

Machine Learning I

Developer Meeting

Agenda

Theorie (40')

Einführung in Machine Learning
Lineare Regression
Logistische Regression
Klassifizierung

Praxis (10')

Gradientenverfahren
Klassifikation

Fragen (10')

**Wer hat schon Erfahrungen
mit Machine Learning gemacht?**



Einführung in Machine Learning

Machine Learning bietet eine neue Möglichkeit, Probleme zu lösen und komplexe Fragen zu beantworten. Machine Learning kann das Wetter vorhersagen, Fahrtzeiten schätzen, Songs automatisch vervollständigen, Artikel zusammenfassen und bisher nicht gesehene Bilder identifizieren,...

Im Grunde ist Machine Learning der Prozess des Trainings einer Software, die als Modell bezeichnet wird, um nützliche Vorhersagen zu treffen oder Inhalte aus Daten zu generieren.

4 Typen von Machine Learning

1

Supervised
learning

Das Modell lernt aus gelabelten Daten, um Vorhersagen für neue, unbekannte Daten zu treffen.

2

Unsupervised
learning

Das Modell entdeckt Muster oder Strukturen in ungelabelten Daten ohne menschliche Anleitung.

3

Reinforcement
learning

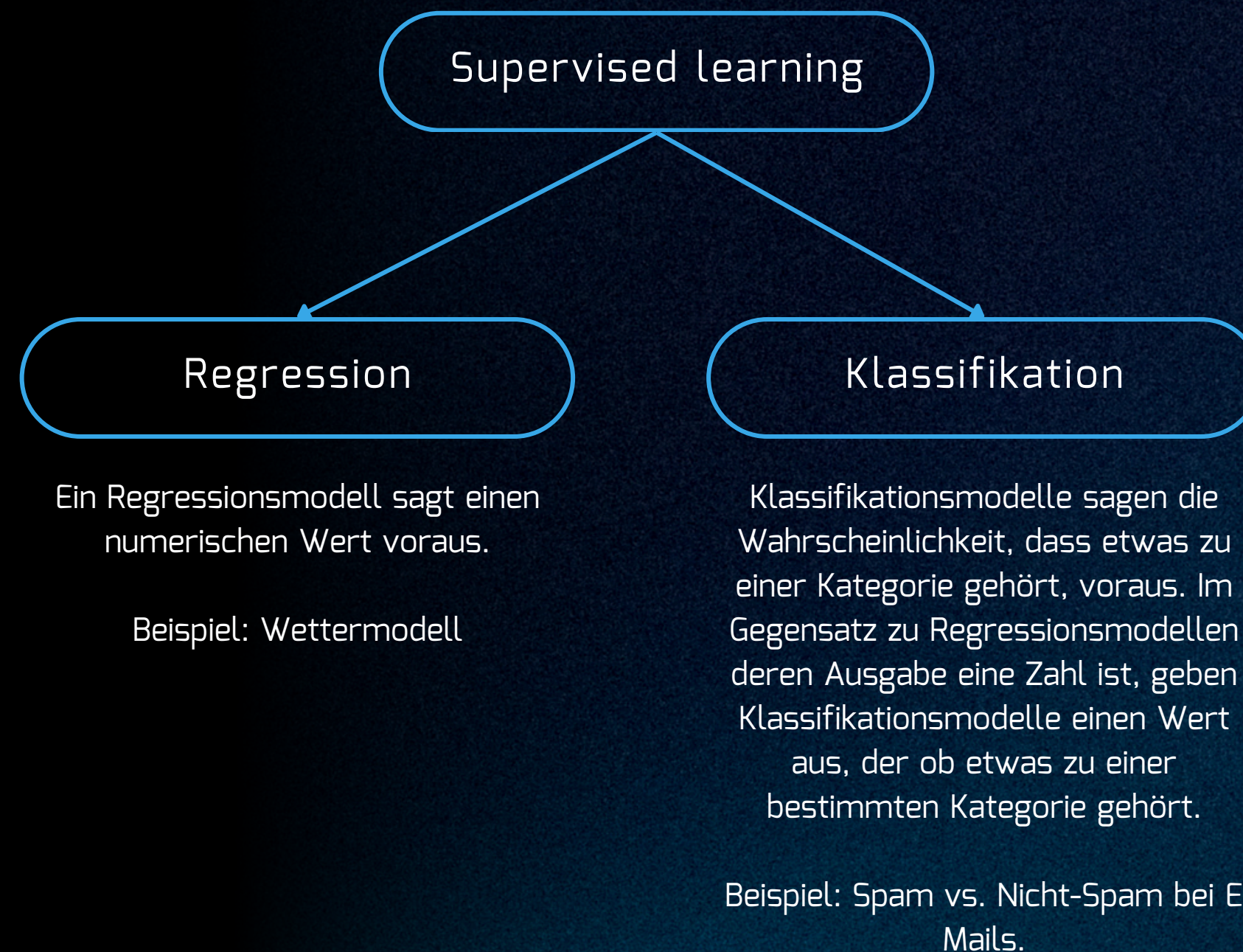
Das Modell lernt durch Belohnung und Bestrafung, wie es optimale Entscheidungen in einer Umgebung treffen kann.

4

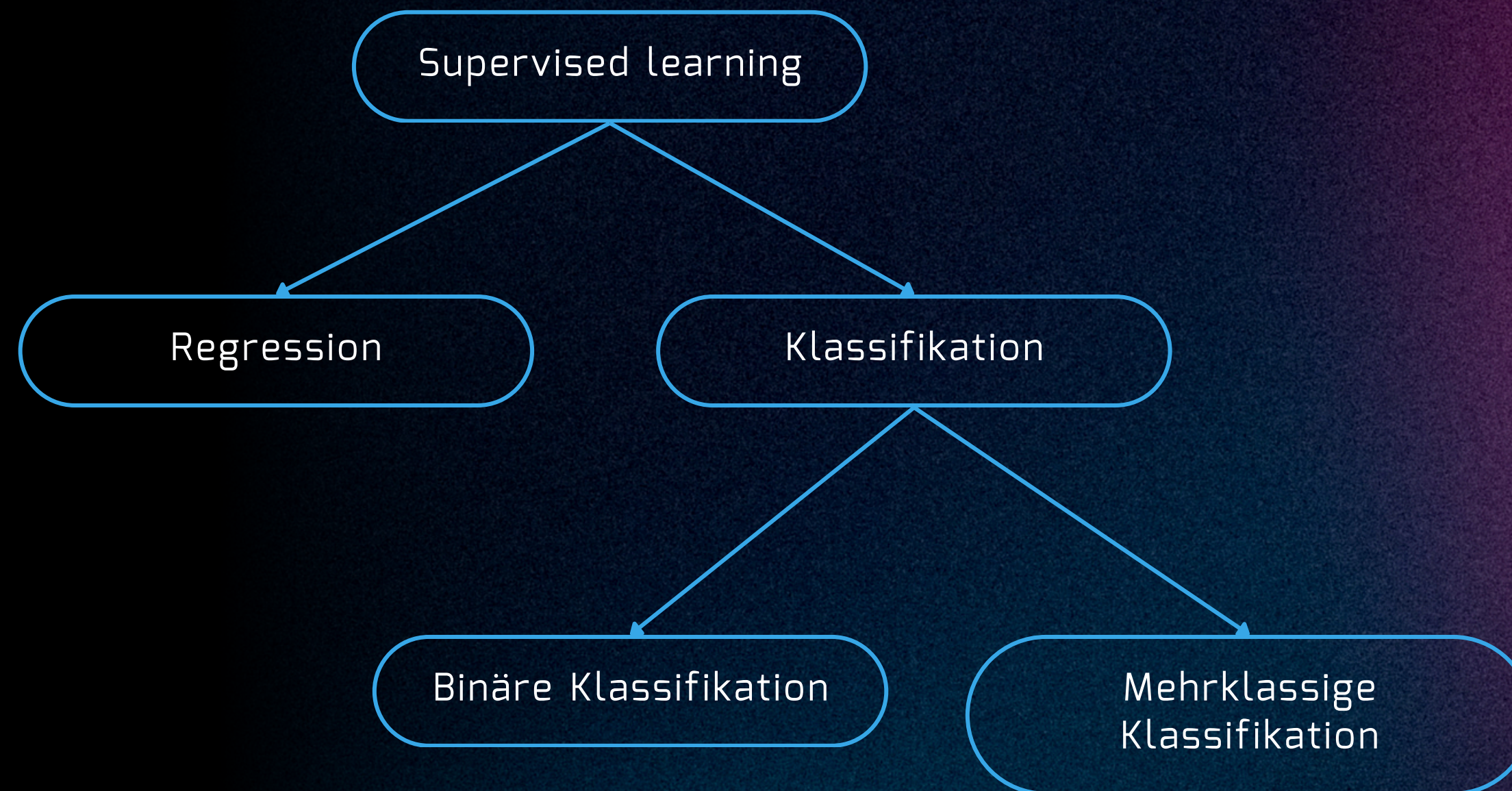
Generative AI

Das Modell erstellt neue Inhalte, wie Bilder, Texte oder Musik, basierend auf zuvor gelernten Daten.

