ingenieur
wissenschaften
htw saar

Hochschule für
Technik und Wirtschaft
des Saarlandes
University of
Applied Sciences

Bachelor-Thesis

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)
an der Hochschule für Technik und Wirtschaft des Saarlandes
im Studiengang Praktische Informatik
der Fakultät für Ingenieurwissenschaften

Effiziente Generierung von Trainingsdaten in der Bildklassifikation

vorgelegt von
Jan Rauber

betreut und begutachtet von
Prof. Dr.-Ing. Klaus Berberich

Saarbrücken, 08. 06. 2025

Selbständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit (bei einer Gruppenarbeit: den entsprechend gekennzeichneten Anteil der Arbeit) selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich erkläre hiermit weiterhin, dass die vorgelegte Arbeit zuvor weder von mir noch von einer anderen Person an dieser oder einer anderen Hochschule eingereicht wurde.

Darüber hinaus ist mir bekannt, dass die Unrichtigkeit dieser Erklärung eine Benotung der Arbeit mit der Note „nicht ausreichend“ zur Folge hat und einen Ausschluss von der Erbringung weiterer Prüfungsleistungen zur Folge haben kann.

Saarbrücken, 08. 06. 2025

_____

Jan Rauber

Zusammenfassung

Kurze Zusammenfassung des Inhaltes in deutscher Sprache, der Umfang beträgt zwischen einer halben und einer ganzen DIN A4-Seite.

Orientieren Sie sich bei der Aufteilung bzw. dem Inhalt Ihrer Zusammenfassung an Kent Becks Artikel: `http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html`.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth [14]

Danksagung

Hier können Sie Personen danken, die zum Erfolg der Arbeit beigetragen haben, beispielsweise Ihren Betreuern in der Firma, Ihren Professoren/Dozenten an der htw saar, Freunden, Familie usw.

Inhaltsverzeichnis

1 Theoretische Grundlagen

Im Folgenden werden die theoretischen Grundlagen erläutert, die zum Verständnis dieser Arbeit notwendig sind.

1.1 Lernen: Eine Definition

Lernen ist die Fähigkeit des Geistes, sich seiner Umwelt anzupassen und sie sich zunutze zu machen. Zum Beispiel durch Werkzeuge wie Sprache, Mathematik, Physik, Musik und so weiter. Die Physis, die dem biologischen Lernen zugrunde liegt, zum Beispiel ein Gehirn, ist weitgehend unerforscht.[11]

1.2 Maschinelles Lernen

Maschinelles Lernen ist eine Disziplin in der Informatik. Hierbei wird versucht, die biologische Fähigkeit zu lernen, auf eine Maschine zu übertragen. Ziel der Wissenschaft ist es, in Zukunft das biologische Pendant zu übertreffen. Maschinelles Lernen gleicht einer komplizierten Excel-Tabelle und hat mit seinem biologischen Pendant nichts gemeinsam.[16]

1.3 Passives Lernen

Bei diesem Konzept bekommt die Maschine einen Datensatz und studiert diesen Datensatz vollständig, ohne zu verstehen, welche Daten relevant sind. Es gleicht eher einem Auswendiglernen des Datensatzes. Die Performance kann daher auf einen anderen Datensatz derselben Kategorie stark abweichen, weil der Fokus nicht auf dem Verstehen, sondern auf dem Auswendiglernen lag.[40]

1.4 Aktives Lernen

Hierbei lernt das Modell deutlich schneller, also mit weniger Labels, weil es sich aktiv am Lernprozess beteiligt, zum Beispiel durch das Herauspicken informativer Datenpunkte, um die Modellgüte zu verbessern.[6].

1.5 Der aktive maschinelle Lernprozess

- Der Mensch, der in diesem Fall die Rolle des Orakels einnimmt, labelt initial einen Bruchteil des Datensatzes, aber nicht den vollständigen Datensatz, wie es beim passiven Lernen der Fall wäre. [23]
- Daraufhin trainiert die Maschine auf diesen Bruchteil der Daten und wählt anschließend aus einem Pool aus den restlichen ungelabelten Daten aus dem Datensatz interessante Daten aus, die für das Modell den größten Informationsgehalt versprechen, um den größten Lernfortschritt zu erzielen. Das Modell fragt also nach. Es

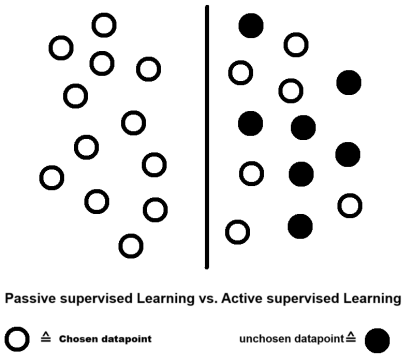

Abbildung 1.1: Diese Grafik beschreibt den iterativen Prozess, bei dem sich der Active Learner die Daten mit dem höchsten Informationsgewinn auswählt, um Labels einzusparen gegenüber dem passiven Ansatz. Dieser Prozess wird so lange wiederholt, bis keine ungelabelten Daten mehr vorhanden sind, die vom Active Learner ausgewählt und an das Orakel zur Annotation geschickt werden können, oder bis ein Abbruchkriterium erreicht wurde. Zum Beispiel wenn der Active Learner die gewünschte Accuracy bereits erreicht hat. Der Active Learner erreicht in der Regel die gewünschte Accuracy mit deutlich weniger Labels, als beim passiven Ansatz notwendig wären.

werden somit viel schneller Verknüpfungen für das Verständnis der Daten erstellt, als das beim passiven Lernen der Fall ist, und somit wird der Annotationsaufwand für gleichbleibende oder bessere Modellleistung verringert. [23]

- Dieser Prozess wiederholt sich iterativ, bis keine zu labelnden Daten aus dem Pool mehr vorhanden sind oder ein Abbruchkriterium erreicht wurde. Die folgende Grafik illustriert den Prozess des aktiven Lernens. [23]

1.6 Auswahl der Daten mit hoher Informationsdichte

Dem Active Learner stehen mehrere Strategien zur Verfügung, um die Informationsgüte unbeschrifteter Beispiele zu schätzen. Nur die Beispiele mit der höchsten Informationsgüte sollen für das Orakel zur Annotation ausgewählt werden, um den Active Learner effizient zu trainieren. Anbei werden die Querystrategien erörtert, die für die Evaluierung im Rahmen dieser Arbeit verwendet werden. [31]

1.6.1 Unsicherheitsbasiertes Sampling

Hierbei wählt das Modell diejenigen Datenpunkte aus, bei denen sich das Modell am unsichersten über die korrekte Klasse ist und aufgrund dieser Unsicherheit bei diesen Datenpunkten nicht in der Lage ist, korrekt zu klassifizieren. Unterkategorien dieses unsicherheitsbasierten Verfahrens sind Least Confidence, Margin Sampling und Entropy Sampling. Diese Unterkategorien werden nachfolgend erläutert. [1]

1.6.1.1 Least Confidence

Bei diesem Verfahren wählt das Modell das Beispiel für das Orakel zur Annotation aus, dessen höchste Klassenwahrscheinlichkeit am geringsten ist. [1]

Die mathematische Definition für Least Confidence lautet:

$$x^*_{LC} = \text{argmax}_x \left(1 - P(\hat{y}|x)\right)$$

wobei $\hat{y} = \text{argmax}_y P(y|x)$ die wahrscheinlichste Klasse darstellt.

