

IB031 Úvod do strojového učení

Tomáš Brázdil

Course Info

Resources:

- ▶ Lectures & tutorials (the **main** source)
- ▶ Many books, few perfect for introductory level
One relatively good, especially the first part:
A. Géron. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media; 3rd edition, 2022
- ▶ (Almost) infinitely many online courses, tutorials, materials, etc.

Evaluation

The evaluation is composed of three parts:

- ▶ Mid-term exam: Written exam from the material of the first half of the semester.
- ▶ End-term exam: The "big" one containing everything from the semester (with possibly more stress in the second half).
- ▶ Projects: During tutorials, you will work on larger projects (in pairs or triples).

Each part contributes the following number of points:

- ▶ Mid-term exam: 25
- ▶ End-term exam: 50
- ▶ Project: 25

To pass, you need to obtain at least 60 points.

Distinguishing Properties of the Course

- ▶ Introductory, prerequisites are held to a minimum
- ▶ Formal and precise: Be prepared for a complete and “mathematical” description of presented methods.

Distinguishing Properties of the Course

- ▶ Introductory, prerequisites are held to a minimum
- ▶ Formal and precise: Be prepared for a complete and “mathematical” description of presented methods.

I assume that you have basic knowledge of

- ▶ Elementary mathematical notation (operations on sets, logic, etc.)
- ▶ Linear algebra: Vectors in \mathbb{R}^n , operations on vectors (including the dot product). Geometric interpretation!
- ▶ Calculus: Functions of multiple real variables, partial derivatives, basic differential calculus.
- ▶ Probability: Notion of probability distribution, random variables/vectors, expectation.

What Is Machine Learning?

Machine learning is the science (and art) of programming computers so they can **learn from data**.

What Is Machine Learning?

Machine learning is the science (and art) of programming computers so they can **learn from data**.

Here is a slightly more general definition:

Arthur Samuel, 1959

Machine learning is the field of study that allows computers to learn without being explicitly programmed.

What Is Machine Learning?

Machine learning is the science (and art) of programming computers so they can **learn from data**.

Here is a slightly more general definition:

Arthur Samuel, 1959

Machine learning is the field of study that allows computers to learn without being explicitly programmed.

And a more engineering-oriented one:

Tom Mitchell, 1997

A computer program is said to learn from experience E concerning some task T and some performance measure P if its performance on T , as measured by P , improves with experience E .

Example

In the context of spam filtering:

- ▶ The task T is to flag spam in new emails.
- ▶ The experience E is represented by a set of emails labeled either spam or ham by hand (the training data).
- ▶ The performance measure P could be the accuracy, which is the ratio of the number of correctly classified emails and all emails.

There are many more performance measures; we will study the basic ones later.

Example

In the context of spam filtering:

- ▶ The task T is to flag spam in new emails.
- ▶ The experience E is represented by a set of emails labeled either spam or ham by hand (the training data).
- ▶ The performance measure P could be the accuracy, which is the ratio of the number of correctly classified emails and all emails.

There are many more performance measures; we will study the basic ones later.

In the context of housing price prediction:

- ▶ The task T is to predict prices of new houses based on their basic parameters (size, number of bathrooms, etc.)
- ▶ The experience E is represented by information about existing houses.
- ▶ The performance measure P could be, e.g., an absolute difference between the predicted and the real price.

Examples (cont.)

In the context of game playing:

- ▶ The task T is to play chess.
- ▶ The experience E is represented by a series of self-plays where the computer plays against itself.
- ▶ The performance measure P is winning/losing the game.

Here, the trick is to spread the delayed and limited feedback about the result of the game throughout the individual decisions in the game.

Examples (cont.)

In the context of game playing:

- ▶ The task T is to play chess.
- ▶ The experience E is represented by a series of self-plays where the computer plays against itself.
- ▶ The performance measure P is winning/losing the game.
Here, the trick is to spread the delayed and limited feedback about the result of the game throughout the individual decisions in the game.

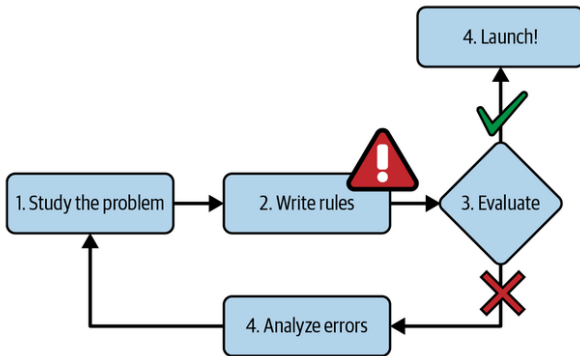
In the context of customer behavior:

- ▶ The task T is to group customers with similar shopping habits in an e-shop.
- ▶ The experience E consists of lists of items individual customers bought in the shop.
- ▶ The performance measure P ?
Measure how "nicely" the customers are grouped.
(whether people with similar habits, as seen by humans, fall into the same group).

Comparison of Programming and Learning

How to code the spam filter?

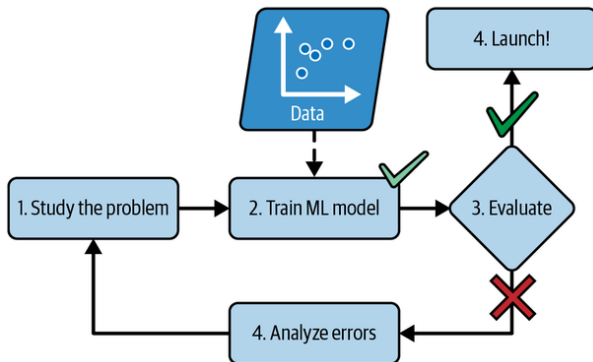
- ▶ Examine what spam mails typically contain: Specific words ("Viagra"), sender's address, etc.
- ▶ Write down a rule-based system that detects specific features.
- ▶ Test the program on new emails and (most probably) go back to look for more spam features.



Comparison of Programming and Learning

The machine learning way:

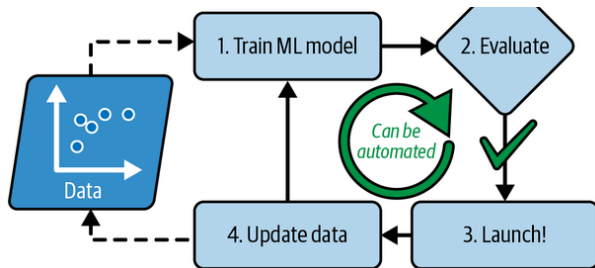
- ▶ Study the problem and collect lots of emails, labeling them spam or ham.
- ▶ Train a machine learning model that reads an email and decides whether it's spam or ham.
- ▶ Test the model and (most probably) go back to collect more data and adjust the model.



ML Solutions are Adaptive

Spam filter: Authors of spam might and will adapt to your spam filter (possibly change the wording to pass through).