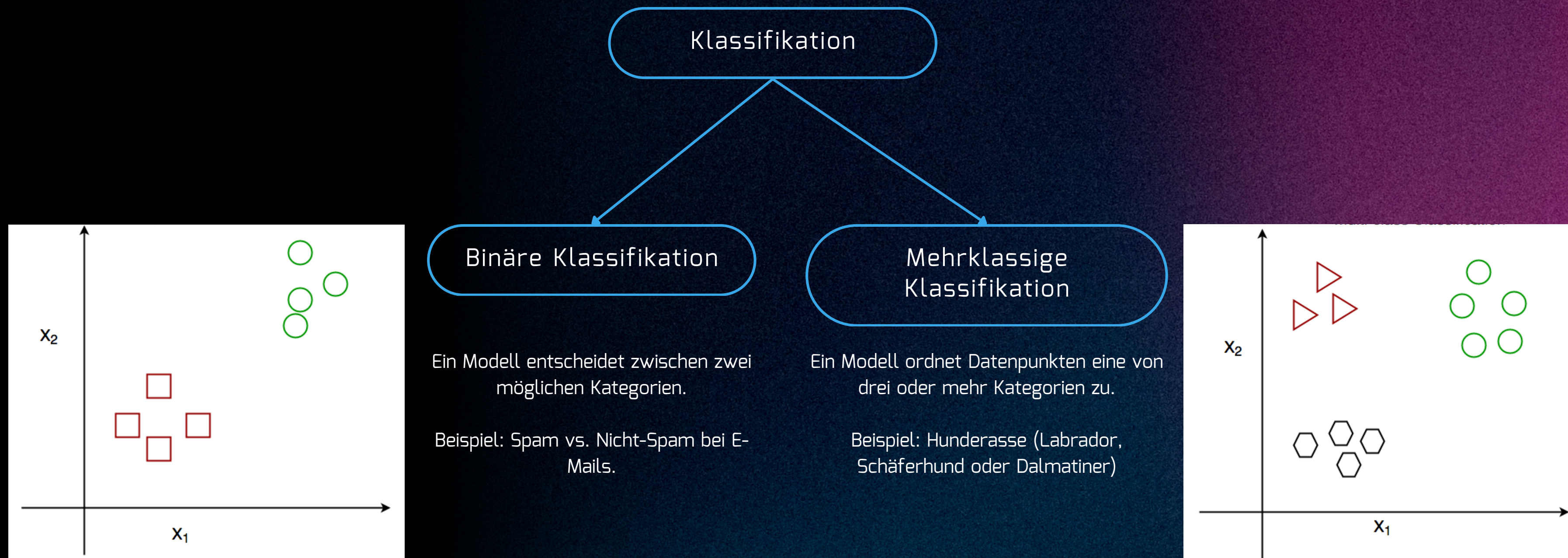
Supervised learning



Supervised learning



Supervised learning



Kernkonzepte

Das **supervised learning** basiert auf den folgenden **Kernkonzepten**:

Daten

Datensätze bestehen aus den **Features** und den **Labels**. Features sind die Werte, die ein Modell verwendet, um das Label vorherzusagen. Das Label ist die „Antwort“, also der Wert, den das Modell vorhersagen soll.

Features								Label
date	lat	long	temp	humidity	cloud_coverage	wind_direction	atmp_pressure	rainfall
2021-09-09	49.71N	82.16W	74	20	3	N	18.6	.01
2021-09-09	32.71N	117.16W	82	42	6	SW	29.94	.23

Example

Gute Daten haben eine grosse Grösse und hohe Vielfalt.

Kernkonzepte

Das **supervised learning** basiert auf den folgenden **Kernkonzepten**:

Daten

Modell

Ein Modell ist eine komplexe Sammlung von Zahlen, die die mathematische Beziehung zwischen bestimmten Eingabemustern und bestimmten Ausgabewerten definieren. Das Modell findet diese Muster durch Training heraus.

Kernkonzepte

Das **supervised learning** basiert auf den folgenden **Kernkonzepten**:

Daten

Modell

Training

Ein überwachtes Modell wird mit gelabelten Daten trainiert, um die Beziehung zwischen Merkmalen und Labels zu lernen. Es vergleicht seine Vorhersagen mit den tatsächlichen Werten, passt sich anhand des Fehlers (Verlust) an und verbessert so schrittweise seine Genauigkeit.

Beispiel: Wenn das Modell 1,15 Zoll Regen vorhersagt, der tatsächliche Wert jedoch 0,75 Zoll beträgt, passt das Modell seine Lösung an, damit die Vorhersage näher an 0,75 Zoll liegt. Nachdem das Modell alle Beispiele im Datensatz durchlaufen hat – oft sogar mehrmals –, findet es eine Lösung, die im Durchschnitt die besten Vorhersagen für die Beispiele liefert.

Kernkonzepte

Das **supervised learning** basiert auf den folgenden **Kernkonzepten**:

Daten

Modell

Training

Evaluierung

Das trainierte Modell wird bewertet, um festzustellen, wie gut es gelernt hat.

Bei der Bewertung eines Modells werden diesem nur die Features des Datensatzes geben. Anschliessend wird verglichen wie die Vorhersagen des Modells mit den wahren Werten entsprechen.

Kernkonzepte

Das **supervised learning** basiert auf den folgenden **Kernkonzepten**:

Daten

Modell

Training

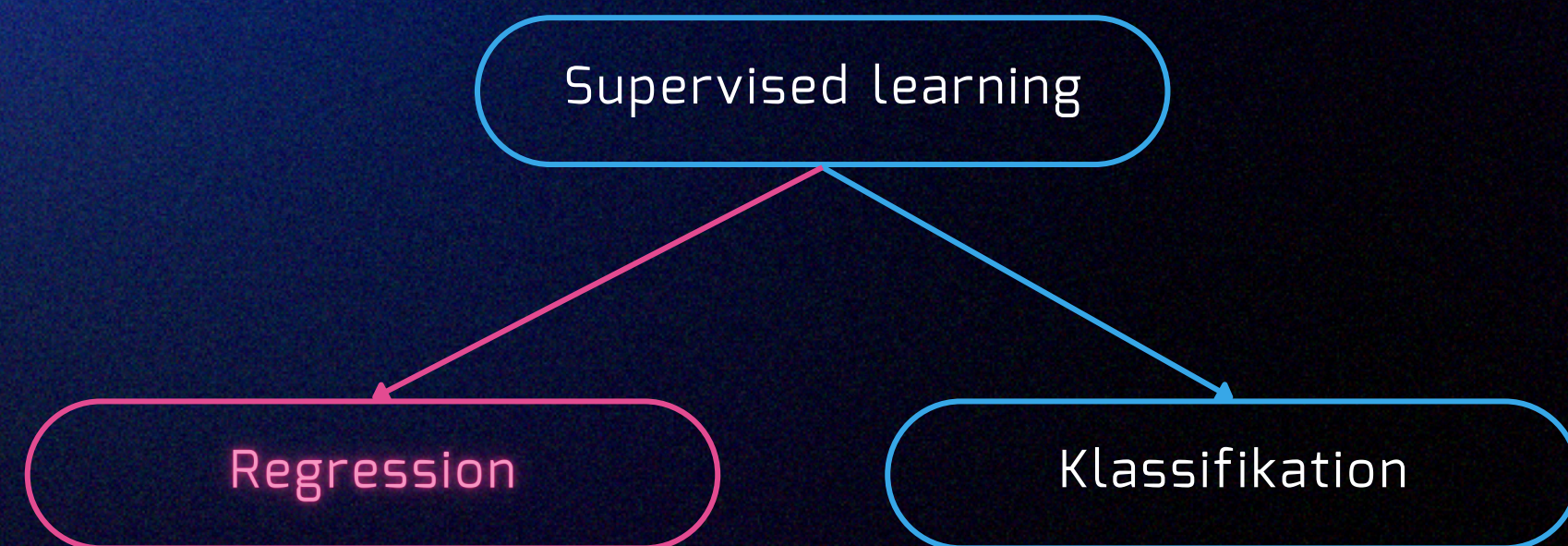
Evaluierung

Interferenz

Wenn wir mit den Ergebnissen der Auswertung des Modells zufrieden sind, können wir das Modell verwenden, um Vorhersagen, so genannte Inferenzen, für nicht beschriftete Beispiele zu treffen.

Im Beispiel der Wetter-App würden wir dem Modell die aktuellen Wetterbedingungen wie Temperatur, Luftdruck und relative Luftfeuchtigkeit vorgeben, und es würde die Menge des Niederschlags vorhersagen.

