



Übungslektion 12 – Machine Learning

Informatik II

6. / 7. Mittwoch 2025

Heutiges Programm

Wiederholung der Vorlesung

Theoretische Übungen

Praktische Übungen

Homework

1. Wiederholung der Vorlesung

Machinelles Lernen

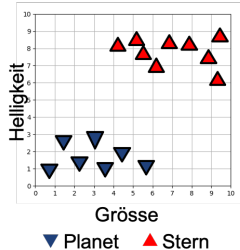
Trainieren von Modellen mithilfe von Beispielen.

ML hat Anwendungen in:

- Astronomie
- Physik
- Medizin
- Etc.

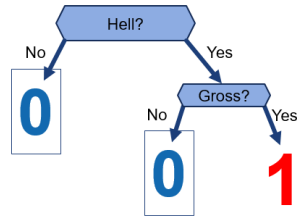
Entscheidungsbäume

Sterne und Planeten Datensatz



Sterne sind gross und hell.
Planeten sind kleiner und weniger hell.

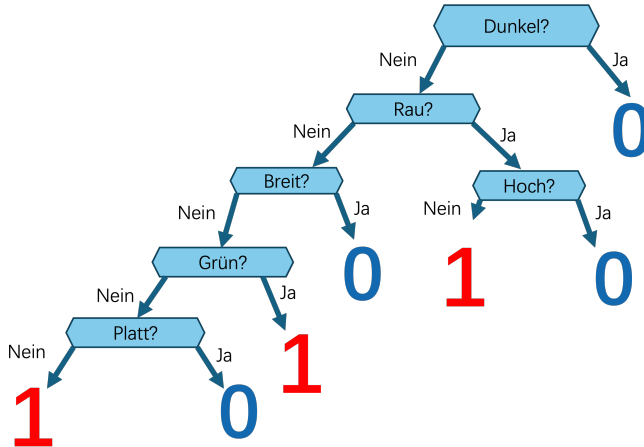
Entscheidungsbaum



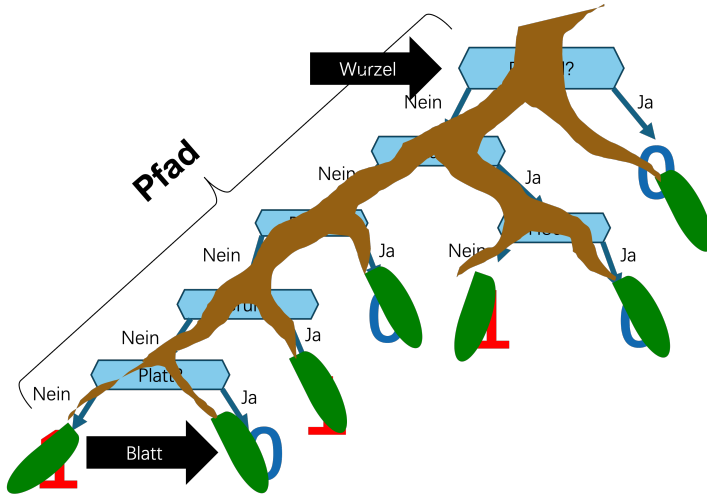
Planeten: 0, Sterne: 1

Tiefe des Baumes: Anzahl von Knoten im längsten Pfad (zwischen Wurzel und Blatt).

Entscheidungsbäume

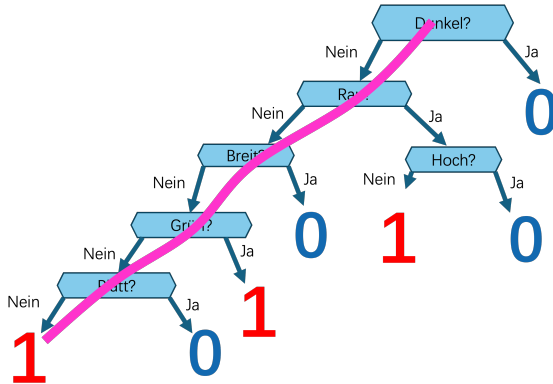


Entscheidungsbäume



Entscheidungsbäume

Tiefe = 5



Trainieren von Modellen in ML

- **Datensatz** Erstellen Sie einen Datensatz und teilen Sie ihn in den Trainingssatz D und den Validierungssatz D' auf.

