#PragyaNotebook

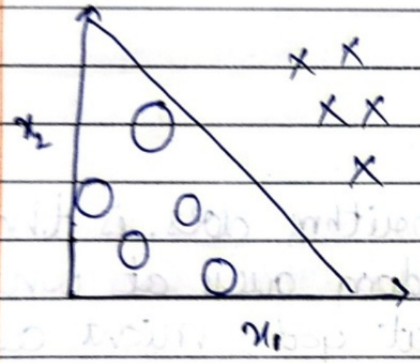
DATE
PAGE

UNSUPERVISED LEARNING, RECOMMENDERS, REINFORCEMENT LEARNING

⇒ Clustering

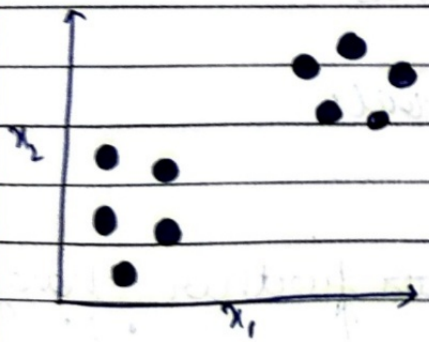
Video 1: What is clustering?

Supervised Learning



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised Learning



we train to algorithm to find something interesting about the data.

#Pragyahotebook

DATE
PAGE

Applications of clustering

- grouping news articles
- market segmentation.
- analyse DNA data
- astronomical data analysis

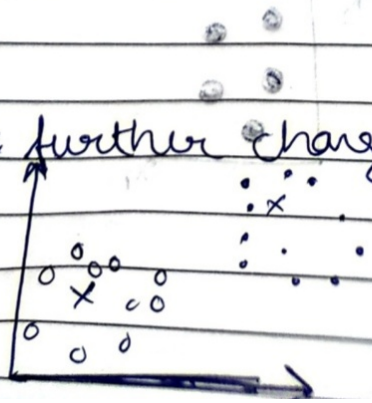
Video 2: K-means intuition

The first thing that ~~the~~ this algorithm does, is that it ~~finds the~~ will take a random guess at where the centers of the 2 clusters that you might ask it to find centre.

It will randomly pick 2 points

assign points to cluster centroids
move cluster centroids.

it will iterate this till no ~~four~~ further changes are required to be made.

