

# Introduction to machine learning

Victor Kitov

[v.v.kitov@yandex.ru](mailto:v.v.kitov@yandex.ru)

# Course information

- Instructor - Victor Vladimirovich Kitov
  - MSU, NES
  - practical experience
  - academic experience
  - ensemble learning
- Tasks of the course
- Structure: lectures, seminars
- Practice:
  - theoretical tasks
  - programming using python
    - ipython notebook, numpy, scipy, pandas, scikit-learn.

## Recommended materials

- Лекции К.В.Воронцова (видео-лекции и материалы на [machinelearning.ru](http://machinelearning.ru))
- **Statistical Pattern Recognition**. 3rd Edition, Andrew R. Webb, Keith D. Copsey, John Wiley & Sons Ltd., 2011.
- **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2nd Edition, Springer, 2009. <http://statweb.stanford.edu/~tibs/ElemStatLearn/>.
- **Machine Learning: A Probabilistic Perspective**. Kevin P. Murphy. Massachusetts Institute of Technology. 2012.
- **Pattern Recognition and Machine Learning**. Christopher M. Bishop. Springer. 2006.
- **Any additional public sources** - wikipedia, articles, tutorials, video-lectures.

# Table of Contents

- 1 Tasks solved by machine learning
- 2 Main concepts of machine learning.
- 3 Practical applications of machine learning

## Formal definitions of machine learning

- Machine learning is a field of study that gives computers the ability to learn without being explicitly programmed.
- A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P**, if its performance  $P$  at tasks in  $T$  improves with experience  $E$ .
- Examples: spam filtering, speech recognition, image recognition (face detection, eyes detection, pose detection, person identification).

## Major niches of ML

- dealing with huge datasets with many attributes (text categorization)
- hard to formulate explicit rules (image recognition)
- further adaptation to usage conditions is required (voice detection)
- fast adaptation to changing conditions (stock prices prediction)

## Connections with other fields

- Computer science
- Pattern recognition
  - recognize patterns and regularities in the data
- Artificial intelligence
  - create devices capable of intelligent behavior
- Time-series analysis
- Theory of probability, statistics
  - rely on probabilistic model
- Optimization methods
- Theory of algorithms

## General problem statement

- Set of objects  $O$
- Each object is described by a vector of known characteristics  $\mathbf{x} \in \mathcal{X}$  and predicted characteristics  $y \in \mathcal{Y}$ .

$$o \in O \longrightarrow (\mathbf{x}, y)$$

- Usually  $\mathcal{X} = \mathbb{R}^D$ ,  $\mathcal{Y}$  - a scalar, but they may be any structural descriptors of objects in general.



## General problem statement

- Task: find a mapping  $f$ , which could accurately approximate  $\mathcal{X} \rightarrow \mathcal{Y}$ .
  - using a finite «training» set of objects with known  $(x, y)$ .
  - to apply on a set of objects of interest
- Questions solved in ML:
  - how to select object descriptors - features
  - in what sense a mapping  $f$  should approximate true relationship
  - how to construct  $f$

# Examples

- Spam filtering
- Document classification
- Web-page ranking
- Sentimental analysis
- Intrusion/fraud detection
- Churn prediction
- Target detection / classification
- Handwriting recognition
- Part-of-speech tagging
- Credit scoring
- Particle classification

## Variants of problem statement

- For each new object  $x$  need to associate  $y$ .
- What is known:
  - $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  - supervised learning:
  - $x_1, x_2, \dots, x_N$  - unsupervised learning
    - dimensionality reduction
    - clustering
  - $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N), x_{N+1}x_{N+2}, \dots, x_{N+M}$  - semi-supervised learning.
- If predicted objects  $x'_1, x'_2, \dots, x'_K$  for which  $y$  is forecasted, are known in advance, then this is «transductive» learning.

# Generative and discriminative - models

## Generative model

Full distribution  $p(x, y)$  is modeled.