Trainieren von Modellen in ML

- **Datensatz** Erstellen Sie einen Datensatz und teilen Sie ihn in den Trainingssatz D und den Validierungssatz D' auf.
- **Modell auswählen** Modell $\mathcal{H}$: eine Menge von Entscheidungsfunktionen (z.B. Entscheidungsbäume der Tiefe ≤ 3 , logistische Regression, oder spezifische NN Architektur)

Trainieren von Modellen in ML

- **Datensatz** Erstellen Sie einen Datensatz und teilen Sie ihn in den Trainingssatz D und den Validierungssatz D' auf.
- **Modell auswählen** Modell $\mathcal{H}$: eine Menge von Entscheidungsfunktionen (z.B. Entscheidungsbäume der Tiefe ≤ 3 , logistische Regression, oder spezifische NN Architektur)
- **Verlustfunktion (loss)** Funktion $L(D, f)$, wobei $f \in \mathcal{H}$, die misst, wie gut f in D abschneidet.

Trainieren von Modellen in ML

- **Datensatz** Erstellen Sie einen Datensatz und teilen Sie ihn in den Trainingssatz D und den Validierungssatz D' auf.
- **Modell auswählen** Modell $\mathcal{H}$: eine Menge von Entscheidungsfunktionen (z.B. Entscheidungsbäume der Tiefe ≤ 3 , logistische Regression, oder spezifische NN Architektur)
- **Verlustfunktion (loss)** Funktion $L(D, f)$, wobei $f \in \mathcal{H}$, die misst, wie gut f in D abschneidet.
- **Training** Finden einer Entscheidungsfunktion $f^* \in \mathcal{H}$, für die $L(D, f^*)$ klein ist.

Trainieren von Modellen in ML

- **Datensatz** Erstellen Sie einen Datensatz und teilen Sie ihn in den Trainingssatz D und den Validierungssatz D' auf.
- **Modell auswählen** Modell $\mathcal{H}$: eine Menge von Entscheidungsfunktionen (z.B. Entscheidungsbäume der Tiefe ≤ 3 , logistische Regression, oder spezifische NN Architektur)
- **Verlustfunktion (loss)** Funktion $L(D, f)$, wobei $f \in \mathcal{H}$, die misst, wie gut f in D abschneidet.
- **Training** Finden einer Entscheidungsfunktion $f^* \in \mathcal{H}$, für die $L(D, f^*)$ klein ist.
- **Validierung** Auswerten der Entscheidungsfunktion f^* an einem Dataset $D' \neq D$.

Initialisierung des Pseudo-Zufallszahlen-Generators

Zufälligkeit spielt beim maschinellen Lernen eine entscheidende Rolle.

- Wir haben Zufälligkeit bei der Datenerfassung, bei der Datenaufteilung (in Trainings-/Testsätze), bei den Trainingsmodellen usw.

Als Quelle der Zufälligkeit wird normalerweise ein

Pseudozufallszahlengenerator verwendet.

- Es handelt sich um eine mathematische Funktion, die eine Folge nahezu zufälliger Zahlen generiert.
- Die Sequenz ist deterministisch und wird mit einer (oder mehreren) Anfangszahl(en) initialisiert, die *seed* genannt wird.

In Code Expert legen wir den Ausgangspunkt fest und machen alles deterministisch. Sie erhalten reproduzierbare Ergebnisse.

Datensatz erstellen

Features oder Merkmale				Class label
Rot	Flackern	Gross	Hell	Stern
1	1	1	1	1
0	0	1	1	1
0	1	0	1	0
1	1	1	0	0
0	0	0	1	0
...	...	...	...	...
...	...	...	...	...

