Fachhochschule
Dortmund
University of Applied Sciences and Arts

Data Science

11: Machine Learning in a nutshell

Klaus Kaiser
WiSe 2024 / 2035

Data Science

Recap

In the following we consider the general test procedure.

$X_1, \ldots, X_n$ independent and identically distributed random variables.

1. Formulation of the **test-problem**:
   - H_0: $\mu = \mu_0$ vs. H_1: $\mu \neq \mu_0$ (two-sided)
   - H_0: $\mu \leq \mu_0$ vs. H_1: $\mu > \mu_0$ (right-sided)
   - H_0: $\mu \geq \mu_0$ vs. H_1: $\mu < \mu_0$ (left-sided)

   The rejection of H_0 is a hard conclusion for which the probability of a wrong decision is limited by α. Therefore, the **important statement to be verified is placed in the alternative**.

2. Chose a proper **significance level** α
3. **Test-statistic**: *TS*: Choice depends on the distribution of the random variable.

Data Science

Recap

4. Determination of the **area of rejection** for selected α: Reject H_0, if
   - $|ts| > ts_{1-\alpha/2}$ for a two-sided test
   - $ts > ts_{1-\alpha}$ for a right-sided test
   - $ts < -ts_{1-\alpha}$ for a left-sided test
5. Compute the **value of the test statistic** for an observed sample: ts
6. **Decision:**
   - **Reject** H_0 if the value of z is in the rejection area
     Do **not reject** H_0 if the value of z is **not** in the rejection area
   - Name the used significance level
   - Formulate the significance of the test decision for the original question

Data Science

Recap

Null hypothesis | **Alternative hypothesis** | **Test-statistics** | **Rejection area**

(approximate) Gaussian test ($X \sim \mathcal{N}(\mu, \sigma^2)$ or $n \geq 30$, σ known)

$\mu = \mu_0$	$\mu \neq \mu_0$	$Z = \frac{\overline{X} - \mu_0}{\sigma / \sqrt{n}}$	$\lvert z \rvert > z_{1-\frac{\alpha}{2}}$
$\mu \geq \mu_0$	$\mu < \mu_0$		$z < -z_{1-\alpha}$
$\mu \leq \mu_0$	$\mu > \mu_0$		$z > z_{1-\alpha}$

t-test on location ($X \sim \mathcal{N}(\mu, \sigma^2)$, σ unknown)

$\mu = \mu_0$	$\mu \neq \mu_0$	$T = \frac{\overline{X} - \mu_0}{S / \sqrt{n}}$	$\lvert t \rvert > t_{n-1,1-\frac{\alpha}{2}}$
$\mu \geq \mu_0$	$\mu < \mu_0$		$t < -t_{n-1,1-\alpha}$
$\mu \leq \mu_0$	$\mu > \mu_0$		$t > t_{n-1,1-\alpha}$

approximate Gaussian test ($n \geq 30$, σ unknown)

$\mu = \mu_0$	$\mu \neq \mu_0$	$Z = \frac{\overline{X} - \mu_0}{S / \sqrt{n}}$	$\lvert z \rvert > z_{1-\frac{\alpha}{2}}$
$\mu \geq \mu_0$	$\mu < \mu_0$		$z < -z_{1-\alpha}$
$\mu \leq \mu_0$	$\mu > \mu_0$		$z > z_{1-\alpha}$

t-Test for location difference

Assumption: $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$, $Y \sim \mathcal{N}(\mu_Y, \sigma_Y^2)$, σ_X, σ_Y unknown

Null-hypothesis	Alternative hypothesis	Test-statistic	Rejection area
$H_0 : \mu_X - \mu_Y = \delta_0$	$H_1 : \mu_X - \mu_Y \neq \delta_0$	$T = \frac{\overline{X} - \overline{Y} - \delta_0}{\sqrt{S_X^2/n + S_Y^2/m}}$	$\lvert t \rvert > t_{k,1-\alpha/2}$
$H_0 : \mu_X - \mu_Y \geq \delta_0$	$H_1 : \mu_X - \mu_Y < \delta_0$		$t < -t_{k,1-\alpha}$
$H_0 : \mu_X - \mu_Y \leq \delta_0$	$H_1 : \mu_X - \mu_Y > \delta_0$		$t > t_{k,1-\alpha}$

with $k = \lfloor (S_X^2/n + S_Y^2/m)^2 / (\frac{1}{n-1}(S_X^2/n)^2 + \frac{1}{m-1}(S_Y^2/m)^2) \rfloor$

