement Learning

K-means algorithm

μ_1, μ_2 $x^{(1)}, x^{(2)}, \ldots x^{(30)}$

$n = 2$ $K = 2$

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$

Repeat {

 # Assign points to cluster centroids

for i = 1 to m

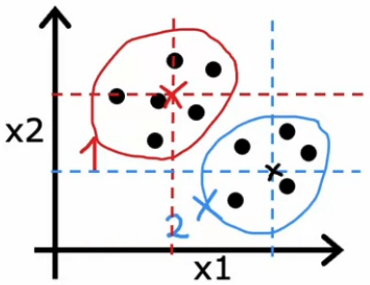
$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

 # Move cluster centroids

for k = 1 to K

μ_k := average (mean) of points assigned to cluster k

}



K-means optimization objective

$c^{(i)}$ = index of cluster $(1, 2, \ldots, K)$ to which example $x^{(i)}$ is currently assigned

μ_k = cluster centroid k

$\mu_{c^{(i)}}$ = cluster centroid of cluster to which example $x^{(i)}$ has been assigned

$x^{(10)}$ $c^{(10)}$ $\mu_c^{(10)}$

Cost function

$$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, .., \mu_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

$$\min_{\substack{c^{(1)}, \ldots, c^{(m)} \\ \mu_1, \ldots, \mu_K}} J(c^{(1)}, \ldots, c^{(m)}, \mu_1, .., \mu_K)$$

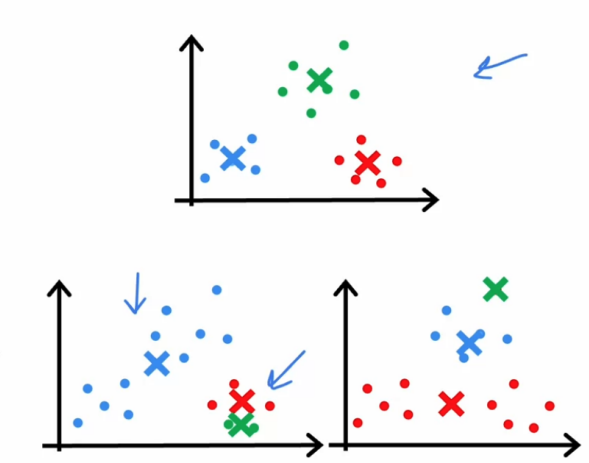
Distortion

Random initialization

Choose $K < m$

Randomly pick K training examples.

Set $\mu_1, \mu_1, \ldots, \mu_k$ equal to these K examples.



With random initialization using some of the data points themselves, depending on which points you choose, we can end up with different clusters. Local minima of the cost function may be found. We can run it with multiple random initializations and then pick the one with lowest final cost function.

Random initialization

50-1000

For $i = 1$ to 100 {

k random examples

Randomly initialize K-means.