$D =$

Aufteilen des Datensatzes

- Den Datensatz in einen Trainingsdatensatz D und einen Validierungsdatensatz D' aufteilen.

R	F	G	H	S
1	1	1	1	1
0	0	1	1	1
0	1	0	1	0
1	1	1	0	0
0	0	0	1	0
1	0	1	0	0
0	1	1	1	1
0	1	1	0	0

$D =$

R	F	G	H	S
1	1	1	1	1
0	0	1	1	1
0	1	0	1	0
1	1	1	0	0
0	0	0	1	0

$D' =$

R	F	G	H	S
1	0	1	0	0
0	1	1	1	1
0	1	1	0	0

Modell auswählen

Modell $\mathcal{H}$: eine Menge von Entscheidungsfunktionen

Ein Beispiel eines Modells ist die Menge aller Bäume mit Tiefe 3

$$\mathcal{H} = \left\{ \begin{array}{c} \text{[Tree 1]} \quad \text{[Tree 2]} \quad \dots \quad \text{[Tree N]} \end{array} \right\}$$

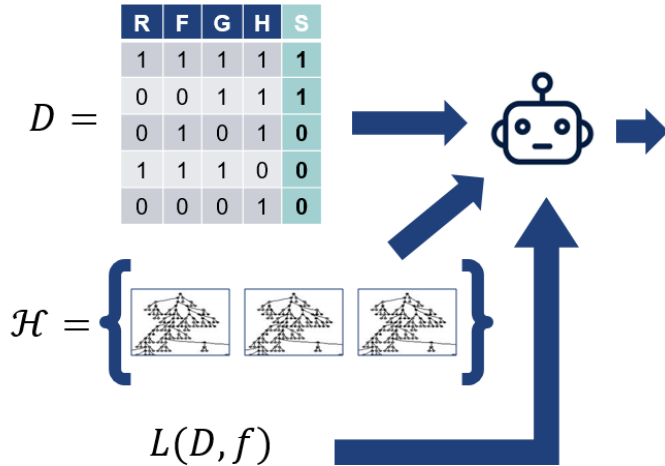
Verlustfunktion

- Funktion $L(D, f)$, wobei $f \in \mathcal{H}$, die misst, wie gut f in D abschneidet.
- Normalerweise benutzt man die 0/1 Verlustfunktion für das Trainieren von Bäumen.

$$L(D, f) = \frac{\text{\#Beispiele in } D, \text{ die von } f \text{ falsch klassifiziert wurden}}{\text{\#Beispiele in } D}$$

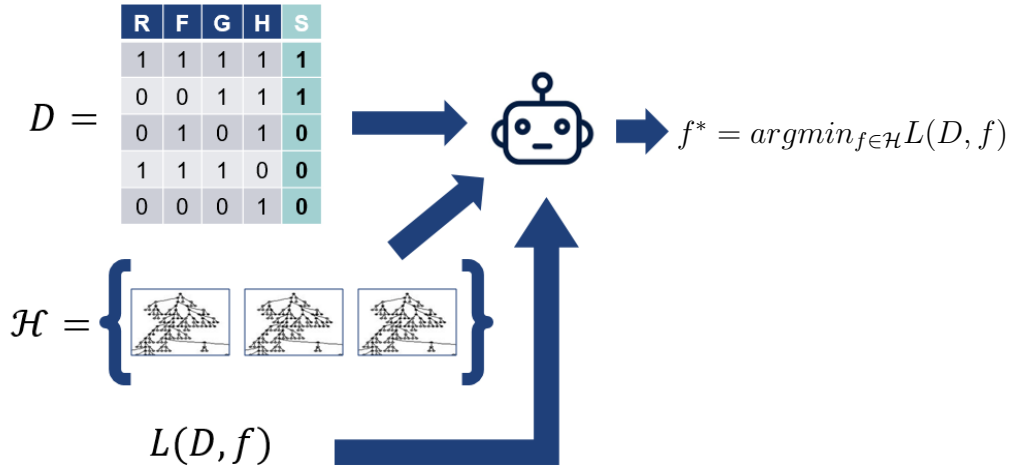
Trainieren

Finden Sie einen Schätzer $f^* \in \mathcal{H}$, für die der Wert von $L(D, f^*)$ niedrig ist.



