

# Stats Components

## Contents

- `GaussianKDE`
- `MultiVariateGaussian`

Statistical functions.

**`class`** `anomalib.models.components.stats.GaussianKDE(dataset=None)`

Bases: `DynamicBufferMixin`

Gaussian Kernel Density Estimation.

### Parameters:

**dataset** (*Tensor* | *None*, *optional*) – Dataset on which to fit the KDE model. Defaults to *None*.

**`static cov(tensor)`**

Calculate the unbiased covariance matrix.

### Parameters:

**tensor** (*torch.Tensor*) – Input tensor from which covariance matrix is computed.

### Return type:

`Tensor`

### Returns:

Output covariance matrix.

**`fit(dataset)`**

Fit a KDE model to the input dataset.

**Parameters:**

**dataset** (*torch.Tensor*) – Input dataset.

**Return type:**

None

**Returns:**

None

**forward**(*features*)

Get the KDE estimates from the feature map.

**Parameters:**

**features** (*torch.Tensor*) – Feature map extracted from the CNN

**Return type:**

Tensor

Returns: KDE Estimates

**class** `anomalib.models.components.stats`.MultiVariateGaussian

Bases: `DynamicBufferMixin`, `Module`

Multi Variate Gaussian Distribution.

**fit**(*embedding*)

Fit multi-variate gaussian distribution to the input embedding.

**Parameters:**

**embedding** (*torch.Tensor*) – Embedding vector extracted from CNN.

**Return type:**

list [ Tensor ]

**Returns:**

Mean and the covariance of the embedding.

## **forward**(*embedding*)

Calculate multivariate Gaussian distribution.

### Parameters:

**embedding** (*torch.Tensor*) – CNN features whose dimensionality is reduced via either random sampling or PCA.

### Return type:

**list** [**Tensor**]

### Returns:

mean and inverse covariance of the multi-variate gaussian distribution that fits the features.

< [Previous](#)  
[Cluster](#)

[Image Models](#) > Next