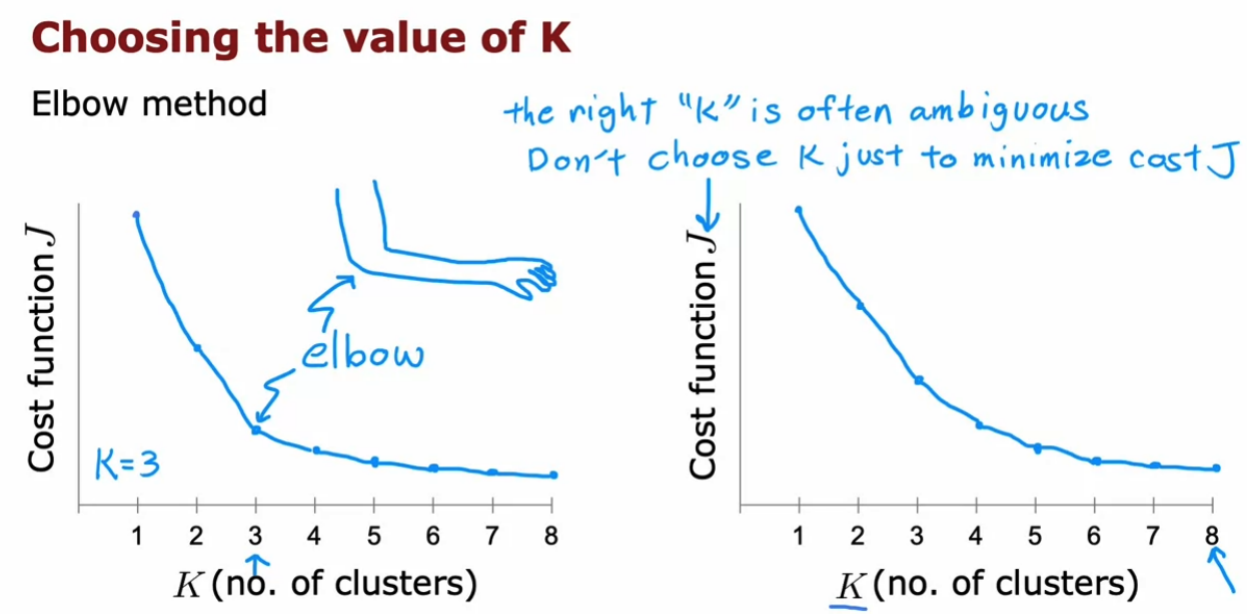
Run K-means. Get $c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_1, \ldots, \mu_k$

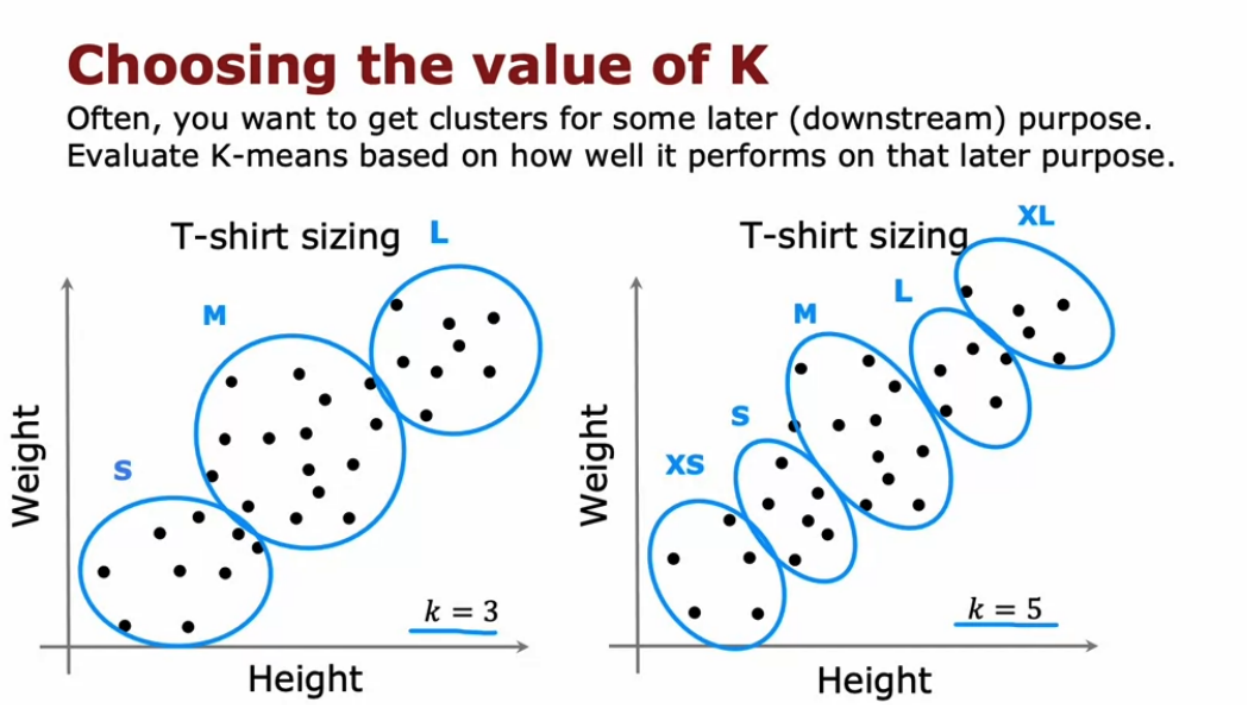
Computer cost function (distortion)