Trainieren

Finden Sie einen Schätzer $f^* \in \mathcal{H}$, für die der Wert von $L(D, f^*)$ niedrig ist.



Validierung

- Der Schätzer f^* auf dem Validierungsdatensatz $D' \neq D$ auswerten.

R	F	G	H	S
1	1	1	1	1
0	0	1	1	1
0	1	0	1	0
1	1	1	0	0
0	0	0	1	0
1	0	1	0	0
0	1	1	1	1
0	1	1	0	0

$D =$

R	F	G	H	S
1	1	1	1	1
0	0	1	1	1
0	1	0	1	0
1	1	1	0	0
0	0	0	1	0

$D' =$

R	F	G	H	S
1	0	1	0	0
0	1	1	1	1
0	1	1	0	0

Verwechslungsmatrix

- Eine Verwirrungsmatrix ist eine Tabelle mit der Verteilung der Klassifikatorleistung auf die Daten.
- Es handelt sich um eine $N \times N$ -Matrix, die zur Bewertung der Leistung eines Klassifizierungsmodells verwendet wird.

		Predicted	
		Ungiftig	Giftig
True Label	Ungiftig	True Positive (TP)	False Positive (FP)
	Giftig	False Negative (FN)	True Negative (TN)

Balanced Accuracy

- Die "Balanced Accuracy" ist die durchschnittliche Genauigkeit aller Klassen.
- Sie kann als Durchschnitt der Diagonalen der normalisierten Verwirrungsmatrix berechnet werden.
- Dies ist hilfreich beim Umgang mit unausgeglichene Daten.

		Predicted	
		Ungiftig	Giftig
True Label	Ungiftig	20	70
	Giftig	30	5000

Ausgewogene Genauigkeit

		Predicted	
		Ungiftig	Giftig
True Label	Ungiftig	20	70
	Giftig	30	5000

- Berechnen wir die Genauigkeit dieser Vorhersage:

$$\frac{TP+TN}{TP+FN+FP+TN} \approx 98,05\%.$$

- Dieses Ergebnis ist beeindruckend, allerdings wird die Ungiftig-Spalte in der Vorhersage selbst nicht richtig behandelt.
- Verwenden Sie die "Balanced Accuracy": $\frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}) \approx 60,80\%$
- Dadurch ist die Punktzahl niedriger als von der Genauigkeit vorhergesagt, da beide Klassen das gleiche Gewicht erhalten.

Lineare Regression

Regression: Finden einer Beziehung (Funktion) zwischen einer abhängigen und unabhängigen Variable.

Lineare Regression: Finden einer Linearen Abbildung, die den Vektor $\mathbf{x}$ von unabhängigen Variablen auf eine abhängige Variable y abbildet.

Lineare Regression

Regression: Finden einer Beziehung (Funktion) zwischen einer abhängigen und unabhängigen Variable.

Lineare Regression: Finden einer Linearen Abbildung, die den Vektor $\mathbf{x}$ von unabhängigen Variablen auf eine abhängige Variable y abbildet.

- **Datensatz:** Jeder Datenpunkt besteht aus einem Vektor $\mathbf{x}$ und einem Skalar y .

Lineare Regression

Regression: Finden einer Beziehung (Funktion) zwischen einer abhängigen und unabhängigen Variable.

Lineare Regression: Finden einer Linearen Abbildung, die den Vektor $\mathbf{x}$ von unabhängigen Variablen auf eine abhängige Variable y abbildet.

- **Datensatz:** Jeder Datenpunkt besteht aus einem Vektor $\mathbf{x}$ und einem Skalar y .
- **Modell:** Die Menge der linearen Abbildungen $\mathbb{R}^n \rightarrow \mathbb{R}$. Wir finden den Vektor $\mathbf{w}$: $\mathbf{w} \cdot \mathbf{x} = y$.