Lineare Regression

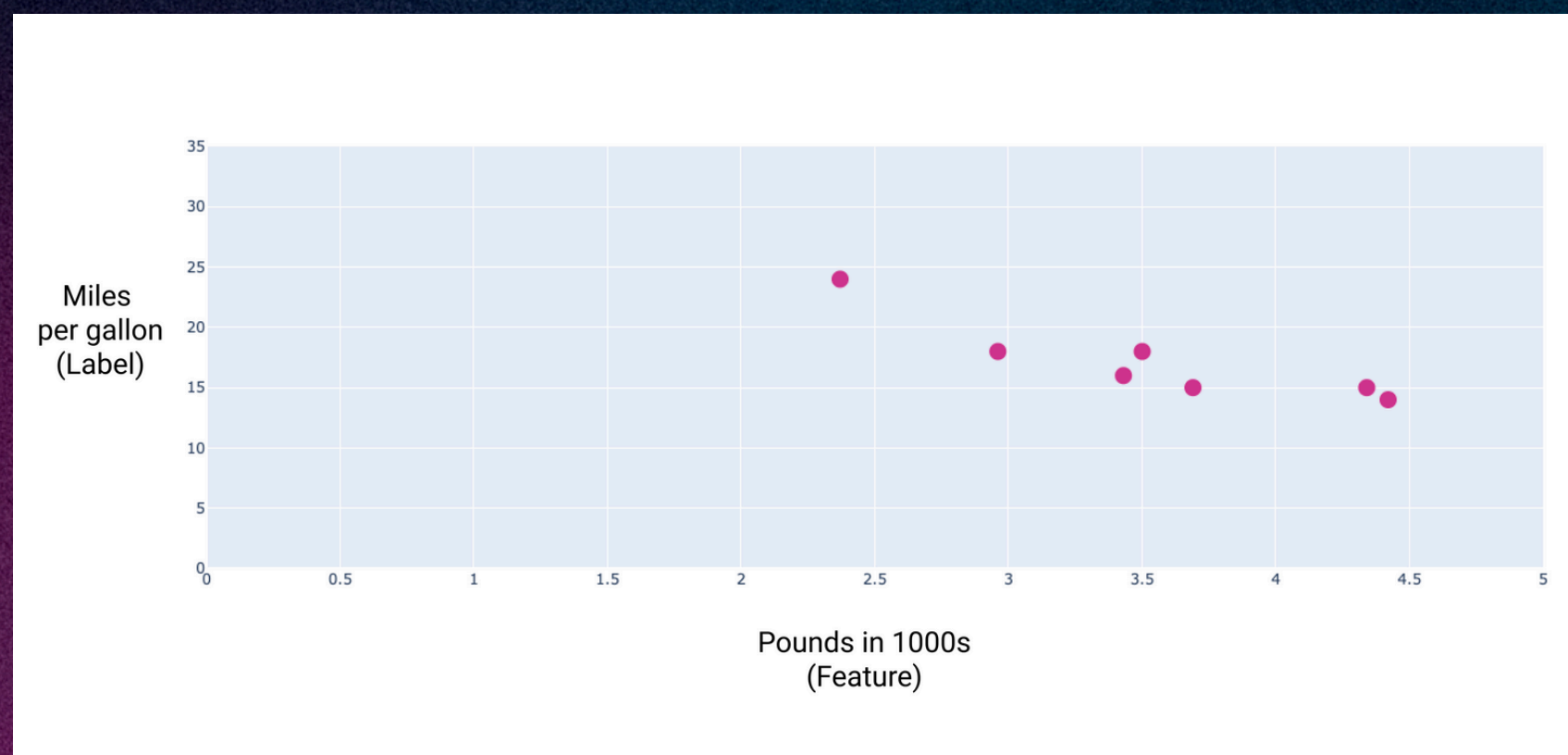


Lineare Regression

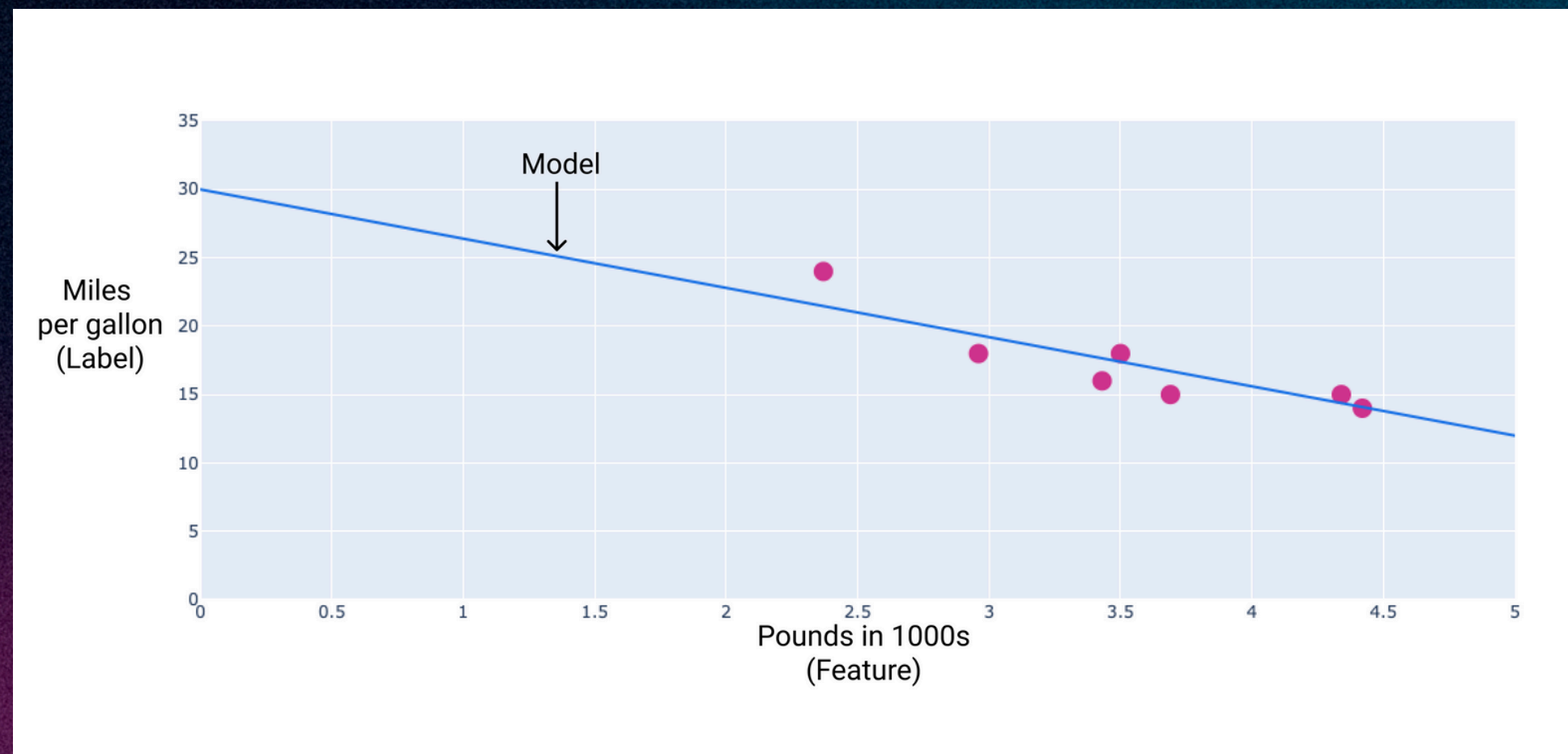
Lineare Regression ist ein statistisches Verfahren, mit dem die Beziehung zwischen den Variablen (Features und Label) ermittelt wird.

Angenommen, wir möchten den Kraftstoffverbrauch eines Autos in Meilen pro Gallone basierend auf dem Gewicht des Autos, und wir haben folgendes Dataset:

Pfund (Feature)	Meilen pro Gallone (Label)
3,5	18
3,69	15
3,44	18
3,43	16
4,34	15
4,42	14
2,37	24