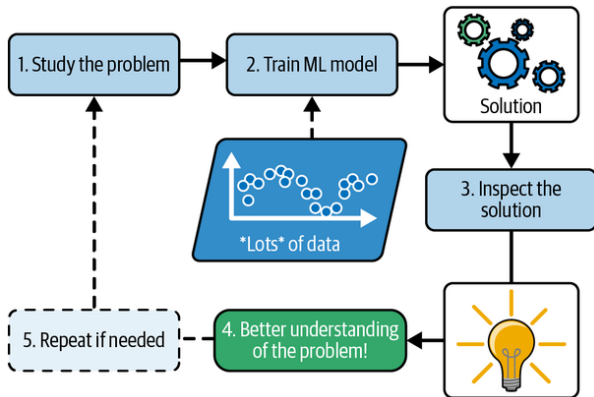
ML systems can be adjusted to new situations by retraining on new data (unless the data becomes ugly).



ML for Human Understanding

Spam filter: A trained system can be inspected for notorious spam features.

Some models allow direct inspection, such as decision trees or linear/logistic regression models.



Usage of Machine Learning

Machine learning suits various applications, especially where traditional methods fall short. Here are some areas where it excels:

- ▶ Solving complex problems where fine-tuning and rule-based solutions are inadequate.
- ▶ Tackling complex issues that resist traditional problem-solving approaches.
- ▶ Adapting to fluctuating environments through retraining on new data.
- ▶ Gaining insights from large and complex datasets.

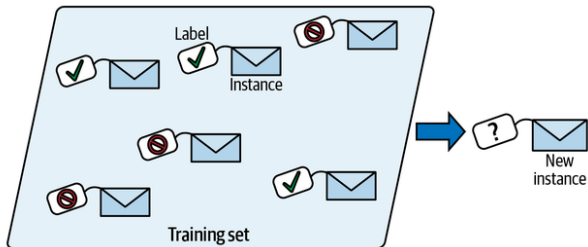
In summary, machine learning offers innovative solutions and adaptability for today's complex and ever-changing problems, (sometimes) providing insights beyond the reach of traditional approaches.

Types of Learning

There are main categories based on information available during the training:

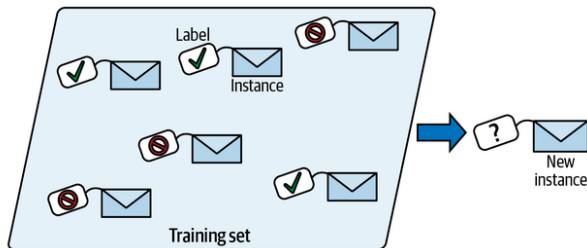
- ▶ Supervised learning
- ▶ Unsupervised learning
- ▶ Semi-supervised learning
- ▶ Self-supervised learning
- ▶ Reinforcement learning

Supervised Learning



Labels are available for all input data.

Supervised Learning

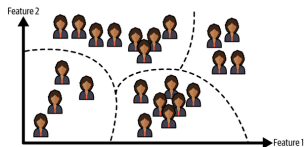
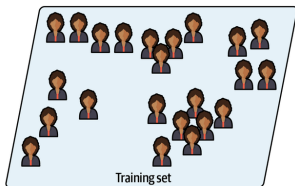


Labels are available for all input data.

Typical supervised learning tasks are

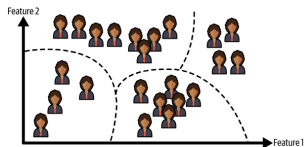
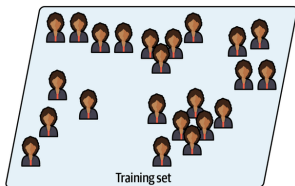
- ▶ *Classification* where the aim is to classify inputs into (typically few) classes
(e.g., the spam filter where the classes are spam/ham)
- ▶ *Regression* where a numerical value is output for a given input
(e.g., housing prices)

Unsupervised Learning



No labels are available for input data.

Unsupervised Learning

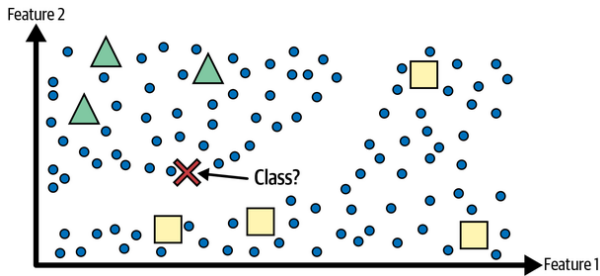


No labels are available for input data.

Typical unsupervised learning tasks are

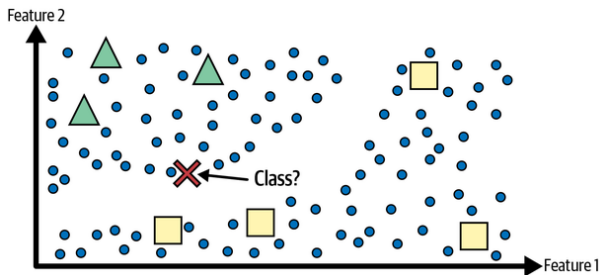
- ▶ *Clustering* where inputs are grouped according to their features
(e.g., clients of a bank grouped according to their age, wealth, etc.)
- ▶ *Association* where interesting relations and rules are discovered among the features of inputs
(e.g., market basket mining where associations between various types of goods are being learned from the behavior of customers)
- ▶ *Dimensionality reduction* reduce high-dimensional data to few dimensions (e.g., images to few image features)

Semi-Supervised Learning



Labels for some data.

Semi-Supervised Learning

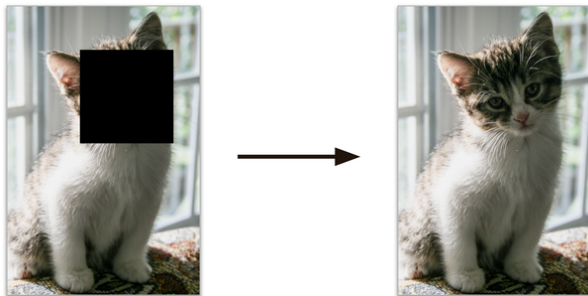


Labels for some data.

For example, Medical data, where elaborate diagnosis is available only for some patients.

Combines supervised and unsupervised learning: e.g., clusters all data and labels the unlabeled inputs with the most common labels in their clusters.

Self-Supervised Learning

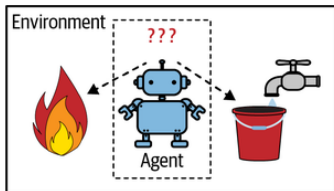


Generate labels from (unlabeled) inputs.

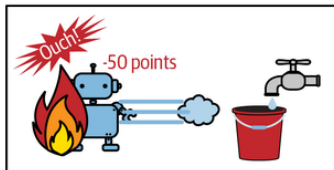
The goal is to learn typical features of the data.

It can be later modified to generate images, classify, etc.

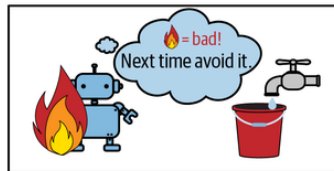
Reinforcement Learning



- 1 Observe
- 2 Select action using policy



- 3 Action!
- 4 Get reward or penalty



- 5 Update policy (learning step)
- 6 Iterate until an optimal policy is found

Learn from performing *actions* and getting feedback from *environment*.

ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)

ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)
- ▶ Machine translation, image captioning
 - ▶ Google translate, etc.
 - ▶ Typically (semi)-supervised learning,

ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)
- ▶ Machine translation, image captioning
 - ▶ Google translate, etc.
 - ▶ Typically (semi)-supervised learning,
- ▶ Various image recognition and processing tasks
 - ▶ In medicine, such systems are slowly making their way into hospitals as assistance tools
 - ▶ Automotive, quality control, advertising, etc., etc., etc.

ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)
- ▶ Machine translation, image captioning
 - ▶ Google translate, etc.
 - ▶ Typically (semi)-supervised learning,
- ▶ Various image recognition and processing tasks
 - ▶ In medicine, such systems are slowly making their way into hospitals as assistance tools
 - ▶ Automotive, quality control, advertising, etc., etc., etc.
- ▶ Science
 - ▶ Chemistry & biology: E.g., prediction of features of chemical compounds (Alpha-fold)

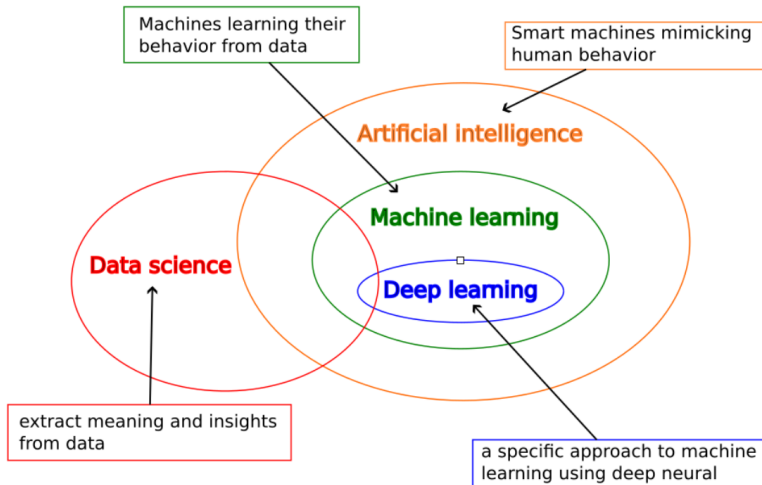
ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)
- ▶ Machine translation, image captioning
 - ▶ Google translate, etc.
 - ▶ Typically (semi)-supervised learning,
- ▶ Various image recognition and processing tasks
 - ▶ In medicine, such systems are slowly making their way into hospitals as assistance tools
 - ▶ Automotive, quality control, advertising, etc., etc., etc.
- ▶ Science
 - ▶ Chemistry & biology: E.g., prediction of features of chemical compounds (Alpha-fold)
- ▶ Various "table" data processing in finance, management, etc.
 - ▶ Often straightforward methods (linear/logistic regression)
 - ▶ Essential but not fancy

ML Applications Highlights

- ▶ ChatGPT (and similar generative models)
 - ▶ The basis forms a generative language model, i.e., a text-generating model trained on texts in a self-supervised way
 - ▶ Recently extended to multimodal versions (text, image, sound)
- ▶ Machine translation, image captioning
 - ▶ Google translate, etc.
 - ▶ Typically (semi)-supervised learning,
- ▶ Various image recognition and processing tasks
 - ▶ In medicine, such systems are slowly making their way into hospitals as assistance tools
 - ▶ Automotive, quality control, advertising, etc., etc., etc.
- ▶ Science
 - ▶ Chemistry & biology: E.g., prediction of features of chemical compounds (Alpha-fold)
- ▶ Various "table" data processing in finance, management, etc.
 - ▶ Often straightforward methods (linear/logistic regression)
 - ▶ Essential but not fancy
- ▶ Game playing: More fancy than useful, learning models beating humans in several difficult games.

ML in Context



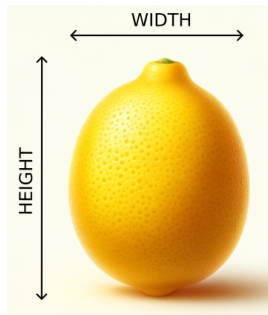
Supervised Learning

Example - Fruit Recognition

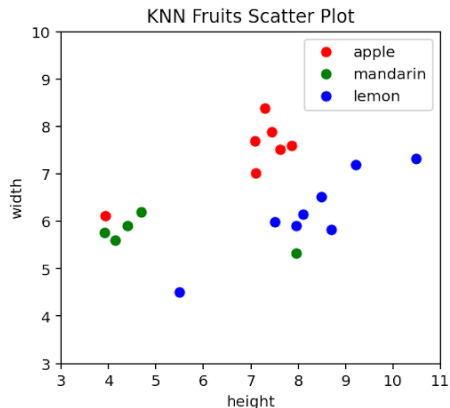
The goal: Create an automatic system for fruit recognition, concretely apple, lemon, and mandarin.

Inputs: Measures of *height* and *width* of each fruit.

Suppose we have a dataset of dimensions of several fruits labeled with the correct class.



Data



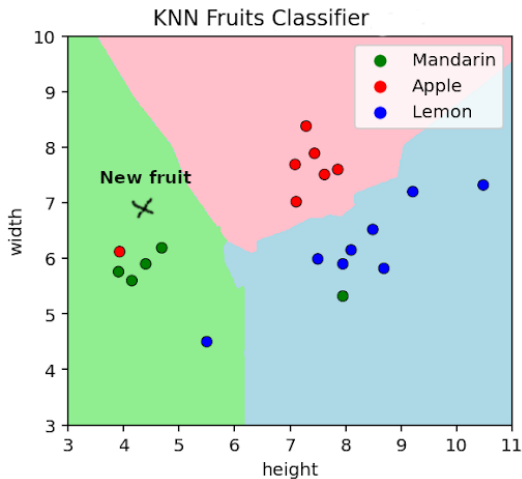
Use similarity to solve the problem.

	height	width	fruit
0	3.91	5.76	Mandarin
1	7.09	7.69	Apple
2	10.48	7.32	Lemon
3	9.21	7.20	Lemon
4	7.95	5.90	Lemon
5	7.62	7.51	Apple
6	7.95	5.32	Mandarin
7	4.69	6.19	Mandarin
8	7.50	5.99	Lemon
9	7.11	7.02	Apple
10	4.15	5.60	Mandarin
11	7.29	8.38	Apple
12	8.49	6.52	Lemon
13	7.44	7.89	Apple
14	7.86	7.60	Apple
15	3.93	6.12	Apple
16	4.40	5.90	Mandarin
17	5.50	4.50	Lemon
18	8.10	6.15	Lemon
19	8.69	5.82	Lemon

KNN Classification

Given a new fruit.
What is it?

Find five closest
examples



Where is the machine learning?

