

# Programming

## Applied Machine Learning

Luna Pianesi

Faculty of Technology, Bielefeld University

```
332
333
334     if extrapolate is None:
335         extrapolate = self.extrapolate
336     x = np.asarray(x)
337     x_shape, x_ndim = x.shape, x.ndim
338     x = np.ascontiguousarray(x.ravel(), dtype=np
339
340     # With periodic extrapolation we map x to the
341     # [self.t[k], self.t[n]].
342     if extrapolate == 'periodic':
343         n = self.t.size - self.k - 1
344         x = self.t[self.k] + (x - self.t[self.k]) *
345
346         extrapolate = False
347
348     out = np.empty((len(x), prod(self.c.shape[1:])),
349                   dtype=self._evaluate(x, nu, extrapolate, out))
350     self._ensure_c_contiguous()
351     out = out.reshape(x_shape + self.c.shape[1:])
352     if self.axis != 0:
353         # transpose to move the calculated values to t
354         l = list(range(out.ndim))
355         l = l[x_ndim:x_ndim+self.axis] + l[:x_ndim] + l[x_ndim+self.axis:]
356         out = out.transpose(l)
357     return out
358
359 def _evaluate(self, xp, nu, extrapolate, out):
360     _bspl.evaluate_spline(self.t, self.c.reshape(self.c
361
362     self.k, xp, nu, extrapolate, out)
363
364
365 def _ensure_c_contiguous(self):
366     """
367     C and t may be modified by the user. The Cython code
368     c and t are C contiguous.
369     """
370     self.c = np.ascontiguousarray(self.c)
371     self.t = np.ascontiguousarray(self.t)
```

***Machine  
Learning***

***Scikit-Learn***

***Applications***

# ***Machine Learning***

- ❖ Branch of artificial intelligence
- ❖ Combination of statistics, optimization theory, computer science, information theory, ...

## ***Unsupervised Learning***

- ❖ Dimensionality reduction
- ❖ Clustering

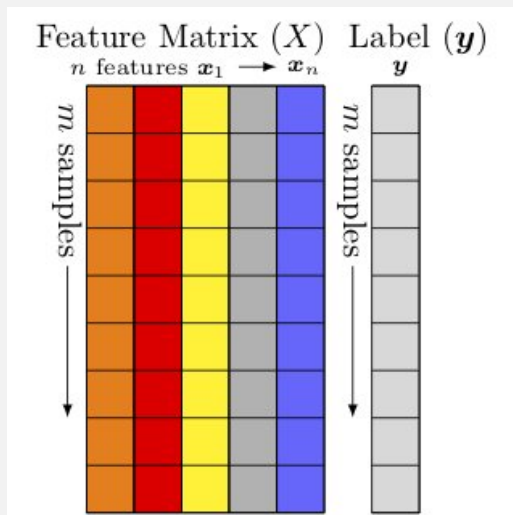
## ***Supervised Learning***

- ❖ Classification
- ❖ Regression

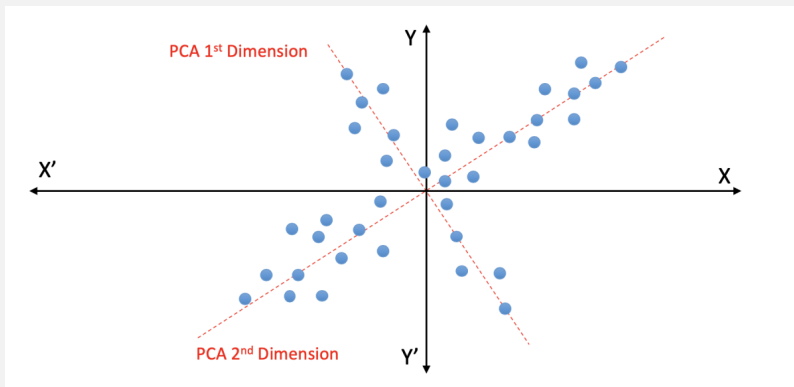
# Data representation: feature matrix

	Feature-1	Feature-2	Feature-3	Feature-4	...	...	Feature-n	
	$x_1^1$	$x_2^1$	$x_3^1$	$x_4^1$	...	...	$x_n^1$	Sample-1
	$x_1^2$	$x_2^2$	$x_3^2$	$x_4^2$	...	...	$x_n^2$	Sample-2
	$x_1^3$	$x_2^3$	$x_3^3$	$x_4^3$	...	...	$x_n^3$	Sample-3
	...	...	...	...	...	...	...	
	$x_1^m$	$x_2^m$	$x_3^m$	$x_4^m$	...	...	$x_n^m$	Sample-m

