Machine Learning

Amin Golzari Oskouei

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

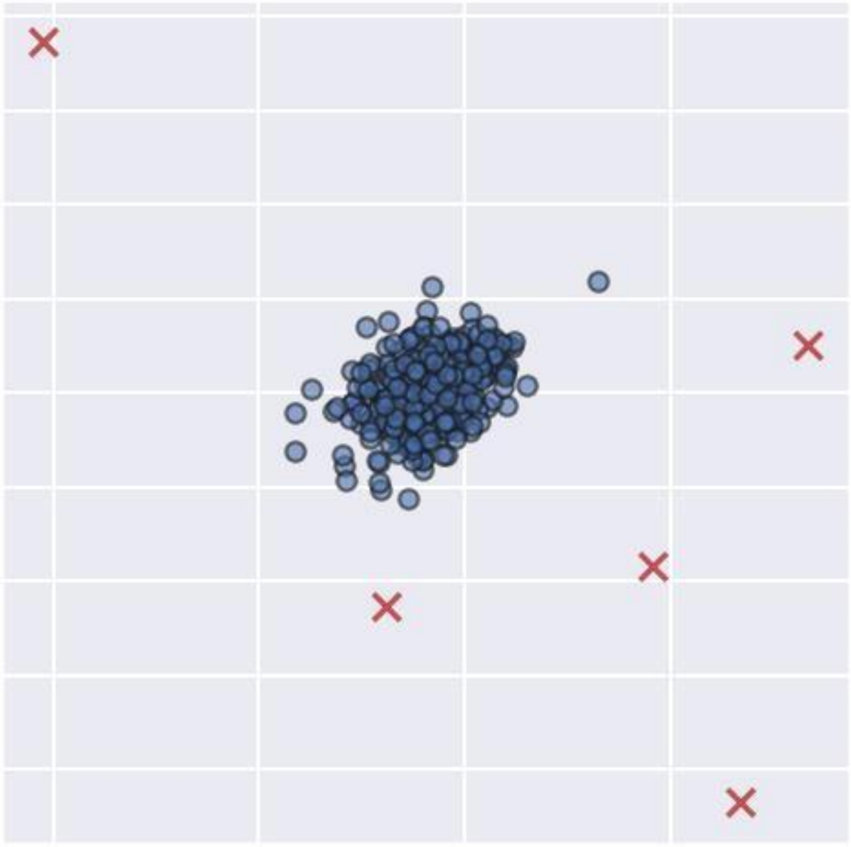
https://github.com/Amin-Golzari-Oskouei

Azarbaijan Shahid Madani University
2023

Anomaly Detection

Anomaly Detection [Outlier Data Detection]

- Anomaly Detection. Identifying observations that differ greatly from most observations.



- Scam Detection.
  - Detection of highly improbable transactions by the credit cardholder.
- Network Security.
  - Detecting activities with very low probability by legal user is done.

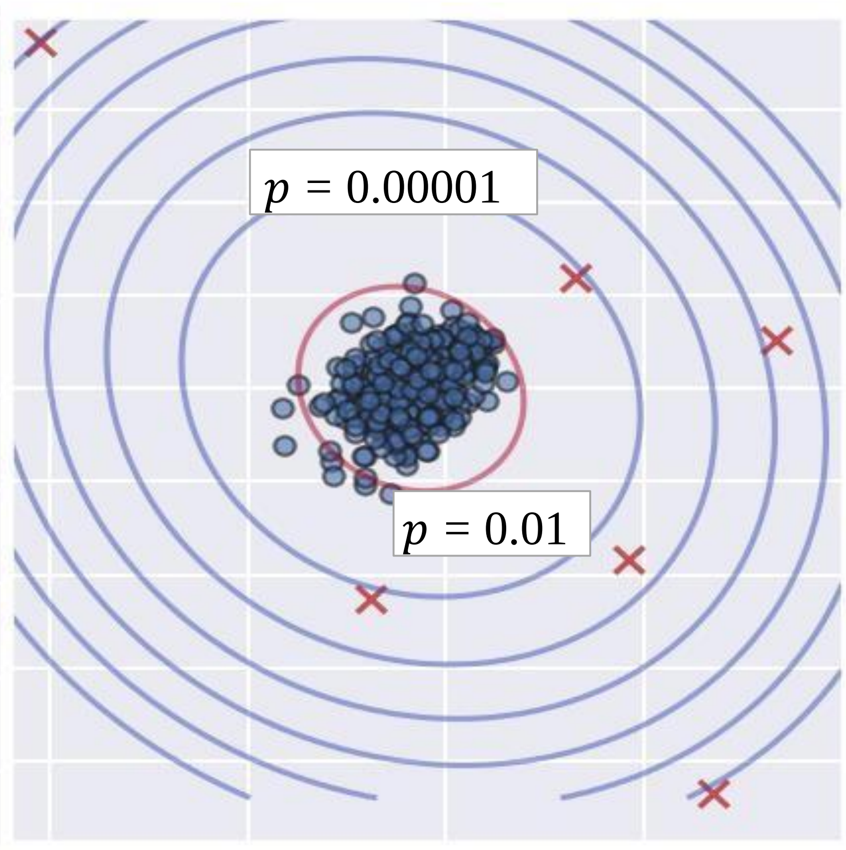
Anomaly Detection [Outlier Data Detection]

- Anomaly Detection. Identifying observations that differ greatly from most observations.

- A probabilistic approach to anomaly detection.
  - Creating a probabilistic model from the data

    [Expressing the probability of seeing any possible event]

  - Specify observations that are very unlikely to occur.

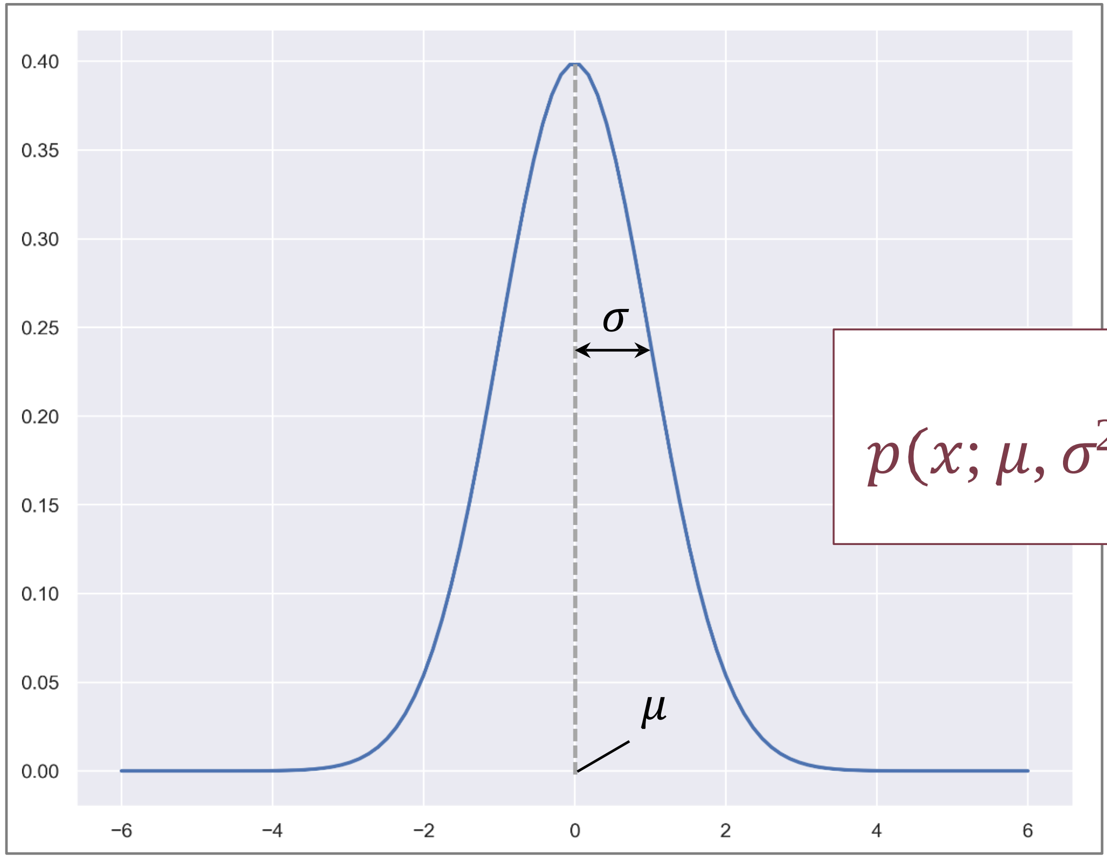


$$p(x) < \epsilon$$

Gaussian Distribution(Normal)