$J(c^{(1)}, \ldots, c^{(m)}, \mu_1, \mu_1, \ldots, \mu_k)$

}

Pick set of clusters that gave lowest cost J

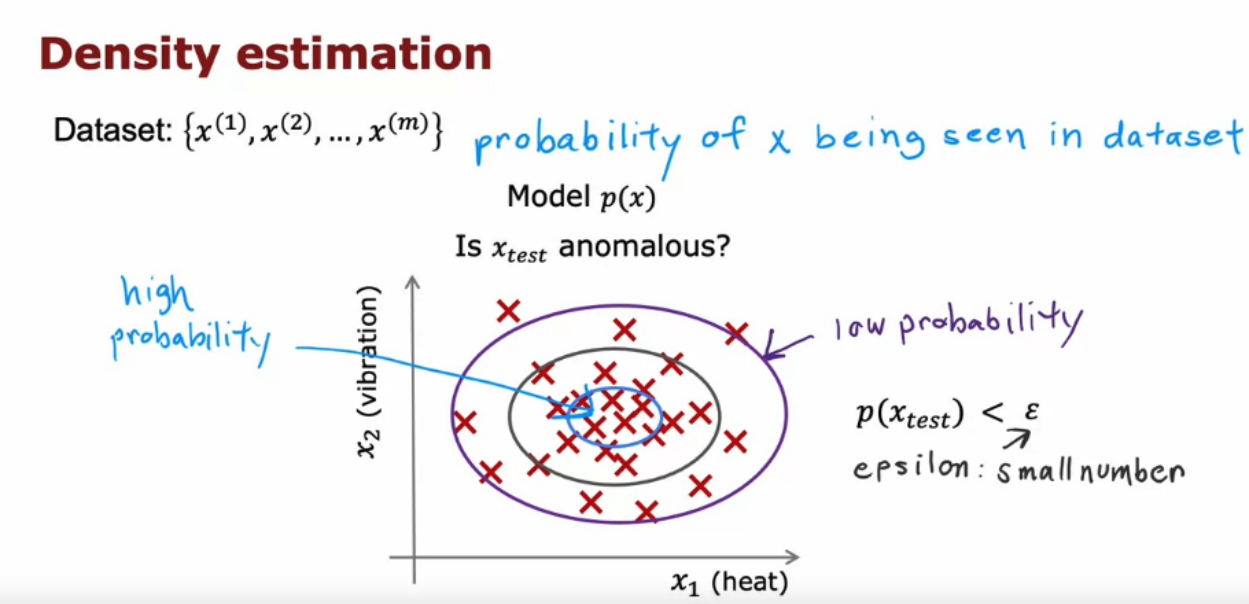


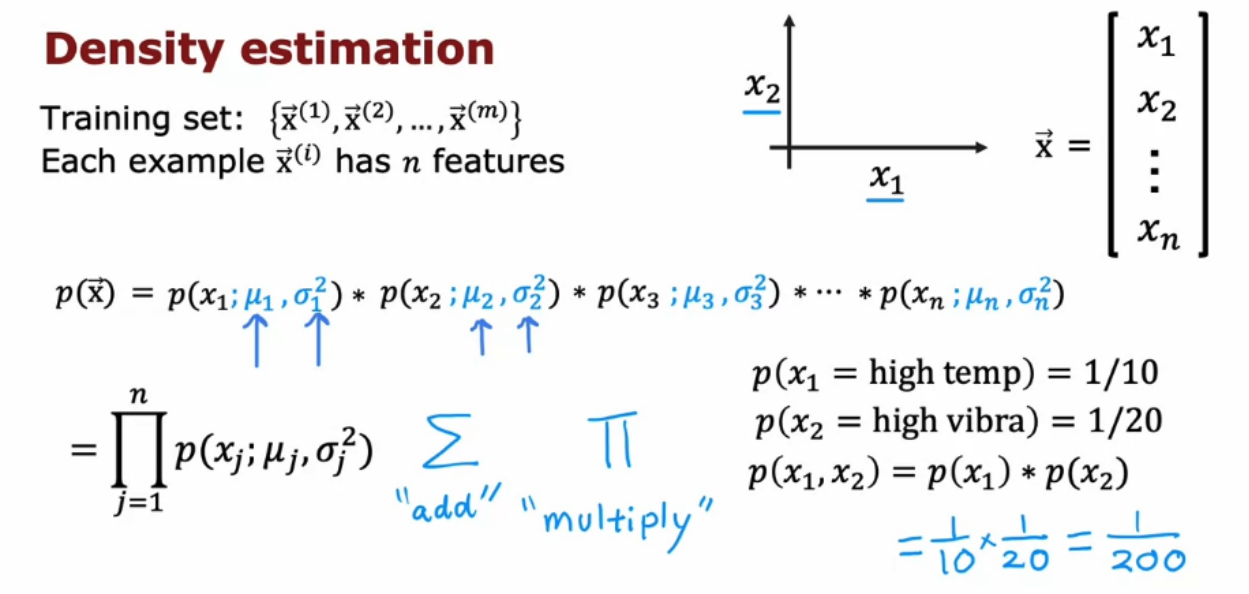


Anomaly Detection

Majority of training examples are normal. Does the test example look like them?

Uses: fraud (e.g. financial), manufacturing defects, computer monitoring in data centers





We fit a Gaussian distribution for each feature that could be indicative of anomaly. The intuition is that is any feature is too small or large, then it's likely an anomaly...

Anomaly detection algorithm