Lineare Regression



In der linearen Algebra:

$$y = mx + b$$

$$y = b + mx$$

y steht für Meilen pro Gallone (der Wert, den wir vorhersagen möchten)

m ist die Steigung der Linie

x sind Pfund (unser Eingabewert)

b ist der y -Achsenabschnitt

Lineares Regressionsmodell (in ML):

$$y' = b + w_1 x_1$$

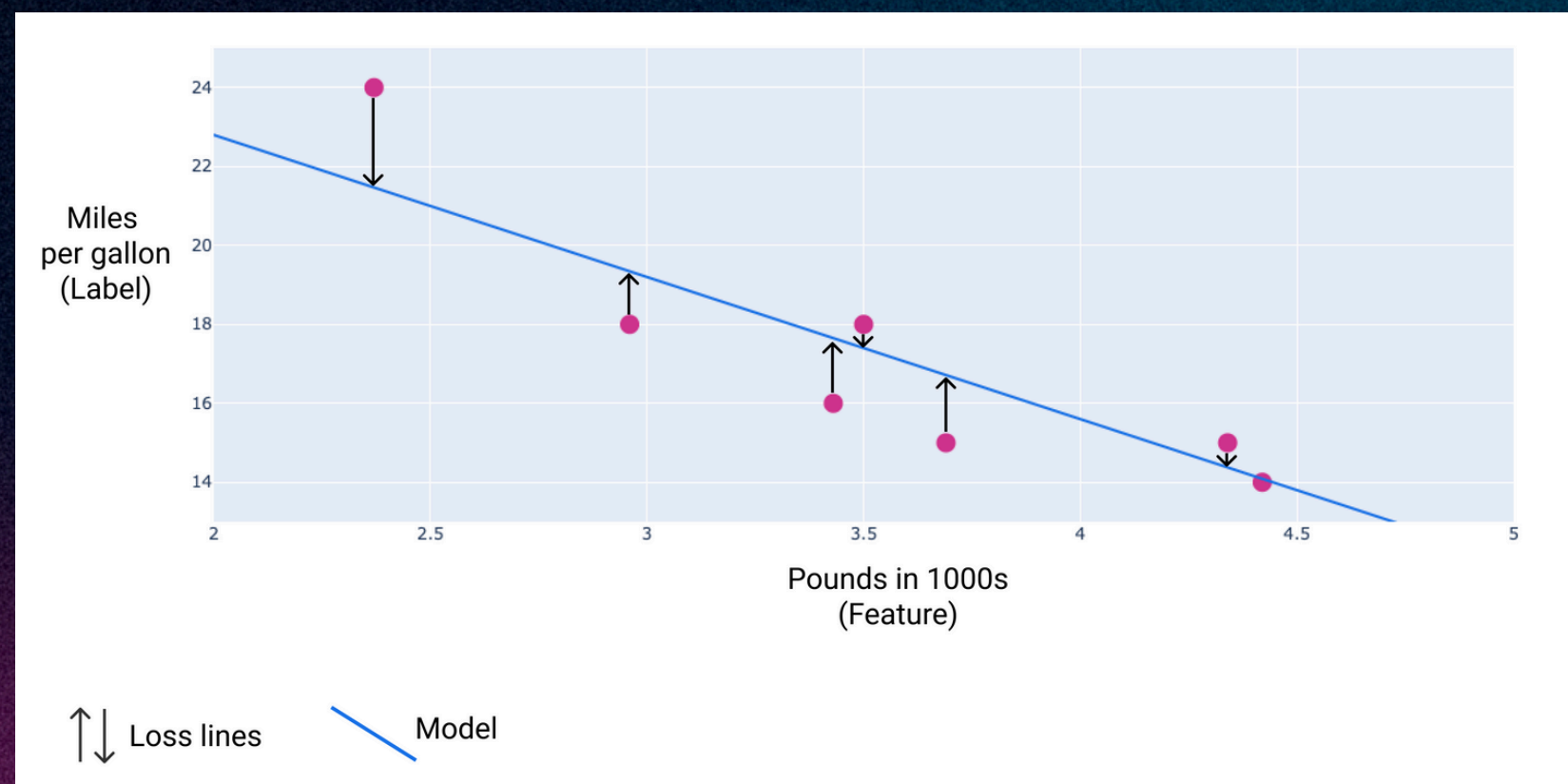
y' ist das vorhergesagte Label (die Ausgabe)

b ist der Bias (während Training berechnet)

w_1 ist die Gewichtung (während Training berechnet)

x_1 ist ein Feature Wert

Lineare Regression - Verlust



Der Verlust ist ein numerischer Messwert, der beschreibt, wie falsch die Vorhersagen eines Modells sind.

Der Verlust misst die Entfernung zwischen den Vorhersagen des Modells und den tatsächlichen Labels. Ziel des Modelltrainings ist es, den Verlust zu minimieren und auf den niedrigstmöglichen Wert zu reduzieren.

Bei der linearen Regression gibt es vier Haupttypen von Verlusten:

- L1-Verlust: $\sum |\text{tatsächlicher Wert} - \text{prognostizierter Wert}|$
- Mean absolute error (MAE): $1/N \sum |\text{tatsächlicher Wert} - \text{vorhergesagter Wert}|$
- L2-Verlust: $\sum (\text{tatsächlicher Wert} - \text{vorhergesagter Wert})^2$
- Mean squared error (MSE): $1/N \sum (\text{tatsächlicher Wert} - \text{vorhergesagter Wert})^2$

Um die Auswahl der besten Verlustfunktion zu treffen wird geschaut, wie das Modell Ausreißer behandeln soll. Beispielsweise verschiebt der MSE das Modell stärker in Richtung der Ausreißer, während dies beim MAE nicht der Fall ist. Ein L2-Verlust hat für einen Ausreißer einen viel höheren Abzug als ein L1-Verlust.

Optimierungsverfahren

Wir wollen das Minimum von der Verlustfunktion finden. Hierfür gibt es verschiedene Möglichkeiten:

- Gradientenverfahren
- Newton-Verfahren
- ...