Gaussian Distribution

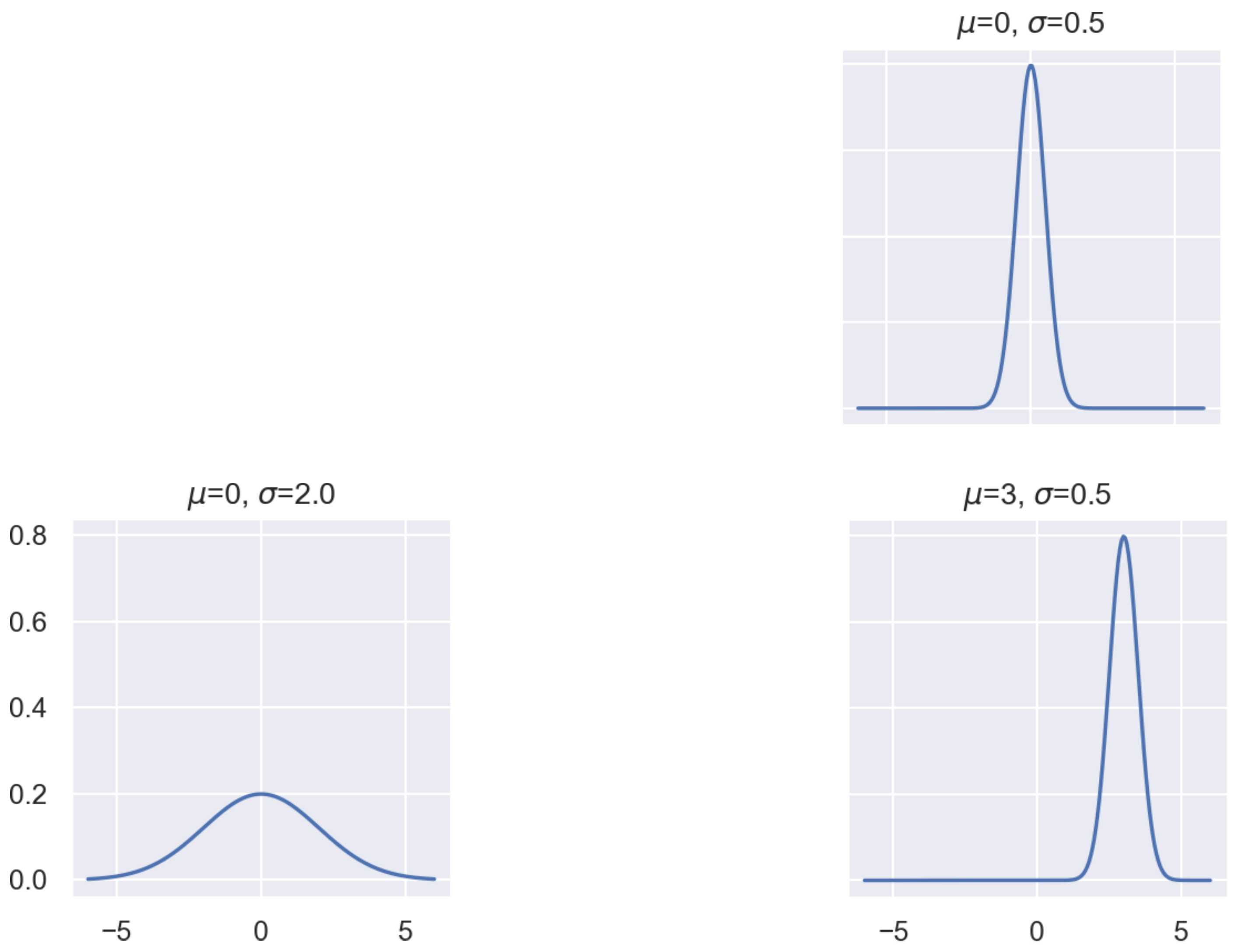
- **Gaussian Distribution.** Suppose x has Gaussian Distribution with average of μ and variance is σ^2.



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

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

Univariate Gaussian Distribution

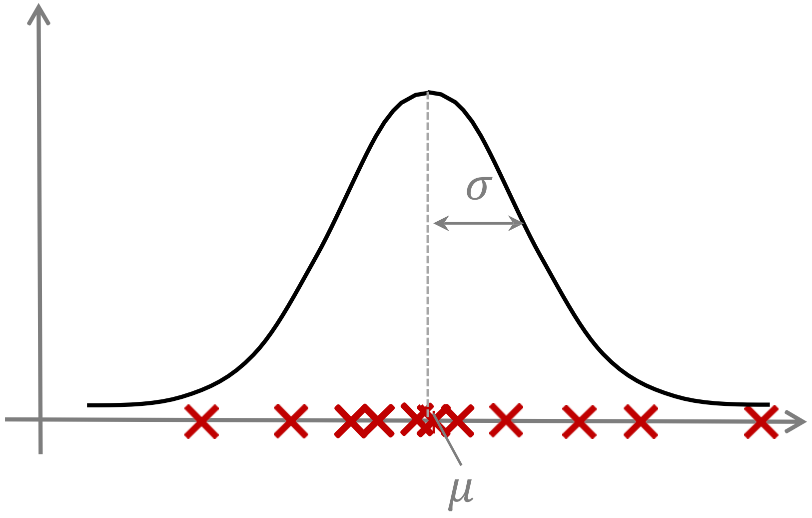


Parameter Estimation

- Data collection.

$$\{x^{(1)}, x^{(2)}, x^{(3)}, \cdots, x^{(m)}\}$$

- Purpose. Estimation of values μ and σ



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

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

Anomaly Detection Algorithm

Estimation of Distribution

- Training set. $\{x^{(1)}, x^{(2)}, x^{(3)}, \cdots, x^{(m)}\}, \quad x^{(i)} \in \mathbb{R}^n$
- Assumptions. $x_j \sim N(\mu_j, \sigma_j^2)$
  - Features follow a normal distribution.
  - There is no correlation between features.[Diagonal covariance matrix]

$$p(\boldsymbol{x}) = p(x_1; \mu_1, \sigma_1^2)\, p(x_2; \mu_2, \sigma_2^2)\, p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)$$

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

Anomaly Detection Algorithm

- Determining features that can be useful in anomaly detection.