Data Science

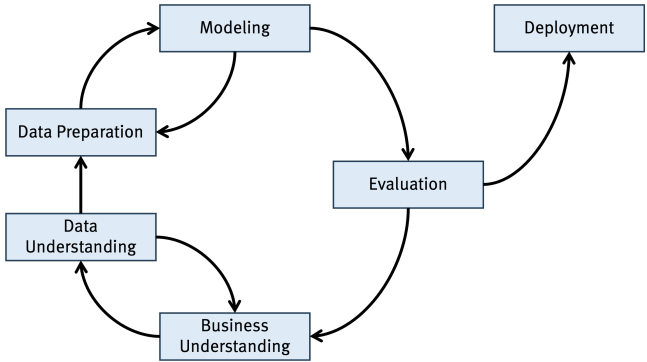
Today

A common machine learning model and more

Machine Learning in a nutshell

Data Science

Today



CRISP-DM process model gives an overview on the steps needed for developing a machine learning model.

Data Science

Today

- Business understanding: Talk with the data holder!
- Data understanding: Analyze the given data (see previous lectures)
- Modeling: Linear regression - **more models?**
- **Evaluation?**

The deployment step is very important to bring machine learning models to production, unfortunately we neglect this step...

Machine Learning in a nutshell

Data preparation

Data Science

Data preparation

Already seen: Depending on the source of data, the quality can differ extremely

- There could be different formats (e.g. German or international date format)
- There could be different sources which needs to be joined
- There could be missing values
- ...

The process of preparing one or multiple data sets for machine learning is called **data preparation**.

Data Science

Data preparation

Missing data is quite common, unfortunately they can occur in different ways:

- No values given
- A value like *N/A*, *NULL*, *NONE*, – ...
- Non-typical values like 9999 for a year or -1 for a height.

Example

- **Survey:** In a survey, some questions could be led open or are only answered if needed
- **Sensor data:** A sensor might be broken for some time or had transmission issue

There is no general strategy to handle missing values, it is common to **filter**, **mark** or **impute** these values.

Filter missing values

- If there are only a few observations with missing values, these could be dropped.
- If an attribute mainly contains missing value, the attribute could be dropped.

Deleting the observations with missing values could introduce a bias into the data!

Mark missing values

- For categorical attributes, one could include a class for the missing values.
- For other attributes, one could add a categorical variable which identifies if a value is missing or not.

The process of replacing missing values with a substitute value is called **imputation**.

Imputation strategies

There are several strategies to impute missing values. Some examples:

- Replace all missing values with the mean, median or modal value of the other observations
- Replace one missing value with the value of an observation which is most likely
- Create a model to predict the missing value based on the remaining attributes

For finding the most likely observation a proper metric is needed. For mixed variables (qualitative and quantitative), the Grover-metric can be used.

Data Science

Data preparation

Task: Consider the following data set[1] concerning the prices of laptops. There are several variables with missing values, how to impute these missing values?

variable	number missing values
Laptop	0
Status	0
Brand	0
Model	0
CPU	0
RAM	0
Storage	0
Storage type	42
GPU	1371
Screen	4
Touch	0
Final Price	0

Data Science

Data preparation

False data is quite common, unfortunately they can occur in different ways:

- Typographical errors (e.g. 10 or 10 instead of 100)
- Different notation (e.g. *str* instead of *street*)
- Duplicates
- Systematically errors (e.g. false variable types)

In contrast to missing values, it is not directly clear if a value is wrong. In some cases it is impossible to identify false data!

Data Science

Data preparation

- Analysis of the data - see previous lectures - to identify suspicious data
- Analysis of outlier (e.g. using a boxplot for metric data)

Discuss with a **domain expert** for better data understanding and identify false data!