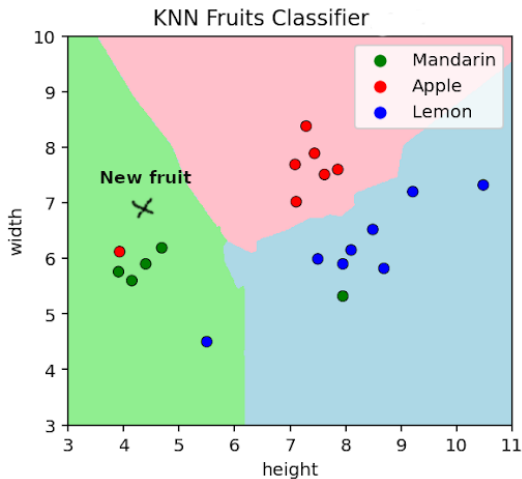
KNN Classification

Given a new fruit.
What is it?

Find five closest
examples

Among the five closest:

- ▶ $M = 4$ mandarins
- ▶ $A = 1$ apples
- ▶ $L = 0$ lemons



Where is the machine learning?

KNN Classification

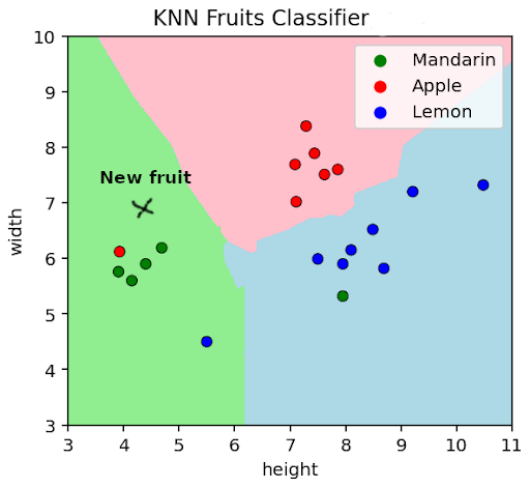
Given a new fruit.
What is it?

Find five closest
examples

Among the five closest:

- ▶ $M = 4$ mandarins
- ▶ $A = 1$ apples
- ▶ $L = 0$ lemons

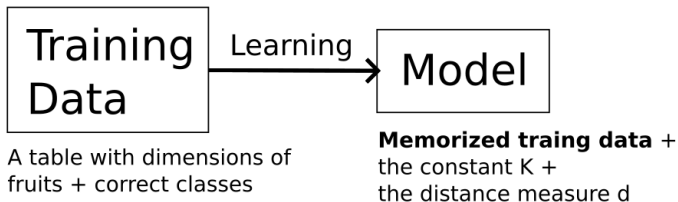
It is a **mandarin**!



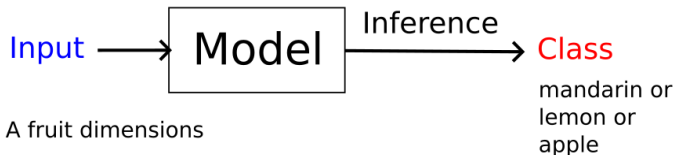
Where is the machine learning?

Learning in Fruit Classification with KNN

Learning:



Inference:



Fruit Classification Algorithm

Input: A fruit F with dimensions *height*, *width*

Output: *mandarin*, *lemon*, *apple*

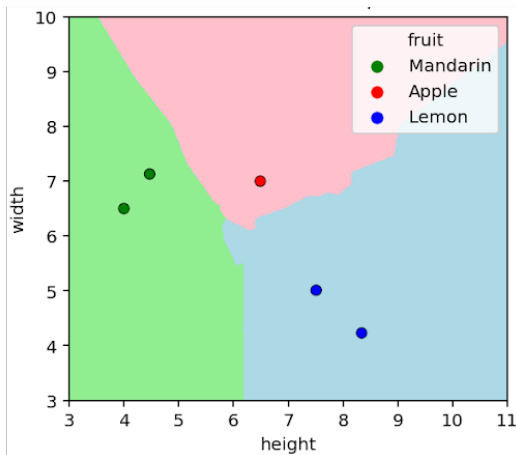
- 1: Find K examples $\{E_1, \dots, E_K\}$ in the dataset whose dimensions are closest to the dimensions of the fruit F
- 2: Count the number of examples of each class in $\{E_1, \dots, E_K\}$
 - M mandarins in $\{E_1, \dots, E_K\}$
 - L lemons in $\{E_1, \dots, E_K\}$
 - A apples in $\{E_1, \dots, E_K\}$
- 3: **if** $M \geq L$ and $M \geq A$ **then return** *mandarin*
- 4: **else if** $L \geq A$ **then return** *lemon*
- 5: **else return** *apple*
- 6: **end if**

Does it work?

Testing the Model for Fruit Classification

Consider a test set of new instances ($K = 5$, d is Euclidean):

height	width	fruit
4.0	6.5	Mandarin
4.47	7.13	Mandarin
6.49	7.0	Apple
7.51	5.01	Lemon
8.34	4.23	Lemon



Perfect classification of new data! Just deploy and sell!!

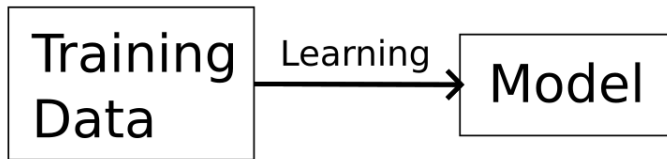
K Nearest Neighbors

... on ideal data

Learning and Inference

Two crucial components of machine learning are the following:

Learning:
Creating model



Inference:
Using model



Training Data

Assume table training data, i.e., of the form

x_{11}	x_{12}	\cdots	x_{1n}	c_1
x_{21}	x_{22}	\cdots	x_{2n}	c_2
\vdots	\vdots	\ddots	\vdots	\vdots
x_{p1}	x_{p2}	\cdots	x_{pn}	c_p

height	width	fruit
4.0	6.5	Mandarin
4.47	7.13	Mandarin
6.49	7.0	Apple
7.51	5.01	Lemon
8.34	4.23	Lemon

Formally, we define **training dataset**

$$\mathcal{T} = \{(\vec{x}_k, c_k) \mid k = 1, \dots, p\}$$

Here each $\vec{x}_k \in \mathbb{R}^n$ is an input vector and $c_k \in C$ is the correct class.

$$\mathcal{T} = \{(4.0, 6.5), M), \\ (4.47, 7.13), M), \\ (6.49, 7.0), A), \\ \dots\}$$

KNN: Learning

Consider the training set:

$$\mathcal{T} = \{(\vec{x}_k, c_k) \mid k = 1, \dots, p\}$$

and *memorize it* exactly as it is.

Store in a table.

Possibly use a clever representation allowing fast computation of nearest neighbors such as KDTrees (out of the scope of this lecture).

Also,

- ▶ determine the number of neighbors $K \in \mathbb{N}$,
- ▶ and the distance measure d .

Inference in KNN

Assume a KNN "trained" by memorizing

$\mathcal{T} = \{(\vec{x}_k, c_k) \in \mathbb{R}^n \times C \mid k = 1, \dots, p\}$, a constant $K \in \mathbb{N}$ and a distance measure d .

For d , consider Euclidean distance, but different norms may also be used to define different distance measures.