Lineare Regression

Regression: Finden einer Beziehung (Funktion) zwischen einer abhängigen und unabhängigen Variable.

Lineare Regression: Finden einer Linearen Abbildung, die den Vektor $\mathbf{x}$ von unabhängigen Variablen auf eine abhängige Variable y abbildet.

- **Datensatz:** Jeder Datenpunkt besteht aus einem Vektor $\mathbf{x}$ und einem Skalar y .
- **Modell:** Die Menge der linearen Abbildungen $\mathbb{R}^n \rightarrow \mathbb{R}$. Wir finden den Vektor $\mathbf{w}$: $\mathbf{w} \cdot \mathbf{x} = y$.
- **Verlustfunktion:** Die Summe der Quadrate: $\sum_{\mathbf{x} \in D} (\mathbf{w} \cdot \mathbf{x} - y)^2$

Lineare Regression

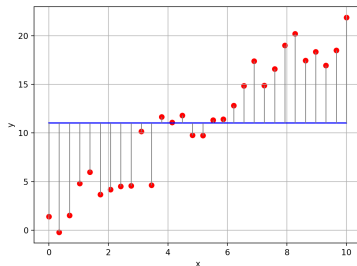
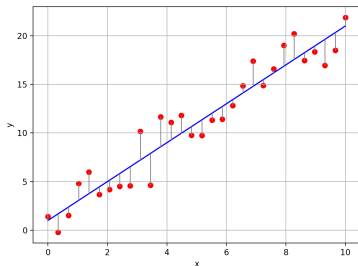
Regression: Finden einer Beziehung (Funktion) zwischen einer abhängigen und unabhängigen Variable.

Lineare Regression: Finden einer Linearen Abbildung, die den Vektor $\mathbf{x}$ von unabhängigen Variablen auf eine abhängige Variable y abbildet.

- **Datensatz:** Jeder Datenpunkt besteht aus einem Vektor $\mathbf{x}$ und einem Skalar y .
- **Modell:** Die Menge der linearen Abbildungen $\mathbb{R}^n \rightarrow \mathbb{R}$. Wir finden den Vektor $\mathbf{w}$: $\mathbf{w} \cdot \mathbf{x} = y$.
- **Verlustfunktion:** Die Summe der Quadrate: $\sum_{\mathbf{x} \in D} (\mathbf{w} \cdot \mathbf{x} - y)^2$
- **Trainieren:** Mit der Methode der kleinsten Quadrate (nicht klausurrelevant).
- **Validierung:** Messen der Summe der Quadrate auf dem Validierungsdatensatz D' .

R2-Score

- Wie gut ist ein Schätzer verglichen mit dem Mittelwertschätzer?



$$R^2 = 1 - \frac{MSE(D, \text{Schätzer})}{MSE(D, \text{Mittelwertschätzer})} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}$$

- $0 < R^2 < 1$: Besser als Mittelwertschätzer
- $-\infty < R^2 < 0$: Schlechter als Mittelwertschätzer

Gini Index

Wie unrein sind die Partitionen eines Entscheidungsbaums?

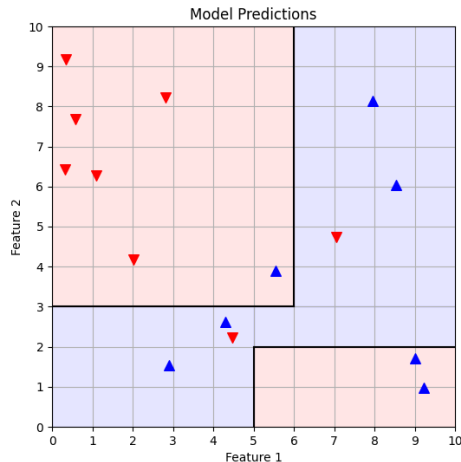
■ Einzelne Partition:

$$G(D, \text{Part}_m) = \sum_k \hat{p}_{m,k} \cdot (1 - \hat{p}_{m,k})$$

$\hat{p}_{m,k}$: Anteil der Klasse k in Part_m .