Handling missing data

- Correcting false data if possible (e.g. with the help of domain experts)
- Handling false data as missing data

Data Science

Data preparation

Often, a sample contains information which are not needed (could be dropped) or information that should be processed further.

The process of transforming raw data into effective and meaningful data is called **feature engineering**.

- Create new features based on the given data
- Replace data with new computed features
- New values should be more general or show concrete information
- Given attributes could be combined to new ones
- Rescale values

Data Science

Data preparation

Examples

- Compute the age of a patient from birthdate
- Compute Weekday from a date
- Derive the address from geo coordinates
- Compute the relative sales price based on the purchasing power in a year

Data Science

Data preparation

Task: Consider the following dataset, what are possible features one could derive?

Lot Area	NBHD	Type	Qual	Cond	Built	1st 1st Flr SF	Mo Sold	SalePrice
11160	NAmes	1Fam	7	5	1968	2110	4.2010	244000
4920	StoneBr	TwnhsE	8	5	2001	1338	4.2010	213500
7500	Gilbert	1Fam	7	5	1999	1028	6.2010	189000
7980	Gilbert	1Fam	6	7	1992	1187	3.2010	185000
12537	NAmes	1Fam	5	6	1971	1078	4.2010	149900
1680	BrDale	Twnhs	5	5	1971	525	3.2010	105500
2280	NPkVill	Twnhs	7	6	1975	836	6.2010	120000
11520	NridgHt	1Fam	9	5	2005	1698	6.2010	275000
10171	NridgHt	1Fam	7	5	2004	1535	3.2010	214000
7132	NridgHt	TwnhsE	8	5	2006	1370	4.2010	205000
3203	Blmngtn	TwnhsE	7	5	2006	1145	1.2010	160000
13300	Gilbert	1Fam	7	5	2004	744	6.2010	184500

Data Science

Data preparation

With the help of data preparation, a dataset is prepared for

There are much more things, one need to consider:

- Balancing of classes
- Sparse data
- too large or too few data
- ...

Machine Learning in a nutshell

Decision trees

Data Science

Decision trees

Simple linear regression

Computing the best fitting linear relation between two variables. Result gives the expected value for a given input.

Extension: Multiple linear regression - linear relation between multiple variables and one target variable.

Often: Target variable is a categorical variable and not a continuous variable.

As **classification**, we denote the process of applying a class (category) to an object based on their properties.

- Categories are predefined
- A classifier is a process / function / model that applies the class for a given object

Linear regression can also be used as classifier e.g. by applying class A if the predicted value is larger than a specific threshold

Examples

- Car category (van, sports car) based on the properties of the Car
- Disease based on the symptoms
- Spam mails based on the content

Data Science

Decision trees

No free lunch theorem

The quality of a classifier, highly depend on the given data to be classified. Therefore, there is no classifier which is optimal for all types of data or problems.

Task: How does a human classify objects?

Supervised learning is the process to learn a predictor based on data where the desired output is given.

Data Science

Decision trees

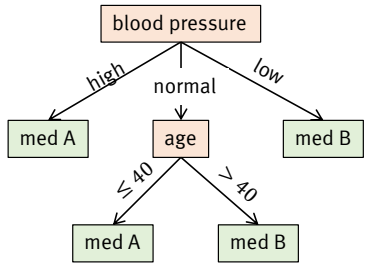
A decision tree is an ordered directed tree, by following the tree to leafs decisions are done.

- In every node a test on an attribute of the data is done (e.g. height of a person is larger than 1.7 m)
- Each leaf describes an outcome, i.e. a class (e.g. van, sports car etc.)

Example: How to choose the right medicine

The goal is to choose the right medicine based on the age and blood pressure of the patient.

- Measure blood pressure and age
- Check blood pressure:
  - high: medicine A
  - low: medicine B
  - normal: Check age:
    - older than 40: medicine B
    - younger than 40: medicine A



Data Science

Decision trees

A decision tree can be build by hand or learned with the help of data!

