

Support Vector Machines



Gliederung

- Was ist eine SVM?
- Was ist eine Hyperebene?
- Trainingsphase
- Kernel-Trick
- Klassifizieren unbekannter Datenpunkte
- Quellen

Was ist eine SVM ?

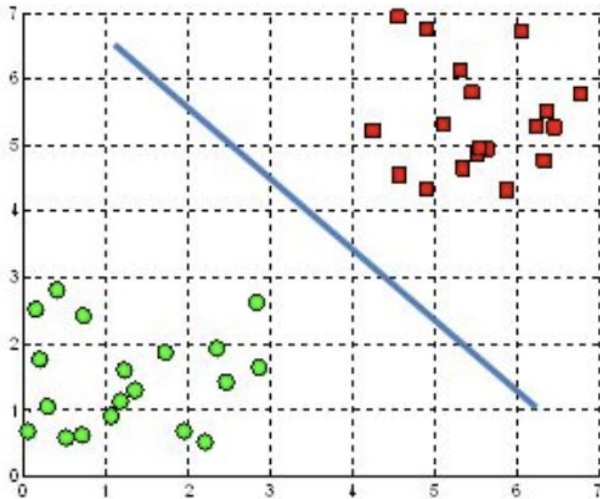
Was ist eine SVM

- Support Vector Machines sind Classifier. Sie unterteilen eine Menge von Objekten in zwei Klassen
- Bei Support Vector Machines handelt es sich um einen Large Margin Classifier
- Es handelt sich bei der normalen SVM um einen linearen Classifier, dieser kann aber durch Verwendung des Kernel-Tricks auch als nicht-linearer Classifier fungieren

Was ist eine Hyperebene ?

Hyperebene

A hyperplane in \mathbb{R}^2 is a line



A hyperplane in \mathbb{R}^3 is a plane

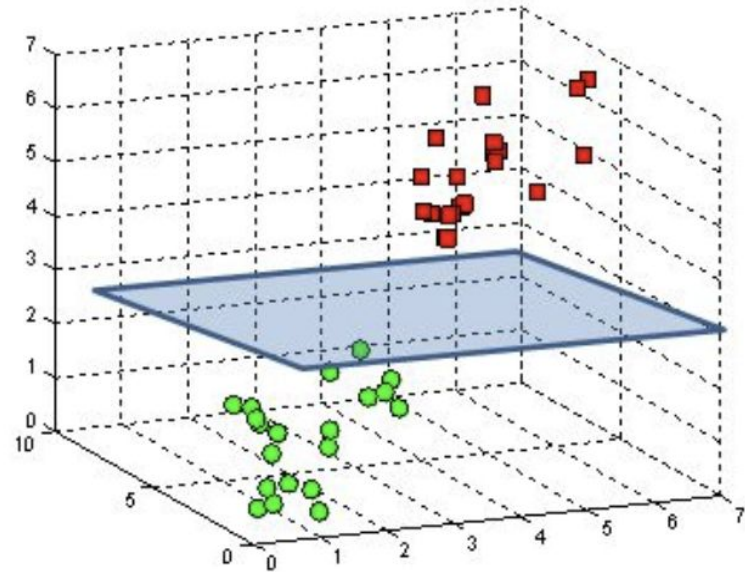


Abb. 1: Definition Hyperbene - <https://medium.com/@gallettilance/support-vector-machines-16241417ee6d>

Hyperebene bestimmen

- Um die Hyperebene zu bestimmen, ist es wichtig dass eine lineare Separierbarkeit vorliegt
- Idealerweise sind unsere Daten bereits linear separierbar, falls nicht muss der Kernel-Trick angewendet werden