■ Insgesamt:

$$G(D, f) = \sum_m \frac{|\text{Part}_m|}{|\text{Part}|} \cdot G(D, \text{Part}_m)$$

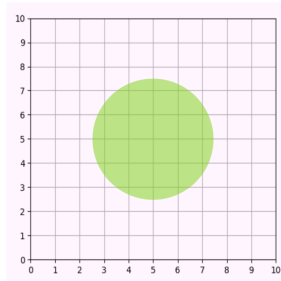


2. Theoretische Übungen

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:



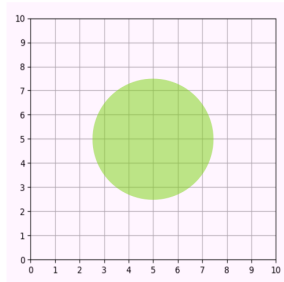
Das Innere sollte als
1 klassifiziert werden,
das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)



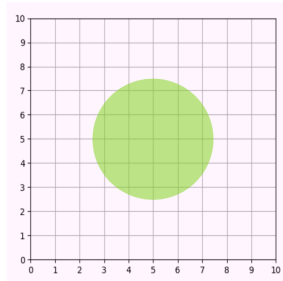
Das Innere sollte als 1 klassifiziert werden, das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)
2. Wählen Sie ein Modell $\mathcal{H}$ aus.



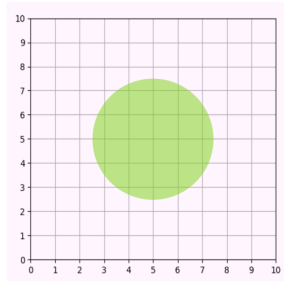
Das Innere sollte als 1 klassifiziert werden, das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)
2. Wählen Sie ein Modell $\mathcal{H}$ aus.
3. Wählen Sie eine Verlustfunktion $L(D, f)$ aus.



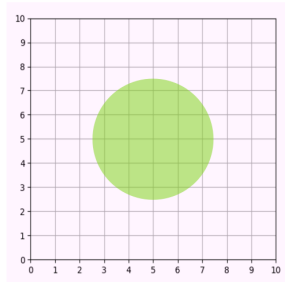
Das Innere sollte als 1 klassifiziert werden, das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)
2. Wählen Sie ein Modell $\mathcal{H}$ aus.
3. Wählen Sie eine Verlustfunktion $L(D, f)$ aus.
4. Erstellen Sie einen Validierungsdatensatz D' mit ca. 5 Datenpunkten.



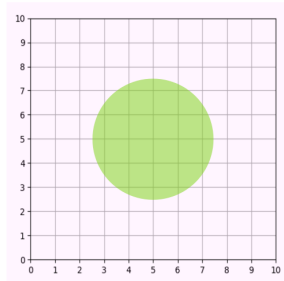
Das Innere sollte als 1 klassifiziert werden, das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)
2. Wählen Sie ein Modell $\mathcal{H}$ aus.
3. Wählen Sie eine Verlustfunktion $L(D, f)$ aus.
4. Erstellen Sie einen Validierungsdatensatz D' mit ca. 5 Datenpunkten.
5. Finden Sie einen Schätzer f^* , der $L(D, f)$ minimiert.



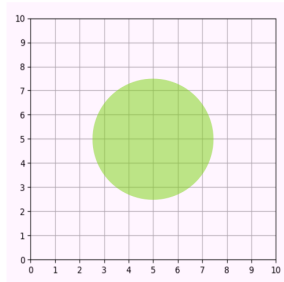
Das Innere sollte als 1 klassifiziert werden, das Äussere als 0.

Übung 1: Training Reproduzieren

Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben sei der Kreis auf dem Bild.