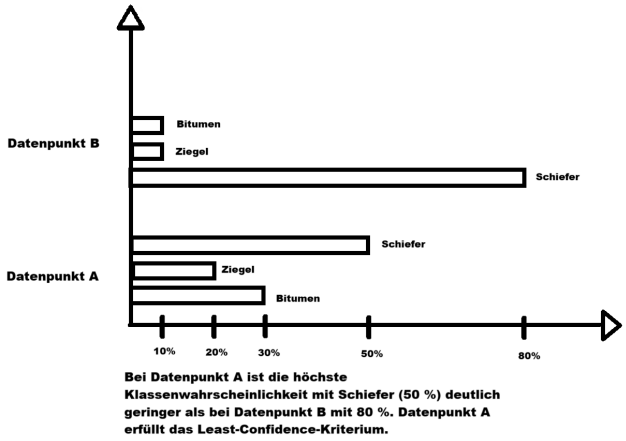

Abbildung 1.2: Visualisierung des Least Confidence Samplings

Beispiel für Least Confidence Angenommen, es gibt in unserem Beispiel drei Möglichkeiten für das Modell, einen Datenpunkt zu labeln. Rein fiktiv in unserem Dachmaterialdatensatz zwischen den Möglichkeiten Schiefer, Ziegel und Bitumen als Dächer. Angenommen, unser Modell klassifiziert das Dach mit einer Wahrscheinlichkeit von 60 % als Schieferdach. Zu 20 % könnte es aber auch ein Dach aus Ziegeln oder Bitumen sein. Die höchste Klassenwahrscheinlichkeit beträgt hier 60 %. Ist aber zugleich die niedrigste in diesem Durchlauf, weil sich das Modell bei allen anderen Datenpunkten in dieser Iteration sicherer war. Zum Beispiel, wenn sich das Modell für einen anderen Datenpunkt zu 98 % für Ziegeln entscheidet, ist es sich viel sicherer in Relation zu einem Datenpunkt, in dem die höchstwahrscheinliche Klasse nur 60% beträgt.

$$P(\text{Ziegel}) = 0{,}33, \quad P(\text{Beton}) = 0{,}33, \quad P(\text{Schiefer}) = 0{,}34 \tag{1.1}$$

1.6.1.2 Entropy Sampling

Bei Anwendung dieser Strategie gilt ein Datenpunkt dann als informativ, wenn die vorhergesagten Wahrscheinlichkeiten zur Klassenzugehörigkeit in diesem Datenpunkt gleich verteilt sind. Man spricht in diesem Fall von einer hohen Entropie. [1]

$$P(A) = 0{,}33, \quad P(B) = 0{,}34, \quad P(C) = 0{,}33 \tag{1.2}$$

Beipiel zu Entropy Sampling Angenommen, das Modell prophezeit in unserem Dachmaterialdatensatz eine Klassenzugehörigkeit für einen Datenpunkt von 34 % für Bitumen, 33 % für Ziegel und 33 % für Schiefer als Dach. Somit ist die Entropie sehr hoch und der Datenpunkt wird vom Modell zur Annotation an das Orakel geschickt.

$$H = -\sum_{i} P(i) \cdot \log(P(i))$$

Konkret ergibt sich für das obige Beispiel:

$$H = -[0,33 \cdot \log(0,33) + 0,34 \cdot \log(0,34) + 0,33 \cdot \log(0,33)] \approx 1,0986$$

Dieser Wert liegt nahe am Maximum (bei drei Klassen maximal $\log(3) \approx 1,0986$), was die hohe Unsicherheit verdeutlicht.

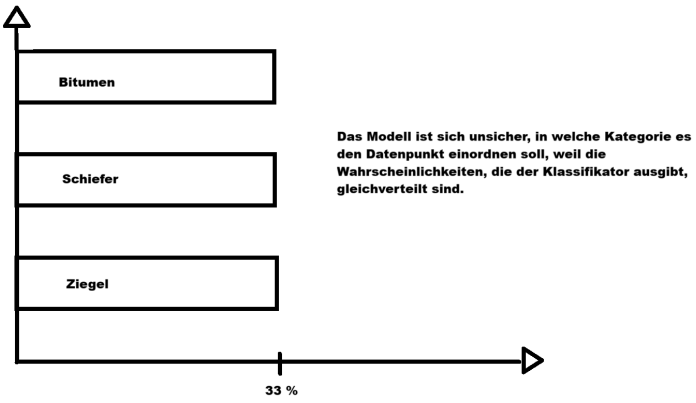

Abbildung 1.3: Visualisierung des Auswahlkriteriums für Entropy Sampling im Balkendiagramm

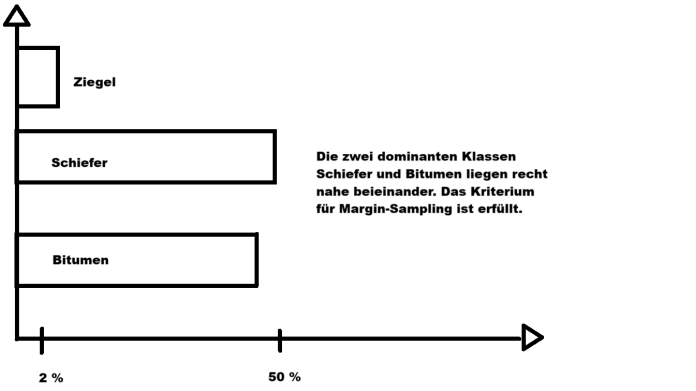

Abbildung 1.4: Visualisierung mit Balkendiagramm anhand welches Kriteriums Margin Sampling Datenpunkte auswählt

1.6.1.3 Margin Sampling

Das Modell misst seine Unsicherheit in Bezug auf den Datenpunkt daran, wie knapp es zwischen den beiden wahrscheinlichsten Klassen entscheidet. Bei dieser Differenz in der Vorhersagewahrscheinlichkeit zwischen den beiden wahrscheinlichsten Klassen spricht man auch von Margin. Diese Margin soll möglichst klein sein, um vom Modell ausgewählt zu werden. [1]

$$\text{Margin} = P(\hat{y}_1|x) - P(\hat{y}_2|x)$$

wobei $\hat{y}_1$ und $\hat{y}_2$ die beiden wahrscheinlichsten Klassen sind.

Beispiel für Margin Sampling Angenommen, das Modell vergibt in unserem Dachmaterialdatensatz Wahrscheinlichkeiten in einem Datenpunkt, dass das Dach zu 50 % ein Ziegeldach ist, zu 48 % ein Dach aus Bitumen sein könnte und zu 2 % ein Schieferdach darstellen könnte. Hier beträgt die Margin zwischen den beiden wahrscheinlichsten Klassen Ziegel und Bitumen 2 % und der Datenpunkt wird wahrscheinlich zur Annotation an das Orakel weitergeleitet.

1.7 Wilcoxon Signed Rank Test

Der Wilcoxon-Signed-Rank-Test wird im Rahmen der Evaluation dieser Arbeit eingesetzt. Er wird eingesetzt, um paarweise Vergleiche zwischen der passiven Baseline (Random Sampling) und den Active-Learning-Strategien durchzuführen, um herauszufinden, ob die Query-Strategien des aktiven Lernens besser abschneiden als Random Sampling. Der Wilcoxon signed-rank-Test prüft also die Hypothese, dass aktives Lernen bei weniger verwendeten Labels besser abschneiden kann als passives Lernen. Der Vorteil dieses Hypothesentests ist, dass die Daten nicht metrisch und nicht normalverteilt vorliegen müssen. [10]

Beispiel für den Wilcoxon Signed Rank Test in Bezug auf die spätere Evaluation Angenommen, die Accuracy wird in Bezug auf Random Sampling zu Margin Sampling unter Verwendung des Klassifikators Random Forest verglichen. Die Berechnung wird in diesem Beispiel dreimal wiederholt und wir erhalten somit drei verschiedene Wertepaare. Die Differenzen der jeweiligen Wertepaare werden berechnet und es wird eine Rangliste erstellt, wo die Differenz am größten ist. Die verwendeten Werte in diesem Beispiel haben keine Bedeutung.