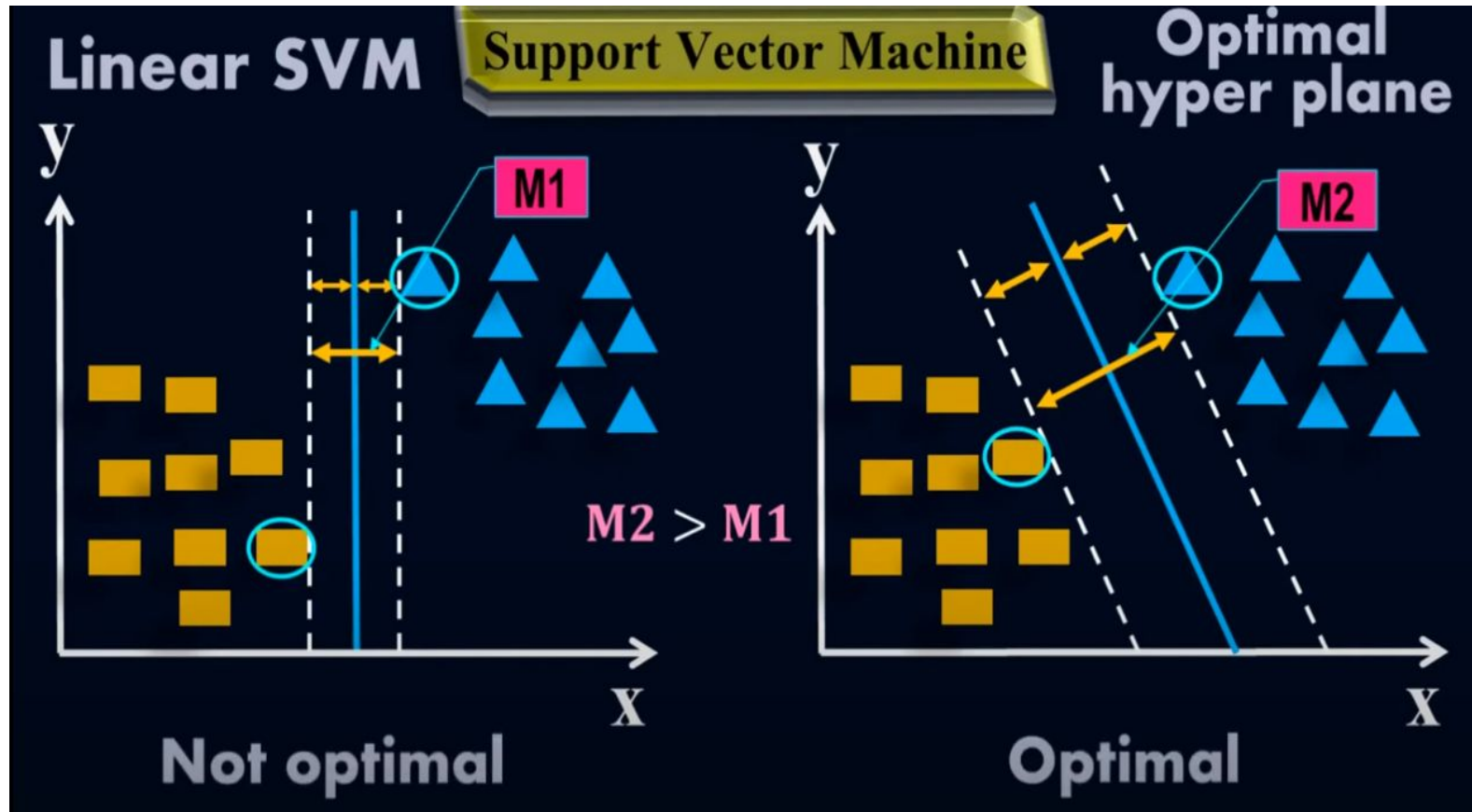


Abb. 2: Vergleich Hyperebenen - <https://images.app.goo.gl/pc2JxY5SzLFSmmR79>

Trainingsphase

Trainingsphase

- In der Trainingsphase wird eine geeignete Hyperebene mit Hilfe der Trainingsdaten ermittelt
- Diese Hyperebene wird durch die Parameter w (Gewichtsvektor) und b (Bias) eindeutig bestimmt
- Für diese beiden Parameter möglichst optimale Werte zu finden, ist das Ziel der Trainingsphase
- Sofern eine Hyperebene existiert, existieren unendlich viele