Führen Sie die folgenden Schritte aus:

1. Erstellen Sie einen Trainingsdatensatz D mit ca. 10 Datenpunkten. (Geben Sie die x- und y-Koordinaten und die Klasse an.)
2. Wählen Sie ein Modell $\mathcal{H}$ aus.
3. Wählen Sie eine Verlustfunktion $L(D, f)$ aus.
4. Erstellen Sie einen Validierungsdatensatz D' mit ca. 5 Datenpunkten.
5. Finden Sie einen Schätzer f^* , der $L(D, f)$ minimiert.
6. Validieren Sie den Schätzer f^* an D' .



Das Innere sollte als 1 klassifiziert werden,
das Äussere als 0.

Beispiel Antwort 1

Beispiel Antwort 1

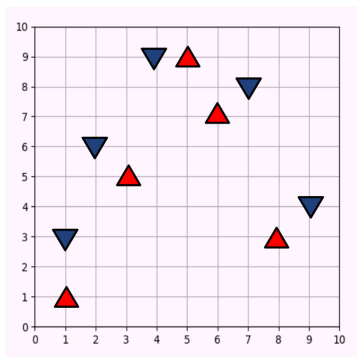
Übung 2: Training

Machen Sie diese Übung nur mit Bleistift und Papier.

Gegeben sei der Trainingsdatensatz D für ein dreieckiges Muster im Bild.

$D =$

x-Koord	y-Koord	Klasse
1	1	1
3	5	1
5	9	1
6	7	1
8	3	1
1	3	0
2	6	0
4	9	0
7	8	0
9	4	0



Finden Sie einen Baum mit der Tiefe 3, der alle Punkte im obigen Datensatz richtig klassifiziert.

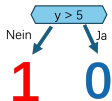
Beispiel Antwort 2

Übung 3: Validierung

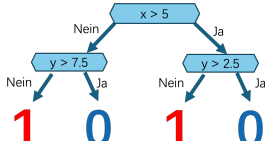
Machen Sie diese Übung nur mit Bleistift und Papier.
Gegeben seien drei Entscheidungsbäume T1, T2, T3 mit Tiefe 1, 2, und 6, und ein Validierungsdatensatz D' , der dem Muster im Bild entspricht.

- Berechnen Sie für jeden Baum, wie viele Punkte in D' durch den Baum inkorrekt klassifiziert werden.
- Entscheiden Sie, welcher Baum der Beste ist.

T1:

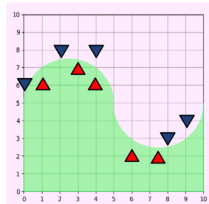


T2:

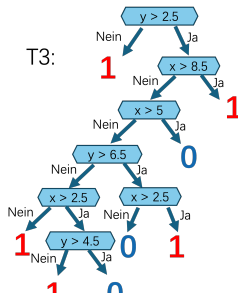


$D' =$

x-Koord	y-Koord	Klasse
1	6	1
3	7	1
4	6	1
6	2	1
7.5	2	1
0	6	0
2	8	0
4	8	0
8	3	0
9	4	0



T3:



Antwort 3

3. Praktische Übungen

Übersicht

Code Expert:

<https://expert.ethz.ch/enrolled/SS25/mavt2/codeExamples>

Exercise 12 - In-class

Zwei Programmierübungen:

1. **Iris**: Klassifikation mit Entscheidungsbäumen
2. **House prices**: Regression mit der Linearen Regression

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie einen Entscheidungsbaum für die Klassifikation der Arten von Iris aus den Messungen.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie einen Entscheidungsbaum für die Klassifikation der Arten von Iris aus den Messungen.
4. Verwenden Sie den Baum, um die Blumen im Testdatensatz zu klassifizieren.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie einen Entscheidungsbaum für die Klassifikation der Arten von Iris aus den Messungen.
4. Verwenden Sie den Baum, um die Blumen im Testdatensatz zu klassifizieren.
5. Lesen Sie X_final.csv mit pandas ein, verwenden Sie den Baum um die Blumen zu klassifizieren, und geben Sie die Vorhersagen zurück.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie einen Entscheidungsbaum für die Klassifikation der Arten von Iris aus den Messungen.
4. Verwenden Sie den Baum, um die Blumen im Testdatensatz zu klassifizieren.
5. Lesen Sie X_final.csv mit pandas ein, verwenden Sie den Baum um die Blumen zu klassifizieren, und geben Sie die Vorhersagen zurück.
6. Die Vorhersagen werden von Code Expert automatisch bewertet.