Inference in KNN

Assume a KNN "trained" by memorizing

$\mathcal{T} = \{(\vec{x}_k, c_k) \in \mathbb{R}^n \times C \mid k = 1, \dots, p\}$, a constant $K \in \mathbb{N}$ and a distance measure d .

For d , consider Euclidean distance, but different norms may also be used to define different distance measures.

Input: A vector $\vec{z} = (z_1, \dots, z_n) \in \mathbb{R}^n$

Output: A class from C

- 1: Find K indices of examples $X = \{i_1, \dots, i_K\} \subseteq \{1, \dots, p\}$ with minimum distance to \vec{z} , i.e., satisfying

$$\max\{d(\vec{z}, \vec{x}_\ell) \mid \ell \in X\} \leq \min\{d(\vec{z}, \vec{x}_\ell) \mid \ell \in \{1, \dots, p\} \setminus X\}$$

- 2: For every $c \in C$ count the number $\#c$ of elements ℓ in X such that $c_\ell = c$
- 3: Return some

$$c_{\max} \in \arg \max_{c \in C} \#c$$

A class $c_{\max} \in C$ which maximizes $\#c$.

The resulting model

What exactly constitutes the model? The *model* consists of

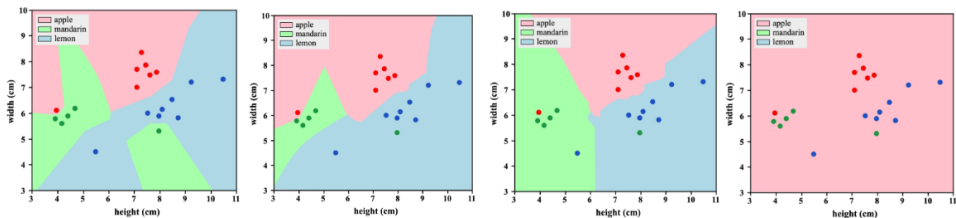
- ▶ The *trained parameters*: In this case the memorized training data.
- ▶ The *hyperparameters* set “from the outside”: In this case, the number of neighbors K and the distance measure d .

The resulting model

What exactly constitutes the model? The *model* consists of

- ▶ The *trained parameters*: In this case the memorized training data.
- ▶ The *hyperparameters* set “from the outside”: In this case, the number of neighbors K and the distance measure d .

Note that different settings of K lead to different classifiers (for the same d):



In Practice

...to get an efficient solution:

In Practice

...to get an efficient solution:

- ▶ Deal with issues in the data
 - ▶ Data almost always comes in weird formats, with inconsistencies, missing values, wrong values, etc.
 - ▶ Data rarely have the ideal form for a given learning model.

We need to ingest, validate, and preprocess the data.

In Practice

... to get an efficient solution:

- ▶ Deal with issues in the data
 - ▶ Data almost always comes in weird formats, with inconsistencies, missing values, wrong values, etc.
 - ▶ Data rarely have the ideal form for a given learning model.

We need to ingest, validate, and preprocess the data.

- ▶ Deal with issues in the model
 - ▶ In KNN, the training memorizes the example, but at least the K can be tuned.

We need to tune the model.

In Practice

... to get an efficient solution:

- ▶ Deal with issues in the data
 - ▶ Data almost always comes in weird formats, with inconsistencies, missing values, wrong values, etc.
 - ▶ Data rarely have the ideal form for a given learning model.

We need to ingest, validate, and preprocess the data.

- ▶ Deal with issues in the model
 - ▶ In KNN, the training memorizes the example, but at least the K can be tuned.

We need to tune the model.

- ▶ Deal with the wrong model by testing and validation in as realistic conditions as possible.

In Practice

... to get an efficient solution:

- ▶ Deal with issues in the data

- ▶ Data almost always comes in weird formats, with inconsistencies, missing values, wrong values, etc.
- ▶ Data rarely have the ideal form for a given learning model.

We need to ingest, validate, and preprocess the data.

- ▶ Deal with issues in the model

- ▶ In KNN, the training memorizes the example, but at least the K can be tuned.

We need to tune the model.

- ▶ Deal with the wrong model by testing and validation in as realistic conditions as possible.
- ▶ Deal with deployment - real-world application issues involving, e.g., implementation in embedded devices with limited resources.

Models Considered in This Course

Throughout this course, we will meet the following models:

- ▶ KNN (already did)
- ▶ Decision trees
- ▶ (Naive) Bayes classifier
- ▶ Clustering: K-means and hierarchical
- ▶ Linear and logistic regression
- ▶ Support Vector Machines (SVM)
- ▶ Kernel linear models
- ▶ Neural networks (light intro to feed-forward networks)
- ▶ Ensemble methods + random forests
- ▶ (maybe some reinforcement learning)

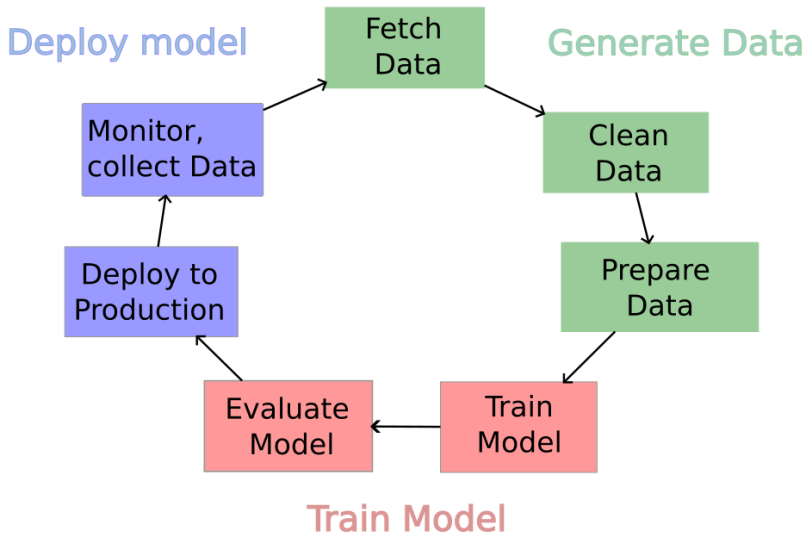
Models Considered in This Course

Throughout this course, we will meet the following models:

- ▶ KNN (already did)
- ▶ Decision trees
- ▶ (Naive) Bayes classifier
- ▶ Clustering: K-means and hierarchical
- ▶ Linear and logistic regression
- ▶ Support Vector Machines (SVM)
- ▶ Kernel linear models
- ▶ Neural networks (light intro to feed-forward networks)
- ▶ Ensemble methods + random forests
- ▶ (maybe some reinforcement learning)

...but first, let us see the whole machine-learning pipeline.

Machine Learning Pipeline



Fetch Data

Always start with

Fetch Data

Always start with

- ▶ The problem formulation & understanding.

For example, diagnosis of diabetes from medical records. What info is (possibly) sufficient for such a diagnosis?

Fetch Data

Always start with

- ▶ The problem formulation & understanding.

For example, diagnosis of diabetes from medical records. What info is (possibly) sufficient for such a diagnosis?

- ▶ Find data sources.