Aufbau der Trainingsdaten

- Die Trainingsdaten bestehen aus einer Liste von Tupeln
 - Ein Tupel besteht aus einem Merkmalsvektor und dem zugehörigen Label
- Hierbei gibt es für das Label **genau zwei** Möglichkeiten
 - Zum Beispiel: negativ oder positiv
- Damit die Daten von der SVM verarbeitet werden können, müssen die Labels passend umgewandelt werden
 - Hierbei wird jedem Label entweder der Wert “**1**” oder “**-1**” zugewiesen
 - Zum Beispiel: **positiv = 1** und **negativ = -1**

Grafische Darstellung Hard Margin Case

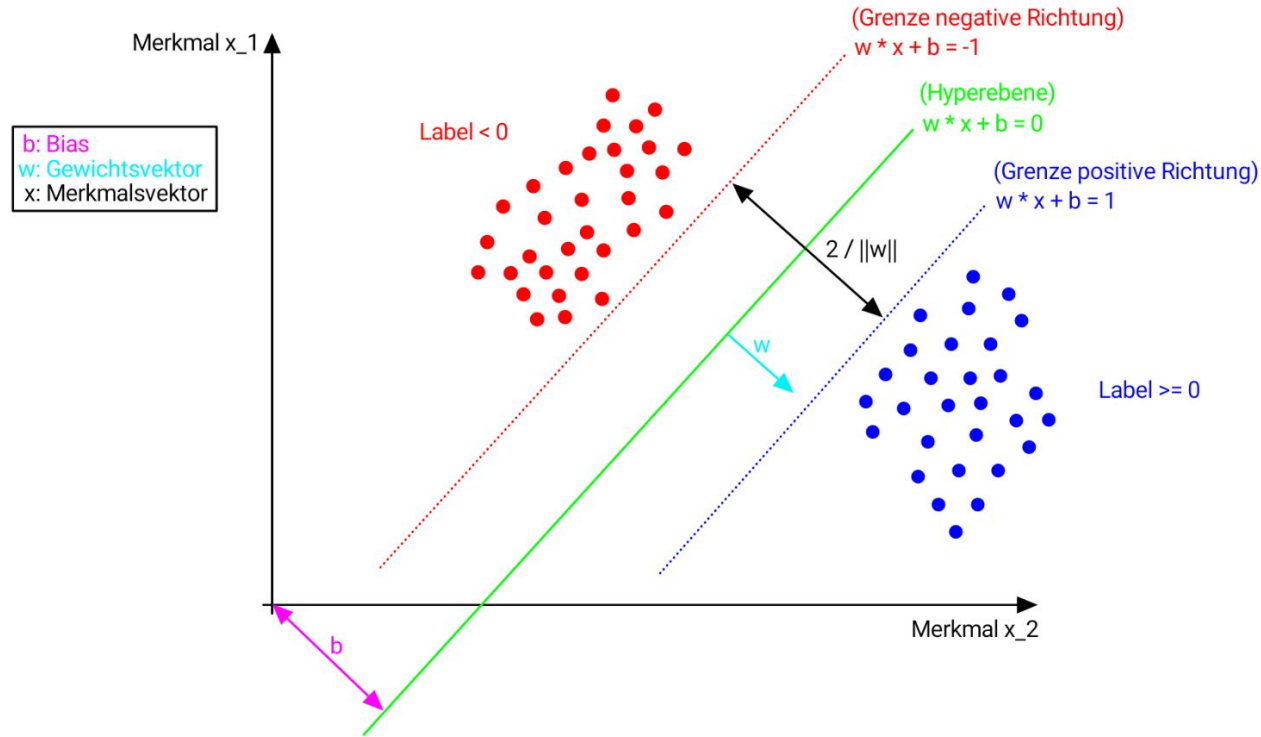


Abb. 3: Darstellung Hard Margin Case

Mathematische Bedingung des Trainings

- Die beiden betrachteten Grenzen in positive bzw. negative Richtung lassen sich zu einer Bedingung zusammenfassen
- Bedingung: $\mathbf{y} * (\mathbf{w} * \mathbf{x} + \mathbf{b}) \geq 1$ (F1: Entscheidungsfunktion)
 - \mathbf{y} : Label von \mathbf{x} (kann entweder 1 oder -1 sein)
 - \mathbf{w} : Gewichtsvektor der SVM
 - \mathbf{x} : Aktuell betrachteter Merkmalsvektor aus den Trainingsdaten
 - \mathbf{b} : Bias der SVM
- Ist diese Bedingung erfüllt, so wurde der Merkmalsvektor \mathbf{x} korrekt klassifiziert