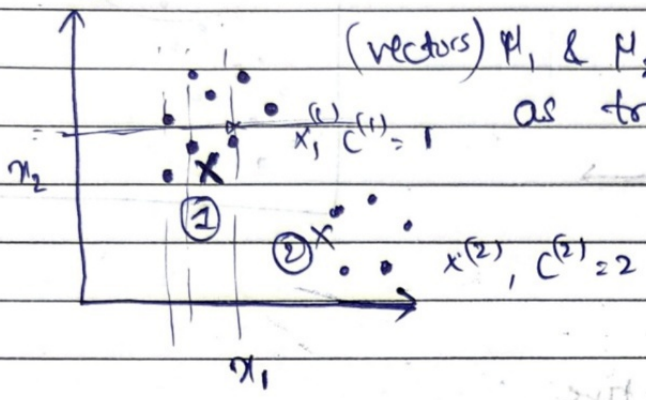


#PragyaNotebook

DATE
PAGE

Video 3: K-means algorithm

Randomly initialise K cluster centroids from $\mu_1, \mu_2, \ldots \mu_K$



Repeat {

Assign points to cluster centroids

for $i = 1$ to m

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

$\min_k \|x^{(i)} - \mu_k\|^2$

Move cluster centroids

for $k = 1$ to K

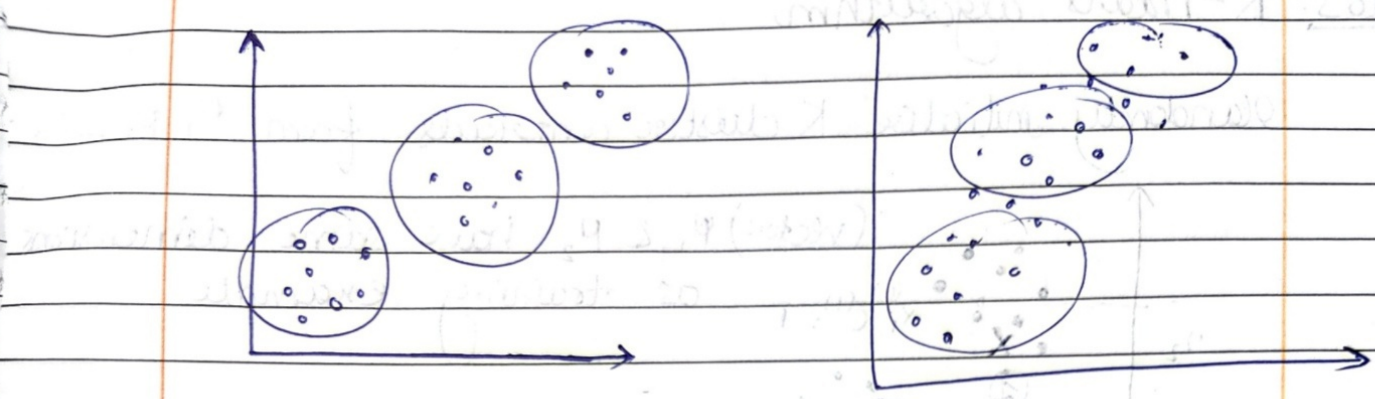
μ_k := average (mean) of points assigned to cluster k

}

* If there are 0 training examples assigned to a cluster eliminate it.

#PragyaNotebook

DATE
PAGE



Video 4: Optimisation objective

$c^{(i)}$ = index of the 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 / Distortion function

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

The K-means clustering algorithm is trying to find assignments of points of clusters centroid as well as find locations of cluster centroids that minimise the squared distance

#PragyaNotebook

DATE
PAGE

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

Repeat {

assign points to cluster centroids
for $i = 1$ to m
$c^{(i)}$: index of cluster centroid closest to $x^{(i)}$

move cluster centroids
for $K = 1$ to K
μ_K := average of points in cluster K

}

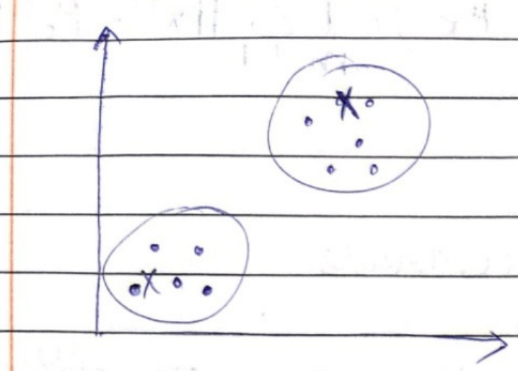
Video 5 : Initialising K-means

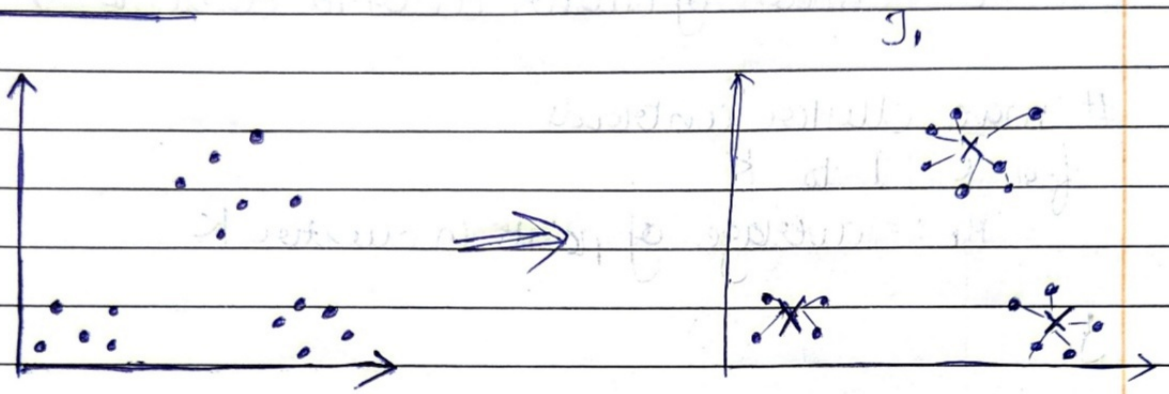
Step 0 : Randomly initialise K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$