- Estimation of parameters (for n ≥ j ≥ 1)

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

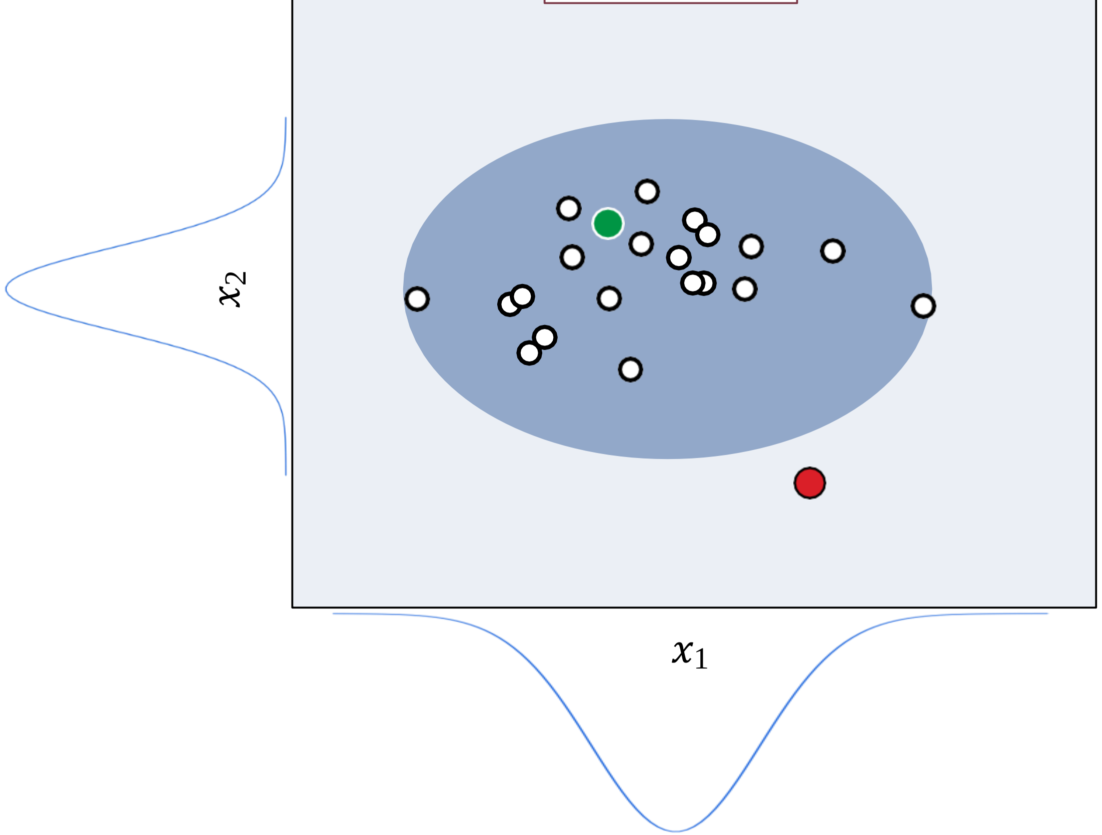
- Calculation p(x) for the new data of x

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

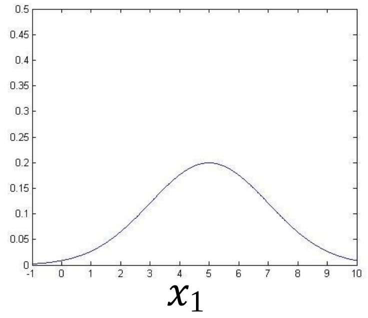
- Printing the <<yes>> output if $p(x) < \epsilon$

Example

$\varepsilon = 0.02$

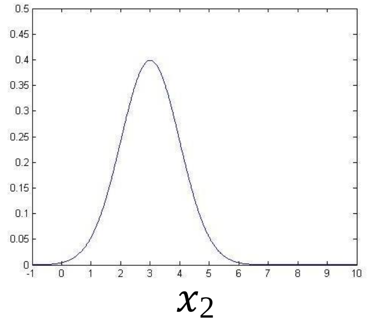


$\mu_1 = 5, \sigma_1 = 2$



$p(x_1; \mu_1, \sigma_1^2)$

$\mu_2 = 3, \sigma_2 = 1$



$p(x_2; \mu_2, \sigma_2^2)$

$p(x_{test}^{(1)}) = 0.0426$

$p(x_{test}^{(2)}) = 0.0021$

Development and Measurement of Anomaly Detection Systems

Numerical Evaluation

- Importance.

  - During the process of developing learning systems, if we have a method to evaluation the system, then Many decisions (such as feature selection, etc.) will become much simpler.

- Suppose we have some labeled data that (y = 0) is its normality and its abnormality is (y = 1).

- Training collection $\{x^{(1)}, x^{(2)}, x^{(3)} \cdots, x^{(m)}\}$

- Validation set. $\left\{\left(x_{cv}^{(1)}, y_{cv}^{(1)}\right), \left(x_{cv}^{(2)}, y_{cv}^{(2)}\right), \cdots, \left(x_{cv}^{(m_{cv})}, y_{cv}^{(m_{cv})}\right)\right\}$

- Test set. $\left\{\left(x_{test}^{(1)}, y_{test}^{(1)}\right), \left(x_{test}^{(2)}, y_{test}^{(2)}\right), \cdots, \left(x_{test}^{(m_{test})}, y_{test}^{(m_{test})}\right)\right\}$

Example

- Data collection. Engine performance information
  - 10000 Unbroken engine
  - 20 broken engine

- Data assortment.
  - Training set. 6000 unbroken engine[Single Category Assortment]
  - Validation set. 2000 unbroken engine and 10 broken engine
  - Experimental set. 2000 unbroken engine and 10 broken engine

Algorithm Evaluation

- Instruction. Development of the p(x) model according to the training set
- Forecasting. For samples in the validation or training set

$$y = \begin{cases} 1, & \mathrm{p(x)} < \varepsilon \\ 0, & \mathrm{p(x)} \geq \varepsilon \end{cases}$$

- Possible evaluation factor.
  - true positive, false positive, true negative, false negative
  - Accuracy rate and reminder rate
  - F1 score

- Attention. Validation set can be used to choose a suitable value for ε.

Evaluation Factor

- **Evaluation factor.** For unbalanced data

real

predict	$y = 1$	$y = 0$
$y = 1$	TP	FP
$y = 0$	FN	TN

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2\frac{P \cdot R}{P + R}$$

Anomaly Detection or Supervised Learning?

Anomaly detection or supervised learning?

Monitored Learning

- Number of samples.
  - Large numbers of positive and negative samples
- Positive sample.
  - Number of positive examples for the algorithm to understand them, is enough.
  - New positive samples are similar to positive ones that the algorithm was previously faced, during the training process.

Anomaly Detection

- Number of samples.
  - The ratio of the number of positive to negative samples is very low
- Different “Types" of anomalies.
  - For any algorithm, learning anomalies from small numbers of positive samples is very difficult.
  - New anomalies may not be similar to anomalies that have been seen before.

Anomaly detection or monitored learning?