Trainingsphase Pseudocode

```
1 algorithm fit(data_train[1...n], iterations, learning_rate)
2     //initialisiere Gewichtsvektor w mit Nullvektor
3     w = Nullvektor
4     //initialisiere Bias b mit 0
5     b = 0
6
7     for i = 0; i < iterations; i++
8         for sample in data_train
9             if not (sample.label * (w * sample.vector + b)) >= 1
10                 //Datenpunkt verletzt Bedingung: Passe w und b an
11                 w += sample.vector * (sample.label * learning_rate)
12                 b += sample.label * learning_rate
```

Abb. 4: Trainingsphase Pseudocode

Visualisierung Training

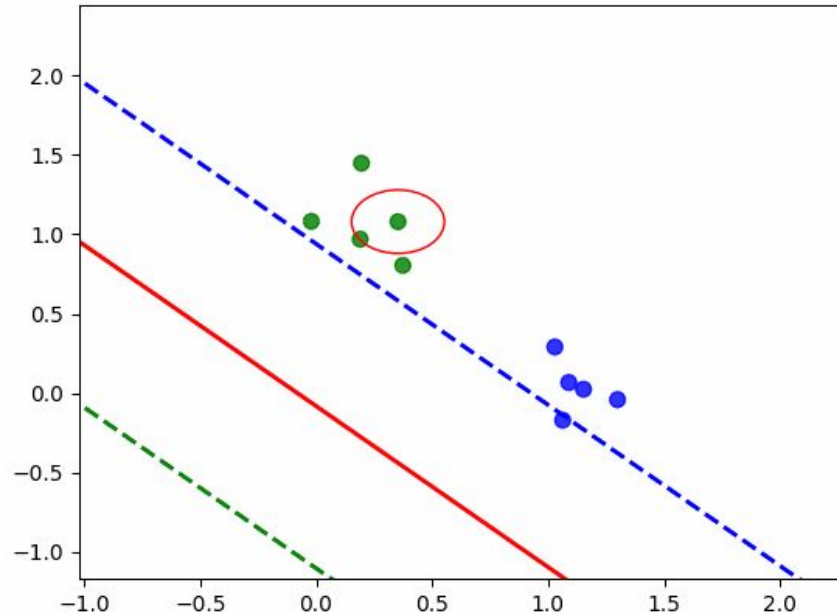


Abb. 5: Visualisierung Training - <https://medium.com/@gallettilance/support-vector-machines-16241417ee6d>

Beobachtung

- Die Hyperebene pendelt sich ab einer gewissen Anzahl an Iteration an einer bestimmten Stelle ein und ändert sich nicht mehr signifikant
- Somit muss das Training mit einer ausreichenden Anzahl an Iterationen durchgeführt werden