# Data representation: feature matrix



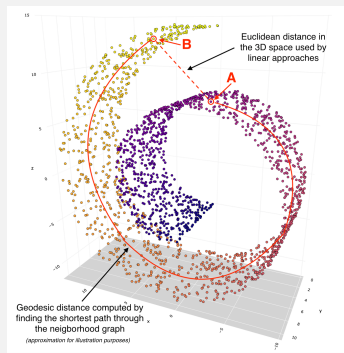
# Dimensionality reduction



Some methods:

- ❖ Principal Component Analysis (PCA)
- ❖ Isomap

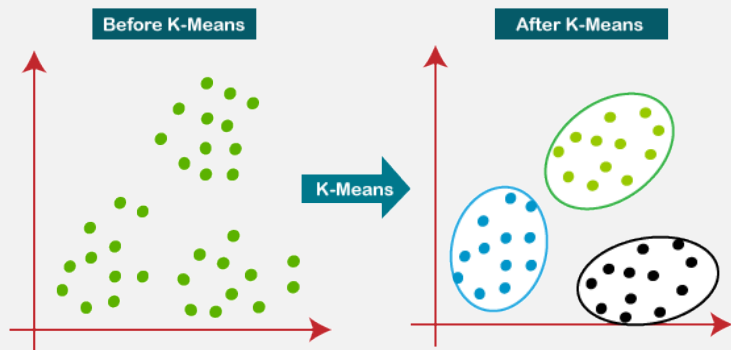
# Dimensionality reduction



Some methods:

- ❖ Principal Component Analysis (PCA)
- ❖ Isomap

# Clustering

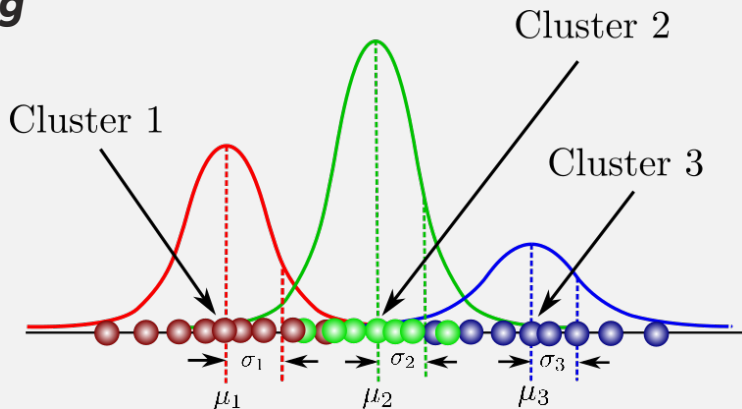


Some methods:

- ❖ K-means
- ❖ Gaussian Mixture Models (GMM)
- ❖ Spectral Clustering



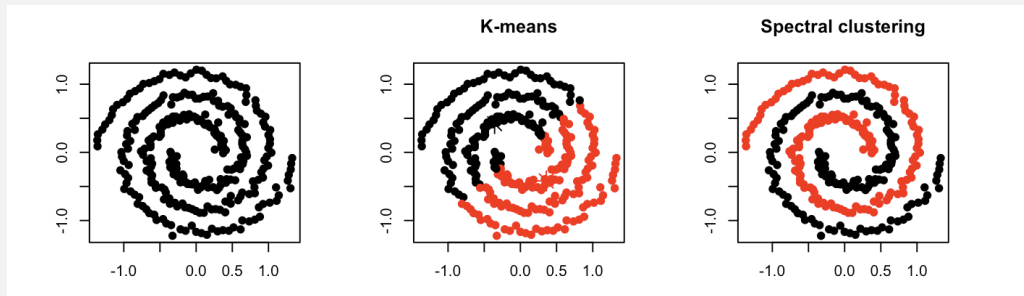
# Clustering



Some methods:

- ❖ K-means
- ❖ Gaussian Mixture Models (GMM)
- ❖ Spectral Clustering

# Clustering



Some methods:

- ❖ K-means
- ❖ Gaussian Mixture Models (GMM)
- ❖ Spectral Clustering

# Classification

The diagram illustrates Bayes' Theorem with the following components:

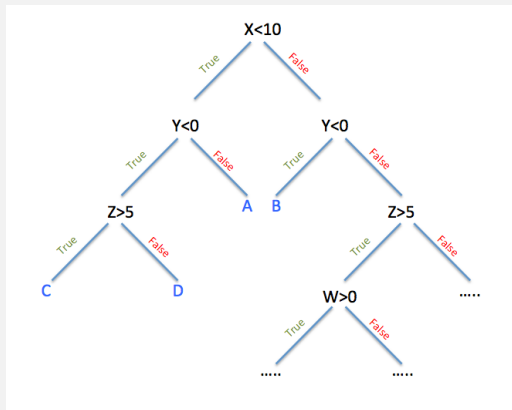
- Likelihood of the Evidence given that the Hypothesis is True** (yellow text, top left):  $P(E|H)$
- Prior Probability of the Hypothesis** (red text, top right):  $P(H)$
- Posterior Probability of the Hypothesis given that the Evidence is True** (blue text, bottom left):  $P(H|E)$
- Prior Probability that the evidence is True** (green text, bottom right):  $P(E)$

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Some methods:

- ❖ Naive Bayes
- ❖ Decision Trees

# Classification



Some methods:

- ❖ Naive Bayes
- ❖ Decision Trees

# *Regression*

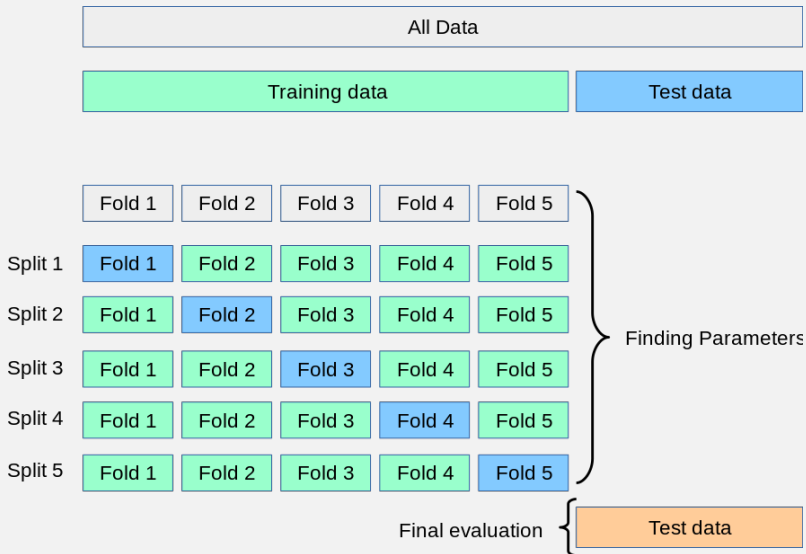
Some methods:

- ❖ Linear regression
- ❖ Ridge regression, Lasso regression
- ❖ Multiple regression
- ❖ Multivariate regression

# *Feature representation*

- ❖ Categorical: can be stored and identified by names or labels
- ❖ Numerical: simply numbers

# Cross-validation



# Quiz

- Assign the following methods to their categories:
  - Naive Bayes
  - Kmeans
  - PCA
  - Decision Tree
  - Gaussian Mixture Models
  - Isomap
  - Spectral Clustering
  
- *True or false?*
  - Cross validation can only be performed on labeled data
  - Gaussian Mixture Models assumes that data points follow a normal distribution



# Quiz

➤ Assign the following methods to their categories:

- |                           |                     |
|---------------------------|---------------------|
| ➤ Naive Bayes             | Classification      |
| ➤ Kmeans                  | Clustering          |
| ➤ PCA                     | Dimensionality red. |
| ➤ Decision Tree           | Classification      |
| ➤ Gaussian Mixture Models | Clustering          |
| ➤ Isomap                  | Dimensionality red. |
| ➤ Spectral Clustering     | Clustering          |

➤ *True or false?*

- |   |      |
|---|------|
| ➤ Cross validation can only be performed on labeled data                        | true |
| ➤ Gaussian Mixture Models assumes that data points follow a normal distribution | true |

***Machine  
Learning***

***Scikit-Learn***

***Applications***

# *The Estimator API*

Estimators of the Scikit-Learn package share a common API.