1. Choose n features x_i that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \ldots, \mu_n, \sigma_1^2, \ldots, \sigma_n^2$

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)} \qquad \sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

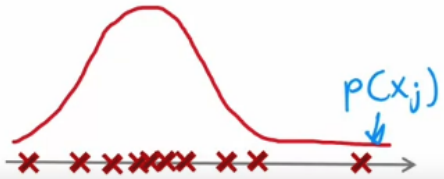
Vectorized formula

$$\vec{\mu} = \frac{1}{m}\sum_{i=1}^{m} \vec{x}^{(i)} \qquad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \ldots \\ \mu_n \end{bmatrix}$$

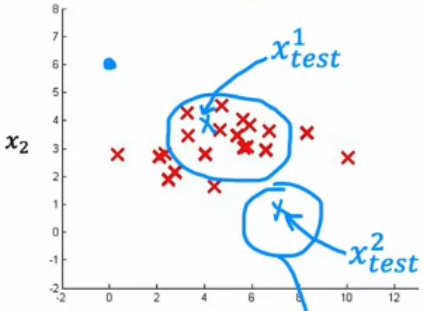
3. Given new example x, compute $p(x)$:

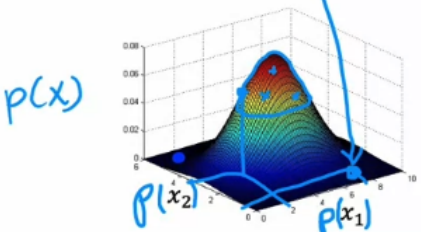
$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_j} exp(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2})$$