Kernel-Trick

Nicht lineare Daten

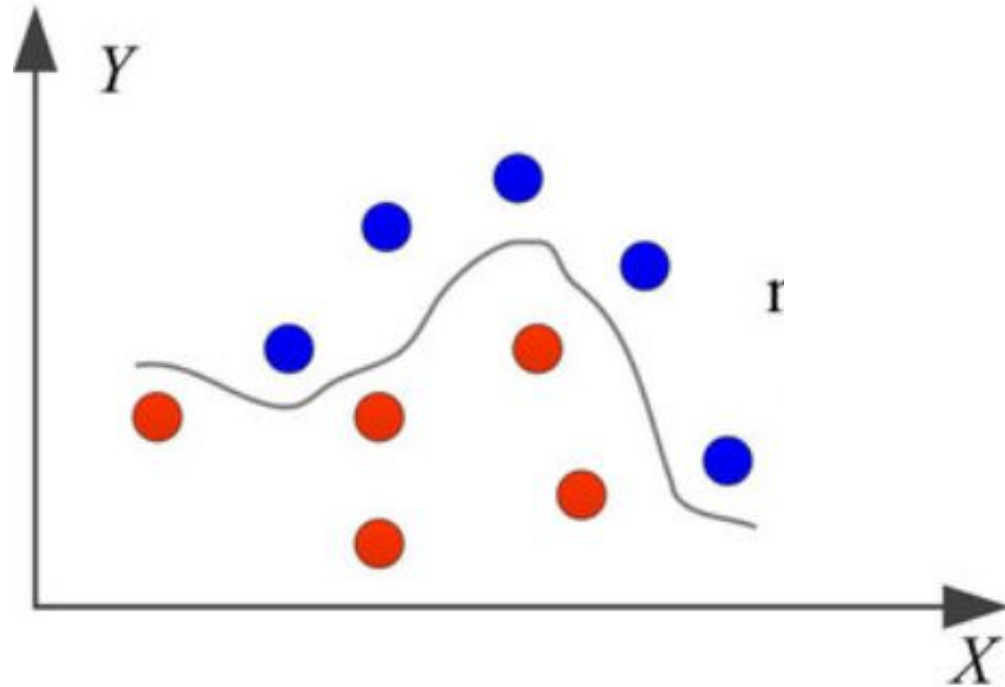


Abb. 6: Nicht lineare Daten - <https://dida.do/de/blog/was-ist-ein-kernel-beim-maschinellen-lernen>

Kernel-Trick

- Bei Kernel-Methoden gibt es einen Trick, wodurch ein linearer Klassifikator wie die SVM auch auf nichtlineare Daten angewendet werden kann
- Im Grunde wird dies erreicht, indem man die Daten in höhere Dimensionen transformiert, bis eine lineare Separierbarkeit erreicht ist

Wie funktioniert der Kernel-Trick

- Beim Kernel-Trick werden die genauen Koordinaten im höheren Raum nicht berechnet
- Es werden stattdessen Ähnlichkeiten bzw. Produktwerte zwischen den Datenpaarpunkten berechnet, als würden diese sich in dem höheren Raum befinden. Dafür wird ein Kernel-Funktionswert verwendet.
- Durch das Umgehen der Transformation unserer Datenpaare wird der Rechenaufwand verringert