Use of estimators:

- ❖ Choose model (Estimator)
- ❖ Choose model hyperparameters
- ❖ Instantiate model with hyperparameters
- ❖ Call `fit()` to train the model on a given data set
- ❖ Apply model to new data:
  - ❖ Supervised learning: call `predict()`
  - ❖ Unsupervised learning: call `transform()` or `predict()` (depending on the estimator)

# Quiz

- ❖ True or false?
  - ❖ The basic steps are *model, fit, predict/transform*
  - ❖ `LinearRegression.coef_` returns slope and intercept of line
  - ❖ Scikit-Learn can generate artificial datasets
  - ❖ Scikit-Learn doesn't provide real world data sets
  - ❖ transformers uses the `predict()` to transform data.
- ❖ Explain the function of the following estimators:
  - ❖ `OneHotEncoder`
  - ❖ `ColumnTransformer`
  - ❖ `DictVectorizer`
  - ❖ `CountVectorizer`

# Quiz

## ❖ True or false?

- ❖ The basic steps are *model, fit, predict/transform* true
- ❖ `LinearRegression.coef_` returns slope and intercept of line false
- ❖ Scikit-Learn can generate artificial datasets true
- ❖ Scikit-Learn doesn't provide real world data sets false
- ❖ transformers uses the `predict()` to transform data. false

## ❖ Explain the function of the following estimators:

- ❖ `OneHotEncoder` Transforms one categorical feature with  $n$  possible values into  $n$  binary features
- ❖ `ColumnTransformer` Transforms all columns of a `DataFrame`
- ❖ `DictVectorizer` Transforms `dict` with categorical variables into numeric features
- ❖ `CountVectorizer` Tokenizes strings and constructs word count frequency matrix

***Machine  
Learning***

***Scikit-Learn***

***Applications***

# Quiz

- ❖ In which order does function `train_test_split` return test/train data?
  - ❖ `Xtrain, Ytrain, Xtest, Ytrain`
  - ❖ `Xtest, Ytest, Xtrain, Ytrain`
  - ❖ `Xtrain, Xtest, Ytrain, Ytest`
  - ❖ `Xtest, Xtrain, Ytest, Ytrain`
- ❖ What data is stored in
  - ❖ `digits.images`
  - ❖ `digits.data`
  - ❖ `digits.target`

# Quiz

- ❖ In which order does function `train_test_split` return test/train data?
  - ❖ `Xtrain, Ytrain, Xtest, Ytrain`
  - ❖ `Xtest, Ytest, Xtrain, Ytrain`
  - ❖ `Xtrain, Xtest, Ytrain, Ytest` ✓
  - ❖ `Xtest, Xtrain, Ytest, Ytrain`
- ❖ What data is stored in
  - ❖ `digits.images`    bitmap data of all images
  - ❖ `digits.data`    feature matrix
  - ❖ `digits.target`    labels (ground truth digits)



# *Recap*

# Summary

- ❖ Machine Learning
  - ❖ Dimensionality reduction
  - ❖ Clustering
  - ❖ Classification
  - ❖ Regression
- ❖ Scikit-Learn
  - ❖ Estimator API
  - ❖ Feature representation
  - ❖ Crossvalidation
- ❖ Applications
  - ❖ Handwritten digits dataset
  - ❖ Text comparison

## *What comes next?*

- ❖ Have a look at the Jupyter Notebook of this lecture
- ❖ Further reading about Pandas: Chapter 5 of the “Python Data Science Handbook”:  
<https://jakevdp.github.io/PythonDataScienceHandbook/>
- ❖ Have a look at the in-depth analyses that are provided in the handbook