Übung 1: Iris (Klassifikation)

Iris Datensatz: klassischer Datensatz aus dem Jahr 1936. Er enthält Messungen von Blütenblättern (petal) und Kelchblättern (sepal) von 150 Blüten, die zu einer von 3 Arten von Iris (*Iris setosa*, *Iris versicolor*, *Iris virginica*) gehören.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz data.csv mit pandas ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie einen Entscheidungsbaum für die Klassifikation der Arten von Iris aus den Messungen.
4. Verwenden Sie den Baum, um die Blumen im Testdatensatz zu klassifizieren.
5. Lesen Sie X_final.csv mit pandas ein, verwenden Sie den Baum um die Blumen zu klassifizieren, und geben Sie die Vorhersagen zurück.
6. Die Vorhersagen werden von Code Expert automatisch bewertet.

Detaillierte Anweisungen befinden sich auf Code Expert.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie ein lineares Regressionsmodell mit dem Trainingsdatensatz.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie ein lineares Regressionsmodell mit dem Trainingsdatensatz.
4. Verwenden Sie das Modell, um die Hauspreise vorherzusagen. Bewerten Sie die Qualität der Vorhersage mit dem R^2 -Wert.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie ein lineares Regressionsmodell mit dem Trainingsdatensatz.
4. Verwenden Sie das Modell, um die Hauspreise vorherzusagen. Bewerten Sie die Qualität der Vorhersage mit dem R^2 -Wert.
5. Lesen Sie `X_final.csv` mit `pandas` ein, verwenden Sie das Modell, um die Preise vorherzusagen und geben Sie die Vorhersagen zurück.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie ein lineares Regressionsmodell mit dem Trainingsdatensatz.
4. Verwenden Sie das Modell, um die Hauspreise vorherzusagen. Bewerten Sie die Qualität der Vorhersage mit dem R^2 -Wert.
5. Lesen Sie `X_final.csv` mit `pandas` ein, verwenden Sie das Modell, um die Preise vorherzusagen und geben Sie die Vorhersagen zurück.
6. Die Vorhersagen werden von Code Expert automatisch bewertet.

Übung 2: House prices (Regression)

In dieser Übung trainieren wir eine Funktion mithilfe eines fiktiven Datensatzes, die den Preis eines Hauses anhand von seiner Fläche schätzen kann.

Machen Sie die folgenden Übungen in Code Expert:

1. Lesen Sie den Datensatz `data.csv` mit `pandas` ein.
2. Teilen Sie den Datensatz in einen Trainingsdatensatz und einen Testdatensatz auf.
3. Trainieren Sie ein lineares Regressionsmodell mit dem Trainingsdatensatz.
4. Verwenden Sie das Modell, um die Hauspreise vorherzusagen. Bewerten Sie die Qualität der Vorhersage mit dem R^2 -Wert.
5. Lesen Sie `X_final.csv` mit `pandas` ein, verwenden Sie das Modell, um die Preise vorherzusagen und geben Sie die Vorhersagen zurück.
6. Die Vorhersagen werden von Code Expert automatisch bewertet.

Detaillierte Anweisungen befinden sich auf Code Expert.

4. Homework

Übungslektion 10: Intro ML I

On <https://expert.ethz.ch/enrolled/SS25/mavt2/exercises>

- Cancer Detection
- Diabetes Prediction
- Circles

Abgabedatum: Montag 19.05.2025, 20:00 MEZ Das Grading funktioniert wie bei den praktischen Übungen.

NO HARDCODING

Fragen oder Anregungen?