Random Sampling	Margin Sampling	Differences	Ranks
5	16	-11	3
5	3	2	1
5	12	-7	2

Jetzt wird notiert, ob der Rang aus einer negativen oder aus einer positiven Differenz stammt. Dann wird jeweils die Summe der Werte in der Rangspalte, die jeweils aus negativen Differenzen stammen, und der Werte, die aus positiven Differenzen stammen, getrennt gebildet. In diesem Beispiel beträgt die Summe der positiven Ränge 1 und die Summe der negativen Ränge 5. Wenn jetzt hier in dem Beispiel kein Unterschied vorliegt, müssten beide Summen identisch sein. Das entspricht der Nullhypothese. In diesem Fall trifft die Nullhypothese nicht zu, da die Summen unterschiedlich sind. In der späteren Evaluation werden die Experimente fünfmal wiederholt anstatt nur dreimal.

1.8 Der P-Wert

Der P-Wert wird später in der Evaluation darüber aussagen, ob die ermittelten Ergebnisse, ob aktives Lernen besser sein könnte als passives Lernen in Bezug auf Labelaufwand und Accuracy, rein zufällig zustande kamen und womöglich keine Aussagekraft haben. Bei einem P-Wert größer als fünf Prozent sagt man, die Ergebnisse kamen zufällig zustande. Wenn dieser Fall eintritt, müssen wir von der Nullhypothese ausgehen. [7]

1.8.1 Berechnung des P-Werts

Um den P-Wert zu bestimmen, muss erst der Z-Wert berechnet werden. Die Formel für den Z-Wert lautet [7]:

$$z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

Dabei ist $\bar{x}$ der Mittelwert der Stichprobe, μ ist der Wert der Nullhypothese, σ ist die Standardabweichung und n repräsentiert die Anzahl der Fälle in der Stichprobe. Nach

Einsetzen der Werte erhält man:

$\mu = \frac{n(n+1)}{4} = \frac{3(3+1)}{4} = 3$

$\sigma = \sqrt{\frac{n(n+1)(2n+1)}{24}}$

$\sigma = \sqrt{\frac{3(3+1)(2\cdot 3+1)}{24}}$

$\delta = 1,87$

$z = \frac{1-3}{1,87} \approx -1,069$

Den p-Wert hierfür lässt sich in einer Tabelle nachschlagen. Es resultiert ca. 0,285 als p-Wert. Der P-Wert in diesem Beispiel ist somit größer als fünf Prozent und es gibt keinen statistischen Grund, die Nullhypothese abzulehnen.

1.9 Bonferroni-Korrektur

Die Bonferroni-Korrektur wird in der Evaluation verwendet, um das Signifikanzniveau alpha von fünf Prozent anzupassen. Um die Bonferroni-Korrektur durchzuführen, wird das gewünschte Signifikanzniveau von fünf Prozent durch die Anzahl der Tests geteilt. Als Ergebnis resultiert das korrigierte Signifikanzniveau für jeden einzelnen Test. Die Bonferroni-Korrektur ist notwendig wegen der Alpha-Fehler-Kumulierung. Das Alpharisiko berechnet sich durch [34]:

$1 - (1 - \alpha)^j$

Dabei ist j die Anzahl der durchgeführten Tests. Diese Formel gibt das Risiko an, ob ein Test fälschlicherweise signifikant werden könnte. Um dieses Risiko zu minimieren, wird die Bonferroni-Korrektur angewendet.

1.10 Effektstärke

Die Effektstärke gibt an, wie stark der beobachtete Effekt ist. In diesem Fall, ob aktives Lernen deutlich besser ist als Random Sampling. Der Effekt steht also in diesem Fall für den Unterschied. Die Effektstärke wird berechnet, indem der z-Wert des Wilcoxon-Tests durch die Quadratwurzel der Stichprobengröße geteilt wird. Werte, die sich der Null annähern, werden als geringe Effektstärke interpretiert, während Werte, die sich minus eins oder eins annähern, auf eine hohe Effektstärke hindeuten. Die Effektstärke gibt somit die Stärke des Unterschieds an, während der p-Wert lediglich aussagt, dass die beobachtete Messung kein Zufall ist. Die Effektstärke ist in dieser Arbeit von Nutzen, um zu prüfen, ob die Implementierung aktiven Lernens in der Praxis gegenüber passivem Lernen eine spürbare positive Veränderung in Bezug auf Labelaufwand und Accuracy bewirkt. Eine geringe Effektstärke würde zur Nullhypothese führen. [13].

Nachdem die Methoden zur statistischen Auswertung der Ergebnisse erläutert wurden, werden nun die Klassifikatoren vorgestellt, deren Leistung im Rahmen dieser Arbeit evaluiert wird.

1.11 Verwendete Klassifikatoren

Dieser Abschnitt erläutert die Funktionsweise der in der Evaluierung verwendeten Klassifikatoren. Zudem gibt es eine Erwartung zu den Ergebnissen, die der jeweilige Klassifikator auf den Datensätzen MNIST, Fashion-MNIST und dem Dachmaterialiendatensatz erzielen wird.

1.11.1 Random Forest

Random Forest basiert auf einem Ensemble von Entscheidungsbäumen. Ein Entscheidungsbaum arbeitet mit diskreten Werten. Zum Beispiel, ob der Wert größer oder gleich sieben ist. Wenn das nicht der Fall ist, wird der Wert zum entsprechenden Ast in dem Baum weitergeleitet. Die Daten werden also durch das interne „Wenn Dann“ immer weiter aufgeteilt. Man kann dadurch anschaulich betrachten, wie eine Entscheidung zustande kommt. Der Nachteil der Entscheidungsbäume ist, dass sie nur diskrete Durchschnittswerte vorhersagen können. Sie bieten keine feinen Abstufungen, um kontinuierliche Werte vorhersagen zu können. Der Baum müsste für feine Abstufungen extrem tief sein, was zu Overfitting führt. Das heißt, das Modell lernt einfach nur die Daten auswendig. Das Modell kann schlecht auf neue, unbekannte Daten in so einem Fall generalisieren. Ein Entscheidungsbaum hat eine hohe Modellvarianz. Das heißt, dass sich ein Entscheidungsbaum aufgrund der Stichprobe, die sich jedes Mal unterscheidet, ganz verschieden entwickelt und somit aufgrund der unterschiedlichen Daten in der Stichprobe jedes Mal stark abweichende Vorhersagen für dieselbe Kategorie von Daten liefern wird. Der Random Forest löst dieses Varianzproblem, indem ein ganzer Wald von unterschiedlichen Bäumen anstatt nur ein Baum implementiert wird. Durch Bootstrap-Aggregating werden die Bäume verschieden durch Bagging. Hierbei wird für jeden Baum eine Stichprobe aus den Originaltrainingsdaten gezogen, mit Zurücklegen. Die Out-of-bag-Samples, die in keinem Baum landen, werden zum Testen des Modells verwendet. Außerdem variieren im Random Forest auch die Merkmale, die die einzelnen Bäume betrachten. Der einzelne Baum darf nicht alle vorhandenen Merkmale in max. features prüfen, sondern er bekommt eine zufällig ausgewählte Teilmenge aus allen verfügbaren Merkmalen vorgegeben. Das macht die Bäume diverser und weniger korreliert untereinander. Der Random-Forest-Algorithmus nimmt den Durchschnittswert seiner einzelnen Bäume als Vorhersage. Random-Forest-Modelle können nicht extrapolieren. Das heißt, sie können keine Werte vorhersagen, die weit außerhalb des Bereichs der Daten liegen, die sie in dem Training gesehen haben. [2]