Beispiel

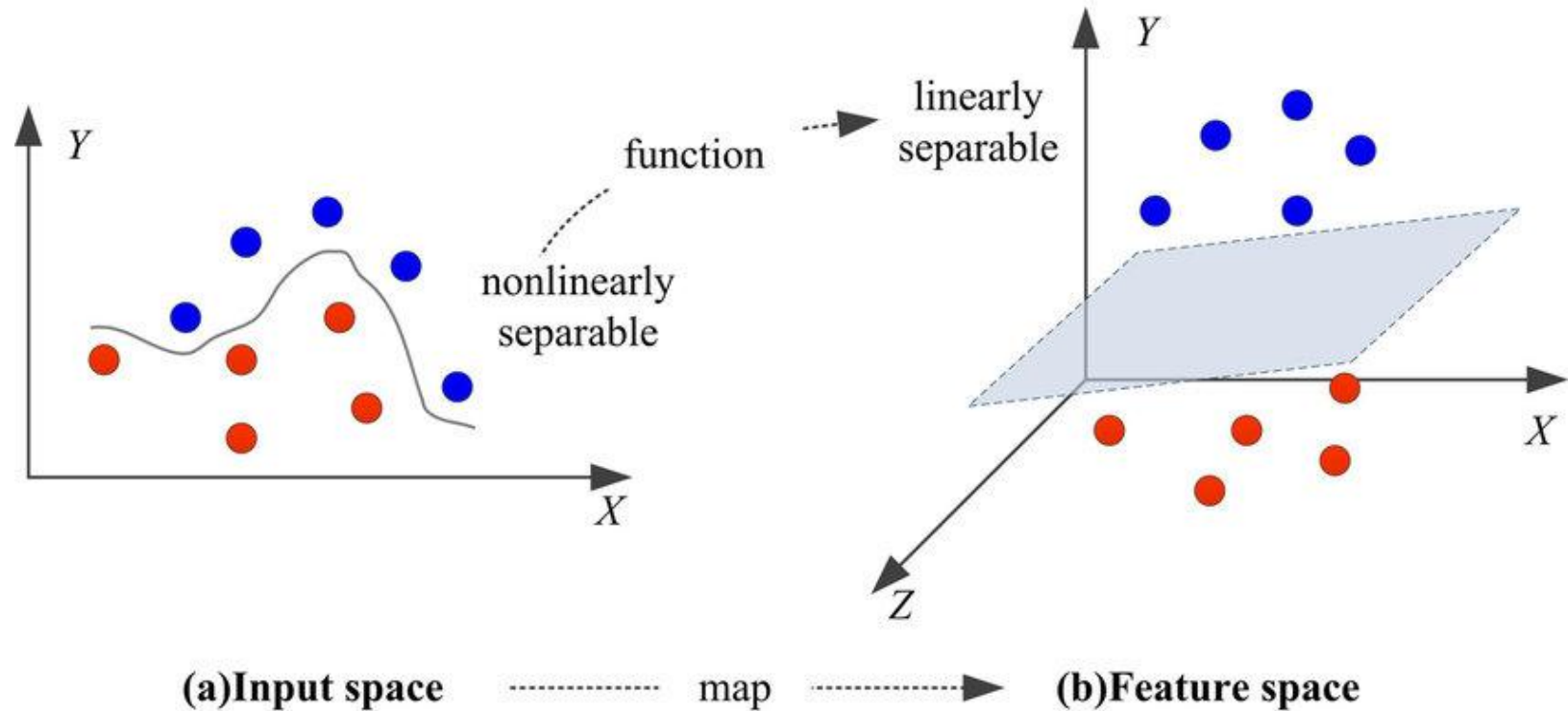


Abb. 7: Beispiel Kernel-Trick Anwendung - <https://dida.do/de/blog/was-ist-ein-kernel-beim-maschinellen-lernen>