Monitored learning

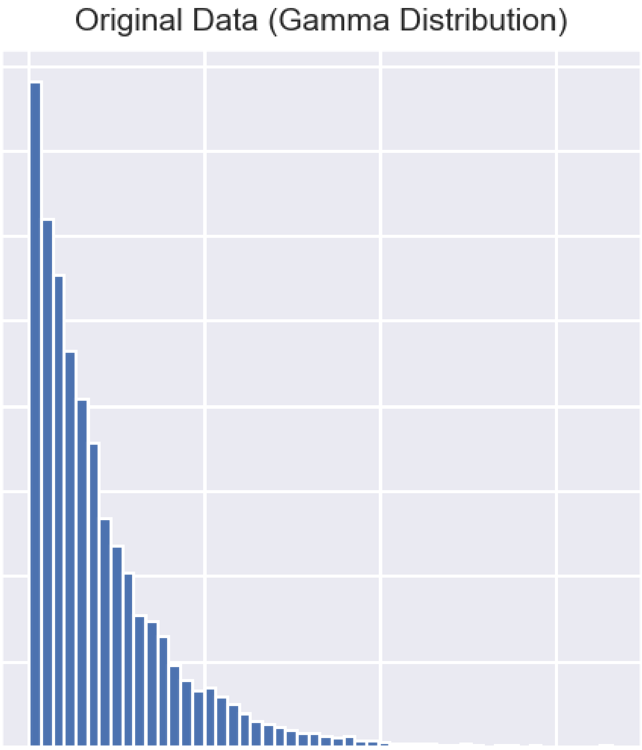
- Spam detection.
- Weather forecast.
- Diagnosis of malignant cancerous tumors.
- …

Anomaly detection

- Scam detection
- Construction and production (making airplane engines).
- Monitoring machines in data centers.
- …

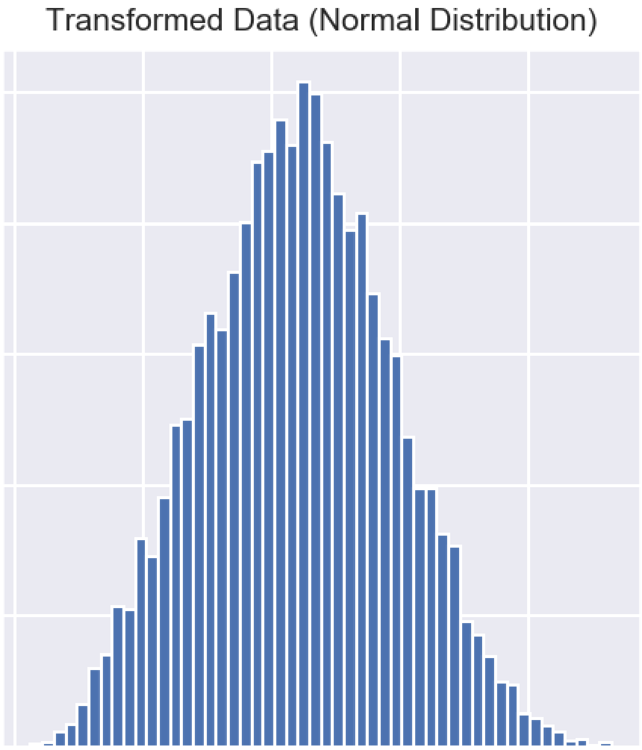
Select Features

Converting the Feature with Abnormal Distribution to the Feature with Normal Distribution




```
x = np.random.gamma(1, 2, (10000, 1))
plt.hist(x, 50)
```

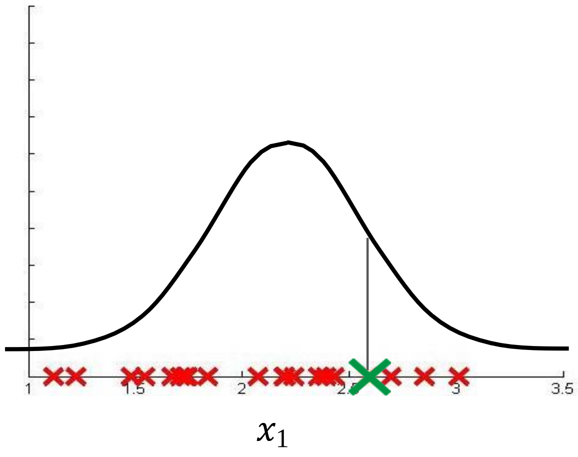
$x^{0.3}$



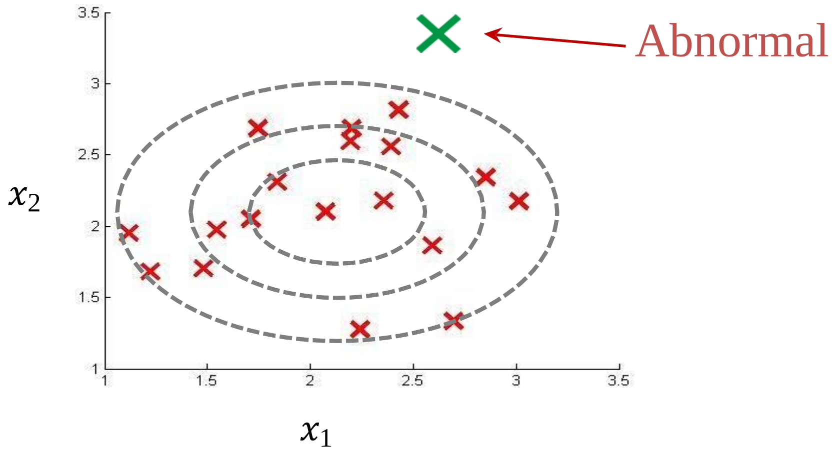

```
plt.hist(x ** 0.3, 50)
```

Error Analysis for Helping in Anomaly Detection

- Purpose. We want p(x) value:
  - Be large for normal data.
  - Be small for abnormal data.



- A common problem.
  - There is no differences between normal and abnormal for p(x).



Monitor Computers in Data Centers

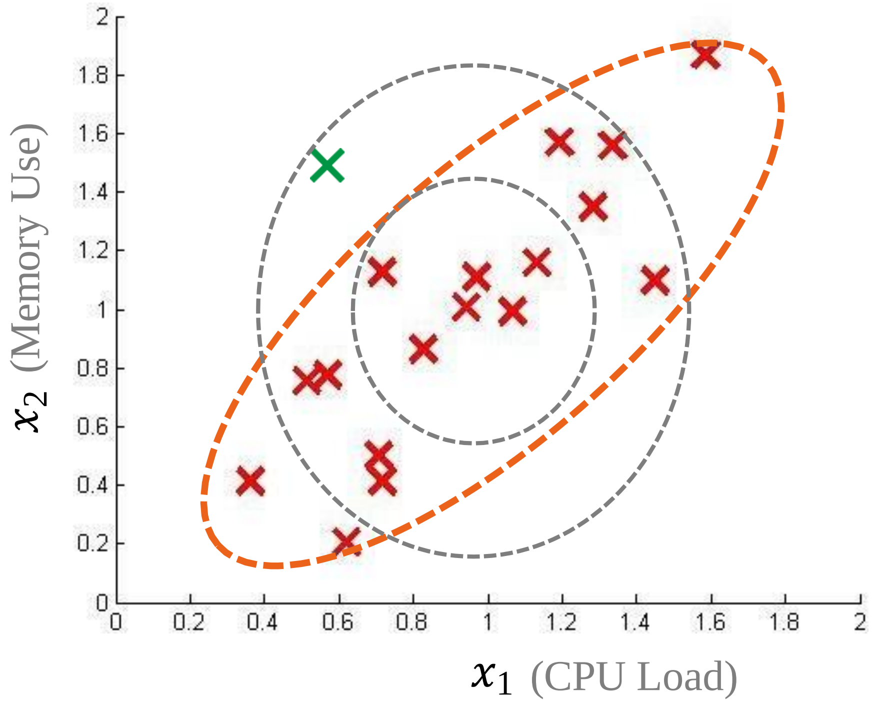
- Features selecting. Selection of features that are very small or very large if there is an anomaly.
  - Memory usage
  - Number of disk accesses per second
  - Processor load
  - Network traffic
- Add new features to detect abnormal conditions.
  - The ratio of processor load to network traffic