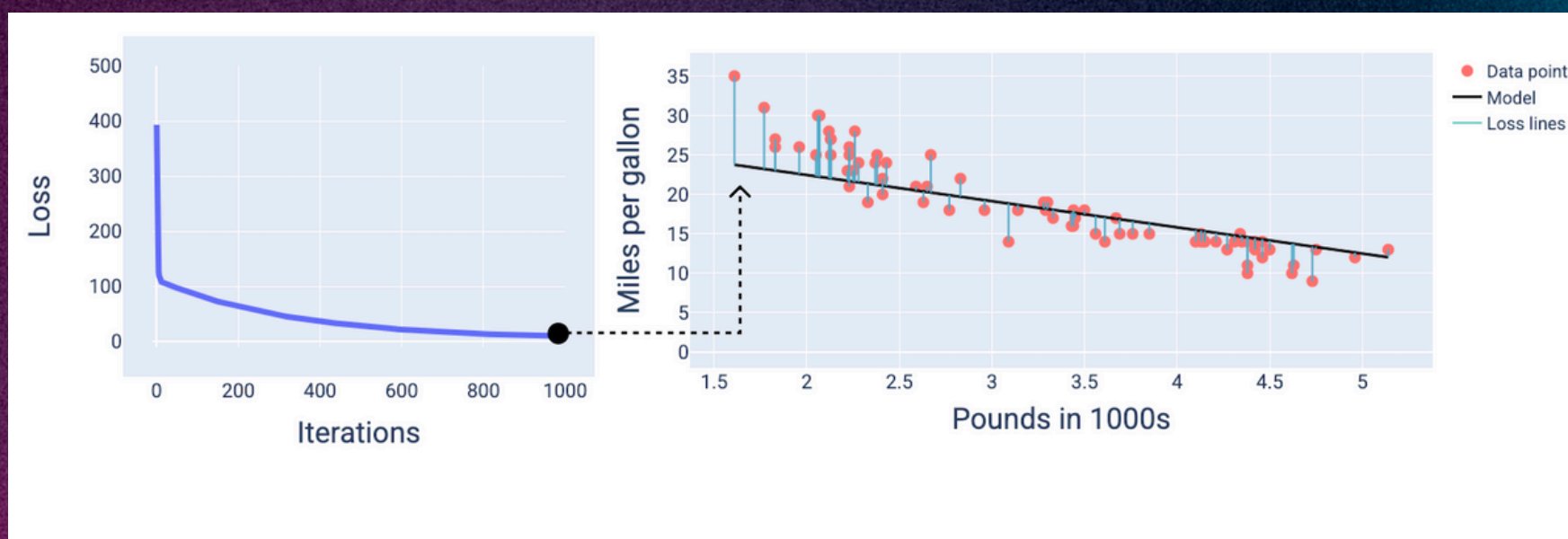
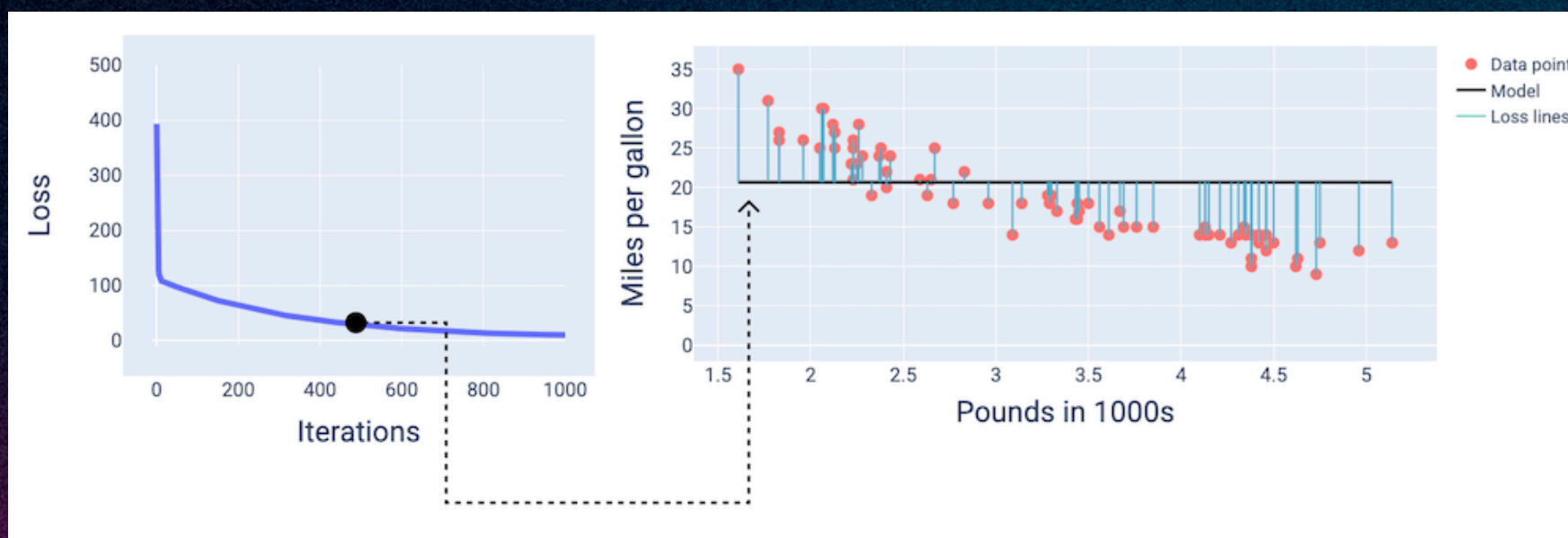
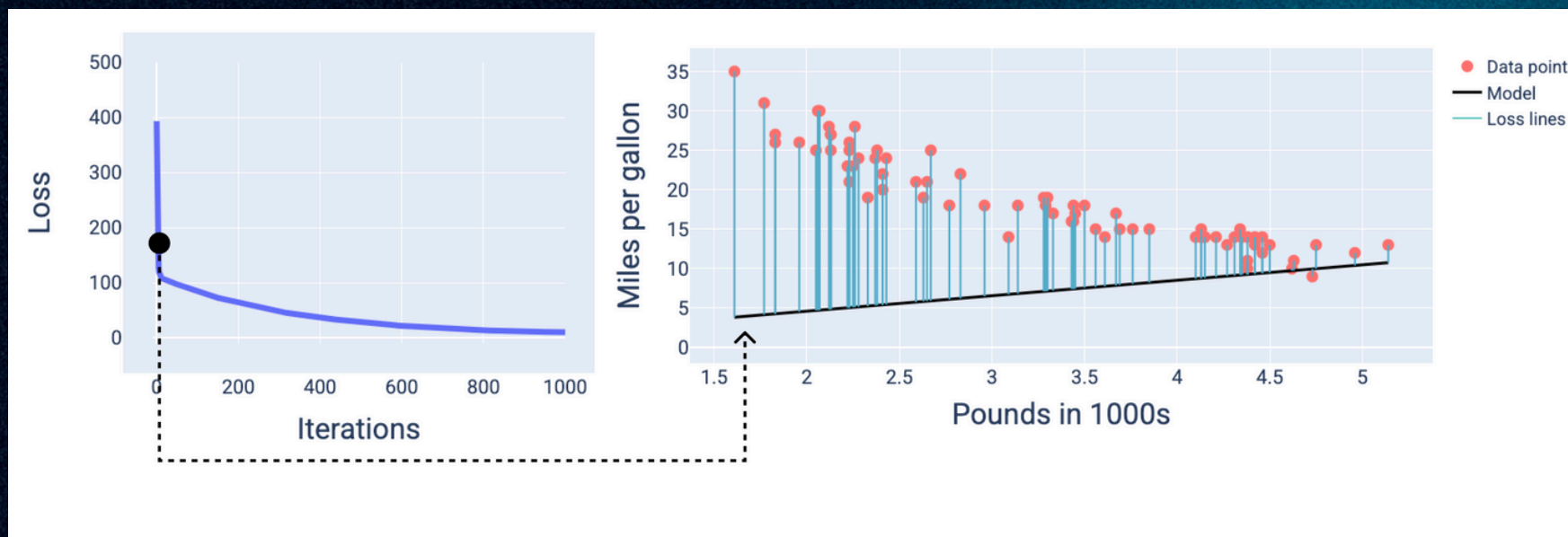
Gradientenverfahren

Das Gradientenverfahren ist eine mathematische Methode, mit der iterativ die Gewichtungen und “Verzerrungen” ermittelt werden, die das Modell mit dem geringsten Verlust erzeugen.

Der Gradientenabstieg ermittelt das beste Gewicht und der beste Bias, indem der folgende Prozess für eine Reihe von benutzerdefinierten Iterationen wiederholt wird.

Das Modell beginnt das Training mit zufälligen Werten für Gewichtungen und Bias nahe Null und wiederholt dann die folgenden Schritte:

1. **Berechnen des Verlusts** basierend auf den aktuellen Gewichtungen und dem Bias.
2. **Bestimmen der Richtung**, in die Gewichtungen und Bias angepasst werden müssen, um den Verlust zu minimieren.
3. **Anpassen der Gewichtungs- und Bias-Werte** um einen kleinen Betrag in die Richtung, die den Verlust reduziert.
4. **Wiederholen des Vorgangs**, bis keine weitere signifikante Reduktion des Verlusts mehr möglich ist.

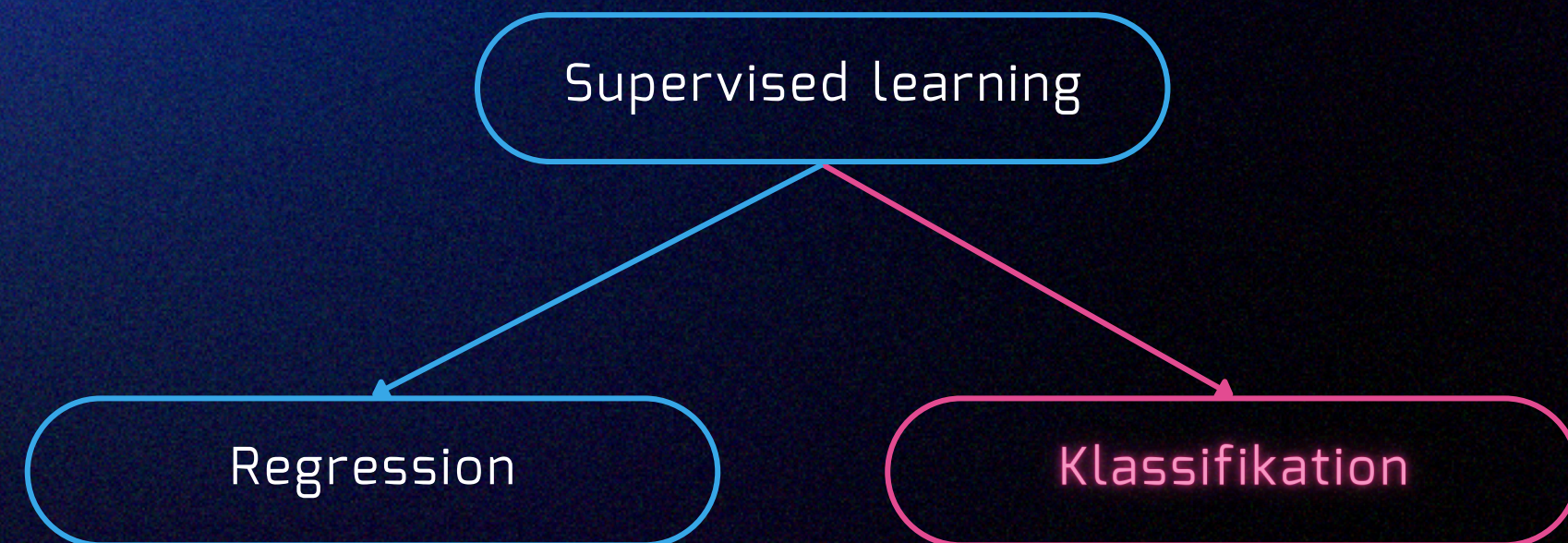


Hier sehen wir, dass der Gradientenabstieg das Gewicht und den Bias gefunden hat, die ein besseres Modell ergeben.