Wahrscheinlichkeitsberechnung: Die Wahrscheinlichkeit einer Klasse k ergibt sich als Anteil der Bäume, die k vorhersagen:

$$\hat{P}(Y = k|x) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{\text{Baum}_t(x) = k\}$$

Erwartung von Random Forest in Bezug auf Performance auf den Datensätzen Erwartung für Random Forest auf den Datensätzen: Aufgrund der Mittelwertbildung in den Endknoten wird erwartet, dass Random-Forest-Modelle häufig vertretene Klassen in unbalancierten Datensätzen gut klassifizieren. Unterrepräsentierte Klassen werden jedoch möglicherweise vernachlässigt, was die Praxistauglichkeit einschränkt. Trotz erwarteter Überlegenheit gegenüber anderen Klassifikatoren auf dem Dachmaterialdatensatz bleibt die Gesamtleistung aufgrund der Datenqualität limitiert. Zusätzliche Features wie

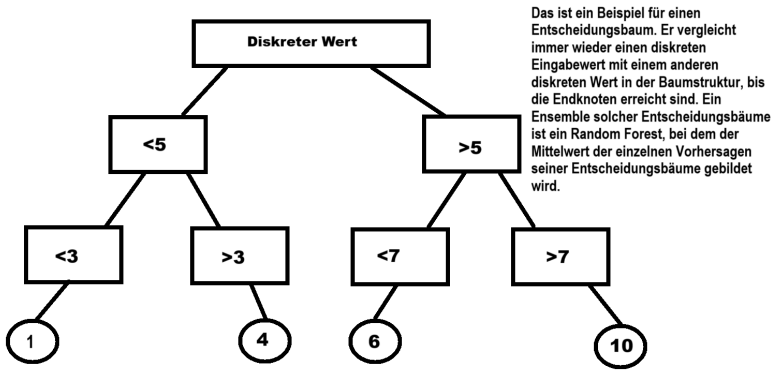

Abbildung 1.5: Visuelle Darstellung eines einzelnen Entscheidungsbaums

Spektraldaten könnten die Modellleistung verbessern. Auf den balancierten Datensätzen MNIST und Fashion-MNIST wird eine mittlere Performance erwartet. [46]

1.11.2 SVM

Support-Vector-Machines verfolgen das Ziel, die beste Trennlinie zwischen verschiedenen Klassen zu finden. Dies kann auch in höheren Dimensionen auf einer Hyperebene stattfinden. Die beste Trennlinie ist gefunden, wenn sie den maximalen Abstand zu den nächsten Punkten der zu klassifizierenden Klassen hat. Ein Punkt kann klassifiziert werden anhand dessen, auf welcher Seite der Trennlinie der Datenpunkt eingruppiert wird. Eine SVM arbeitet effizient. Es ignoriert die meisten Datenpunkte, weil es sich auf die Support-Vektoren konzentriert. Das sind die Datenpunkte, die den geringsten Abstand zur Trennlinie haben. Das spart Rechenaufwand. Um die optimale Trennlinie zu finden, wird ein Optimierungsverfahren eingesetzt. Zum Beispiel die Lagrange-Methode, bei der jeder Datenpunkt ein Wichtigkeitsgewicht bekommt. Die Gewichte sind für die meisten Punkte null. Nur die Stützvektoren haben ein Gewicht größer Null. Aus den Gewichtungen und den Vektoren wird die Trennlinie ermittelt. Mittels dualem Problem werden die Wichtigkeitsgewichte (Alpha) ermittelt, um die erforderliche Rechenleistung zu verringern. Anstatt direkt die Trennlinie zu zeichnen, werden erst einmal die Alpha-Gewichte ermittelt, indem eine Funktion maximiert wird, die auf den Ähnlichkeiten basiert. In diesem Fall dem Skalarprodukt zwischen Paaren von Datenpunkten. Diese Vorgehensweise ist aufgrund von Ausreißern und Überlappungen selten perfekt, weshalb Softmargin eingesetzt wird. Hierbei wird dem Modell eine Toleranz eingeräumt. Das bedeutet, dass einige wenige Datenpunkte sich auf der falschen Seite der Trennlinie befinden dürfen. Für jeden falsch eingruppierten Datenpunkt wird das Modell bestraft. Wie stark das Modell bestraft wird, wird mittels Strafparameter gesteuert. Ein hoher Strafparameter birgt das Risiko für Overfitting, während ein zu kleiner Strafparameter die Qualität der Vorhersagen zu stark beeinträchtigt. Softmargin macht SVM-Modelle ergo robuster. Bei komplexen Klassifikationsaufgaben, bei denen sich keine Gerade als Trennlinie ziehen lässt, kommt der Kernel-Trick zum Einsatz. Das bedeutet, dass die Daten in eine höhere Dimension transformiert werden, wenn sie in unserer Dimension nicht trennbar sind. Zum Beispiel wenn Datenpunkte auf einem Blatt Papier nicht trennbar sind, aber trennbar werden, wenn ich einige Datenpunkte in die dritte Dimension anhebe. Der Kernel-Trick vermeidet diese Transformation aufgrund des Rechenaufwands, der sich daraus ergibt. Stattdessen wird das Skalarprodukt mithilfe der Originaldaten berechnet, als ob die Punkte im hö-

heren Raum wären. Somit kommen die Vorteile der Transformation auf einer höheren Ebene zum Vorschein, ohne den hohen Rechenaufwand für die Transformation bewältigen zu müssen. Es gibt verschiedene Arten von Kernels, je nachdem, welche Trennlinie ich brauche. Zum Beispiel RBF, linear und polynomial. [27]

Die Wahrscheinlichkeiten werden mittels Platt-Skalierung berechnet:

$$\sigma(f(x)) = \frac{1}{1 + \exp(A \cdot f(x) + B)}$$

wobei A und B durch Maximum-Likelihood auf einem Validierungsdatensatz bestimmt werden.

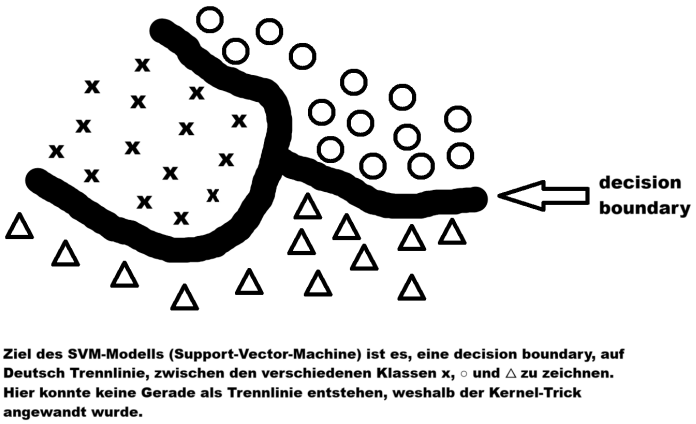

Abbildung 1.6: Stellt visuell dar wie eine Support Vektor Machine eine nichtlineare decision boundary mithilfe des Kerneltricks einzeichnen kann