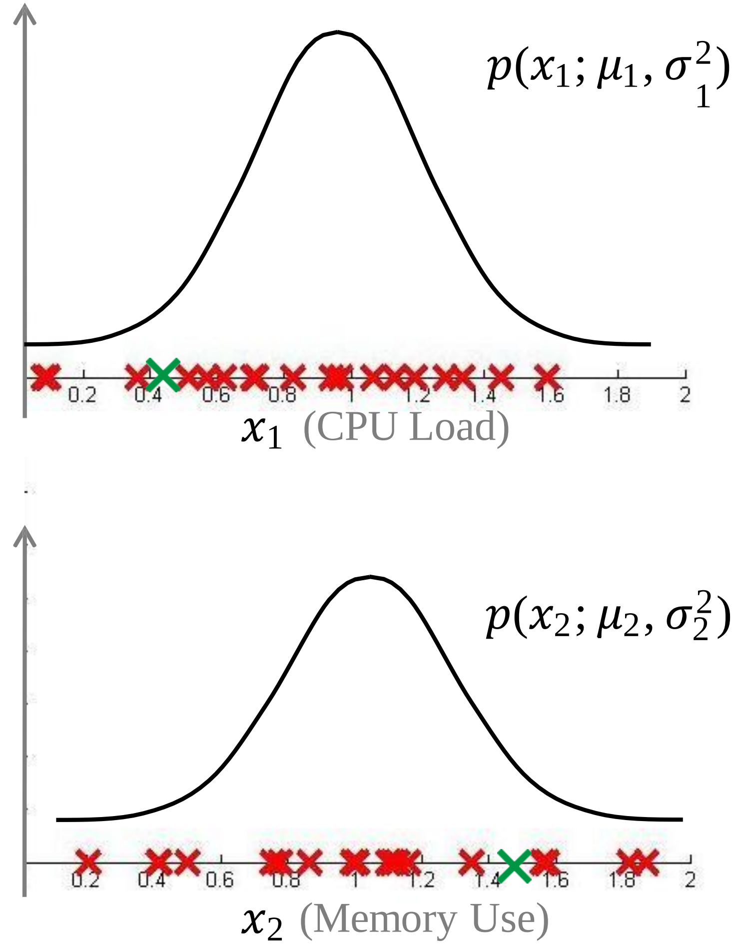
    [For example, if the processor is stuck in an infinite loop, the value of this feature will be very large.]

Multivariate Gaussian Distribution

Introductory Example

Bivariate Gaussian function





As the processor load increases, memory consumption normally increases increase.

Bivariate Gaussian function

- Bivariate Gaussian function.

$$p(\mathrm{X};\mu,\Sigma)=\frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

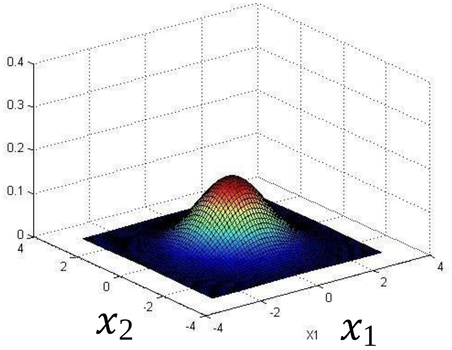
- Parameters

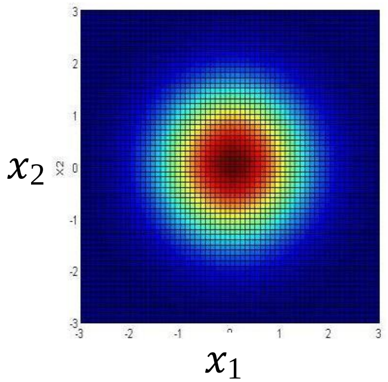
Covariance Matrix

$\mu \in \mathbb{R}^n$ $\Sigma \in \mathbb{R}^{n\times n}$

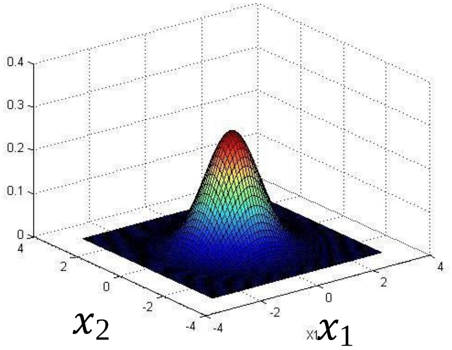
Diagonal Covariance Matrix, the Variance of Features is Equal

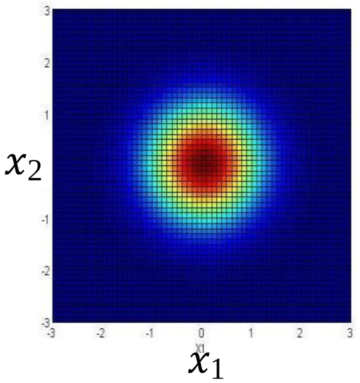
$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$



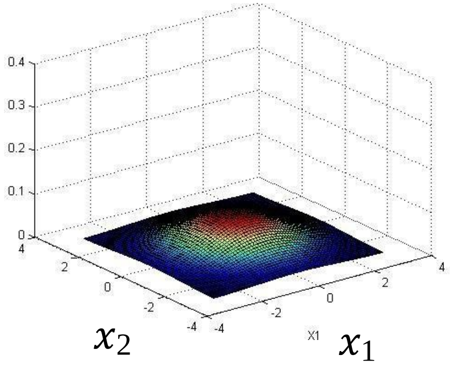


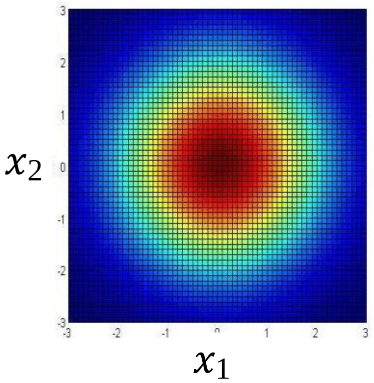
$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}0.6 & 0\\0 & 0.6\end{bmatrix}$$





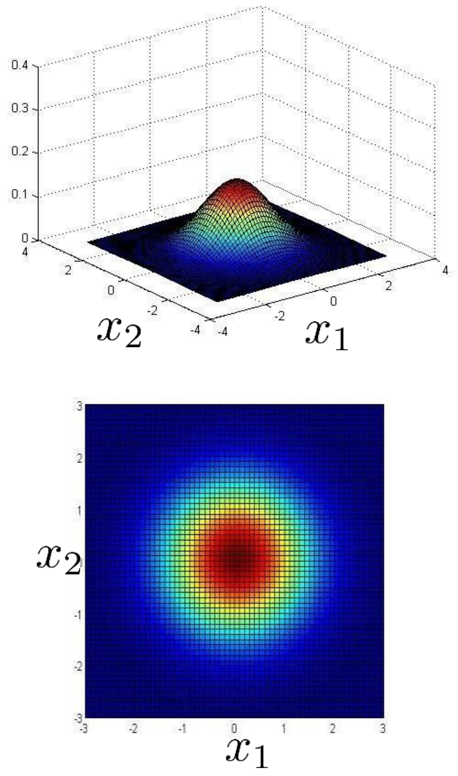
$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}2 & 0\\0 & 2\end{bmatrix}$$