- Can generate new observations  $(x, y)$

$$\begin{aligned}\hat{y}(x) &= \arg \max_y p(y|x) = \arg \max_y \frac{p(x, y)}{p(x)} = \arg \max_y p(y)p(x|y) \\ &= \arg \max_y \{\log p(y) + \log p(x|y)\}\end{aligned}$$

## Generative and discriminative - models

### Generative model

Full distribution  $p(x, y)$  is modeled.

- Can generate new observations  $(x, y)$

$$\begin{aligned}\hat{y}(x) &= \arg \max_y p(y|x) = \arg \max_y \frac{p(x, y)}{p(x)} = \arg \max_y p(y)p(x|y) \\ &= \arg \max_y \{\log p(y) + \log p(x|y)\}\end{aligned}$$

### Discriminative model

- **Discriminative with probability:** only  $p(y|x)$  is modeled
- **Reduced discriminative:** only  $y = f(x)$  is modeled.

# Generative and discriminative - discussion

- **Disadvantages of generative models:**
  - Discriminative models are more general
  - $p(x|y)$  may be inaccurate in high dimensional spaces

# Generative and discriminative - discussion

- **Disadvantages of generative models:**
  - Discriminative models are more general
  - $p(x|y)$  may be inaccurate in high dimensional spaces
- **Advantages of generative models:**
  - Generative models can be adjusted to varying  $p(y)$
  - Naturally adjust to missing features (by marginalization)
  - Easily detect outliers (small  $p(x)$ )

# Types of target variable

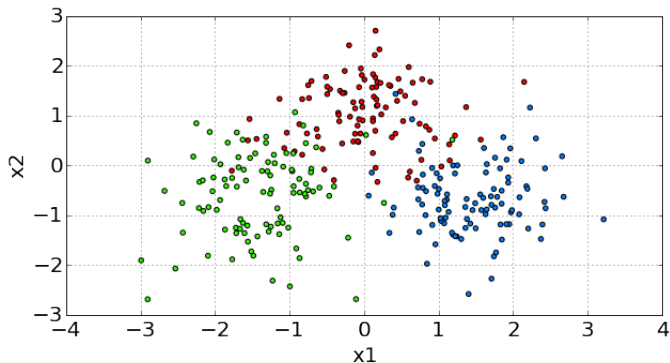
- Types of target variable:
  - $\mathcal{Y} = \mathbb{R}$  - regression (in supervised learning)
  - $\mathcal{Y} = \mathbb{R}^M$  - vector regression (in supervised learning) or feature extraction (in unsupervised learning)
  - $\mathcal{Y} = \{\omega_1, \omega_2, \dots, \omega_C\}$  - classification (in supervised learning) or clustering (in unsupervised learning).
    - $C=2$ : binary classification, encoding -  $\mathcal{Y} = \{+1, -1\}$  or  $\mathcal{Y} = \{0, 1\}$ .
    - $C>2$ : multiclass classification
  - $\mathcal{Y}$ -set of all sets of  $\{\omega_1, \omega_2, \dots, \omega_C\}$  - labeling
    - $\mathcal{Y} = \{y \in \mathbb{R}^C : y_i \in \{0, 1\}\}$ ,  $y_i = 1 \Leftrightarrow$  object is associated with  $\omega_i$ .



# Types of features

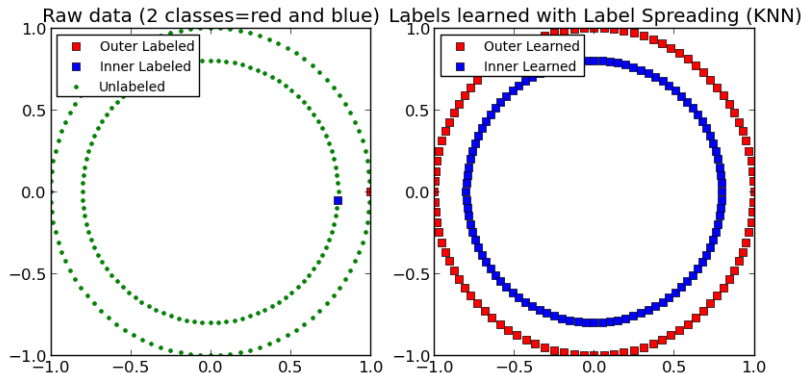
- Full object description  $\mathbf{x} \in \mathcal{X}$  consists of individual features  $x_i \in \mathcal{X}_i$
- Types of feature:
  - $\mathcal{X}_i = \{0, 1\}$  - binary feature
  - $|\mathcal{X}_i| < \infty$  - discrete (nominal) feature
  - $|\mathcal{X}_i| < \infty$  and  $\mathcal{X}_i$  is ordered - ordinal feature
  - $\mathcal{X}_i = \mathbb{R}$  - real feature