- Building the tree starting with the root node - which property first?
- Idea:
  - Choosing the split of the data based on one attribute which reduces the entropy (disorder) most
  - Compute reduction of the entropy for all attributes

The **entropy** describes the disorder in a set. The more objects with different classes are in the set - the larger is the entropy

Data Science

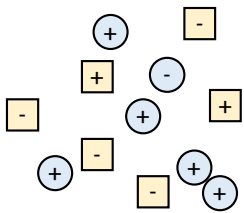
Decision trees

The **entropy** $H(M)$ of a set M can be computed by

$$H(M) = \sum_{x \in M} -p_x \cdot \log_2(p_x)$$

where p_x is the relative frequency of an element x in the set M. Here, we define $0 \cdot \log_2(0) = 0$.

Example



There are different elements given. The goal is to classify the attributes in classes *rectangle / yellow* and *circle / blue*

Task: Compute the Entropy for the classes blue and yellow

$$H_{blue,yellow} = -p_{blue} \cdot \log_2 p_{blue} - p_{yellow} \cdot \log_2 p_{yellow}$$
$$= -\frac{6}{12} \log_2 \left(\frac{6}{12}\right) - \frac{6}{12} \log_2 \left(\frac{6}{12}\right) = 1$$

Data Science

Decision trees

For a split $S = \{s | s \subset M\}$, we can compute the **gain** H_G **of order** by

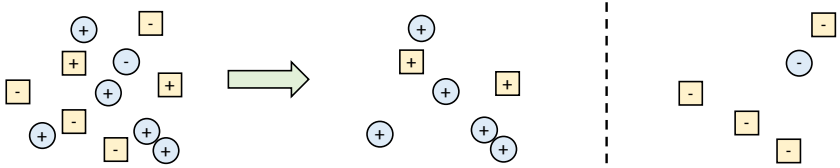
$$H_G = H_M - \oplus(S),$$

where $\oplus(S)$ denotes the weighted sum of the entropy of the splitted subsets.

$$\oplus(S) = \sum_{s \in S} \frac{|s|}{|M|} H(s).$$

Data Science

Decision trees



$$H_G = H_{complete} \quad - \quad H_+ \quad \oplus \quad H_-$$

For an attribute with two possible values ($+$ and $-$) the gain can be computed by

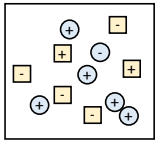
$$H_+ \oplus H_- = \frac{n_+ H_+ + n_- H_-}{n_+ + n_-}$$

where n_- equals to the number of observations of class $+$ and n_- the number of observations of class $+$.

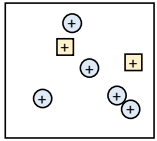
Data Science

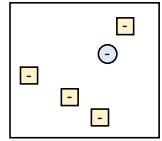
Decision trees

Example



$$H_M = -\frac{6}{12}\log_2\left(\frac{6}{12}\right) - \frac{6}{12}\log_2\left(\frac{6}{12}\right) = 1$$





$$H_+ = -\frac{5}{7}\log_2\left(\frac{5}{7}\right) - \frac{2}{7}\log_2\left(\frac{2}{7}\right)$$

$$\approx 0.863$$

$$H_+ = -\frac{1}{5}\log_2\left(\frac{1}{5}\right) - \frac{4}{5}\log_2\left(\frac{4}{5}\right)$$

$$\approx 0.729$$

Data Science

Decision trees

Example

#	Sex	Age	Blood pressure	Medicine
1	m	20	normal	A
2	f	73	normal	B
3	f	37	high	A
4	m	33	low	B
5	f	48	high	A
6	m	29	normal	A
7	f	52	normal	B
8	m	42	low	B
9	m	61	normal	B
10	f	30	normal	A
11	f	26	low	B
12	m	54	high	A

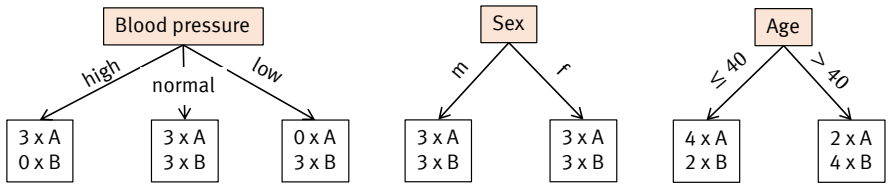
Task: Decide which medicine should be used, based on sex, age and blood pressure of the patients.

