

Preprocessing

Quantile Binning Transformation

Syntax

1. `[Xt,Q] = binning(X,n_bins)`
2. `Xt = binning(X,Q)`

Description

1. Discrete the continuous variables for each column of a matrix based on quantiles.

X is a M-by-N matrix with continuous variables in each column.

n_bins is the number of groupings (n_bins > 2).

Examples

1.

```
X = [16  2;  
     5  11;  
     9   7;  
     4  14];  
n_bins = 3;  
[Xt,Q] = binning(X,n_bins)  
Xt =  
     2   0  
     1   2  
     2   1  
     0   2  
Q =  
     5   7  
     9  11
```

2.

```
X2 = [ 3  13;  
      10  8;  
       6 12;  
      15  1];  
X2t = binning(X2,Q)  
X2t =  
     0   2  
     2   1  
     1   2  
     2   0
```

Principal Component Analysis

Syntax

1. `[Xt,coeff] = principal_component(X,n_components)`
2. `Xt = principal_component(X,coeff)`

Description

Author: David Ferreira
Federal University of Amazonas
PhD student in Electrical Engineering
Contact: ferreirad08@gmail.com

1. Apply the PCA transformation in the features and returns the principal component coefficients.
2. Apply the PCA transformation in the features based on input coefficients.

X is a M-by-N matrix with features (continuous variables) in each column.
n_components is the number of principal component.

Examples

1.

```
X = [1 2 3; 4 5 6; 7 8 9];  
n_components = 2;  
[Xt,coeff] = principal_component(X,n_components)  
Xt =  
    -5.1962    -0.0000  
         0         0  
     5.1962     0.0000  
coeff =  
    0.5774   -0.0332  
    0.5774   -0.6899  
    0.5774    0.7231
```

2.

```
Xt = principal_component(X,coeff)  
Xt =  
    -5.1962   -0.0000  
         0         0  
     5.1962    0.0000
```

Information Gain (IG)

Syntax

1. [Xt,indexes,g] = feature_selection_gain(X,Y,k)

Predictors

k-Nearest Neighbors (kNN)

Syntax

1. label = predict_knnneighbors(X,Y,Xnew,k)

Description

1. Returns the estimated labels of one or multiple test instances.

X is a M-by-N matrix, with M instances of N features.

Y is a M-by-1 matrix, with respective M labels to each training instance.

Xnew is a P-by-N matrix, with P instances of N features to be classified.

k is a scalar with the number of nearest neighbors selected.

Examples

```
1.
load fisheriris
X = meas;
Y = species;
Xnew = [min(meas);mean(meas);max(meas)];
k = 5;
label = predict_knneighbors(X,Y,Xnew,k)
label =
    'setosa'
    'versicolor'
    'virginica'
```

find_knn

Syntax

1. Xnearest = find_knn(X,Y,Xnew,k)
2. [Xnearest,Ynearest,distances] = find_knn(X,Y,Xnew,k)

Description

1. Returns the k nearest training instances.
2. Returns the k nearest training instances, the k nearest training labels and the respective distances.

X is a M-by-N matrix, with M instances of N features.

Y is a M-by-1 matrix, with respective M labels to each training instance.

Xnew is a 1-by-N matrix, with one instance of N features to be classified.

k is a scalar, with the number of nearest neighbors selected.

Examples

```
1.
X = [8 5 1; 3 7 2; 3 6 3; 7 3 1];
Y = {'fruit';'vegetable';'protein';'fruit'};
Xnew = [6 4 1];
k = 3;
[Xnearest,Ynearest,distances] = find_knn(X,Y,Xnew,k)
Xnearest =
    7 3 1
    8 5 1
    3 6 3
Ynearest =
    'fruit'
    'fruit'
    'protein'
distances =
    1.4142
    2.2361
    4.1231
```

Gaussian Naive Bayes (GNB)

Syntax

1. [label,model] = predict_gaussiannb(X,Y,Xnew)
2. label = predict_gaussiannb(model,Xnew)

Description

1. Returns the estimated labels of one or multiple test instances.

X is a M-by-N matrix, with M instances of N features.

Y is a M-by-1 matrix, with respective M labels to each training instance.

Xnew is a P-by-N matrix, with P instances of N features to be classified.

Examples

1.

```
load fisheriris
```

```
X = meas;
```

```
Y = species;
```

```
Xnew = [min(meas);max(meas)];
```

```
[label,model] = predict_gaussiannb(X,Y,Xnew)
```

```
label =
```

```
    'setosa'
```

```
    'virginica'
```

```
model =
```

```
      C      M      S      prior
```

```
'setosa' [1x4 double] [1x4 double] 0.33333
```

```
'versicolor' [1x4 double] [1x4 double] 0.33333
```

```
'virginica' [1x4 double] [1x4 double] 0.33333
```

2.

```
Xnew = mean(meas);
```

```
label = predict_gaussiannb(model,Xnew)
```

```
label =
```

```
    'versicolor'
```

Gaussian Naïve Bayes (GNB)

Syntax

1. [labels,probabilities] = find_gaussiannb(X,Y,Xnew)

Description

1. Returns the labels with their respective probabilities in descending order.

X is a M-by-N matrix, with M instances of N features.

Y is a M-by-1 matrix, with respective M labels to each training instance.

Xnew is a 1-by-N matrix, with one instance of N features to be classified.

Examples

1.

```
load fisheriris
```

```
X = meas;
```

```
Y = species;
```

Author: David Ferreira
Federal University of Amazonas
PhD student in Electrical Engineering
Contact: ferreirad08@gmail.com

```
Xnew = mean(meas);  
[labels,probabilities] = find_gaussiannb(X,Y,Xnew)  
labels =  
    'versicolor'  
    'virginica'  
    'setosa'  
probabilities =  
    1.0000  
    0.0000  
    0.0000
```

Decision Tree (DT)

Syntax

1. label = predict_dtree(X,Y,Xnew)

Description

1. Returns the estimated labels of one or multiple test instances.

X is a M-by-N matrix, with M instances of N features.

Y is a M-by-1 matrix, with respective M labels to each training instance.

Xnew is a P-by-N matrix, with P instances of N features to be classified.

Examples

```
1.  
    load fisheriris  
    X = meas;  
    Y = species;  
    Xnew = [min(meas);mean(meas);max(meas)];  
    label = predict_dtree(X,Y,Xnew)  
    label =  
        'setosa'  
        'versicolor'  
        'virginica'
```