# Example of classification



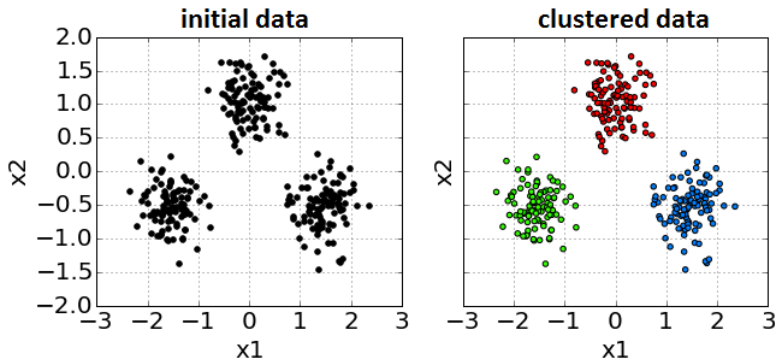
Supervised learning:  $x = (x_1, x_2)$ ,  $y$  is shown with color

# Example of semi-supervised learning



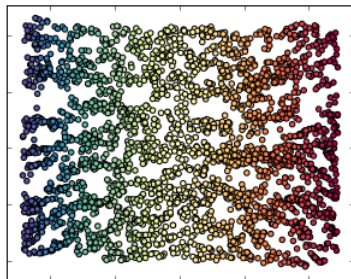
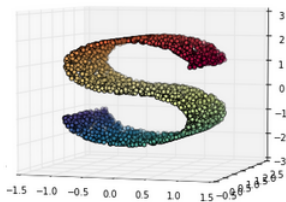
Semi-supervised learning.

# Example of clustering



Unsupervised learning: clustering

# Example of dimensionality reduction



Unsupervised learning: dimensionality reduction

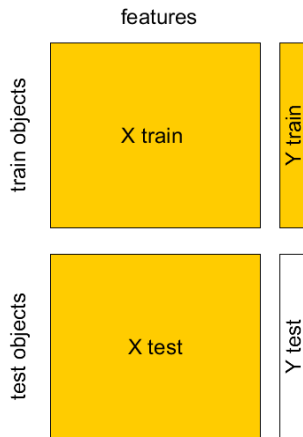
# Table of Contents

- 1 Tasks solved by machine learning
- 2 Main concepts of machine learning.
- 3 Practical applications of machine learning

## Training set

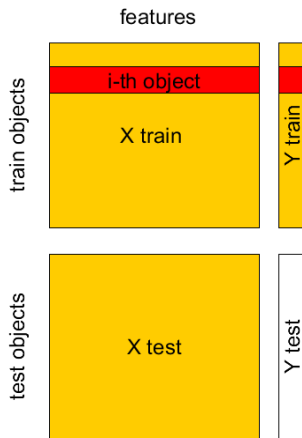
- **Training set:**  $X \in \mathbb{R}^{N \times D}$  - **design matrix**,  $Y \in \mathbb{R}^N$  - predicted outputs (target values)
- Using  $X, Y$  the task is to estimate unknown parameters  $\hat{\theta}$  of mapping  $\hat{y} = f_{\theta}(x)$  so that it will approximate true relationship  $y = y(x)$
- It is assumed that  $z_n = (x_n, y_n)$  for  $n = 1, 2, \dots, N$  - are independent and identically distributed random variables (i.i.d).
- Two steps of ML:
  - **training**
  - **application**

# Train set, test set





# Train set, test set



$N$  - number of objects for which targets ( $Y$ ) are known.