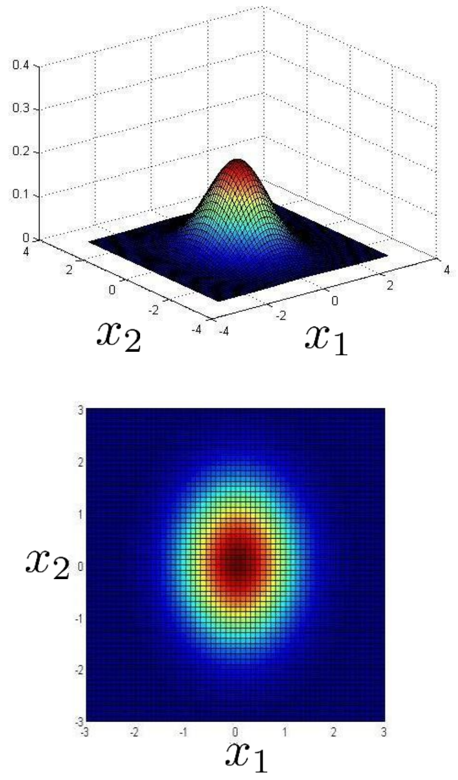


Diagonal Covariance Matrix, the Variance of Features is Equal

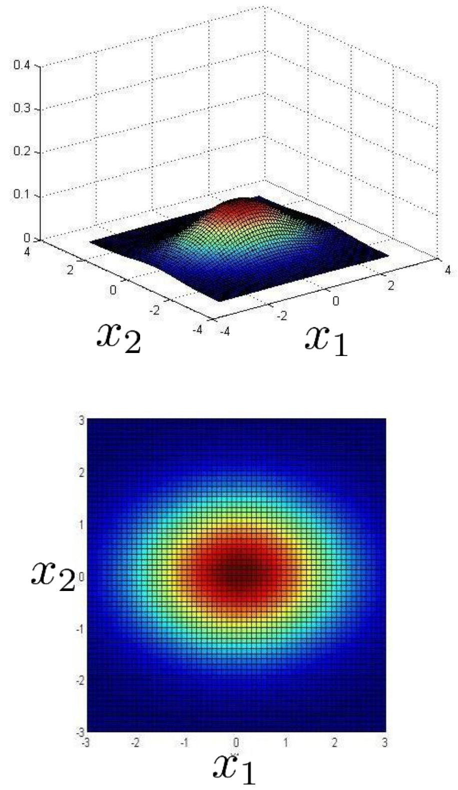
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 0.6 & 0 \\ 0 & 1 \end{bmatrix}$

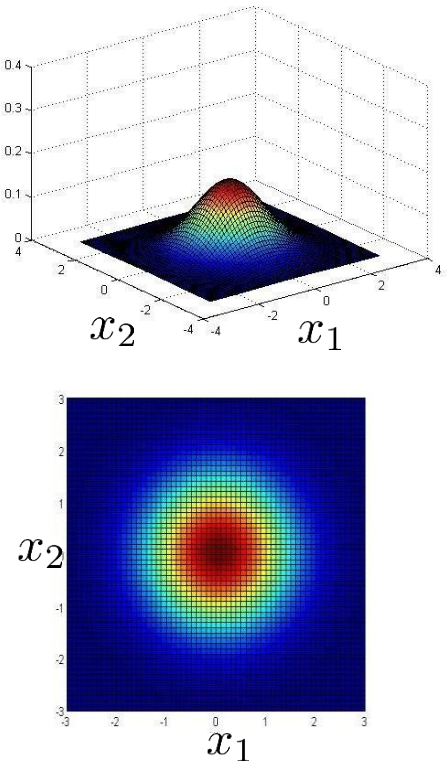


$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; \Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

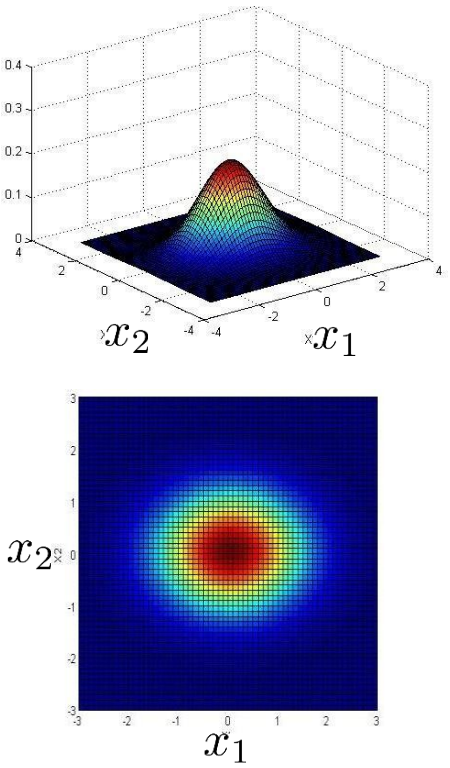


Diagonal Covariance Matrix, the Variance of Features is Equal

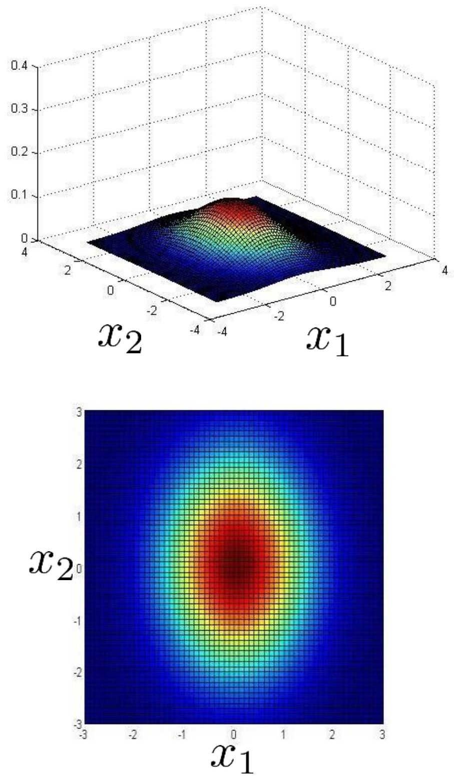
$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.6 \end{bmatrix}$