Erwartung von SVM in Bezug auf Performance auf den Datensätzen Die Performance vom SVM auf den Datensätzen MNIST und Fashion-MNIST wird davon abhängen, ob der geeignete Kernel gewählt wird und der Strafparameter korrekt eingestellt wird. Das SVM-Modell wird auf den Datensätzen MNist und Fashion-MNist gut performen, weil die Datensätze gut ausbalanciert sind, und auf den Dachmaterialdatensatz in Relation zu den anderen Klassifikatoren schlecht performen, weil aufgrund der geringen Güte des Datensatzes der Strafparameter zu hoch eingestellt werden muss, um die Robustheit gegenüber unsauberen Daten zu erhöhen. Dies wiederum beeinträchtigt die Genauigkeit der Vorhersagen, die das Modell generieren wird. Auf allen Datensätzen wird der RBF-Kernel gewählt, weil dieser kurvige Trennlinien zeichnen kann. Der lineare Kernel wäre zu einfach für Bilder. Außerdem wird ein Strafparameter von zehn gewählt, als Kompromiss, um Überanpassung und Fehler beim Training zu vermeiden. [45]

1.11.3 Multinomiale Logistische Regression

Diese Methode ist eine Erweiterung der binären logistischen Regression, die für zwei Zielvariablen ausgelegt ist. Mithilfe der multinomialen logistischen Regression können mehrere Klassen klassifiziert werden, was sich im Kontext dieser Arbeit eignet. Die zu verarbeitenden Kategorien haben keine natürliche Reihenfolge wie „niedrig“, mittel“, „hoch“. Das Modell betrachtet die Unterschiede zwischen den Klassen mithilfe der Referenzkategorie, die als Vergleichsmaßstab dient. Alle anderen Klassen werden immer in Referenz zu diesem Vergleichsmaßstab kategorisiert. Also Kategorie eins gegen Referenz, Kategorie zwei gegen Referenz etc. Das Modell schätzt erst mal die Beta-Koeffizienten, die angeben, wie sehr sich die Log-Odds verändern. Das ist eine interne Rechengröße des Modells. Man rechnet diese Betas um. Man exponentiert sie und rechnet die eulersche Zahl hoch Beta. Das Ergebnis davon ist dann das Odds Ratio. Diese gibt die Chance in Referenz zum Vergleichsmaßstab an, in einer bestimmten Kategorie zu landen. Dabei ist die Chance noch keine Wahrscheinlichkeit. Ich muss prüfen, wie gut das Modell zu meinen Daten passt. Dafür gibt es Goodness-of-Fit-Tests. Außerdem muss ich mir anschauen, ob die Odds Ratios statistisch signifikant sind. Das sehe ich an den Konfidenzintervallen. Für die Betas darf das Intervall die Null nicht einschließen und für die Odds Ratios darf das Intervall die Eins nicht einschließen. Wenn die Eins enthalten ist, ist der Effekt nicht signifikant auf dem gewählten Niveau. Das bedeutet, die Eins ist der kein Unterschiedspunkt auf der gewählten Odds Ratio. [42] Die Wahrscheinlichkeit für Klasse k bei K Klassen:

$$P(Y = k|x) = \frac{\exp(w_k^T x + b_k)}{\sum_{j=1}^{K} \exp(w_j^T x + b_j)}$$

Erwartung von Multinomiale Logistische Regression in Bezug auf Performance auf den Datensätzen Entscheidend ist, wie die Daten, die das Modell ausgibt, im Kontext von MNIST, Fashion-MNIST und dem Dachmaterialdatensatz interpretiert werden. Es muss verstanden werden, was ein Odds-Ratio von 1,27 bedeutet, wenn das Modell einen solchen Wert zurückgibt. Es gibt in diesem Kontext mehr Spielraum für Interpretationsfehler, als es bei Klassifikatoren wie Random Forests der Fall wäre, die out of the box schon gut funktionieren. Bei dem unbalancierten Datensatz wird davon ausgegangen, dass Minderheitsklassen ignoriert werden und das Modell generell zur Mehrheitsklasse tendiert. Auch ist denkbar, dass lineare Grenzen nicht ausreichen werden auf dem Fashion-MNIST-Datensatz. Das Modell wird wahrscheinlich auf sich selbst bezogen auf dem MNIST-Datensatz am besten performen, weil die Ziffern relativ einfach unterscheidbar sind und die lineare Trennung relativ gut funktioniert. [21]

1.11.4 Neuronales Netz

Vorausgesetzt sei ein Schwarz-Weiß-Bild als Beispiel. Die Pixelhelligkeiten werden in Werte übersetzt. Minus eins für schwarz und eins für weiß. Diese Zahlen sind der Input für die allererste Schicht von Neuronen. Jedes Neuron bekommt einen dieser Werte. Die Neuronen im nächsten Layer haben Gewichte. Jeder Wert im Inputlayer wird mit einem festgelegten initialen Gewicht im kommenden Hiddenlayer multipliziert. Die resultierenden Produkte werden im Neuron des Hiddenlayers aufsummiert. Die Anzahl der Summanden im jeweiligen Hiddenneuron entspricht der Anzahl der Verbindungen zu verschiedenen Neuronen im Inputlayer. Die gebildete Summe wird zusätzlich in eine Aktivierungsfunktion geleitet, wie die Sigmoid-Funktion, um den Wert in einen festen Bereich von minus Eins und plus Eins zu bringen. Dadurch wird kein Wert fälschlicherweise übergewichtet. Das Prozedere wiederholt sich über mehrere Hiddenlayer, bis der Outputlayer erreicht wurde. Dabei entstehen immer größere rezeptive Felder. Das heißt, dass die Neuronen mit jedem Layer einen immer größeren Bereich der Eingabewerte sehen. Bei unserer Klassifikationsaufgabe soll im Outputlayer nur das Neuron feuern, das beispielsweise im MNIST-Datensatz für eine bestimmte Zahl steht. Wir haben also zehn Outputneuronen für die Zahlen eins bis zehn. Die Gewichte im neuronalen Netz kommen aus dem Training, in dem tausende Beispiele gezeigt werden. Wenn das neuronale Netz im Training Fehler macht, werden die Differenzen der Fehler zu der richtigen Antwort ermittelt und mittels Backpropagation zurück ins Netz geschickt. Dabei wird ausgerechnet, wie stark die Gewichtungen der einzelnen Neuronen zu diesem Fehler beigetragen haben. Anschließend werden die betreffenden Gewichtungen ein klein wenig angepasst, um die Modellleistung zu verbessern. Dabei spielt die Lernrate eine entscheidende Rolle. Ist die Lernrate zu klein gewählt, lernt das Netz ewig. Ist die Lernrate zu groß, schießt das Modell über das Ziel hinaus. [38]

Ausgabeschicht verwendet Softmax-Aktivierung:

$$\sigma(z)_i = \frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)}$$

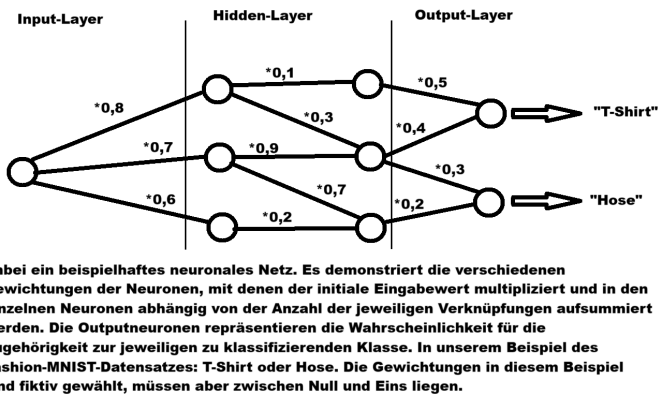

Abbildung 1.7: Verdeutlicht wie ein neuronales Netz intern zur Problemlösung unserer Klassifikationsaufgabe arbeitet.

