WEEK #9
Anomaly Detection.

(1.) DENSITY ESTIMATION.

1.1. Problem Motivatn:

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

Is x_{test} anomalous?

$\boxed{\text{Model } P(x) \text{ from data}}$

$p(x_{test}) < \varepsilon \rightarrow$ flag anomaly.

$p(x_{test}) \geq \varepsilon \rightarrow$ OK

1.2. Gaussian Distributn:

Gaussian (Normal) Distributn

$x \sim \mathcal{N}(\mu, \sigma^2)$

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

1.3. Algorithm:

Density Estimatn:

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

1. Choose features x_i that you think might be indicative of anomalous examples.
2. Fit paramtrs $\mu_1, \mu_2, \dots, \mu_n, \sigma_1^2, \dots \sigma_n^2$

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

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

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

Anomaly if $p(x) < \varepsilon$

(2.) BUILDING AN ANOMALY DETECTN SYSTEM.

2.1. Developing and Evaluatng an Anomaly Detectn Stm:

1000 good (normal) engines
20 flawed engines (anomalous)

Training set: 6000 good engines
CV: 2000 good engines $(y=0)$, 10 anomalous $(y=1)$
Test: 2000 " " $(y=0)$, 10 anomalous $(y=1)$

Fit model $p(x)$ on training set
On a CV, Test set x, predict

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \text{ (anomaly)} \\ 0 & \text{if } p(x) > \varepsilon \text{ (normal)} \end{cases}$$

Possible evaluatn metrics:
- Precision / Recall
- F_1-Score

Use CV set to choose parameter ε

2.2. Anomaly Detectn vs. Supervised Learning:

Anomaly detectn	vs.	Supervised Learning:
very small # of pos've examples ($y=1$) large # of neg've examples ($y=0$)		Large # of pos.ve & neg've examples.
ex/ Fraud detectn Manufacturing (aircraft engines) Monitoring machines in a data center		ex/ Email spam classificatn. Weather predictn

2.3. Choosing what features to Use:

Non-gaussian features:

$$\left.\begin{aligned} x_1 &\leftarrow \log(x_1) \\ x_2 &\leftarrow \log(x_2+c) \\ x_3 &\leftarrow \sqrt{x_3} \\ x_4 &\leftarrow x_4^{1/3} \end{aligned}\right\} \text{Gaussian features.}$$

Error Analysis for anomaly detectn:

Want $p(x)\uparrow$ for normal

$p(x)\downarrow$ for anomalous

Most common problem: $p(x)$ is comparable (both large/small) for normal & anomalous exs.

* Choose feature that might take on unusually large or small values in the event of an anomaly.

x_1, x_2, x_3, x_4 exist, create $x_5 = \frac{\text{CPU load}}{\text{network traffic}}$, $x_6 = \frac{x_3^2}{x_4}$

3. MULTIVARIATE GAUSSIAN DISTRIBUTION.

3.1. Multivariate Gaussian Distributn:

In some cases, actual anomalous examples can be seen as normal.

→ Use modified anomaly detectn algorithm: Multivariate Gaussian distributn.

Don't model $p(x_1), p(x_2), \ldots$ etc separately! Model $p(x)$ all in one go.

Paramtrs $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n\times n}$ (covariance mtx).

3.2. Anomaly Detectn using Multivariate Gaus. Distrn:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^T$$

ORIGINAL MODEL:	MULTIVARIATE GAUSSIAN:
$p(x) = p(x_1;\mu_1,\sigma_1^2)\times\ldots p(x_n;\mu_n,\sigma_n^2)$ ⟷	$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}\lvert\Sigma\rvert^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$
* manually create features to capture anomalies where x_1, x_2 take unusual combinatns of values. * cheaper * OK for small m values.	* Automatically captures correlatns btw features. * more expensive * Must have $m>n$.