# Train set, test set



$D$  - number of features (advanced case: variable feature count).

# Loss function

- **Loss function**  $\mathcal{L}(\hat{y}, y)$
- Examples:
  - classification:

- misclassification rate

$$\mathcal{L}(\hat{y}, y) = \mathbb{I}[\hat{y} \neq y]$$

- regression:
  - MAE (mean absolute error):

$$\mathcal{L}(\hat{y}, y) = |\hat{y} - y|$$

- MSE (mean squared error):

$$\mathcal{L}(\hat{y}, y) = (\hat{y} - y)^2$$

- absolute relative error:  $\frac{|\hat{y}-y|}{|y|}$ , squared relative error:  $\left(\frac{\hat{y}-y}{y}\right)^2$

## Score versus loss

- In machine learning objects, predicted classes, prediction functions, etc. can be assigned:
  - **score, rating** - this should be maximized
  - **loss, cost** - this should be minimized
- We can always transform  $\text{score} \longleftrightarrow \text{loss}$ , using:

$$\text{loss}(z) = -\text{score}(z), \dots$$

$$\text{loss}(z) = \frac{1}{\text{score}(z)} \text{ for } \text{score}(z) > 0$$

...

## Function class. Linear example.

- **Function class** - parametrized set of functions  $F = \{f_\theta, \theta \in \Theta\}$ , from which the true relationship  $\mathcal{X} \rightarrow \mathcal{Y}$  is approximated.

## Function class. Linear example.

- **Function class** - parametrized set of functions  $F = \{f_\theta, \theta \in \Theta\}$ , from which the true relationship  $\mathcal{X} \rightarrow \mathcal{Y}$  is approximated.
- Examples of linear class functions:
  - regression:

$$f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \dots + \theta_D x^D$$

## Function class. Linear example.

- **Function class** - parametrized set of functions  $F = \{f_\theta, \theta \in \Theta\}$ , from which the true relationship  $\mathcal{X} \rightarrow \mathcal{Y}$  is approximated.
- Examples of linear class functions:
  - regression:

$$f(x) = \theta_0 + \theta_1 x^1 + \theta_2 x^2 + \dots + \theta_D x^D$$

- binary classification  $y \in \{+1, -1\}$ :

$$f(x) = \text{sign}\{\theta_0 + \theta_1 x^1 + \theta_2 x^2 + \dots + \theta_D x^D\},$$

## Function class. K-NN example.

- denote for each  $x$ :
  - $i(x, k)$  indexes  $k$ -th most close object to  $x$  in the feature space
- regression:

$$f(x) = \frac{1}{K} (y_{i(x,1)} + \dots + y_{i(x,K)})$$

- classification:

$$f(x) = \operatorname{argmax} \left\{ \sum_{i \in I(x,K)} \mathbb{I}[y_i = 1], \sum_{i \in I(x,K)} \mathbb{I}[y_i = 2], \dots, \sum_{i \in I(x,K)} \mathbb{I}[y_i = \dots] \right\}$$



## Empirical risk

- **Machine learning algorithm** associates  $f_{\hat{\theta}}(\cdot)$  to  $(X, Y)$ 
  - in the function class  $F = \{f_{\theta}, \theta \in \Theta\}$
  - for given loss function  $\mathcal{L}(\hat{y}, y)$
- **Empirical risk:**

$$L(\theta|X, Y) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\theta}(x_n), y_n)$$

- **Method of empirical risk minimization:**

$$\hat{\theta} = \arg \min_{\theta} L(\theta|X, Y)$$

## Estimation of empirical risk

- Generally it holds that:

$$L(\hat{\theta}|X, Y) < L(\hat{\theta}|X', Y')$$

where  $X, Y$  is the training sample and  $X', Y'$  is the new data.