Data Science

Decision trees

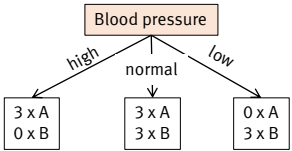
There are three possible splits for the first step

1. based on **sex** in **male** and **female**
2. based on **age** in (e.g.) **older than 40** and **younger than 40**
3. based on **blood pressure** in **low**, **normal** and **high**



Data Science

Decision trees



normal

Sex	Age	Med.
m	20	A
f	73	B
m	29	A
f	52	B
m	61	B
f	30	A

low

Sex	Age	Med.
m	33	B
m	42	B
f	26	B

high

Sex	Age	Med.
f	37	A
f	48	A
m	54	A

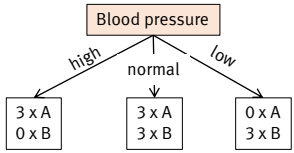
Task: Compute the values

$$H(M) = \sum_{x \in M} -p_x \cdot \log_2(p_x)$$

$$H_G = H_M - \sum_{s \in S} \frac{|s|}{|M|} H(s)$$

Data Science

Decision trees



normal

Sex	Age	Med.
m	20	A
f	73	B
m	29	A
f	52	B
m	61	B
f	30	A

low

Sex	Age	Med.
m	33	B
m	42	B
f	26	B

high

Sex	Age	Med.
f	37	A
f	48	A
m	54	A

Task: Compute the values

$$H(M) = \sum_{x \in M} -p_x \cdot \log_2(p_x)$$

$$H_G = H_M - \sum_{s \in S} \frac{|s|}{|M|} H(s)$$

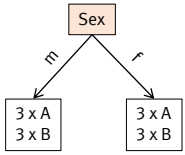
$$H_{normal} = 1, \qquad H_{low} = 0, \qquad H_{high} = 0$$

$$H_{all} = -\frac{6}{12} log_2\left(\frac{6}{12}\right) - \frac{6}{12} log_2\left(\frac{6}{12}\right) = 1$$

$$H_{G,bp} = H_{all} - \left(\frac{3}{12} \cdot 0 + \frac{6}{12} \cdot 1 + \frac{3}{12} \cdot 0\right) = 1 - \frac{1}{2} = \frac{1}{2}$$

Data Science

Decision trees



male

Age	BP	Med.
20	normal	A
33	low	B
29	normal	A
42	low	B
61	normal	B
54	high	A

female

Age	BP	Med.
73	normal	B
37	high	A
48	high	A
52	normal	B
30	normal	A
26	low	B

$$H(M) = \sum_{x \in M} -p_x \cdot \log_2(p_x)$$

$$H_G = H_M - \sum_{s \in S} \frac{|s|}{|M|} H(s)$$

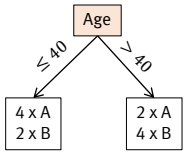
$$H_m = 1, \qquad H_w = 1$$

$$H_{all} = 1$$

$$H_{G,sex} = H_{all} - \left(\frac{6}{12} \cdot 1 + \frac{6}{12} \cdot 1 \right) = 1 - 1 = 0$$

Data Science

Decision trees



Younger than 40

Sex	BP	Med.
m	normal	A
f	high	A
m	low	B
m	normal	A
f	normal	A
f	low	B

Older than 40

Sex	BP	Med.
f	normal	B
f	high	A
f	normal	B
m	low	B
m	normal	B
m	high	A

$$H(M) = \sum_{x \in M} -p_x \cdot \log_2(p_x)$$

$$H_G = H_M - \sum_{s \in S} \frac{|s|}{|M|} H(s)$$

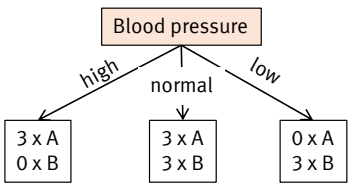
$$H_y \approx 0.918, \qquad H_o \approx 0.918$$