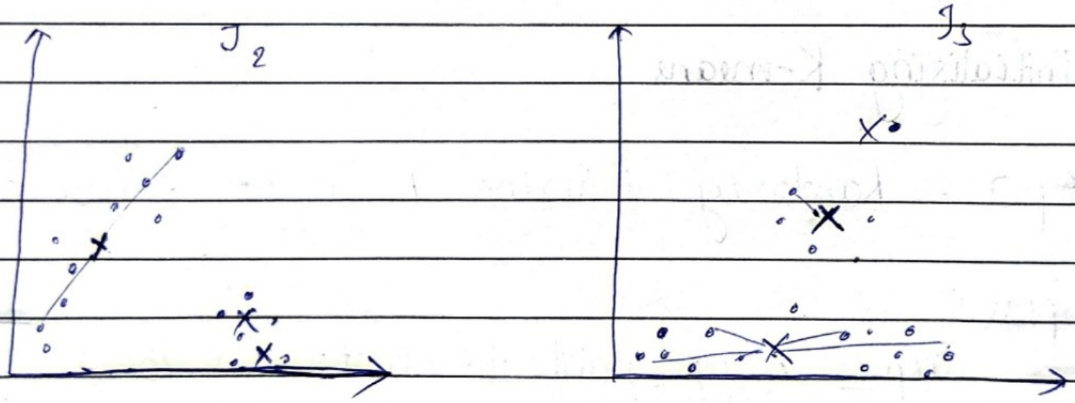
Repeat {
Step 1: Assign points to cluster centroids
Step 2: Move cluster centroids
}

Choose $K < m$

* randomly pick K training examples
* Set $\mu_1, \mu_2, \mu_3, \ldots, \mu_K$ equal to these K examples.





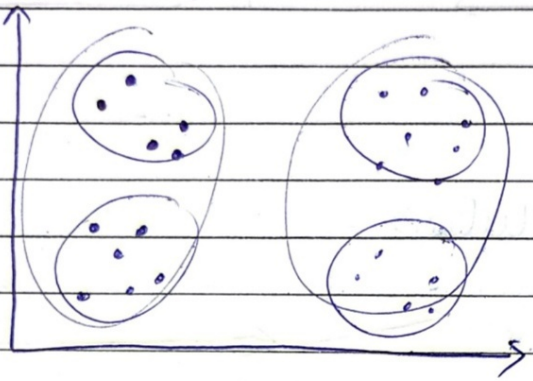


Run K means multiple times to find best local optima, using different random initialisation.

```
for i = 1 to 100{
    randomly initialise K-means
    run K-means. Get c(1), ..., c(m), μ1, μ2, ..., μK
    compute cost function (distortion)
    J(c(1), ..., c(m), μ1, μ2, ..., μK)
}
```

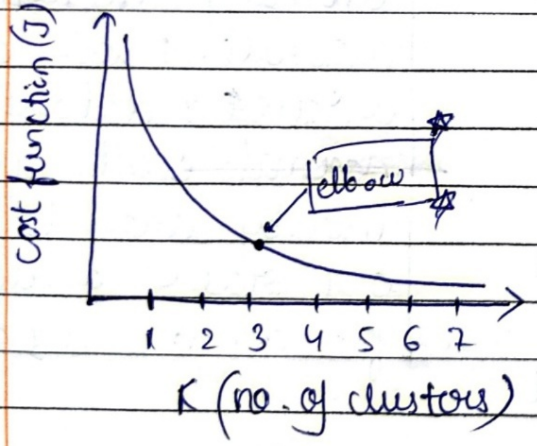
Video 6: Choosing the number of clusters

* one of the inputs is K - how many clusters you want to find.
* What is the right value of K?



* Techniques to choose clusters:

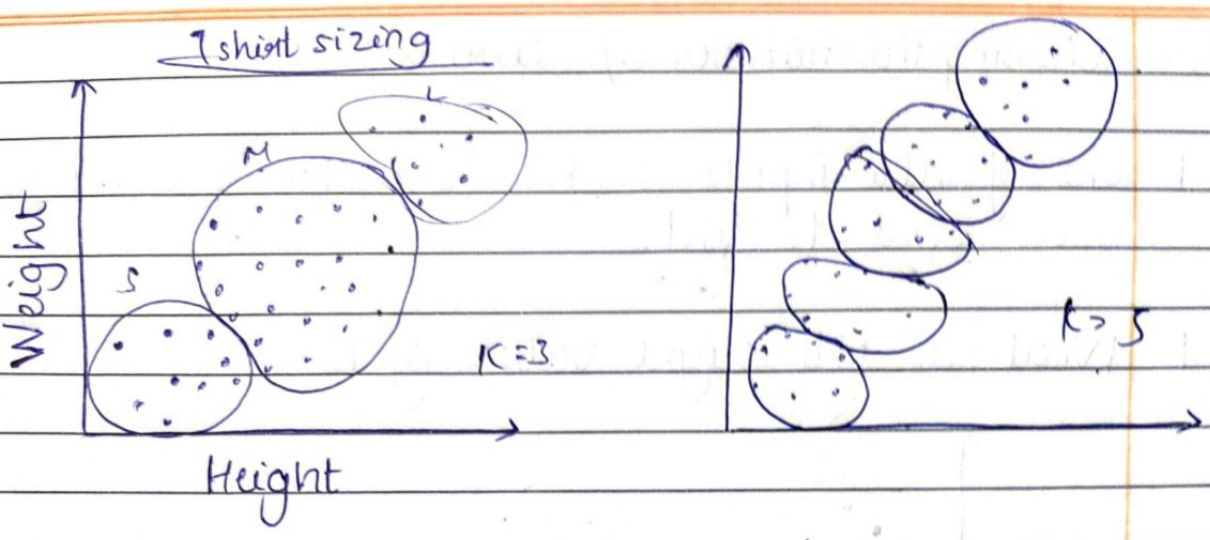
(i) elbow method (hardly used)



don't choose K just to minimise the cost function J

* Choosing the value of K

Often, you want to get clusters for some later (downstream) purpose.
Evaluate K-means based on how well it performs on that later purpose



Run both
look at both solutions

⇒ Anomaly Detection

Video 1: Finding unusual events

Aircraft engine features:

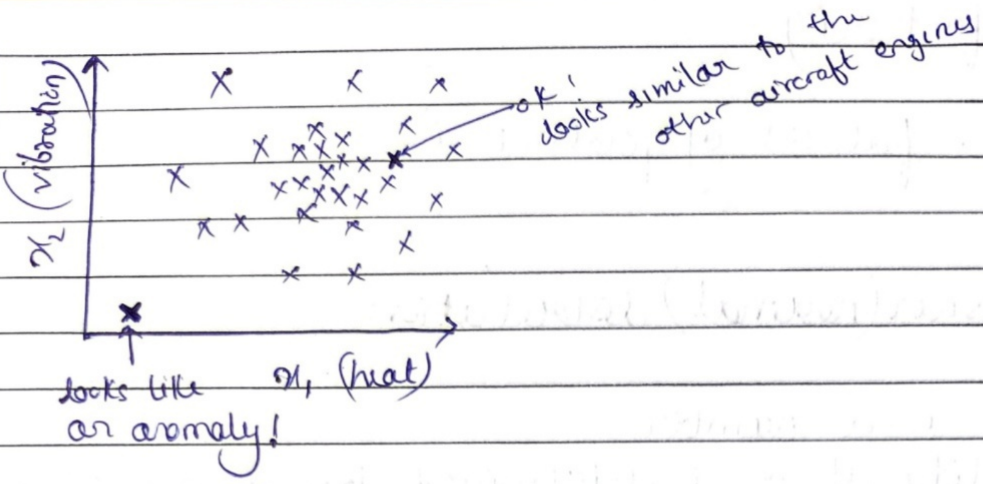
x_1 = heat generated
x_2 = vibration intensity

Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
($x^{(2)}$: no. of manufactured engines)

most of them are normal

New engine: x_{test} ← does this look similar to the ones manufactured before? any features raising suspicion.

anomaly detection algorithms look at an unlabeled dataset of normal events and thereby learn to detect or raise a red flag for an unusual event.