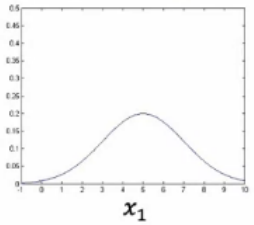
Anomaly if $p(x) < \varepsilon$

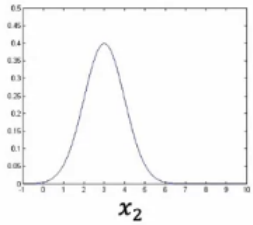


Anomaly detection example









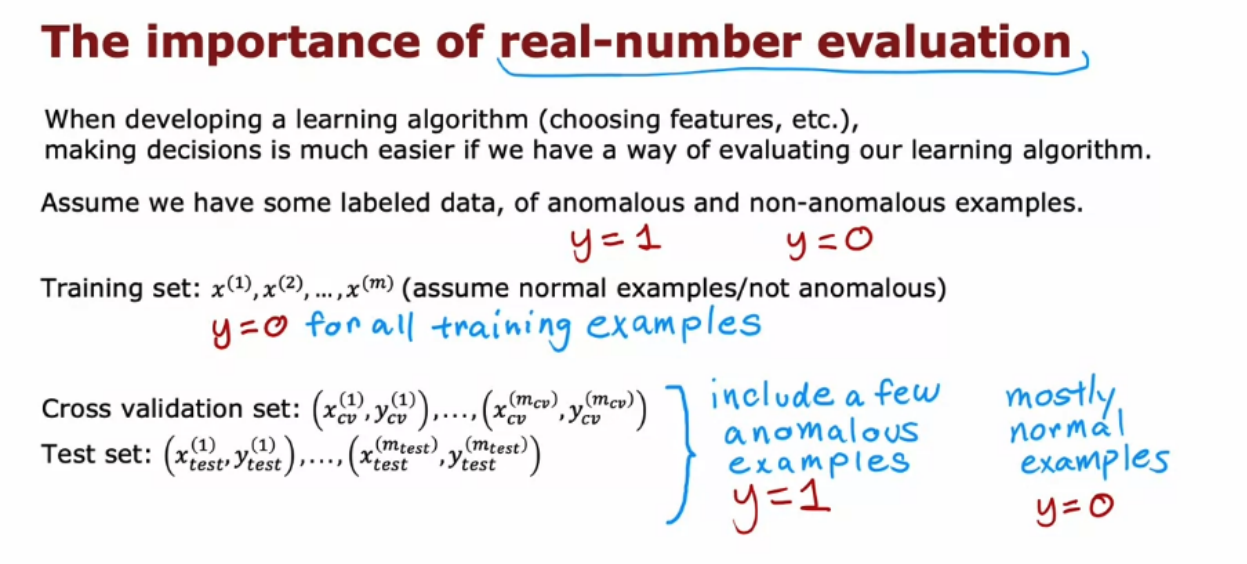
$\mu_1 = 5,\ \sigma_1 = 2$ $\qquad$ $\mu_2 = 3,\ \sigma_2 = 1$