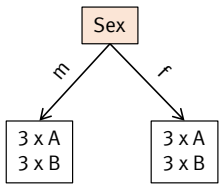
$$H_{all} = 1$$

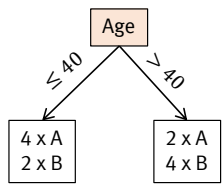
$$H_{G,age} = H_{all} - 0.918 = 0.082$$

Data Science

Decision trees





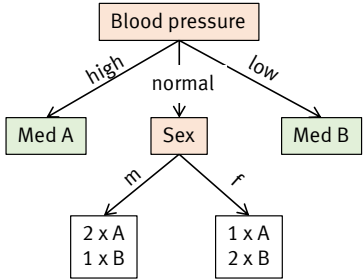


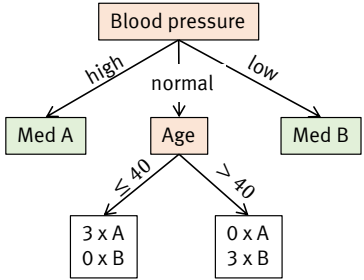
$H_{G,BP} = 0.5$ $H_{G,sex} = 0$ $H_{G,age} = 0.082$

- The gain for using blood pressure to split the data is the largest. Therefore: Using blood pressure for the first split
- The resulting subsets for high and low are pure, therefore the decision for these sets are clear
- Further splitting for the subset normal is needed

Data Science

Decision trees



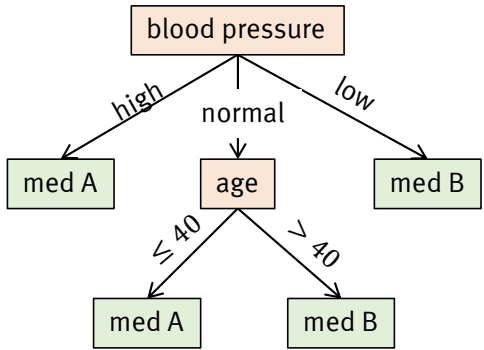


$H_{G,sex} = 0.82$ $H_{G,age} = 1$

- The gain for using age to split the data is the largest.
- The resulting subsets are pure, therefore all decisions are clear
- The attribute sex does not play any role in the decision tree

Data Science

Decision trees



Data Science

Decision trees

Decision trees are a simple method for decision or classification problems.

Pros

- Easy to be implemented (if-then-rules)
- Fast in usage (only follow the path through tree)
- Explainable and easy to understand (no black-box)
- Can be extended to regression problems

Cons

- Not useful for numerical attributes (discretization)
  - Need to define fixed boundaries (e.g. $age < 40$)
- Could lead to small-scale distinctions (overfitting)

Data Science

Decision trees

- Decision trees are one part of *state of the art* machine learning algorithms
- Especially, for tabular data tree based methods are very popular

The **random forest** method learns multiple, ideally uncorrelated, decisions trees. From these trees an ensemble is build to make a prediction.

Similarly to random forest methods, **gradient boosting decision tree** methods learn an ensemble of decisions trees. These are improved due to the boosting technique (technique to reduce bias).

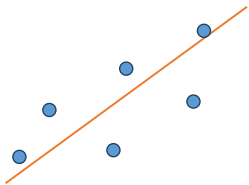
Machine Learning in a nutshell

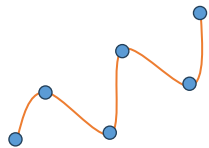
Evaluating models

Data Science
Evaluating models

A machine learning model, like decision trees or linear regression, are fitted to a given dataset.