In our example, the sources are hospitals. It would be best to persuade them to give you the data and sign a contract.

Fetch Data

Always start with

- ▶ The problem formulation & understanding.

For example, diagnosis of diabetes from medical records. What info is (possibly) sufficient for such a diagnosis?

- ▶ Find data sources.

In our example, the sources are hospitals. It would be best to persuade them to give you the data and sign a contract.

- ▶ Collect the data.

In our example, the data is possibly small (just tables with results of tests). But for other diagnoses, you may include huge amounts of data from MRI, CT, etc. Then, the collection itself might be a serious technical problem.

Fetch Data

Always start with

- ▶ The problem formulation & understanding.

For example, diagnosis of diabetes from medical records. What info is (possibly) sufficient for such a diagnosis?

- ▶ Find data sources.

In our example, the sources are hospitals. It would be best to persuade them to give you the data and sign a contract.

- ▶ Collect the data.

In our example, the data is possibly small (just tables with results of tests). But for other diagnoses, you may include huge amounts of data from MRI, CT, etc. Then, the collection itself might be a serious technical problem.

- ▶ Integrate data from various sources.

A serious diagnostic system must be trained/tested on data from many hospitals. You must blend the data from various sources (different formats, etc.).

Fetch Data

For simple “toy” machine learning projects, you may fetch prepared datasets from various databases on the internet.

Fetch Data

For simple “toy” machine learning projects, you may fetch prepared datasets from various databases on the internet.

The data should be stored in an identified location and versioned. You will probably keep adding data and training models on the ever-changing datasets. You have to be able to keep track of the changes and map training data to particular models.

Tools such as ML Flow or Weights & Biases might be helpful.

Fetch Data

For simple “toy” machine learning projects, you may fetch prepared datasets from various databases on the internet.

The data should be stored in an identified location and versioned. You will probably keep adding data and training models on the ever-changing datasets. You have to be able to keep track of the changes and map training data to particular models.

Tools such as ML Flow or Weights & Biases might be helpful.

Data Separation

At this point, you should randomize the ordering of the data and select a test set to be used in model evaluation!

The test data are supposed to simulate the actual conditions, i.e., they should be “unseen”.

Fetch Data

For simple “toy” machine learning projects, you may fetch prepared datasets from various databases on the internet.

The data should be stored in an identified location and versioned. You will probably keep adding data and training models on the ever-changing datasets. You have to be able to keep track of the changes and map training data to particular models.

Tools such as ML Flow or Weights & Biases might be helpful.

Data Separation

At this point, you should randomize the ordering of the data and select a test set to be used in model evaluation!

The test data are supposed to simulate the actual conditions, i.e., they should be “unseen”.

Data Exploration

Compute basic statistics to identify missing values, outliers, etc.

Clean Data

The cleaning usually comprises the following steps:

- ▶ Fix or remove incorrect or corrupted values.
- ▶ Identify outliers and decide what to do with them.
Outliers may harm some training methods and are not “representative”. However, sometimes, they naturally belong to the dataset, and expert insight is needed.
- ▶ Fix formatting.
For example, the Date may be expressed in many ways, and a simple Yes/No answer.
- ▶ Resolve missing values (by either removing the whole examples or imputing)
Many methods have been developed for missing values imputation. It is a susceptible issue because new values may strongly bias the model.
- ▶ Remove duplicates.

The above steps often affect the training and need expertise in the application domain.

Later in this course, we will discuss techniques for data cleaning.

ID	Age	Income	Gender	Customer_Satisfaction
1	38	46641.356413713	nan	Unsatisfied
2	42	49129.0615585107	female	Neutral
3	18	119965.049731014	Male	nan
4	18	66828.0762224329	nan	very unsatisfied
5	58	57422.2721106762	female	very unsatisfied
6	28	59502.8174855665	Other	Satisfied
7	18	42659.6675768587	Other	Neutral
8	18	54019.1173206374	Other	Satisfied
9	40	25429.1604541137	female	Unsatisfied
10	21	15595.5862129548	Other	Satisfied
11	18	58094.2328460069	Other	very unsatisfied
12	18	39097.3278583155	female	Very Satisfied
13	30		Other	Satisfied
14	50	30617.3914472273	Female	Very Satisfied
15	18		nan	Neutral
16	34	39902.4430953214	male	nan
17	49	68381.6997683133	Female	Very Satisfied
18	33	44796.0962271524	Other	Very Satisfied
19	47	39218.9560738814	Female	very unsatisfied
20		14544.9226784447	Other	Satisfied

Prepare Data

Unlike cleaning, which is application-dependent, data preparation/transformation is model-dependent. This usually subsumes:

- ▶ **Scaling:** Settings values of inputs to a similar range.

Some models, especially those utilizing distance, are sensitive to large differences between input sizes.

- ▶ **Encoding:** Encode non-numeric data using real-valued vectors.

Many models, especially those based on geometry, work only with numeric data. Non-numeric data such as Yes/No, Short/Medium/Long must be encoded appropriately.

- ▶ **Binning or Discretization** Convert continuous features into discrete bins to capture patterns in ranges.

Comment: Sometimes **Normalization**, that is changing the distribution of inputs to resemble the normal distribution, is mentioned. However, this step is typically not essential for machine learning itself. However, it is important to use statistical inference to test the significance of learned parameters.

Prepare Data

- ▶ **Feature selection** Throw out input features that are too “similar” to other features.

For example, if the temperature is measured both in Celsius and in Kelvin, keep one of them. The relationship can, of course, be a more complex (non-linear) correlation.

Prepare Data

- ▶ **Feature selection** Throw out input features that are too “similar” to other features.

For example, if the temperature is measured both in Celsius and in Kelvin, keep one of them. The relationship can, of course, be a more complex (non-linear) correlation.

- ▶ **Dimensionality reduction** Transforming data from \mathbb{R}^n to \mathbb{R}^m where $m \ll n$.

Growing dimension means growing difficulty of training for all models. Some models cease to work for high-dimensional data. The reduction typically searches for a few important characteristic features of inputs.

Prepare Data

- ▶ **Feature selection** Throw out input features that are too “similar” to other features.

For example, if the temperature is measured both in Celsius and in Kelvin, keep one of them. The relationship can, of course, be a more complex (non-linear) correlation.

- ▶ **Dimensionality reduction** Transforming data from \mathbb{R}^n to \mathbb{R}^m where $m \ll n$.

Growing dimension means growing difficulty of training for all models. Some models cease to work for high-dimensional data. The reduction typically searches for a few important characteristic features of inputs.

- ▶ **Feature aggregation** Introducing new features using operations on the original ones.

We will see kernel transformations later in this course, allowing simple models to solve complex problems.

Train Model

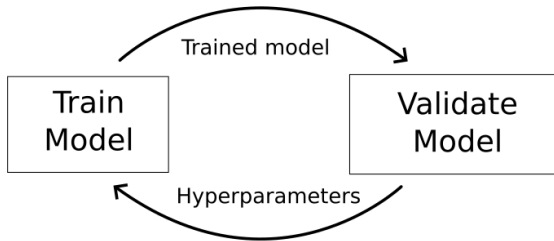
Now the dataset has been cleaned; we may train a model.