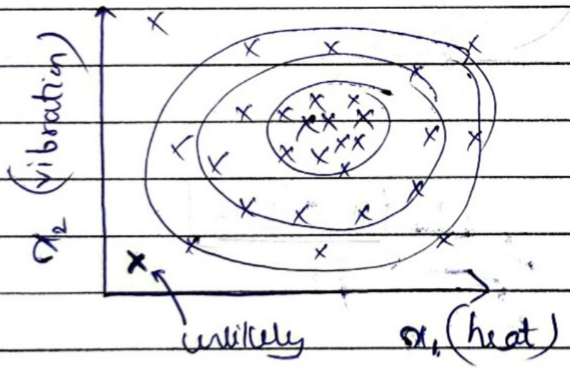
Technique - density estimation

Dataset : $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

probability of x being seen in dataset

Model $p(x)$

Is x_{test} anomalous



anomaly if:

$p(x_{test}) < \varepsilon$

epsilon (threshold)

epsilon: small no.

Applications:

(i) fraud detection

- $x^{(i)}$ = features of user i's activities
- model $p(x)$ from data
- identify unusual users by checking which have $p(x) < \varepsilon$

#PragyaNotebook

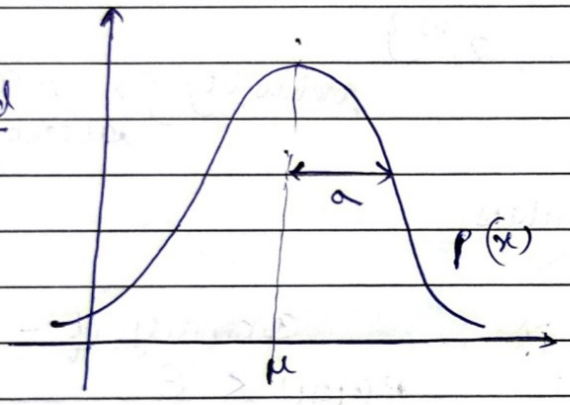
DATE
PAGE

(iii) Manufacturing

$x^{(i)}$ = features of product i

Video 2: Gaussian (normal) distribution

Say x is a number
Probability of x is determined by a Gaussian with mean μ and variance σ^2



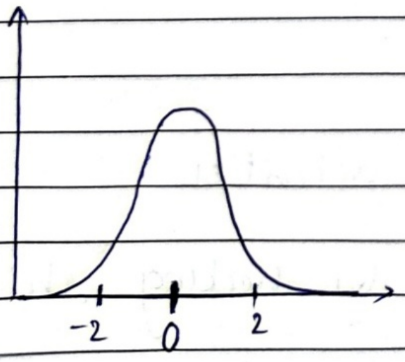
bell shaped curve

σ - standard duration
σ^2 - variance

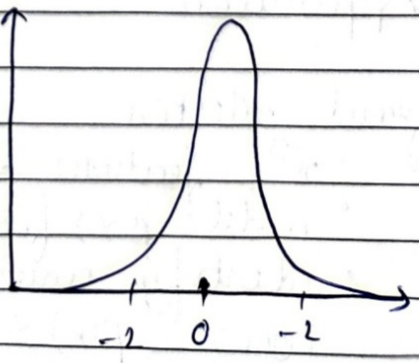
$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$\pi = 3.14$

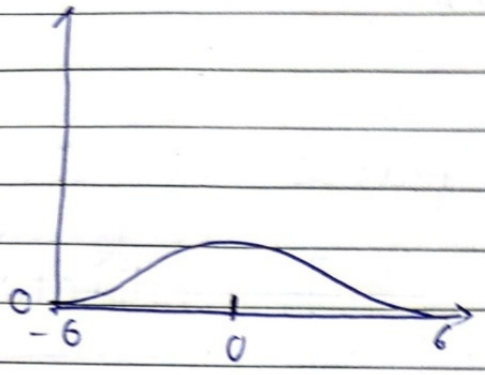
$\mu = 0 \quad \sigma = 1$



$\mu = 0 \quad \sigma = 0.5$

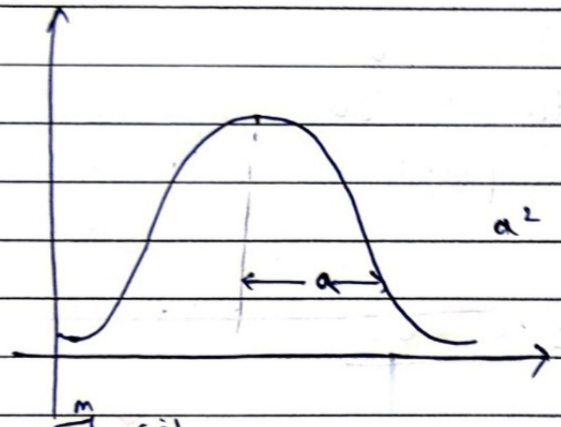


$\mu = 0$ $\sigma = 2$



Parameter Estimation

Dataset = $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$



$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} \left(x^{(i)} - \mu\right)^2$$