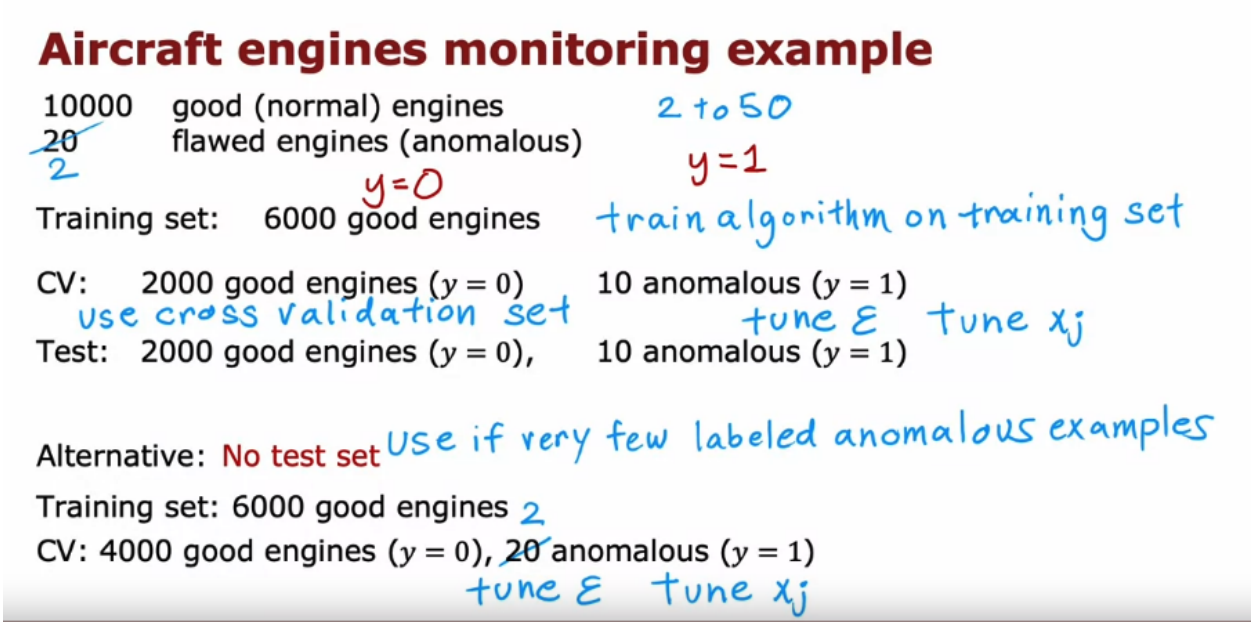
$p(x_1; \mu_1, \sigma_1^2)$ $\qquad$ $p(x_2; \mu_2, \sigma_2^2)$

$\varepsilon = 0.02$

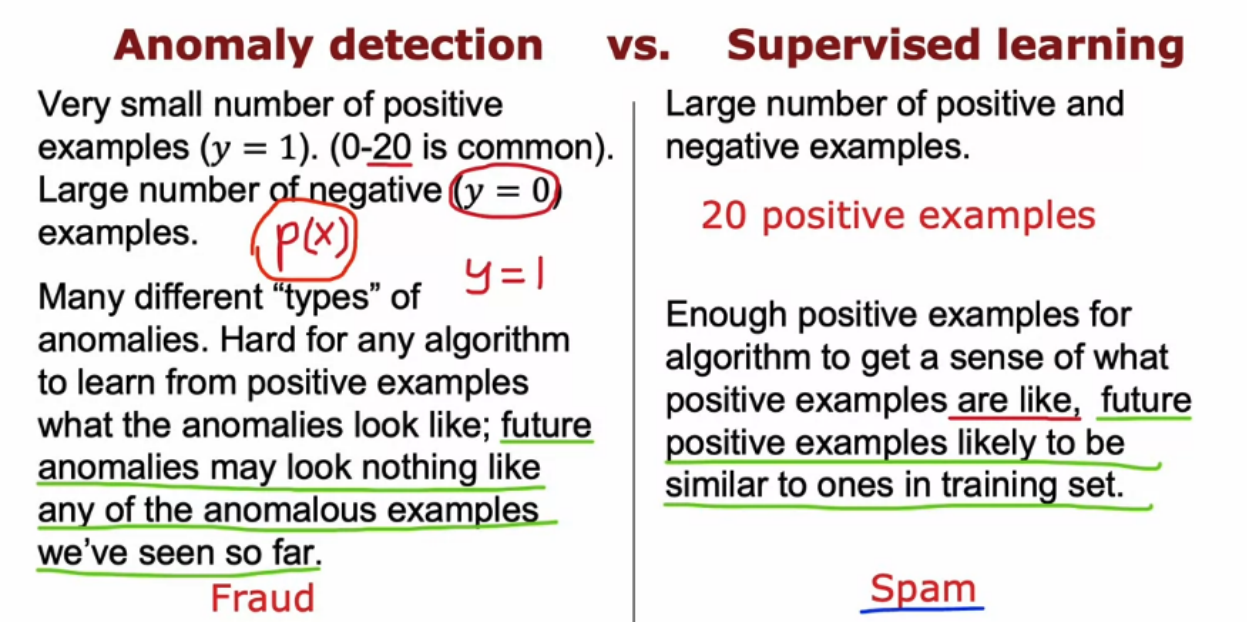
$p\left(x_{test}^{(1)}\right) = 0.0426 \longrightarrow$ "ok"