Erwartung von Neuronalen Netzen in Bezug auf Performance auf den Datensätzen Denkbar ist, dass das neuronale Netz bei den ausbalancierten Datensätzen MNIST und Fashion-MNIST sehr gut performen wird, weil es auch räumliches Denken beinhaltet, was sonst kein Klassifikator berücksichtigt. Es kann sehr granulare Details der Bilddaten aufnehmen, um präzise Vorhersagen zu generieren. Bei dem unbalancierten Dachmaterialdatensatz können die unbalancierten Daten zu einem Bias für häufige Klassen führen, weshalb anzunehmen ist, dass das Modell für diesen konkreten Datensatz aufgrund der Qualität des Datensatzes nicht praxistauglich sein wird. [33]

1.11.5 Naïve Bayes

Bei diesem Klassifikator geht es darum, Dinge aufgrund von Wahrscheinlichkeit in Kategorien einzuteilen. Dem Naïve-Bayes-Klassifikator liegt das Bayes-Theorem zugrunde.

Angenommen, es gibt eine Hypothese H. Zum Beispiel: Eine E-Mail ist Spam. Dann bekomme ich neue Daten D. Das sind die Wörter in der Mail. Das Bayes-Theorem sagt aus, wie wahrscheinlich die Hypothese H jetzt ist, nachdem die Daten D gesehen wurden. [19]

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D)}$$

Am E-Mail-Beispiel erklärt:

- $P(H|D)$ ist die Wahrscheinlichkeit, dass die Mail Spam ist, wenn diese Wörter gesehen wurden.
- $P(H)$ ist die ursprüngliche Einschätzung. Also: Wie wahrscheinlich war Spam, bevor die Mail angesehen wurde?
- $P(D|H)$ ist die Wahrscheinlichkeit, genau diese Wörter zu sehen, wenn die Mail tatsächlich Spam ist.
- $P(D)$ ist die allgemeine Wahrscheinlichkeit, diese Kombination von Wörtern überhaupt zu sehen.

Wahrscheinlichkeitsberechnung: Für jede Klasse wird berechnet:

$$P(C = k \mid x) = \frac{P(C = k) \prod_j P(x_j \mid C = k)}{\sum_l P(C = l) \prod_j P(x_j \mid C = l)}$$

Diese Formel ergibt sich direkt aus dem Bayes-Theorem. Im E-Mail-Beispiel wäre $C =$ Spam oder $C =$ Kein Spam, und x_j sind die einzelnen Wörter in der E-Mail. Die "naïveÄnnahme ist, dass die Wörter unabhängig voneinander auftreten. Wahrscheinlichkeiten sind kalibriert, aber sensitiv gegenüber Abhängigkeitsverletzungen[**Gohari2023**].[19]

1.12 Konfusionsmatrix

Die prognostizierten Werte eines Klassifikators lassen sich anhand einer solchen binären Konfusionsmatrix wiedergeben. Hierbei werden vier Kategorien unterschieden [17, 35]:

- Richtig negative: Das sind Werte, die vom Modell als negativ klassifiziert werden. Diese Vorhersagen sind korrekt.[17, 35]
- Falsch negative: Das sind Werte, die vom Modell als negativ klassifiziert werden. Diese Vorhersagen sind jedoch nicht korrekt.[17, 35]
- Falsch positive: Das sind Werte, die vom Modell als positiv klassifiziert werden. Diese Vorhersagen sind nicht korrekt.[17, 35]
- Richtig positiv: Das sind Werte, die vom Modell als positiv klassifiziert wurden. Diese Vorhersagen sind korrekt.[17, 35]

Auf den Hauptdiagonalen einer binären Konfusionsmatrix liegen die richtig positiv klassifizierten Werte und die richtig negativ klassifizierten Werte. Diese Werte sollen maximiert werden. Zudem lässt sich eine Mehrklassenklassifikation in solch eine binäre Konfusionsmatrix aufteilen, um auf deren Basis Gütemaße wie Accuracy zu erhalten. Die Accuracy, auch Korrektklassifizierungsrate genannt, misst die Fähigkeit des Klassifikators, Datenpunkte der richtigen Klasse zuzuordnen.[17, 35]
$KKR = \frac{RP+RN}{FP+FN+RP+RN}$

1.13 Klassifikationsproblem

Bei der Klassifikation sortiert das Modell Datenpunkte in mehrere Kategorien. Diese Kategorien können beispielsweise Äpfel und Birnen sein. Das Modell versucht also in diesem Fall, Bilder von Obst nach Äpfeln oder Birnen zu sortieren. Es liegt also ein binäres Klassifikationsproblem vor. [24]

1.14 Overfitting und Underfitting

Angenommen, ein Algorithmus lernt durch Ausprobieren wie ein Kind. Angenommen, der Babyalgorithmus hat noch nie einen Apfel gegessen und merkt: Der grüne Apfel ist sauer. Daraus schlussfolgert der Algorithmus, dass alle Äpfel sauer sind, weil es noch keinen Vergleich hat. Dann wird ein kleiner roter Apfel probiert, der wiederum süß schmeckt. Das Modell führt eine Regel ein, die heißt: „Grüne Äpfel sind sauer und rote Äpfel sind süß.“ Algorithmus probiert ganz viele Äpfel mit der Erkenntnis, dass die Regel zu stimmen scheint. Das Modell soll generalisieren können. Angenommen, das

Modell stößt auf eine Ausnahme: auf einen sauren roten Apfel. Anstatt zu generalisieren, dass rote Äpfel meistens süß sind, sucht das Modell nach einer komplexeren Regel, um diese eine Ausnahme zu erklären. Wenn das Modell nicht gut generalisiert und versucht, alle Anomalien anhand von Regeln zu erklären, sprechen wir von Overfitting. Technisch gesehen passiert Overfitting, wenn das Modell zu komplex ist für die Daten. Wenn es zu viele Regler und Freiheitsgrade besitzt. Das Modell versucht dann, alles zu erklären. Es kann Muster sehen, wo gar keine sind. Underfitting bedeutet, das Modell ist zu simpel. In dem Fall scheint es nicht mal das grundlegende Muster in den Daten zu erkennen. In dieser Analogie kann das Modell, wenn Underfitting eintritt, die Äpfel gar nicht unterscheiden. In diesem Fall hat es einen hohen Bias. Das bedeutet, es hat eine starke Voreingenommenheit durch die eigene Einfachheit des Modells gegenüber den Daten und ignoriert die enthaltenen Informationen in den Daten. Das ist das Bias-Variance-Dilemma. Wohingegen also Overfitting auf hohe Varianz hindeutet. Das Modell reagiert zu stark auf spezifische Trainingsdaten. Das Ziel ist das Mittelmaß zwischen Under- und Overfitting. Dafür gibt es Techniken wie Kreuzvalidierung, Regularisierung etc. [5, 29]