Klassifizieren unbekannter Datenpunkte

Bedeutung des Skalarprodukts

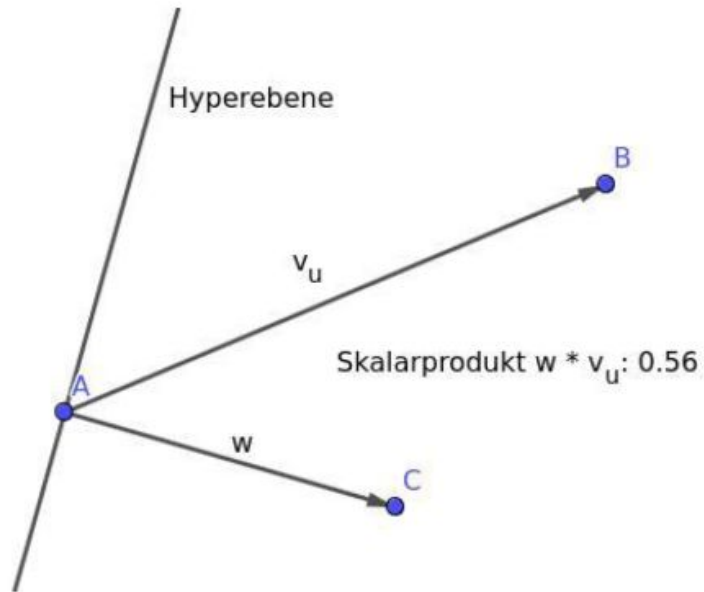


Abb. 8: Skalarprodukt positiv

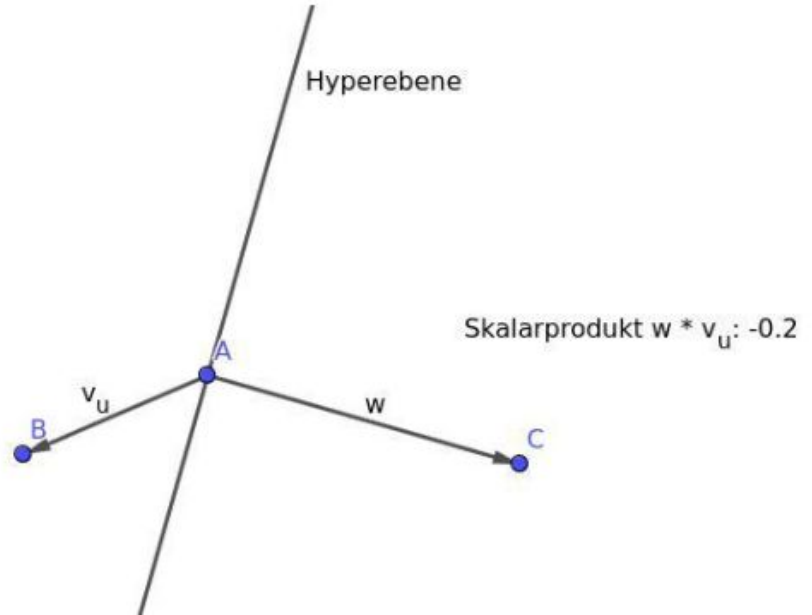


Abb. 9: Skalarprodukt negativ

Label von unbekanntem Datenpunkt bestimmen

Berechne: $y = w * v_U + b$ (F2: Bestimmung des Labels)

$y \geq 0 \rightarrow$ Klasse 1 (Label = 1)

$y < 0 \rightarrow$ Klasse 2 (Label = -1)