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

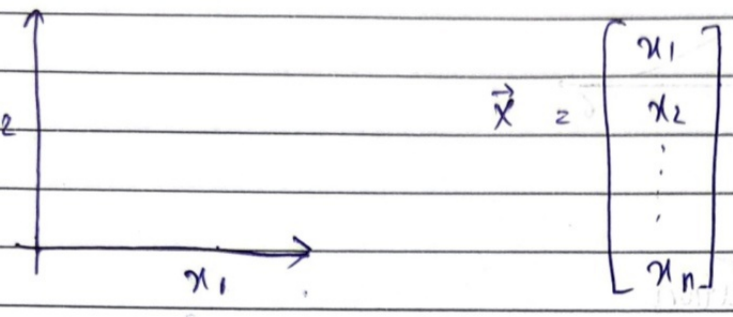
maximum likelihood for μ, σ

$\frac{1}{m-1}$

Video 3: Anomaly Detection Algorithm

Training set: $\{\vec{x}^{(1)}, \vec{x}^{(2)}, \ldots, \vec{x}^{(m)}\}$
Each example $\vec{x}^{(i)}$ has n features

density estimation
we will build a model for estimation of $p(x)$



$$p(\vec{x}) = p(x_1; \mu_1, a_1^2) * p(x_2, \mu_2, a_2^2) * p(x_3, \mu_3, a_3^2) * \ldots * p(x_n, \mu_n, a_n^2)$$

$p(x_1 = \text{high temp}) = 1/10$
$p(x_2 = \text{high vibration}) = 1/20$
$p(x_1, x_2) = p(x_1) * p(x_2)$
$= \frac{1}{200}$
$= 0.005$

$$= \prod_{j=1}^{n} p(x_j, \mu_j, a_j^2)$$

Anomaly Distribution Algorithm

1. Choose n features x_i that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \ldots, \mu_n, a_1^2, \ldots a_n^2$

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

#PragyaNotebook

DATE
PAGE

3. Give new example x, compute $p(x)$

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

anomaly if $p(x) < \varepsilon$

Video 4: Developing & Evaluating Anomaly Detection

⟹ The importance of real number evaluation

* It is a method to evaluate algorithm (choose features et
A number that tells you whether algo got better or worse.

* Assume we have some labeled data of anomalous ($y=1$) and non-anomalous ($y=0$) examples.

* Training set: $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ ($y=0$) (assume normal examples / not anomalous)

* To evaluate we need a small no. of anomalous examples

include few anomalies {
Cross validation set: $(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})})$
Test set: $(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m_{test})}, y_{test}^{(m_{test})})$

#PragyaNotebook

DATE
PAGE

Aircraft engines monitoring example

10,000 → good engines
20 → flawed engines

Training set → 6000 (good engines)
CV → 2000 (good engines) ($y=0$) } tune ε
10 anomalous $y=1$
Test → 2000 good engines ($y=0$)
10 anomalous ($y=1$)

Alternative → no test set

Video 5: Anomaly Detection v/s supervised learning

Anomaly Detection	Supervised Learning
Very small number of positive examples ($y=1$) Large number of negative examples ($y=0$)	Large no. of positive & negative examples
Many different types of anomalies	Enough positive examples to get a sense of what ~~future~~ the examples are like.
fraud detection	email spam classification
finding new previously unseen defects in manufacturing	finding known defects.

#PragyaNotebook

DATE
PAGE

Video 6: Choosing what features to use:

Non Gaussian features ⇒ Gaussian features

$x_1 \leftarrow \log(x_1)$
$x_2 \leftarrow \log(x_2 + 1)$
$x_3 \leftarrow x_3^{1/2}$
$x_4 \leftarrow x_4^{1/3}$

Error analysis for ~~analysis~~ anomaly detection

Want $p(x) \geq \varepsilon$ large for normal examples x
$p(x) < \varepsilon$ small for anomalous examples x

Most common problem:
$p(x)$ is comparable for normal & anomalous examples
$p(x)$ is large for both