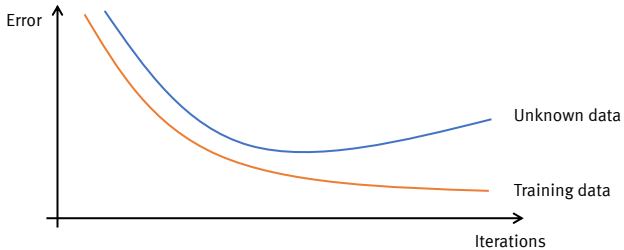
- Training process finds optimal fit for the given data
- **Problem:** The perfect fit for the training data might not be the perfect general model!





Data Science

Evaluating models



The problem of **overfitting** occurs, if a model fits too good to the training-data, and therefore performs worse on other data.

The problem of **underfitting** occurs, if a model can not adequately capture the structure of the data.

Data Science
Evaluating models

- Compute error (e.g. residuum) on independent dataset

In machine learning it is common to split the given data set into a training set and a test set. This is called **train-test splitting**.

- The train-test splitting can be obtained in different ways, depending on the application and data:
  - Random split: Randomly separate the data in train/test
  - Temporal split: For temporal data, split could be obtained by putting data before a date into train and after a date into test
- Often $80/20$ or $70/30$ split, i.e. use 80% of the data for training and 20% for testing

Data Science
Evaluating models

In order to compare machine learning models, it is useful to add a additional dataset, i.e. one performs a **train-validation-test split**.

- **Train set:** Used to train the different models
- **Validation set:** Used to compare the different models
- **Test set:** Used to verify that the best model performs well

How to measure the quality of an algorithm?

Data Science
Evaluating models

The **mean squared error (MSE)** of a predictor $\hat{\theta}$ for an unknown value θ is given by $E\left(\left(\hat{\theta}-\theta\right)^2\right)$. Thus, for predicted values $\hat{y}_i$ with exact values y_i the MSE is given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

There are several alternative

- Mean average error (MAE): $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$
- Mean absolute percentage error: $\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i-\hat{y}_i|}{|y_i|}$

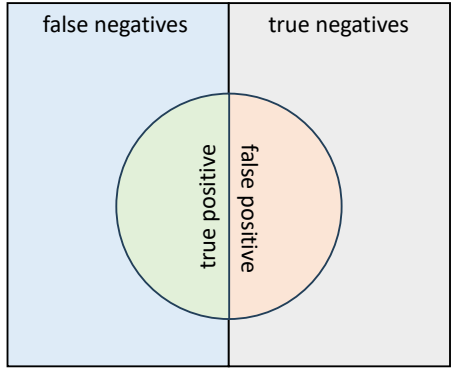
Task: What is the difference of these errors?

The **confusion matrix** is a table, collecting the quality of predictions on a dataset for binary classification problem.

		actual values	
		Positive	negative
predicted values	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Data Science

Evaluating models



Data Science

Evaluating models

Task: Assume that the following table gives the results of a machine learning model on a test-set. What is the confusion matrix?

i	y_i	$\hat{y}_i$
1	1	1
2	1	0
3	1	0
4	1	1
5	1	1
6	0	1
7	0	1
8	0	0
9	1	0
10	1	1

Data Science

Evaluating models

Task: Assume that the following table gives the results of a machine learning model on a test-set. What is the confusion matrix?

i	y_i	$\hat{y}_i$
1	1	1
2	1	0
3	1	0
4	1	1
5	1	1
6	0	1
7	0	1
8	0	0
9	1	0
10	1	1

		actual values	
		1	0
predicted values	1	4	2
	0	3	1

Data Science
Evaluating models

For the TP, FP, TN, FN values of confusion matrix we define

- the **accuracy** as

$$\frac{TP + TN}{TP + TN + FP + FN}$$

- the **precision** as

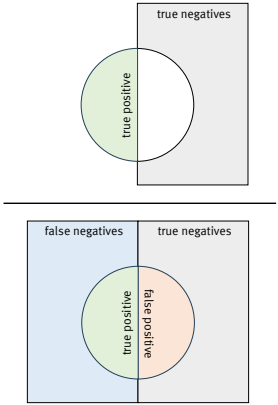
$$\frac{TP}{TP + FP}$$

- the **recall** as

$$\frac{TP}{TP + FN}$$

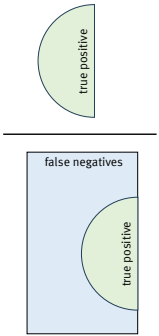
Data Science

Evaluating models



Proportion of all predicted correct.