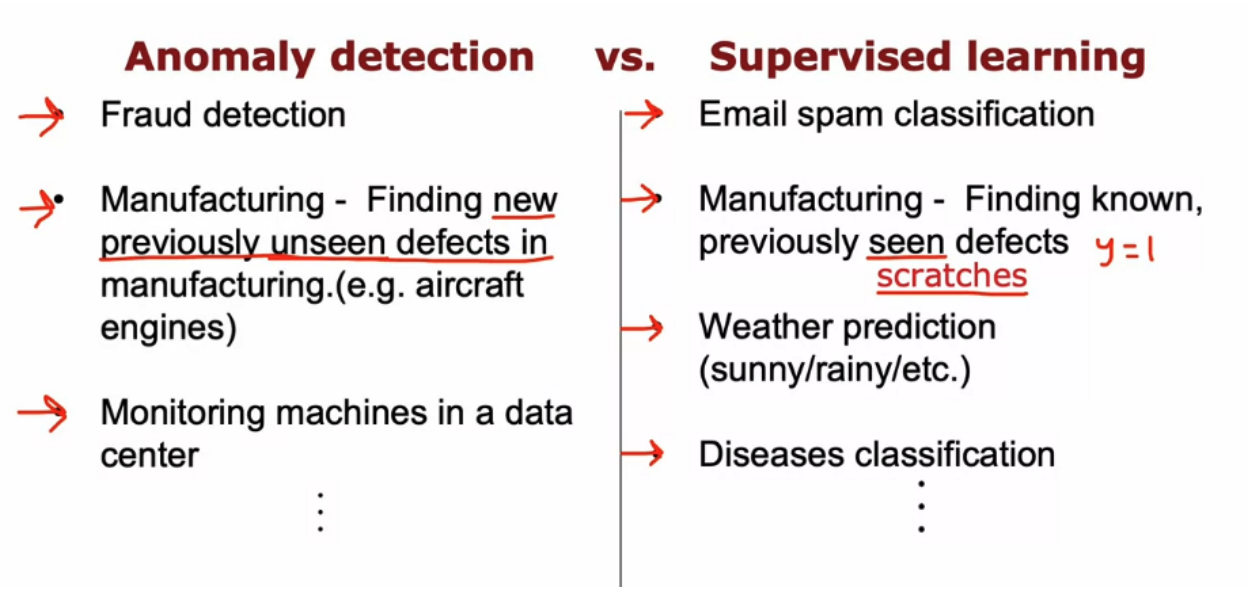
$p\left(x_{test}^{(2)}\right) = 0.0021 \longrightarrow$ anomaly





We use the cross validation to tune parameters of the model and the test set to see how well it's doing. If we have very few anomalous examples, one alternative is to only have a cross validation set, although the chances of overfitting are higher. The algorithm is still unsupervised since the training examples are unlabeled data. We use labels only on the cross validation to evaluate the model that was already fit. We can then get metrics on the cross validation and test sets. We can calculate TP, FP, FN, TN, precision, recall, f1 score, specially if the dataset is really skewed.





Carefully choosing the right features is more important for anomaly detection than supervised learning.

- Pick Gaussian features or adjust non-gaussian features to gaussian (e.g. take the log or the sqrt)
- Error analysis (which anomalies is the algorithm not detecting?) to try and identify a new feature that would help it label the anomaly correctly