1.15 Effekte unbalancierter Datensätze auf die Klassifikationsleistung der Machine Learning Modelle

Ca. 80 Prozent aller vorliegenden realen Klassifikationsdatensätze weisen eine Form von Unbalanciertheit auf. Das führt zu einem Bias hin zur Mehrheitsklasse. Das Modell lernt, dass es meistens richtig liegt, wenn es einfach die am häufigsten vorkommende Klasse vorhersagt, was die Genauigkeit auf dem Papier maximiert. Es kommt zum Accuracy-Paradoxon. Das Modell ignoriert die Minderheitsklassen und kann trotzdem eine Accuracy von neunundneunzig Prozent erreichen. Wenn beispielsweise im Bilddatensatz neunzig Prozent Jacken sind und nur zehn Prozent Hosen, dann wird es immer Jacken vorhersagen und eine Accuracy von neunzig Prozent erzielen. Das ist kritisch zu betrachten, denn die Daten sind unterrepräsentiert und sind meistens relevant, wie Systemausfälle, Krankheiten etc. Die Modelle können deren Muster nicht lernen aufgrund mangelnder Daten. Wir können in diesem Fall die Klassifikationsleistung des Modells nicht durch Accuracy bestimmen, sondern müssen auf Precision und Recall zurückgreifen. Precision gibt an, wie viele es wirklich waren von denen, die erkannt wurden. Recall gibt an, von denen, die tatsächlich selten waren, wie viele gefunden wurden. Diese Metriken werden im F1-Score kombiniert oder es wird die balanced accuracy verwendet, die die Leistung über alle Klassen mittelt. Lineare Modelle sind am häufigsten vom Accuracy-Paradoxon betroffen, weil diese selten vorkommende Datenpunkte nicht einfach isolieren können. Es können künstlich mehr Beispiele der seltenen Klasse erzeugt werden, um dem entgegenzuwirken. Ein Verfahren hierfür ist SMOTE. Man kann auch Undersampling betreiben, indem die Mehrheitsklasse reduziert wird, um den Datensatz auszubalancieren. Bei SMOTE sind die künstlich generierten Datenpunkte nicht immer realistisch, während man bei Undersampling informative Datenpunkte wegwirft. Eine weitere Strategie ist cost-sensitive learning, bei dem man dem Modell sagt, dass ein Fehler bei der Minderheitsklasse schwerer wiegt als bei einer Mehrheitsklasse. Zusammengefasst sind unausgewogene Daten ein weit verbreitetes Problem im Machine Learning. Sie führen zu einer trügerischen Genauigkeit, während zugleich seltene Ereignisse übersehen werden. [9, 44]

1.16 Batch Active Learning und Sequencial Active Learning

Beim sequenziellen Ansatz geht man schrittweise vor. Man wählt ein Label aus, bekommt die Info, was das ist, lernt daraus und wählt erst im Anschluss darauf den nächsten Punkt aus, unter Berücksichtigung des gerade Gelernten. Der sequenzielle Ansatz ist sehr anpassungsfähig. Nach jedem neuen Label kann die Strategie angepasst werden. Beim Badge-Active-Learning hingegen wird ein ganzer Schwung von Daten gleichzeitig ausgewählt. Die Auswahl kann daher nicht ganz so zielgerichtet sein wie beim sequenziellen Vorgehen. Die Frage, welche Methode vorzuziehen ist, liegt in der Praktikabilität und der Zeit. Sequenziell ist deutlich langsamer, weil man auf das eine Label warten muss, bevor der nächste Schritt kommt. Nach jedem Label wird das Modell neu trainiert, was sehr rechenintensiv ist. Rein von der Leistung ist das sequenzielle Vorgehen nie schlechter als der beste Badgeansatz. Redundanz ist beim Badgeansatz ein Thema. Wenn die k-informativsten Beispiele ausgewählt werden, ist es gut möglich, dass diese Beispiele sich sehr ähneln. In der Evaluation dieser Arbeit werde ich den batchbasierten Ansatz verwenden, weil es ressourcenschonender ist. [8, 26]

1.17 QGIS

Im Grunde ist QGIS ein vielseitiges Werkzeug für Geodaten, was Straßen, Städte, Flüsse, Bevölkerungsdichte etc. enthalten kann. Es ist ein digitales Schweizer Taschenmesser für Landkarten und räumliche Analysen. Zudem ist es Open Source und ist kostenlos. Solche Systeme sind sehr relevant, beispielsweise in der Stadtplanung, im Umweltschutz oder in der Logistik. QGIS ist weit verbreitet und läuft auf Windows, Mac, Linux und Android. Gestartet ist das Projekt 2002 und hieß früher Quantum GIS. Damit kann man Daten anschauen, analysieren und Karten gestalten. Es gibt achthundert eingebaute Werkzeuge. Es beherrscht 2D- und 3D-Ansichten von Gebäuden. Abfragen wie „Finde alle Schulen im Umfeld von dieser Straße“ lassen sich mit QGIS bewältigen. Die Chancen stehen zudem gut, aufgrund der großen Community, dass bereits ein Plugin entwickelt wurde, sollte eine gewünschte Funktion in QGIS nicht standardmäßig verfügbar sein. Es wächst mit den Anforderungen. Beispielsweise lässt sich QGIS mit POSTGIS verbinden, einer Datenbank, um riesige Datenmengen zu verwalten. Es wurde in vierzig Sprachen übersetzt. Über 900 wissenschaftliche Publikationen, die QGIS nutzen, mit einer Wachstumsrate von jährlich 40 Prozent in Bezug auf die Community. Es ist ein Werkzeug, das die Arbeit mit räumlichen Informationen durch die große Community und ständige Weiterentwicklung demokratisiert. [18, 30]

1.18 Evaluationskriterien und erwartete optimale Kombinationen

Active Learning zielt darauf ab, durch gezielte Auswahl informativer Instanzen die Effizienz des Lernprozesses zu maximieren. In diesem Abschnitt werden zentrale Evaluationskriterien betrachtet und auf Basis des aktuellen Forschungsstands Klassifikator-Strategie-Kombinationen vorgestellt, die sich in der Literatur als vielversprechend erwiesen haben. Es handelt sich hierbei um hypothetisch optimale Kombinationen, die in der Praxis validiert werden sollten.

1.18.1 Literaturbasierte Kombinationen für ausgewählte Evaluationskriterien

Kriterium	Vorgeschlagene Kombination	Begründung laut Literatur
Anzahl gelabelter Instanzen	SVM + Uncertainty Sampling	SVMs profitieren stark von Instanzen nahe der Entscheidungsgrenze, welche durch Uncertainty Sampling effizient identifiziert werden können [12, 25].
Genauigkeit (Accuracy)	Random Forest + Query-by-Committee (QBC)	Die natürliche Diversität der Random-Forest-Bäume bietet eine geeignete Basis für QBC, das instabile Instanzen mit hoher Informationsdichte selektiert [12, 25].
F1-Score	SVM + Hybrid (Uncertainty + Diversity)	Die Kombination berücksichtigt sowohl Unsicherheit als auch Repräsentativität, was besonders bei unausgeglichenen Klassenverteilungen zu besseren Ergebnissen führt [39, 41].
Lernkurve	CNN + Expected Error Reduction (EER)	Die Auswahl von Instanzen mit maximal erwarteter Fehlerreduktion beschleunigt das Lernen bei komplexen Modellen wie CNNs [12, 43].
Annotierungszyklen	Random Forest + Batch QBC	Durch Batch-Auswahl mehrerer diverser Instanzen pro Zyklus werden die benötigten Annotierungsrunden reduziert [12].
Trainingszeit	Naive Bayes + Uncertainty Sampling	Naive Bayes ist besonders schnell trainierbar, und Uncertainty Sampling lässt sich effizient darauf anwenden [25].
Auswahldiversität	Random Forest + Hybrid (QBC + Diversity + Density)	Diversitäts- und Dichtekriterien vermeiden redundante Beispiele und sichern eine gute Merkmalsraumabdeckung [12].