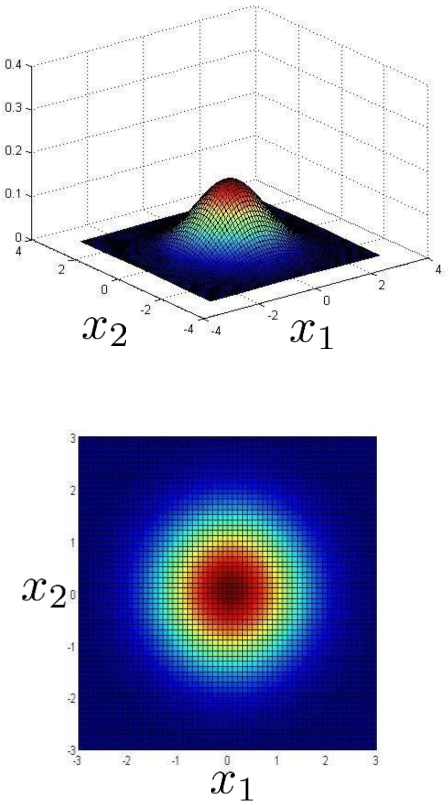


$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$

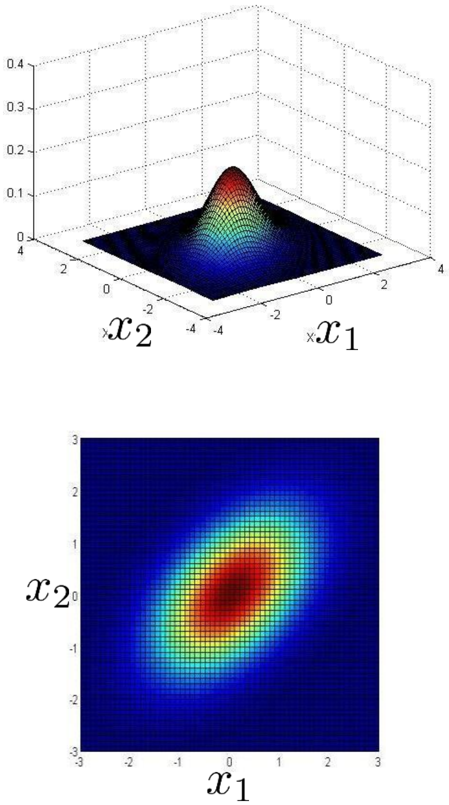


Positive Correlation Between the Features

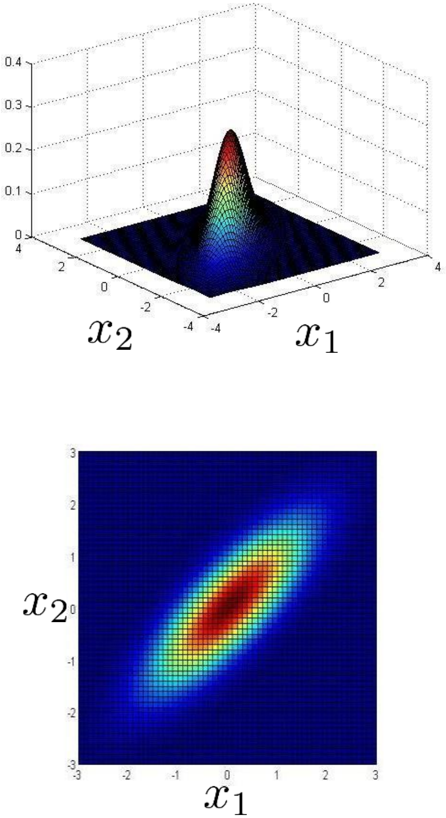
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

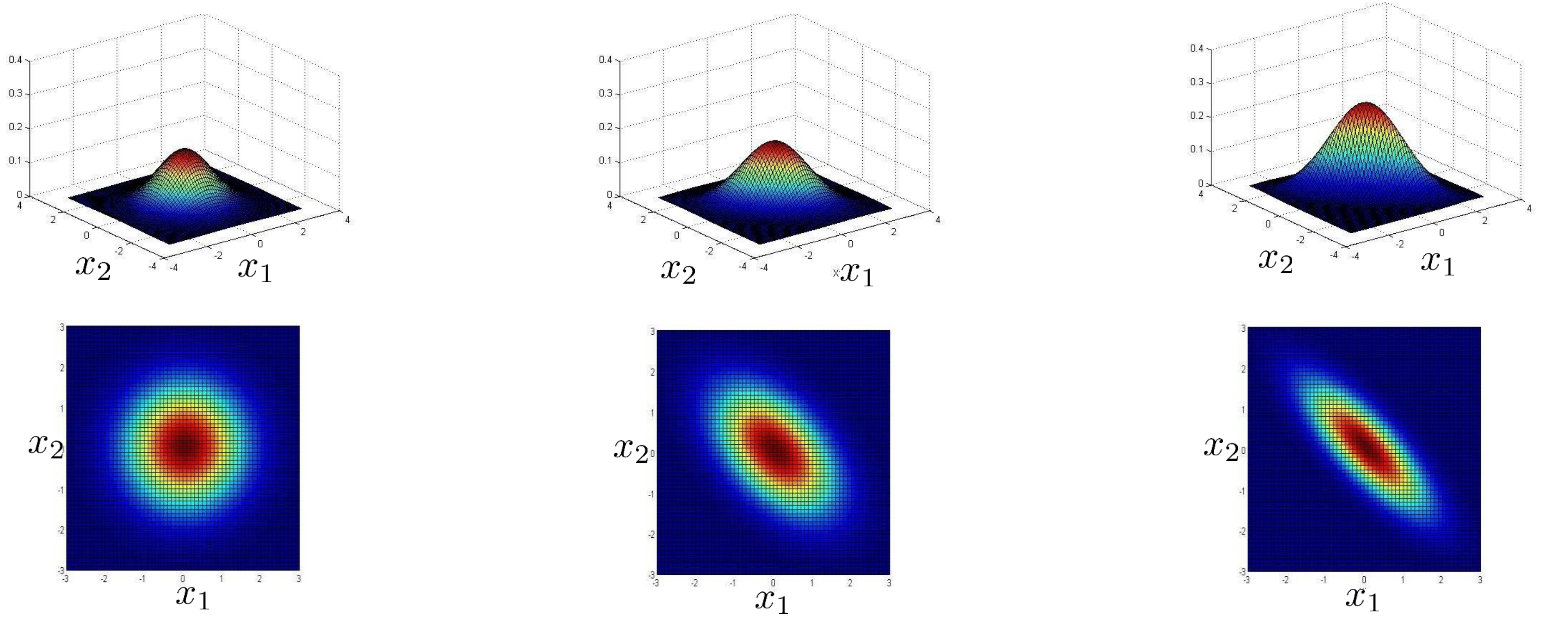


Negative Correlation Between the Features

$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

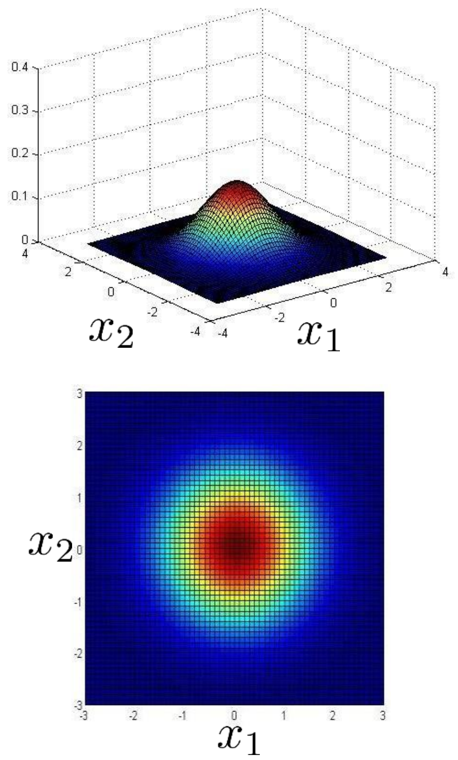
$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}1 & -0.5\\-0.5 & 1\end{bmatrix}$$

$$\mu = \begin{bmatrix}0\\0\end{bmatrix} \quad \Sigma = \begin{bmatrix}1 & -0.8\\-0.8 & 1\end{bmatrix}$$

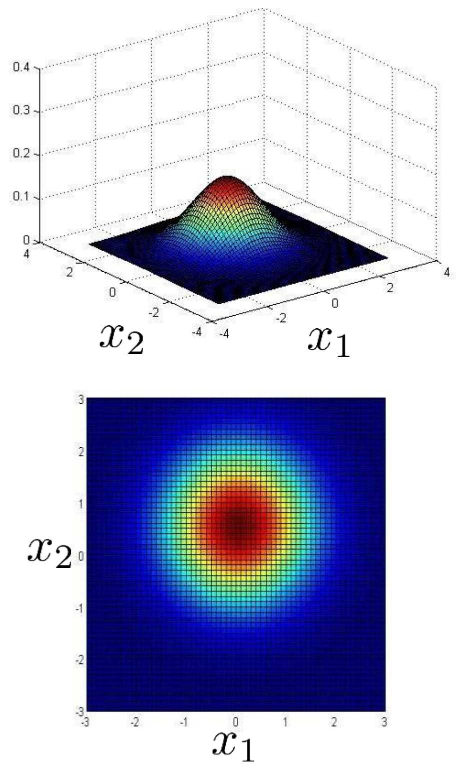


Center (Mean) of Gaussian Distribution

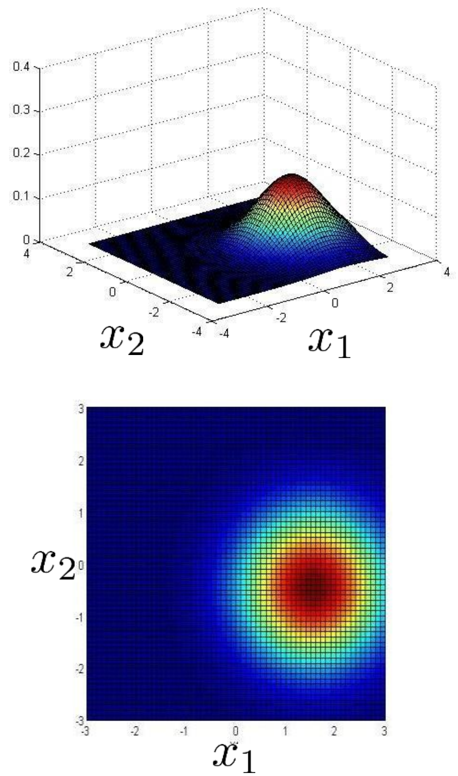
$$\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$



$$\mu = \begin{bmatrix} 1.5 \\ -0.5 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