Beispiel

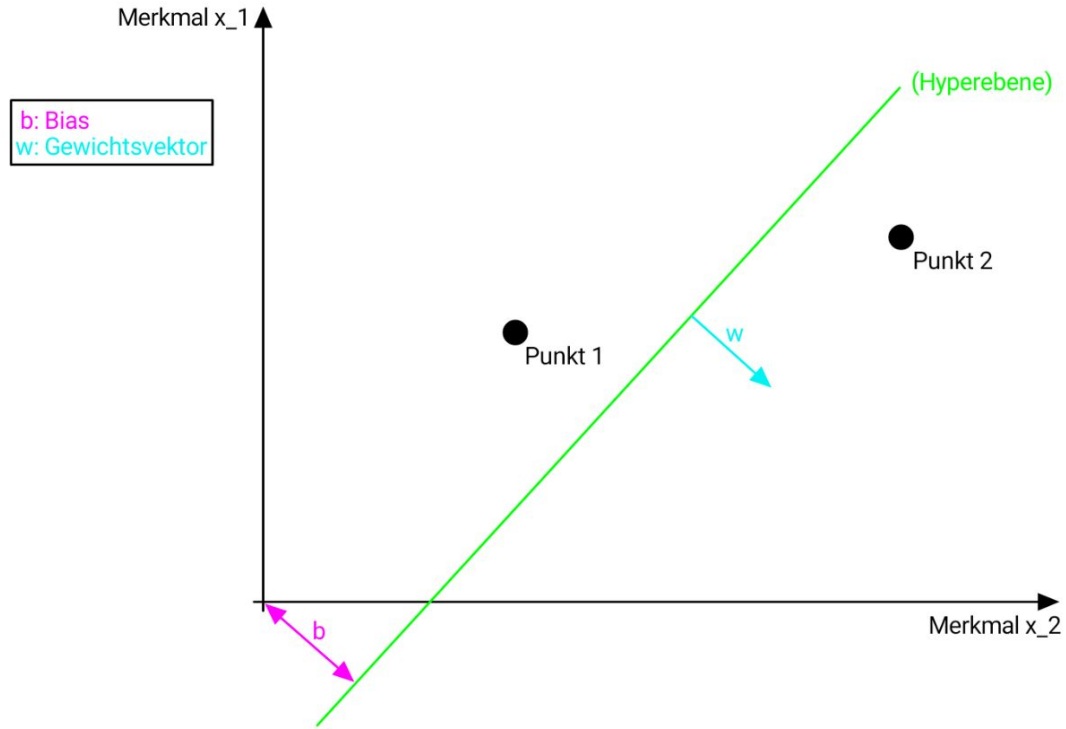


Abb. 10: Beispiel unbekannte Datenpunkte - unbestimmt

Beispiel

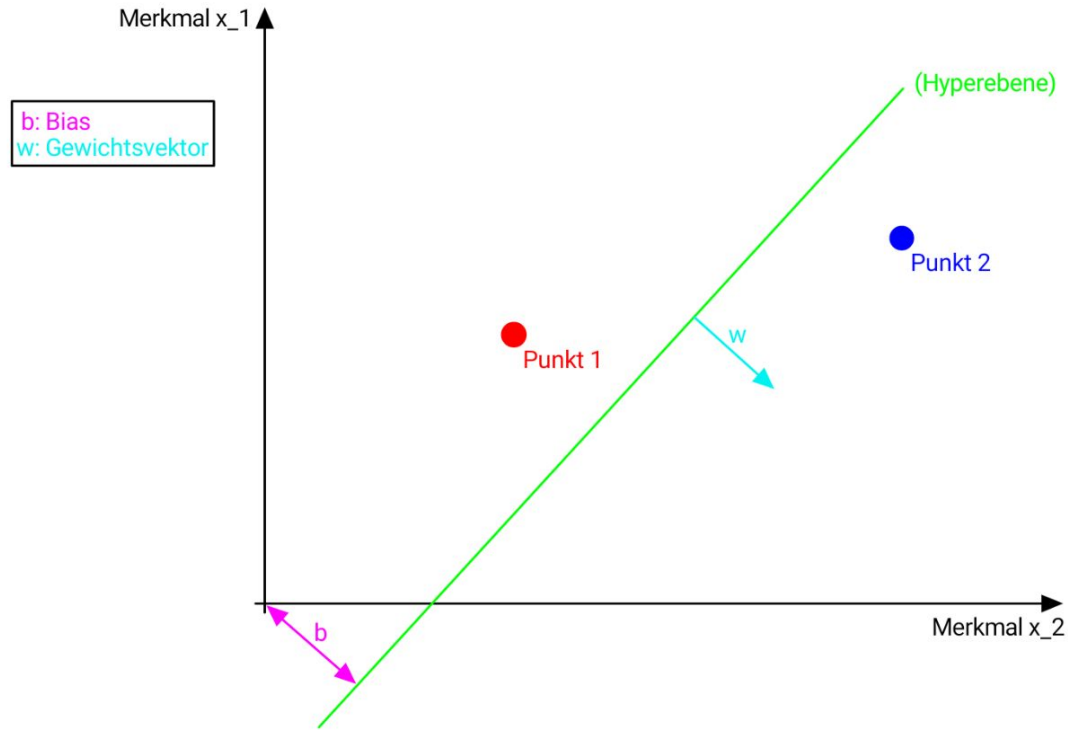


Abb. 11: Beispiel unbekannte Datenpunkte - bestimmt

Quellen

- https://www.youtube.com/watch?v=bM4_AstaBZo
- <https://www.youtube.com/watch?v=T9UcK-TxQGw>
- <https://github.com/AssemblyAI-Examples/Machine-Learning-From-Scratch/blob/main/09%20SVM/svm.py>
- <https://medium.com/@gallettilance/support-vector-machines-16241417ee6d>
- <https://medium.com/@kushaldps1996/a-complete-guide-to-support-vector-machines-svms-501e71aec19e>
- https://de.wikipedia.org/wiki/Support_Vector_Machine
- <https://www.geeksforgeeks.org/using-a-hard-margin-vs-soft-margin-in-svm/>
- https://www.tu-chemnitz.de/urz/ittime/documents/Vortrag_Jens_Poenisch_SVM.pdf
- https://c.wgr.de/f/verlage/westermanngruppe-at/dimensionen-mathematik/materialien/06_Vektorrechnung/06_GeometrischeDeutungDesSkalarprodukts/06_06_GeometrischeDeutungSkalarprodukt.html
- <https://github.com/CihanBosnali/Machine-Learning-without-Libraries/tree/master/Support-Vector-Machine>
- Géron, Aurélien: Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2. Auflage, O'Reilly Media, Inc., Kanada 2019
- <https://images.app.goo.gl/pc2JxY5SzLFSmmR79>