- $L(\hat{\theta}|X', Y')$  can be estimated using :
  - separate **validation set**
  - **cross-validation**
  - **leave-one-out** method

## 4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Divide training set into  $K$  parts, referred as «folds» (here  $K = 4$ ).

Variants:

- randomly
- randomly with stratification (w.r.t target value or feature value).

## 4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,2,3 for model estimation and fold 4 for model evaluation.

## 4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,2,4 for model estimation and fold 3 for model evaluation.

## 4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 1,3,4 for model estimation and fold 2 for model evaluation.

## 4-fold cross validation example

X	Y
1	1
2	2
3	3
4	4

Use folds 2,3,4 for model estimation and fold 1 for model evaluation.

## 4-fold cross validation example

- Denote
  - $k(n)$  - fold to which observation  $(x_n, y_n)$  belongs to:  $n \in I_k$ .
  - $\hat{\theta}^{-k}$  - parameter estimation using observations from all folds except fold  $k$ .



## 4-fold cross validation example

- Denote
  - $k(n)$  - fold to which observation  $(x_n, y_n)$  belongs to:  $n \in I_k$ .
  - $\hat{\theta}^{-k}$  - parameter estimation using observations from all folds except fold  $k$ .

### Cross-validation empirical risk estimation

$$\hat{L}_{total} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\hat{\theta}^{-k(n)}}(x_n), y_n)$$

## 4-fold cross validation example

- Denote
  - $k(n)$  - fold to which observation  $(x_n, y_n)$  belongs to:  $n \in I_k$ .
  - $\hat{\theta}^{-k}$  - parameter estimation using observations from all folds except fold  $k$ .

### Cross-validation empirical risk estimation

$$\hat{L}_{total} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\hat{\theta}^{-k(n)}}(x_n), y_n)$$

- For  $K$ -fold CV we have:
  - $K$  parameters  $\hat{\theta}^{-1}, \dots, \hat{\theta}^{-K}$
  - $K$  models  $f_{\hat{\theta}^{-1}}(x), \dots, f_{\hat{\theta}^{-K}}(x)$ .
  - $K$  estimations of empirical risk:
 
$$\hat{L}_k = \frac{1}{|I_k|} \sum_{n \in I_k} \mathcal{L}(f_{\hat{\theta}^{-k}}(x_n), y_n), \quad k = 1, 2, \dots, K.$$

## 4-fold cross validation example

- Denote
  - $k(n)$  - fold to which observation  $(x_n, y_n)$  belongs to:  $n \in I_k$ .
  - $\hat{\theta}^{-k}$  - parameter estimation using observations from all folds except fold  $k$ .

### Cross-validation empirical risk estimation

$$\hat{L}_{total} = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_{\hat{\theta}^{-k(n)}}(x_n), y_n)$$

- For  $K$ -fold CV we have:
  - $K$  parameters  $\hat{\theta}^{-1}, \dots, \hat{\theta}^{-K}$
  - $K$  models  $f_{\hat{\theta}^{-1}}(x), \dots, f_{\hat{\theta}^{-K}}(x)$ .
  - $K$  estimations of empirical risk:
 
$$\hat{L}_k = \frac{1}{|I_k|} \sum_{n \in I_k} \mathcal{L}(f_{\hat{\theta}^{-k}}(x_n), y_n), k = 1, 2, \dots, K.$$
- Using  $\hat{L}_1, \dots, \hat{L}_K$  we can estimate variance & use statistics!

## Comments on cross-validation

- When number of folds  $K$  is equal to number of objects  $N$ , this is called **leave-one-out method**.
- Cross-validation uses the i.i.d.<sup>1</sup> property of observations
- Stratification by target helps for imbalanced/rare classes.

