

# Package ‘amelie’

July 22, 2025

**Type** Package

**Title** Anomaly Detection with Normal Probability Functions

**Version** 0.2.1

**Description** Implements anomaly detection as binary classification for cross-sectional data. Uses maximum likelihood estimates and normal probability functions to classify observations as anomalous. The method is presented in the following lecture from the Machine Learning course by Andrew Ng: <<https://www.coursera.org/learn/machine-learning/lecture/C8IJp/algorithm/>>, and is also described in: Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, Jaideep Srivastava (2003) <[doi:10.1137/1.9781611972733.3](https://doi.org/10.1137/1.9781611972733.3)>.

**Imports** stats

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat, knitr, rmarkdown

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Dmitriy Bolotov [aut, cre]

**Maintainer** Dmitriy Bolotov <[dbolotov@live.com](mailto:dbolotov@live.com)>

**Repository** CRAN

**Date/Publication** 2019-03-18 06:53:42 UTC

## Contents

ad . . . . .	2
amelie . . . . .	3
pdfunc . . . . .	4
predict.ad . . . . .	4
<b>Index</b>	<b>6</b>

---

ad	<i>ad: anomaly detection with normal probability density functions.</i>
----	---

---

## Description

ad: anomaly detection with normal probability density functions.

## Usage

```
ad(x, ...)

## S3 method for class 'formula'
ad(formula, data, na.action = na.omit, ...)

## Default S3 method:
ad(x, y, univariate = TRUE, score = "f1",
   steps = 1000, ...)

## S3 method for class 'ad'
print(x, ...)
```

## Arguments

x	A matrix of numeric features.
...	Optional parameters to be passed to <code>ad.default</code> .
formula	An object of class "formula": a symbolic description of the model to be fitted.
data	A data frame containing the features (predictors) and target.
na.action	A function specifying the action to be taken if NAs are found.
y	A vector of numeric target values, either 0 or 1, with 1 assumed to be anomalous.
univariate	Logical indicating whether the univariate pdf should be used.
score	String indicating which score to use in optimization: f1 (default) or mcc.
steps	Integer number of steps to take during epsilon optimization, default 1e3.

## Details

`amelie` implements anomaly detection with normal probability functions and maximum likelihood estimates.

Features are assumed to be continuous, and the target is assumed to take on values of 0 (negative case, no anomaly) or 1 (positive case, anomaly).

The threshold `epsilon` is optimized using the either the Matthews correlation coefficient or F1 score.

Variance and covariance are computed using `var` and `cov`, where denominator  $n-1$  is used.

Algorithm details are described in the Introduction vignette.

The package follows the anomaly detection approach in Andrew Ng's course on machine learning.

**Value**

An object of class `ad`:

<code>call</code>	The original call to <code>ad</code> .
<code>univariate</code>	Logical indicating which pdf was computed.
<code>score</code>	The score that was used for optimization.
<code>epsilon</code>	The threshold value.
<code>train_mean</code>	Means of features in the training set.
<code>train_var</code>	Variances of features in the training set. If <code>univariate=FALSE</code>

, holds the covariance matrix for the features.

<code>val_score</code>	The score obtained on the validation data set. 0 to 1 for F1 score, -1 to 1 for Matthews correlation coefficient
------------------------	--

**References**

[Machine learning course](#)  
[Confusion matrix](#)  
[Matthews correlation coefficient](#)

**Examples**

```
x1 <- c(1,.2,3,1,1,.7,-2,-1)
x2 <- c(0,.5,0,.4,0,1,-.3,-.1)
x <- do.call(cbind,list(x1,x2))
y <- c(0,0,0,0,0,0,1,1)
dframe <- data.frame(x,y)
df_fit <- ad(y ~ x1 + x2, dframe)
mat_fit <- ad(x = x, y = y)
```

---

amelie

*amelie: A package for anomaly detection.*

---

**Description**

Anomaly detection with maximum likelihood estimates and normal probability functions.

**Amelie functions**

The package contains a function for running the anomaly detection algorithm.

**More information**

[ad](#) documents the main `ad` function.

For more details and examples, see the vignette.

---

pdfunc	<i>Compute the probability density function of a matrix of features.</i>
--------	--

---

**Description**

Compute the probability density function of a matrix of features.

**Usage**

```
pdfunc(x, univariate = TRUE)
```

**Arguments**

x	A matrix of numeric features.
univariate	Logical indicating whether the univariate pdf should be computed.

**Details**

pdfunc computes univariate or multivariate probabilities for a set of observations.

All columns of a row are used in computing the pdf.

Variance and covariance are computed using var and cov, where the denominator n-1 is used.

**Value**

A vector with values of the density function.

**Examples**

```
dmat <- matrix(c(3,1,3,1,2,3,-1,0),nrow=2)
pdfunc(dmat,TRUE)

# '@importFrom stats cov
```

---

predict.ad	<i>Predict method for ad Objects</i>
------------	--------------------------------------

---

**Description**

Predict method for ad Objects

**Usage**

```
## S3 method for class 'ad'
predict(object, newdata, type = "class",
        na.action = na.pass, ...)
```

**Arguments**

<code>object</code>	An object of class <code>ad</code> , created by the function <code>ad</code> .
<code>newdata</code>	A data frame or matrix containing new data.
<code>type</code>	One of <code>'class'</code> (for class prediction) or <code>'prob'</code> (for probabilities).
<code>na.action</code>	A function specifying the action to be taken if NAs are found; default is to predict NA ( <code>na.pass</code> ).
<code>...</code>	Currently not used.

**Details**

Specifying `'class'` for `type` returns the class of each observation as anomalous or non-anomalous.  
Specifying `'prob'` returns the probability of each observation.

**Value**

A vector of predicted values.

**Examples**

```
x1 <- c(1,.2,3,1,1,.7,-2,-1)
x2 <- c(0,.5,0,.4,0,1,-.3,-.1)
x <- do.call(cbind,list(x1,x2))
y <- c(0,0,0,0,0,0,1,1)
dframe <- data.frame(x,y)
df_fit <- ad(y ~ x1 + x2, dframe)
predict(df_fit, newdata = dframe)
```

# Index

`ad`, [2](#), [3](#)  
`amelie`, [3](#)  
`amelie-package (amelie)`, [3](#)  
  
`pdfunc`, [4](#)  
`predict.ad`, [4](#)  
`print.ad (ad)`, [2](#)