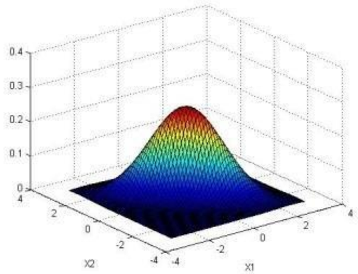


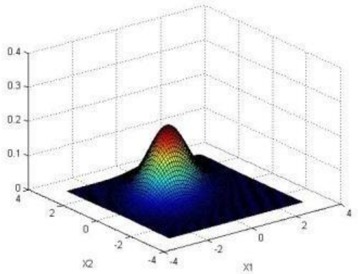
Anomaly Detection with Multivariate Gaussian Function

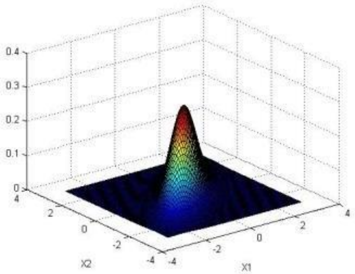
Multivariate Gaussian Distribution

- Multivariate Gaussian distribution function.

$$p(x;\mu,\Sigma)=\frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}}\exp\left(-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)\right)$$







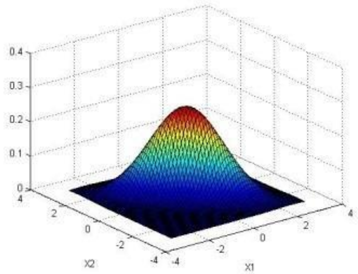
- Estimation of parameters.

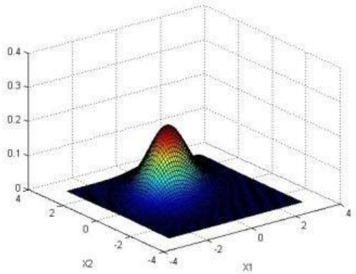
$$\mu=\frac{1}{m}\sum_{i=1}^{m}x^{(i)}\qquad \Sigma=\frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\mu)(x^{(i)}-\mu)^{T}$$

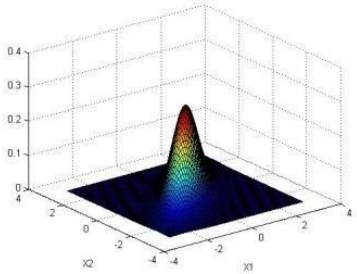
Multivariate Gaussian Distribution

- Multivariate Gaussian distribution function.

$$p(x;\mu,\Sigma)=\frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$







- Estimation of parameters.

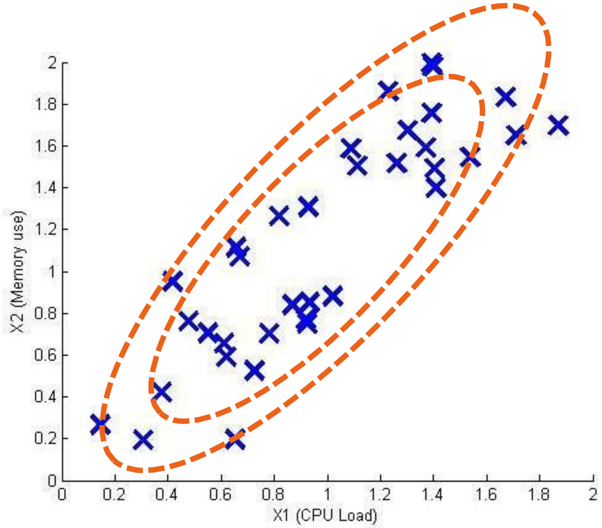
```
mu = np.mean(X, axis=0)
```

```
Sigma = np.cov(X.T)
```

Algorithm

- Estimation of p(x) model parameters

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



- Calculate the value of p(x) for the new data of x

$$p(x;\mu,\Sigma) = \frac{1}{|\Sigma|^{1/2}(2\pi)^{n/2}}\exp\left(-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)\right)$$

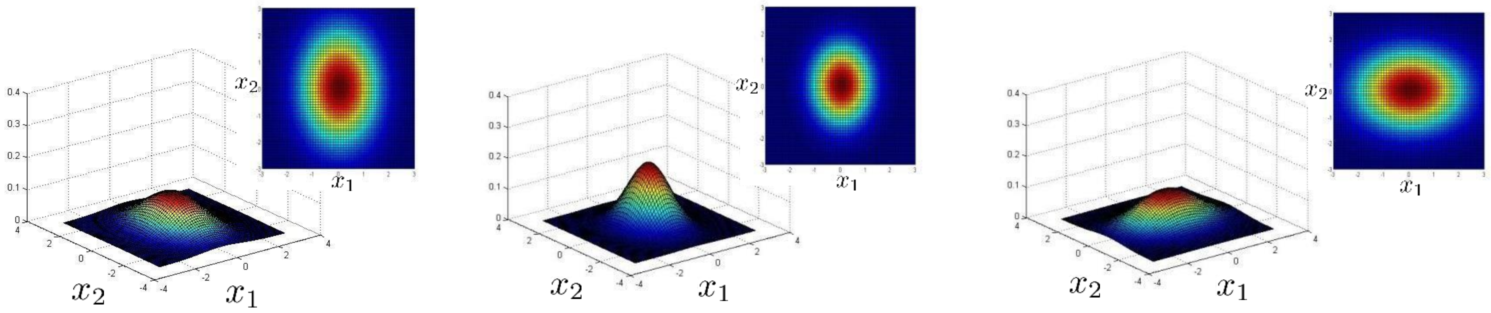
- Printing the <<yes>> output if p(x) < ϵ

Relation with the Primary Model

- Primary model.

$$p(\boldsymbol{x}) = p(x_1; \mu_1, \sigma_1^2)\, p(x_2; \mu_2, \sigma_2^2)\, p(x_3; \mu_3, \sigma_3^2) \cdots p(x_n; \mu_n, \sigma_n^2)$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$



- Relation with multivariate Gaussian distribution.

$$p(x; \mu, \Sigma) = \frac{1}{|\Sigma|^{1/2} (2\pi)^{n/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$

Introductory Model or Multivariate Gaussian Distribution

- Introductory model.
  - Creating features is done manually. (x_1/x_2)
  - Computational costs are relatively low.
  - If the number of training samples is small, it still works correctly. [Number of parameters: 2n]

- Multivariate gaussian distribution.
  - It automatically learns the correlation between features.
  - Computational costs are high. [Calculating the inverse of the covariance matrix]
  - The number of training samples should be more than the number of features. [Invertibility of matrix Σ]