$$y' = 35.94 - 5.44 x$$



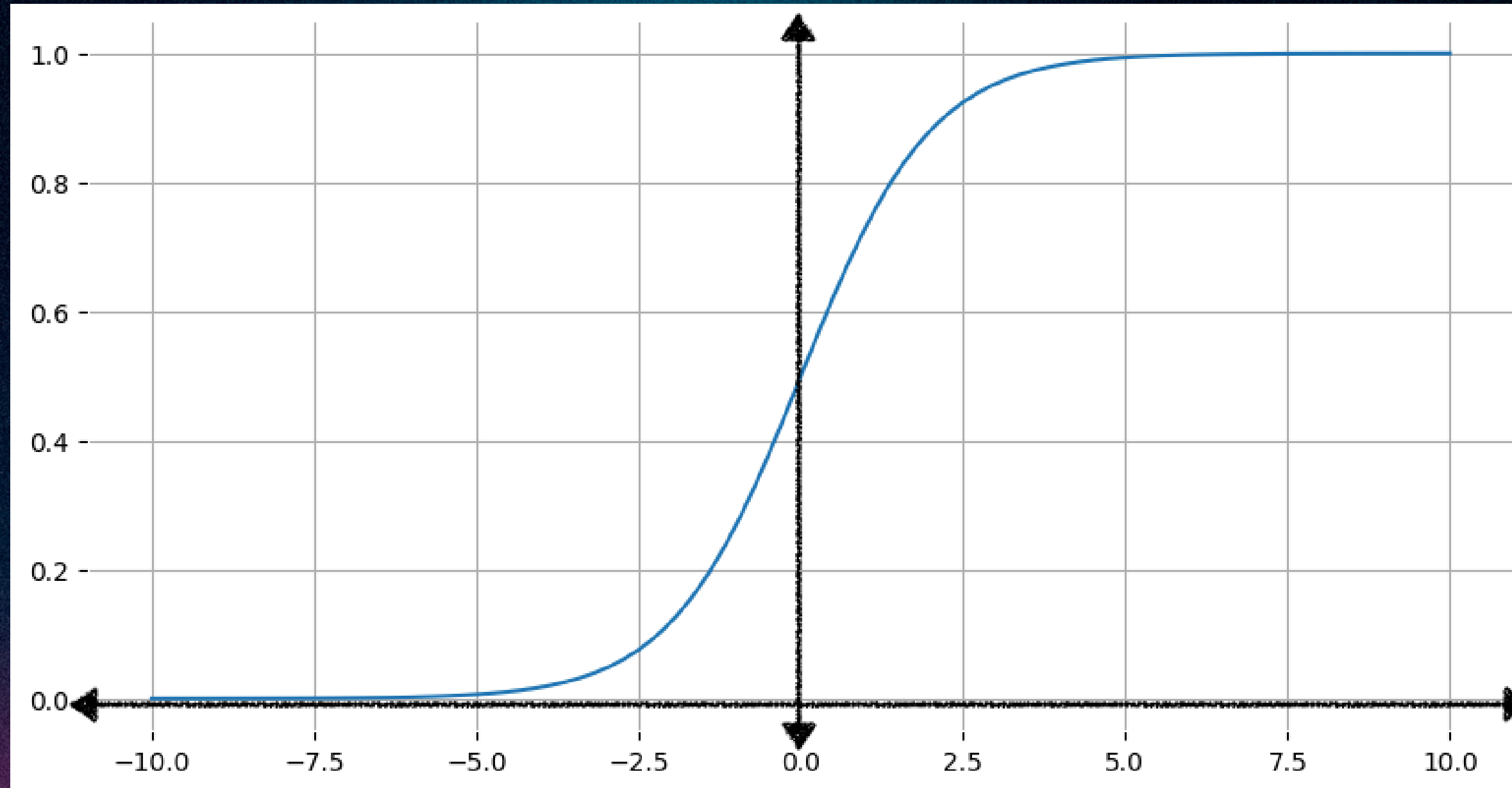
Logistische Regression



Logistische Regression

Viele Probleme erfordern eine Schätzung von Wahrscheinlichkeiten als Ergebnis. Die logistische Regression bietet einen äußerst effizienten Mechanismus zur Berechnung von Wahrscheinlichkeiten. Die zurückgegebene Wahrscheinlichkeit kann auf zwei Arten genutzt werden:

1. **Direkte Verwendung der Wahrscheinlichkeit:** Zum Beispiel gibt ein Modell zur Spam-Erkennung bei einer E-Mail einen Wert von 0,932 zurück, was einer Wahrscheinlichkeit von 93,2 % entspricht, dass es sich um Spam handelt.
2. **Umwandlung in eine binäre Kategorie:** Die Wahrscheinlichkeit wird in eine Klasse wie "Wahr/Falsch" oder "Spam/Nicht-Spam" übersetzt. (Nächstes Kapitel Klassifizierung)



Sigmoid Funktion

$$f(x) = \frac{1}{1 + e^{-x}}$$

Binäre Klassifikation

Softmax Funktion

$$f(z_i) = \frac{e^{z_i}}{\sum_{m=1}^K e^{z_m}}$$

Mehrklassige Klassifikation

Verlustfunktion

Schauen wir uns die Verlustfunktion für die logistische Regression **Log Loss** an für eine binäre Klassifikation. Die Klassifizierung in unserem Fall wäre für 0 oder 1. Die Log Loss-Gleichung liefert den Logarithmus der Größe der Veränderung und nicht nur den Abstand zwischen Daten und Vorhersage.

Log Loss wird wie folgt berechnet:

$$\text{Log Loss} = \sum_{(x,y) \in D} -y \log(y') - (1-y) \log(1-y')$$

wobei:

y ist der wahre Wert (0 oder 1)

y' ist die Vorhersage des Modells (irgendwo zwischen 0 und 1)

Klassifizierung

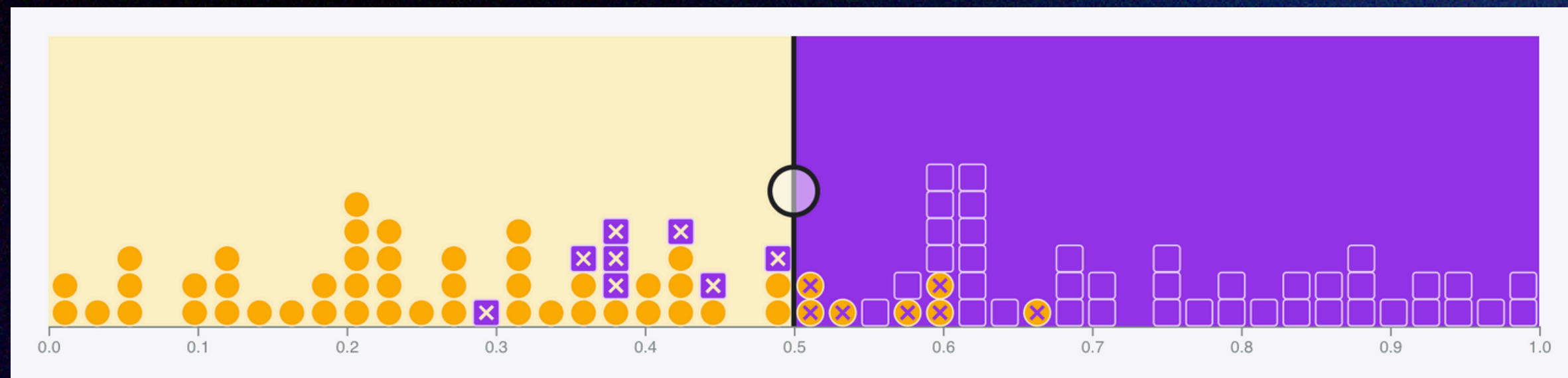
Output als
Wahrscheinlichkeit



Output als Klasse

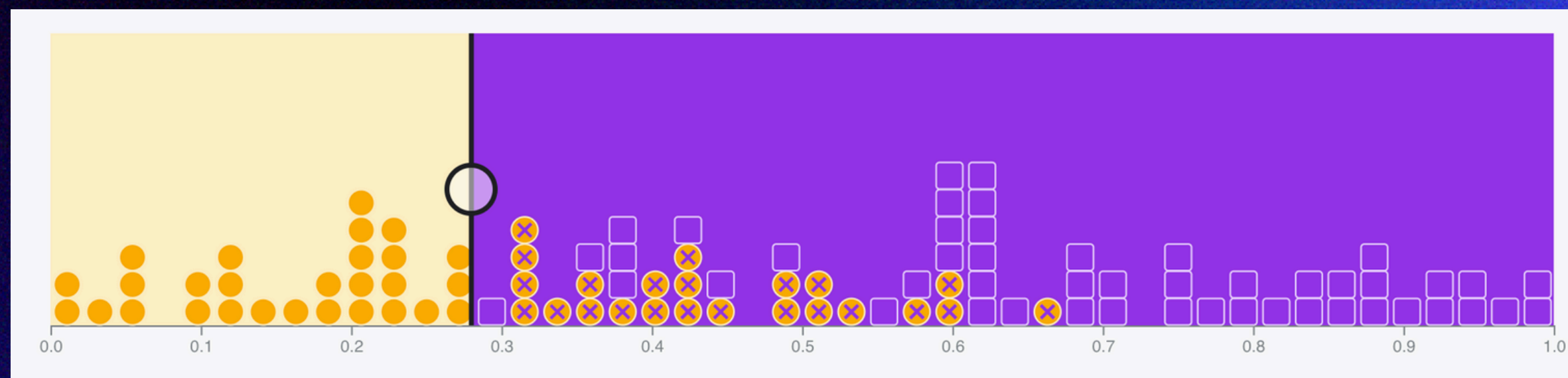
Schwellenwert

Verwirrungsmatrix



Schwellenwert: 0.5

	Actually positive	Actually negative
Predicted positive	TP=40	FP=7
Predicted negative	FN=8	TN=44