---

<sup>1</sup>i.i.d.=independent and identically distributed

## Cross-validation vs. A/B testing

- A/B testing:
  - ① divide objects randomly into two groups - A and B.
  - ② apply model 1 to A
  - ③ apply model 2 to B
  - ④ compare final results

Comparison of cross-validation and A/B test:

cross-validation	A/B test
evaluates forecasting quality	evaluates final business quality <sup>2</sup> (may evaluate forecasting quality as well)
uses available data, only computational costs	requires time and resources for collecting & evaluating feedback from objects of groups A and B

<sup>2</sup>final business quality may be high even when forecasting quality is low.

## Hyperparameters selection

- Suppose we want to select hyperparameters of the model:
  - regression: # of features  $d$ , e.g.  $x, x^2, \dots x^d$
  - K-NN: number of neighbors  $K$
- *What are gotchas when using CV?*

## Hyperparameters selection

- Suppose we want to select hyperparameters of the model:
  - regression: # of features  $d$ , e.g.  $x, x^2, \dots, x^d$
  - K-NN: number of neighbors  $K$
- *What are gotchas when using CV?*
- To assess method with selected hyperparameters we need separate test set.

## Levels of fitting

### Underfitted model

Model that oversimplifies true relationship  $\mathcal{X} \rightarrow \mathcal{Y}$ .

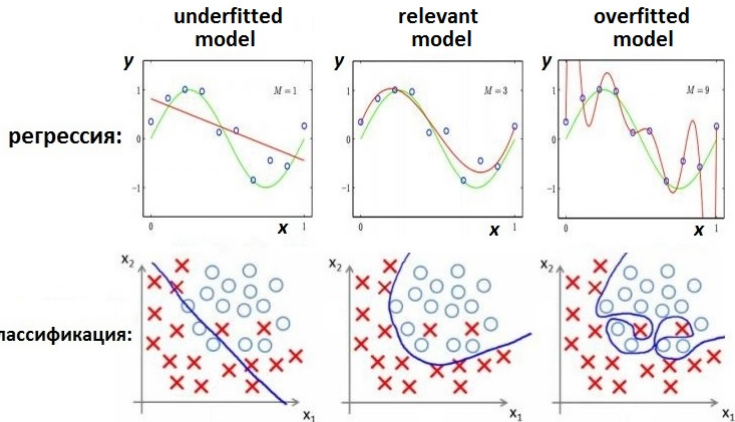
### Overfitted model

Model that is too tuned on particular peculiarities (noise) of the training set instead of the true relationship  $\mathcal{X} \rightarrow \mathcal{Y}$ .

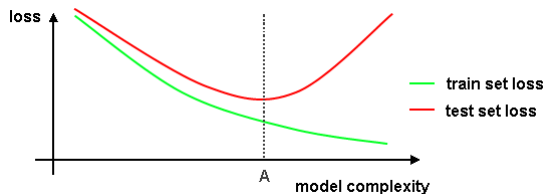


# Examples of overfitted/underfitted models

- true relationship
- estimated relationship with polynimes of order  $M$
- objects of the training sample



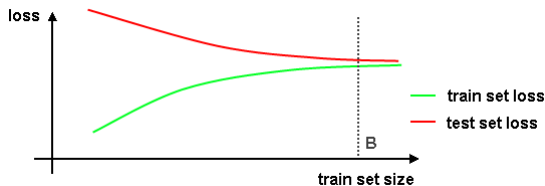
## Loss vs. model complexity



### Comments:

- expected loss on test set is always higher than on train set.
- left to A: model too simple, underfitting, high bias
- right to A: model too complex, overfitting, high variance

## Loss vs. model complexity



### Comments:

- expected loss on test set is always higher than on train set.
- right to B there is no need to further increase training set size

# Table of Contents

- 1 Tasks solved by machine learning
- 2 Main concepts of machine learning.
- 3 Practical applications of machine learning**

## Examples of ML applications

### Classification:

- spam filtering
- search engine: do query and document match each other?
- is series of network transactions regular or a hacking attempt?
- will the client with given characteristics switch his mobile operator?
- will given client of a bank return his debt?
- does the signal correspond to the target or noise in radar detection?