Proportion of the true predicted correct.



Proportion of the true class found.

Data Science
Evaluating models

Task: Compute accuracy, precision and recall for the given confusion matrix

		actual values	
		1	0
predicted values	1	4 (TP)	2 (FP)
	0	3 (FN)	1 (TN)

- **accuracy:** $\frac{TP+TN}{TP+TN+FP+FN}$
- **precision:** $\frac{TP}{TP+FP}$
- **recall:** $\frac{TP}{TP+FN}$

Task: Compute accuracy, precision and recall for the given confusion matrix

		actual values 1	actual values 0
predicted values	1	4 (TP)	2 (FP)
	0	3 (FN)	1 (TN)

- **accuracy:** $\frac{TP+TN}{TP+TN+FP+FN}$
- **precision:** $\frac{TP}{TP+FP}$
- **recall:** $\frac{TP}{TP+FN}$

- **accuracy:** $\frac{TP+TN}{TP+TN+FP+FN} = \frac{4+1}{4+1+3+2} = \frac{5}{10} = 0.5$
- **precision:** $\frac{TP}{TP+FP} = \frac{4}{4+2} = \frac{4}{6} = 0.66$
- **recall:** $\frac{TP}{TP+FN} = \frac{4}{4+3} = \frac{4}{7} = 0.57$

The predictive performance of a classifier can be measured with help of the **F1** score, which is given by

$$F_1 = 2\frac{\mathit{precision} \cdot \mathit{recall}}{\mathit{precision} + \mathit{recall}}$$

- Harmonic mean of precision and recall
- If recall and precision equals to 1, the F_1 score equals to 1.
- If recall or precision equals to 0, the F_1 score equals to 0.

Task: Compute the F_1 score of the given precision and recall values!

- **precision:** $\frac{TP}{TP+FP} = \frac{4}{4+2} = \frac{4}{6} = 0.66$
- **recall:** $\frac{TP}{TP+FN} = \frac{4}{4+3} = \frac{4}{7} = 0.57$

Task: Compute the F_1 score of the given precision and recall values!

- **precision:** $\frac{TP}{TP+FP} = \frac{4}{4+2} = \frac{4}{6} = 0.66$
- **recall:** $\frac{TP}{TP+FN} = \frac{4}{4+3} = \frac{4}{7} = 0.57$

$$F_1 = 2\frac{precision \cdot recall}{precision + recall} = 2\frac{0.66 \cdot 0.57}{0.66 + 0.57} = 0.31$$

Data Science
Evaluating models

Computing the metrics of a machine learning model gives a glimpse on the quality. Unfortunately, this not always gives an information if the results are useful or not.

- Is 90% accuracy good?
  - Assume 100 data points, 90 are of class *A* and 10 of class *B*.
  - Predicting always class *A* would lead to an accuracy of 90%
- Defining a naive reference solution could help in interpreting the results

Data Science
Evaluating models

Consider the following setting: A machine learning model should predict the price of a car, based on the properties brand, style and engine.

Task: What are possible naive reference solutions?

Summary & Outlook

Data Science

Summary & Outlook: Summary

- You understand what a data preparation is and are able to perform some preparation steps
- You are able to compute and use decision tree classifier
- You know how to evaluate machine learning models and compare them

Dashboards and Summary

References

Data Science

Summary & Outlook: Endnotes

[1] https://www.kaggle.com/datasets/juanmerinobermejo/laptops-price-dataset

Data Science

Summary & Outlook: Acknowledgement

Parts of the lecture base on the lectures

- ”Angewandtes Machinelles Lernen” (FH Dortmund) by Prof. Dr. Nadja Bauer
- ”Adaptive Systeme” (FH Dortmund) by Prof. Dr. Inga Saatz, Prof. Dr. Christoph M. Friedrich, Dr. Marcus Frenkel, Prof. Dr. Klaus Kaiser