Train Model

Now the dataset has been cleaned; we may train a model.

Before training, we should split the dataset into

- ▶ *training* dataset on which the model will learn
- ▶ *validation* dataset on which we fine-tune hyperparameters



The resulting model is obtained after several iterations of the above process.

Evaluate Model

Here, we use the test set that we separated during data fetching.

In some cases, a brand new test set can be generated.

patients are examined regularly, creating new records continuously.

In some cases, it is tough to obtain new data.

For example, new expensive and difficult measurements are needed to obtain new data.

Evaluate Model

Here, we use the test set that we separated during data fetching.

In some cases, a brand new test set can be generated.

patients are examined regularly, creating new records continuously.

In some cases, it is tough to obtain new data.

For example, new expensive and difficult measurements are needed to obtain new data.

Critical issue: Make sure that you are truly testing
exactly the whole inference process.

Often, just a model is tested, and the testing and production inference engines are separated. This leads to truly nasty errors in the production!

We will discuss various generic metrics helpful in measuring the quality of the resulting model.

Deploy to Production

Deployment of machine learning models is a complex question, application dependent.

The recently emerging area of MLOps is concerned with the engineering side of the model deployment.

Deploy to Production

Deployment of machine learning models is a complex question, application dependent.

The recently emerging area of MLOps is concerned with the engineering side of the model deployment.

From the technical point of view, the typical issues solved by ML Ops teams are

- ▶ how to extract/process data in real-time
- ▶ how much storage is required
- ▶ how to store/collect model (and data) artifacts/predictions
- ▶ how to set up APIs, tools, and software environments
- ▶ What the period of predictions (instantaneous or batch predictions) should be
- ▶ how to set up hardware requirements (or cloud requirements for on-cloud environments) by the computational resources required
- ▶ how to set up a pipeline for continuous training and parameter tuning

Deploy to Production

From the user's point of view:

- ▶ How to get a sensible and valuable user output?
 - ▶ AI researchers will be satisfied with tons of running text in terminals.
 - ▶ “Normal” people need a graphical interface with understandable output.
 - ▶ Experts working in other domains typically demand speed and clarity at the extreme.

Deploy to Production

From the user's point of view:

- ▶ How to get a sensible and valuable user output?
 - ▶ AI researchers will be satisfied with tons of running text in terminals.
 - ▶ “Normal” people need a graphical interface with understandable output.
 - ▶ Experts working in other domains typically demand speed and clarity at the extreme.
- ▶ How do you persuade users that the AI is working for them?
 - ▶ Especially if safety is at stake, you need to have outstanding arguments and explanations ready for end-users.
 - ▶ In many areas, the devices need to be certified (medicine, automotive) for ML-based systems.

This complex subject will be only touched on in this course.

Monitor, collect Data

Deployed machine learning models must be constantly monitored.

Because of the influx of new data, ML models work in highly dynamic environments.

For example, an image-processing medical diagnostic model suddenly misdiagnosed a patient because a nurse marked the sample with a marker pen.

Every customer has a different infrastructure and may produce data slightly differently.

Data for retraining and improvement should be stored.

Also, many areas allow the *active learning* where users provide feedback for (continuous) retraining of the models.

Data

Data Science Example

You receive data from a medical researcher concerning a project that you are eager to work on.

Data Science Example

You receive data from a medical researcher concerning a project that you are eager to work on.

The data consists of a 1000 lines table with five columns:

012	232	33.5	0	10.7
020	121	16.9	2	210.1
027	165	24.0	0	427.6
...				

The aim is to predict the last field given the others.

Data Science Example

You receive data from a medical researcher concerning a project that you are eager to work on.

The data consists of a 1000 lines table with five columns:

012	232	33.5	0	10.7
020	121	16.9	2	210.1
027	165	24.0	0	427.6
...				

The aim is to predict the last field given the others.

The medical researcher does not elaborate further on the data, but they seem to be pretty easy to work with, right?

Data Science Example

You receive data from a medical researcher concerning a project that you are eager to work on.

The data consists of a 1000 lines table with five columns:

012	232	33.5	0	10.7
020	121	16.9	2	210.1
027	165	24.0	0	427.6
...				

The aim is to predict the last field given the others.

The medical researcher does not elaborate further on the data, but they seem to be pretty easy to work with, right?

After a few days, you have trained a model that predicts numbers resembling the ones in the table.

You contact the medical researcher and discuss the results.

Model Discussion

Researcher: So, you got the data for all the patients?

Model Discussion

Researcher: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Model Discussion

Researcher: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Researcher: Amazing. There were so many data issues with this set of patients that I couldn't do much.

Model Discussion

Researcher: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Researcher: Amazing. There were so many data issues with this set of patients that I couldn't do much.

Data Miner: Oh? I didn't hear about any possible problems.

Model Discussion

Researcher: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Researcher: Amazing. There were so many data issues with this set of patients that I couldn't do much.

Data Miner: Oh? I didn't hear about any possible problems.

Researcher: Well, first, there is field 5, the variable we want to predict. It's common knowledge among people who analyze this type of data that results are better if you work with the log of the values, but I didn't discover this until later. Was it mentioned to you?

Model Discussion

Researcher: So, you got the data for all the patients?

Data Miner: Yes. I haven't had much time for analysis, but I do have a few interesting results.

Researcher: Amazing. There were so many data issues with this set of patients that I couldn't do much.

Data Miner: Oh? I didn't hear about any possible problems.

Researcher: Well, first, there is field 5, the variable we want to predict. It's common knowledge among people who analyze this type of data that results are better if you work with the log of the values, but I didn't discover this until later. Was it mentioned to you?

Data Miner: No.

Model Dicsuccion

Researcher: But surely you heard about what happened to field 4? It's supposed to be measured on a scale from 1 to 10, with 0 indicating a missing value, but because of a data entry error, all 10's were changed into 0's. Unfortunately, since some of the patients have missing values for this field, it's impossible to say whether a 0 in this field is a real 0 or a 10. Quite a few of the records have that problem.

Model Dicsuccion

Researcher: But surely you heard about what happened to field 4? It's supposed to be measured on a scale from 1 to 10, with 0 indicating a missing value, but because of a data entry error, all 10's were changed into 0's. Unfortunately, since some of the patients have missing values for this field, it's impossible to say whether a 0 in this field is a real 0 or a 10. Quite a few of the records have that problem.

Data Miner: Interesting. Were there any other problems?

Model Dicsuccion

Researcher: But surely you heard about what happened to field 4? It's supposed to be measured on a scale from 1 to 10, with 0 indicating a missing value, but because of a data entry error, all 10's were changed into 0's. Unfortunately, since some of the patients have missing values for this field, it's impossible to say whether a 0 in this field is a real 0 or a 10. Quite a few of the records have that problem.

Data Miner: Interesting. Were there any other problems?

Researcher: Yes, fields 2 and 3 are basically the same, but I assume that you probably noticed that.