Schwellenwert: 0.28

	Actually positive	Actually negative
Predicted positive	TP=48	FP=23
Predicted negative	FN=0	TN=28

Nützliche Metriken

Verwirrungsmatrix

	y = positive	y = negative
y' = positive	TP	FP
y' = negative	FN	TN

1. Die **Genauigkeit** (Accuracy) ist der Anteil aller richtigen Klassifizierungen, unabhängig davon, ob sie positiv oder negativ waren.

$$\text{Accuracy} = \frac{\text{Korrekte Klassifizierungen}}{\text{Totale Klassifizierungen}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

2. Die **True-Positive-Rate** (TPR), d.h. der Anteil aller tatsächlich positiven Ergebnisse, die korrekt als positiv eingestuft wurden, wird auch als Recall bezeichnet.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. Die **False-Positive-Rate** (FPR) ist der Anteil aller tatsächlichen Negativbefunde, die fälschlicherweise als Positivbefunde eingestuft wurden, auch bekannt als Fehlalarmwahrscheinlichkeit.

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

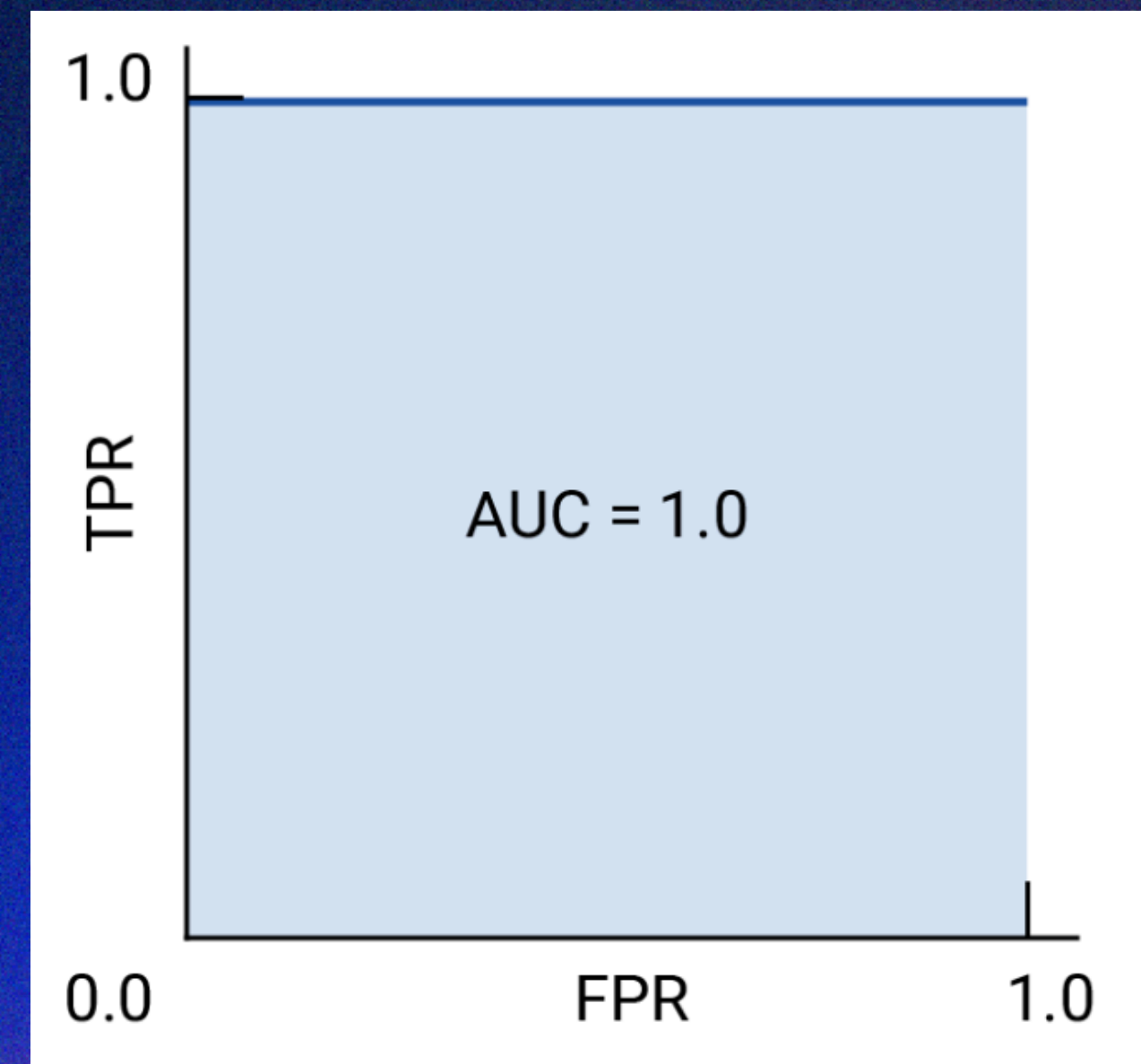
ROC und AUC

Die ROC-Kurve ist eine visuelle Darstellung der Modelleistung über alle Schwellenwerte hinweg.

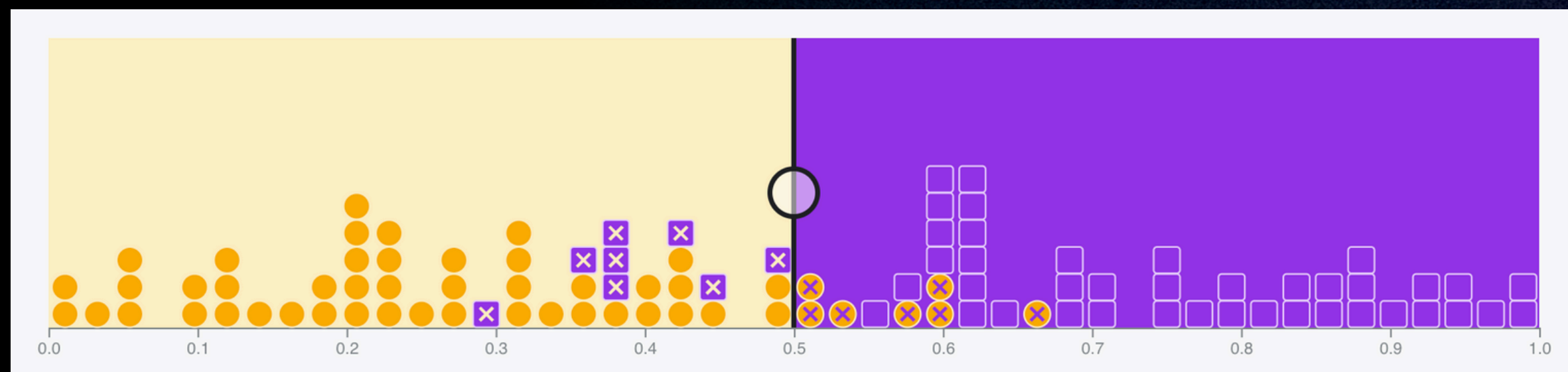
Die ROC-Kurve wird gezeichnet, indem die Rate der echten Positiven (TPR) und die Falsch-Positiven-Rate (FPR) bei jedem möglichen Schwellenwert (in der Praxis bei ausgewählten Intervallen) berechnet und dann die TPR über der FPR grafisch dargestellt wird.

Ein perfektes Modell, das bei einem bestimmten Schwellenwert eine TPR von 1,0 und eine FPR von 0,0 aufweist, kann entweder durch einen Punkt bei (0, 1) dargestellt werden, wenn alle anderen Schwellenwerte ignoriert werden, oder durch die folgende Kurve.