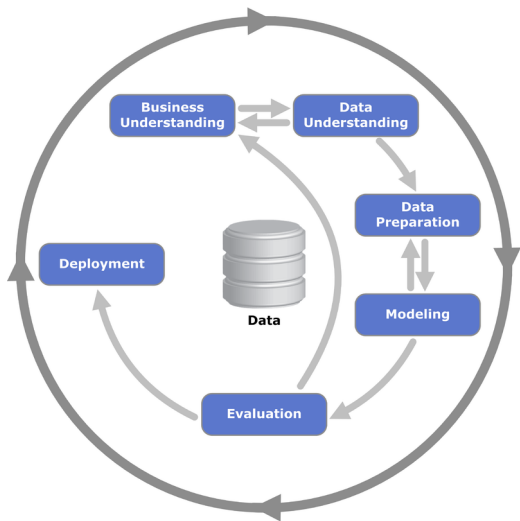
### Labelling:

- assignment of topics to text documents

### Regression:

- determine the flat price by its characteristics
- predict demand for certain product

# CrispDM methodology



## CrispDM general comments

- Log each step
  - quantitative: procedures and results in report.
  - qualitative: explain why certain option was taken and alternative options ignored.

## CrispDM - Business understanding

- Understand business goals and constraints
- State business objective in business terms
- State relevant data mining objective in technical terms
- State success criteria
- Produce plan of project



# CrispDM - Data understanding

- Collect data
- Understand data
  - qualitative meaning (what and how was measured)
  - quantitative distribution (data type, range, variance, skewness)
- Explore data
  - basic dependencies
  - interesting subsets
  - statistical analysis
- Quality check
  - outliers
  - missing data
  - errors in measurements

## CrispDM - Data preparation

usually takes most of the time

- Select data (select datasets, records, attributes)
- Clean data
  - missing values
  - outliers
  - erroneous values
  - inconsistent groups of attributes
- Construct data
  - derive attributes (normalization, aggregation, composition)
  - use background knowledge
  - fill missing values
- Integrate data together into connected structures (e.g. joined tables)
- Format data (uppercase/lowercase, encoding, etc.)

# CrispDM - Modeling

- Select relevant models
  - depending on data mining objective
  - depending on data properties (possibly need to return to data preparation)
- Divide dataset into training/validation/test sets
- Build models
  - choose initial values for model parameters
  - choose parameter estimation techniques
  - estimate parameters
  - post-process results using domain knowledge

## CrispDM - Evaluation

- evaluate model output quality using technical data mining criteria
  - compare to baseline
  - reliability of results (statistical significance, dependence on specific data assumptions)
  - check for systematic errors and interpret them (may be caused by missed factors/constraints)
- evaluate resulting models (interpretability, efficiency, scalability)
- analyze final business effect

## CrispDM - Deployment

- plan deployment
- plan monitoring and maintenance
- produce final report
- review project experience
  - from project team
  - from customers

## Notation used in the course

- If this corresponds the context and there are no redefinitions, then:
  - $x$  - vector of known input characteristics of an object
  - $y$  - predicted target characteristics of an object specified by  $x$
  - $x_i$  -  $i$ -th object of a set,  $y_i$  - corresponding target characteristic
  - $x^k$  -  $k$ -th feature of object specified by  $x$
  - $x_i^k$  -  $k$ -th feature of object specified by  $x_i$
  - $D$  - dimensionality of the feature space:  $x \in \mathbb{R}^D$
  - $N$  - the number of objects in the training set
  - $X$  - design matrix,  $X \in \mathbb{R}^{N \times D}$
  - $Y \in \mathbb{R}^N$  - target characteristics of a training set
  - $\mathcal{L}(\hat{y}, y)$  - loss function, where  $y$  is the true value and  $\hat{y}$  is the predicted value.
  - $\{\omega_1, \omega_2, \dots, \omega_C\}$  - possible classes,  $C$  - total number of classes.
  - $\hat{z}$  defines an estimate of  $z$ , based on the training set: for example,  $\hat{\theta}$  is the estimate of  $\theta$ ,  $\hat{y}$  is the estimate of  $y$ , etc.