Model Dicsuccion

Researcher: But surely you heard about what happened to field 4? It's supposed to be measured on a scale from 1 to 10, with 0 indicating a missing value, but because of a data entry error, all 10's were changed into 0's. Unfortunately, since some of the patients have missing values for this field, it's impossible to say whether a 0 in this field is a real 0 or a 10. Quite a few of the records have that problem.

Data Miner: Interesting. Were there any other problems?

Researcher: Yes, fields 2 and 3 are basically the same, but I assume that you probably noticed that.

Data Miner: Yes, but these fields were only weak predictors of field 5.

Model Discussion

Researcher: Anyway, given all those problems, I'm surprised you were able to accomplish anything.

Data Miner: True, but my results are really quite good. Field 1 is a very strong predictor of field 5. I'm surprised that this wasn't noticed before.

Researcher: What? Field 1 is just an identification number.

Data Miner: Nonetheless, my results speak for themselves.

Researcher: Oh, no! I just remembered. We assigned ID numbers after we sorted the records based on field 5. There is a strong connection, but it isn't very sensible. Sorry.

Model Discussion

Researcher: Anyway, given all those problems, I'm surprised you were able to accomplish anything.

Data Miner: True, but my results are really quite good. Field 1 is a very strong predictor of field 5. I'm surprised that this wasn't noticed before.

Researcher: What? Field 1 is just an identification number.

Data Miner: Nonetheless, my results speak for themselves.

Researcher: Oh, no! I just remembered. We assigned ID numbers after we sorted the records based on field 5. There is a strong connection, but it isn't very sensible. Sorry.

OK, what's the point?

You have to

Understand the task you want to solve and the data!

Data Objects

Data objects represent entities we work with (e.g., classify them).

For example, in cancer prediction, the data objects are patients. In fruit classification, the data objects are individual fruits.

Data Objects

Data objects represent entities we work with (e.g., classify them).

For example, in cancer prediction, the data objects are patients. In fruit classification, the data objects are individual fruits.

Data objects are described by *attributes* (or *features* or *variables*).

For example, the age, weight, genetic profile, and other patient characteristics. Or the width and height of a fruit.

Attributes vs Features vs Variables

The name differs from field to field.

Attributes vs Features vs Variables

The name differs from field to field.

So, the following names are usually used as synonyms:

- ▶ *Attributes* - used mostly by database and data mining experts.
- ▶ *Features* - used mostly by machine learning experts.
- ▶ *Variables* - used mostly by statisticians.

Attributes vs Features vs Variables

The name differs from field to field.

So, the following names are usually used as synonyms:

- ▶ *Attributes* - used mostly by database and data mining experts.
- ▶ *Features* - used mostly by machine learning experts.
- ▶ *Variables* - used mostly by statisticians.

One may make some distinctions

- ▶ *Attributes* represent information about the object without any additional assumptions.
- ▶ *Features* assume that their values are somewhat characteristic of the object.
- ▶ *Variables* assume that there is some process behind them (typically a random process in the case of statistics).

Data Types - Categorical Attributes

Categorical attributes (nominal attributes) are symbols or names of things.

- ▶ Each value represents some kind of category, code, or state.
- ▶ Values are not ordered and should not be used quantitatively (in computer science, the values are known as enumerations).

Data Types - Categorical Attributes

Categorical attributes (nominal attributes) are symbols or names of things.

- ▶ Each value represents some kind of category, code, or state.
- ▶ Values are not ordered and should not be used quantitatively (in computer science, the values are known as enumerations).
- ▶ **Examples:**

$\text{hair_color} \in \{\text{black, brown, blond, red, auburn, gray, white}\}$

$\text{marital_status} \in \{\text{single, married, divorced, widowed}\}$

$\text{customer_ID} \in \{0, 1, 2, \dots\}$

Even though the last one is usually expressed using numbers, it should not be used quantitatively.

Data Types - Categorical Attributes

Categorical attributes (nominal attributes) are symbols or names of things.

- ▶ Each value represents some kind of category, code, or state.
- ▶ Values are not ordered and should not be used quantitatively (in computer science, the values are known as enumerations).
- ▶ **Examples:**

$\text{hair_color} \in \{\text{black, brown, blond, red, auburn, gray, white}\}$

$\text{marital_status} \in \{\text{single, married, divorced, widowed}\}$

$\text{customer_ID} \in \{0, 1, 2, \dots\}$

Even though the last one is usually expressed using numbers, it should not be used quantitatively.

Binary attributes are categorical attributes with only two values.

DataTypes - Ordinal Attributes

Ordinal attribute is an attribute with values that have a meaningful order or ranking among them.

DataTypes - Ordinal Attributes

Ordinal attribute is an attribute with values that have a meaningful order or ranking among them.

Examples:

$\text{drink_size} \in \{\text{small, medium, large}\}$

$\text{grades} \in \{\text{A, B, C, D, E, F}\}$

It can also be obtained by discretizing numeric quantities into series of intervals.

Ordinal attributes do not allow arithmetic operations.

DataTypes - Ordinal Attributes

Ordinal attribute is an attribute with values that have a meaningful order or ranking among them.

Examples:

$\text{drink_size} \in \{\text{small, medium, large}\}$

$\text{grades} \in \{\text{A, B, C, D, E, F}\}$

It can also be obtained by discretizing numeric quantities into series of intervals.

Ordinal attributes do not allow arithmetic operations.

Categorical and ordinal attributes are called *qualitative* attributes.

Next, we look at numeric, i.e., *quantitative* attributes.

Data Types - Numeric Attributes

Numeric attributes are quantities represented by numbers.

Data Types - Numeric Attributes

Numeric attributes are quantities represented by numbers.

Distinguish two types: *Interval-scale* and *ratio-scale*.

	INTERVAL SCALE	RATIO SCALE
Measurement interval	Equal intervals between consecutive points.	Equal intervals with the presence of a true zero.
Absolute zero	Lacks a true zero point.	Possesses a true zero point.
Statistical analysis	Limited to addition and subtraction	Allows for meaningful multiplication and division.
Meaningful ratios	Ratios are not meaningful due to the lack of zero.	Ratios are meaningful due to the presence of zero.
Examples	IQ scores, Celsius temperature, NPS data, etc.	Height, weight, income, etc.

Discrete vs Continuous Attributes

Often, two kinds of numeric attributes are distinguished:

Discrete vs Continuous Attributes

Often, two kinds of numeric attributes are distinguished:

- ▶ *Discrete*

A finite or countably infinite range of values, i.e., integers may represent the values.

Some (but not all) authors count the qualitative (categorical, ordinal) attributes among the discrete attributes.

Discrete vs Continuous Attributes

Often, two kinds of numeric attributes are distinguished:

- ▶ *Discrete*

A finite or countably infinite range of values, i.e., integers may represent the values.

Some (but not all) authors count the qualitative (categorical, ordinal) attributes among the discrete attributes.

- ▶ *Continuous*

An uncountably infinite range of values, typically an interval.

There are several more or less formal definitions of continuous attributes in the literature. For example:

- ▶ All non-discrete variables.
- ▶ Have an infinite number of values between any two values.
- ▶ Their values are measured (??).

Deeper characteristics of data (statistical properties, etc.) will be examined at tutorials.