ROC = Receiver-operating characteristic curve
AUC = Area under curve



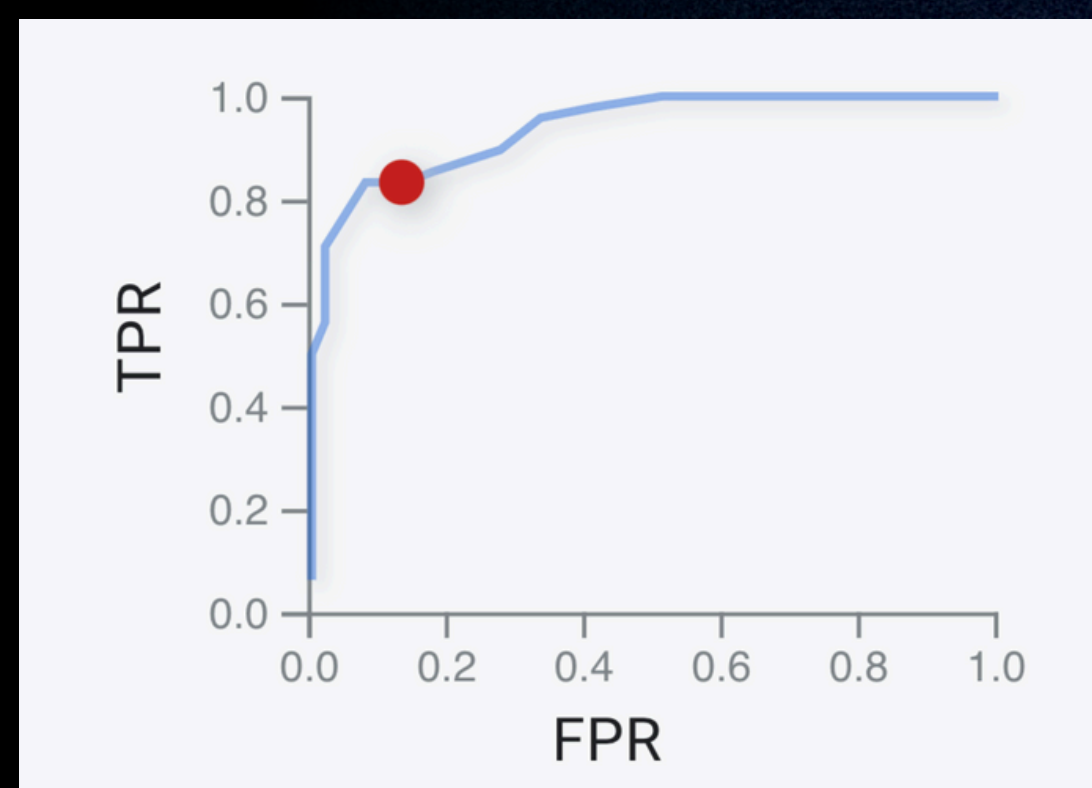
Beispiel



Schwellenwert: 0.5

	Actually positive	Actually negative
Predicted positive	 TP=40	 FP=7
Predicted negative	 FN=8	 TN=44

Accuracy: 0.85
 TPR: 0.83
 FPR: 0.14



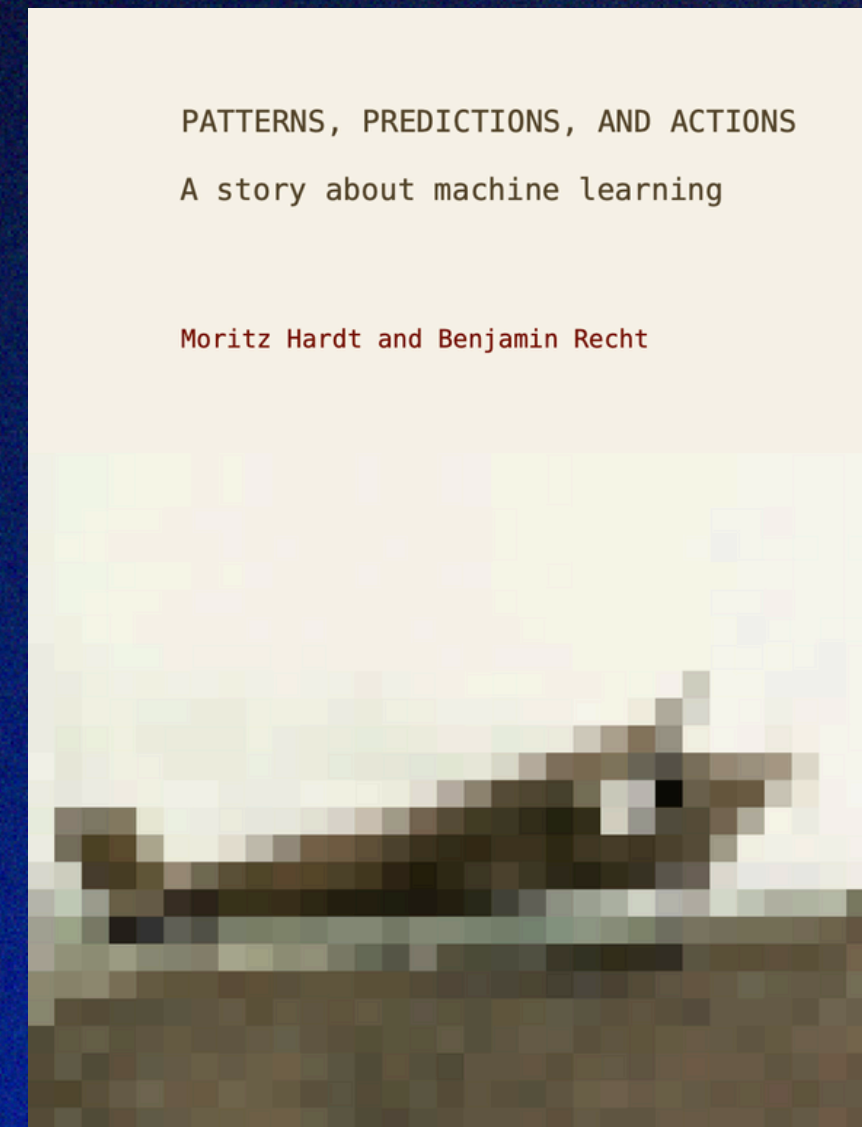
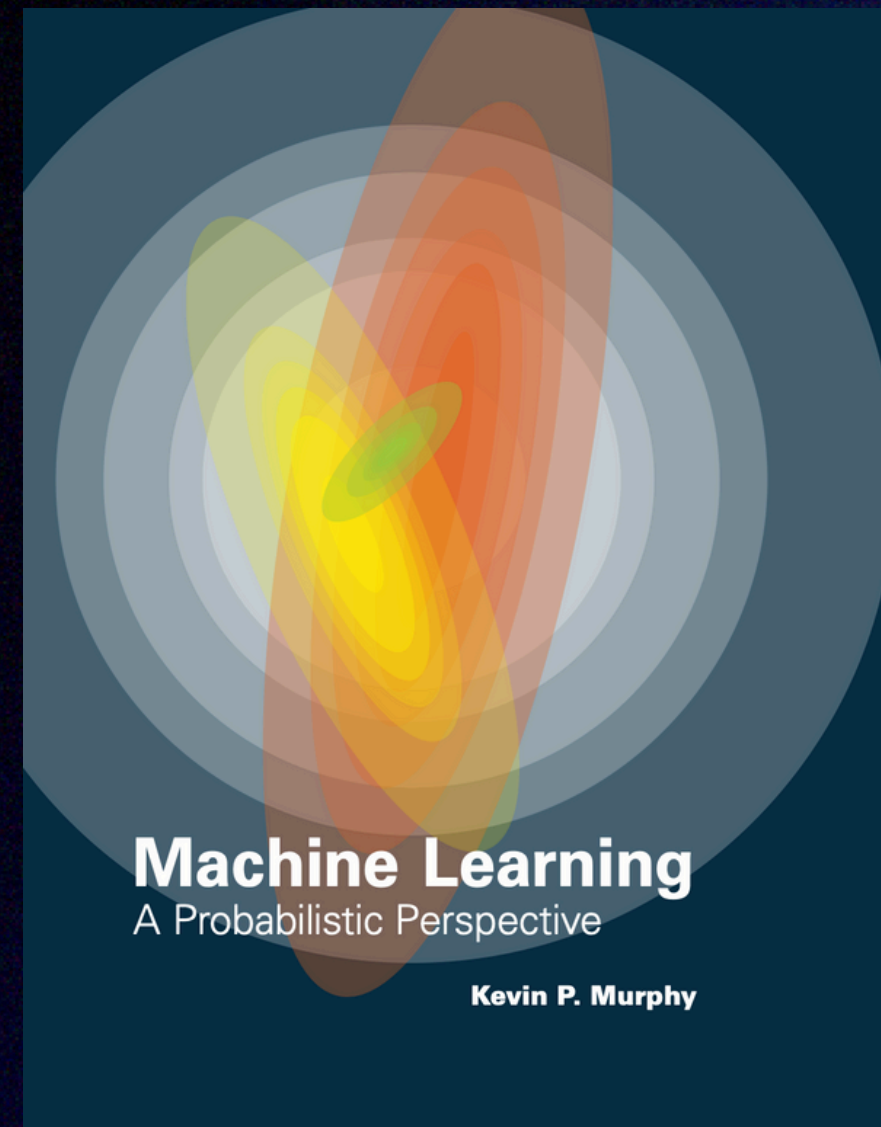
Praxis

Ausblick: Machine Learning II mit Damian 🎉

Themen

- Supervised learning
 - Neural Networks
- Unsupervised learning
 - Clustering
- Generative AI

Literatur



Confluence → iRIXEngineering → Knowledge Base → Machine Learning

Fragen?