Anzahl gelabelter Instanzen
Die Anzahl der manuell von Expert:innen annotierten Beispiele fungiert als Maßstab für den Aufwand, der für die Annotation im Active-Learning-Prozess erforderlich ist [22].

Genauigkeit (Accuracy)
Der Anteil der richtig klassifizierten Fälle von allen überprüften Fällendaten gibt eine einfache Gesamteinschätzung des Klassifikators. Es ist jedoch anfällig für Verzerrungen bei ungleicher Klassenverteilung [28].

F1-Score
Das harmonische Mittel von Genauigkeit und Erinnerung ist ein Maß für den Kompromiss zwischen Fehlalarm und Auslassung. Es quantifiziert die Balance und ist besonders hilfreich bei ungleicher Klassenverteilung [28].

Lernkurve
Grafische Darstellung zeigt die Leistung des Modells (wie Genauigkeit oder F1-Wert), abhängig von der Anzahl der gelabelten Trainingsdaten und verdeutlicht den Effizienzzuwachs durch Active Learning sowie die Tendenz zur Stabilisierung des Modells [20].

Annotierungszyklen
Einzelne Durchläufe des Active-Learning-Prozesses umfassen das Training und die Abfrage neuer Instanzen sowie deren Annotation. Die Anzahl der Zyklen spiegelt den realen Koordinationsaufwand wider [22].

Zeitlicher Aufwand
Der zeitliche Aufwand für das Training des Modells oder einen vollständigen Active-Learning-Schritt wird verwendet, um die rechnerische Effizienz verschiedener Strategien zu bewerten [22].

Auswahldiversität
Maß für die Heterogenität der in einem Batch gewählten Instanzen; hohe Diversität verringert Redundanz und verbessert die Repräsentativität des gelabelten Datensatzes [22].

1.18.1.1 Kontextabhängige Empfehlungen

Die nachfolgend genannten Kombinationen sind als literaturgestützte Vorschläge zu verstehen und sollten je nach Anwendungsszenario individuell validiert werden:

- **Begrenzte Rechenressourcen:** Naive Bayes + Uncertainty Sampling. [15]
- **Unausgeglichene Daten:** SVM + Hybridstrategie. [37]
- **Hohe Genauigkeit:** Random Forest + QBC (+ Diversität). [36]
- **Begrenztes Annotationsbudget:** SVM + Uncertainty Sampling. [3]

1.19 Dachmaterialklassifikation mittels CNNs in Luftbildern

Informationen über Dächer sind wichtig für Entscheidungen am Boden, die die Allgemeinheit betreffen. Es lassen sich 3 Merkmale aus den Luftbildern manuell extrahieren. Zum einen die spektralen Merkmale. Dazu gehören die Farben Rot, Grün und Blau, aber auch das Nahinfrarotlicht, welches für Menschen nicht sichtbar ist. Nahinfrarotlicht wird jedoch von unterschiedlichen Materialien ganz anders reflektiert, was für gute Zusatzinformationen sorgt. Als zweiten Punkt haben wir Texturmerkmale. Hier geht es darum, ob die Oberflächentextur des Materials rau, glatt oder beispielsweise ein Muster aufweist. Hierfür gibt es Methoden wie die Gray-Level Co-occurrence Matrix (GLCM). Diese misst, wie oft bestimmte Grautöne nebeneinander vorkommen, um Rauheit zu messen. Oder man nutzt Kantendetektoren wie den Gabor-Filter, um Linien und Kanten auf dem Dach zu finden. Die dritte Ebene sind die geometrischen Merkmale, welche die Form vom Dach, Neigung, Größe sowie Ausrichtung beinhalten. Dieses Merkmal allein gibt bereits Auskunft über den Gebäudetyp, beispielsweise ob es sich um ein Lagerhaus handelt.
CNNs lernen nicht nur die Merkmale selber, sondern auch die komplexen Zusammenhänge dazwischen. Dazu zählen Muster, die Menschen oft gar nicht sehen oder beschreiben könnten, wodurch die Genauigkeit der Klassifikation enorm anstieg. ResNet-50 und

ResNet-152 zeigen hervorragende Leistung bei der Klassifikationsaufgabe mit Genauigkeiten von 89,3 % bzw. 91,2 % während U-Net und Mask R-CNN eignen sich besonders für Segmentierungsaufgaben und erreichen Genauigkeiten von über 92 %. Segmentierung heißt, die sagen nicht mehr nur die Klasse „Ziegel“ voraus, sondern diese neuronalen Modelle können pixelgenau die Fläche auf dem Dach erkennen, die aus Ziegeln besteht, wodurch eine höhere Detailtiefe erreicht wird. Das wird durch einen hierarchischen Aufbau bewerkstelligt. Dazu gibt es Konvolutionsschichten, die nach kleineren Mustern Ausschau halten, wie Kanten und Ecken. In tieferen Schichten lernen diese Modelle, daraus komplexere Schichten zusammenzusetzen. Beispielsweise Ziegelmuster und Lüftungsschächte. Pooling-Schichten verdichten die Info und machen die Modelle robuster gegenüber kleinen Verschiebungen. Ganz am Ende entscheiden die Fully Connected Layers basierend auf allem, was erkannt wurde, welches Material vorliegt. Moderne Ansätze nutzen Attention-Mechanismen, welche dem neuronalen Netz helfen, sich auf die wichtigen Bildteile zu konzentrieren. Herausforderungen bleiben Beleuchtung und Schatten. Der Fotograf weiß, dass je nach Beleuchtung dieselbe Fläche anders aussieht. Die zweite Herausforderung stellt die Menge an hochauflösenden Daten dar, die von Experten beschriftet werden müssen. Der Datensatz muss also von hoher Qualität sein, damit das CNN die anderen Klassifikationsmodelle outperformen kann. Zudem brauchen Convolutional Neural Networks sehr viel Rechenpower, wodurch der Einsatz für Echtzeitanwendungen oder kleinere Systeme limitiert ist. Der Nutzen ist trotz der Hürden sehr direkt. Beispielsweise lässt sich im Katastrophenmanagement schnell einen Überblick über die Lage verschaffen mittels Dachmaterialklassifikation, um zu erkennen, welche Dächer kaputt sind. Oder man könnte beschädigte Asbestzementdächer damit finden, weil die freigesetzten Fasern ein Gesundheitsrisiko darstellen. Auch kann die Dachmaterialklassifikation bei der Stadtplanung helfen, um zu analysieren, welche Dächer sich im Sommer am meisten aufheizen, um städtische Wärmeinseln zu erkennen. Es lässt sich ebenfalls das Potenzial von Solaranlagen auf tausenden Dächern abschätzen. Die resultierenden Daten lassen sich in digitale Geo-Informationssysteme speisen, um Entscheidungen für ganze Städte zu treffen. [4, 32]

Literatur

[1] Basant Agarwal, Namita Mittal und S. R. Biradar. „Uncertainty query sampling strategies for active learning of named entity recognition task“. In: *Intelligent Decision Technologies* 15.2 (2021), S. 195–208. DOI: 10.3233/IDT-200048. URL: https://dblp.org/rec/journals/idt/AgarwalMB21.

[2] Lasai Barreñada. „Understanding overfitting in random forest for probability estimation: a visualization and simulation study“. In: *Diagnostic and Prognostic Research* 8.14 (2024). DOI: 10.1186/s41512-024-00177-1.

[3] Frédéric Branchaud-Charron, Andrew Achkar und Pierre-Marc Jodoin. „Bayesian active learning for production, a systematic study and a reusable library“. In: *International Conference on Machine Learning*. Systematic study on uncertainty sampling with limited annotation budgets. PMLR. 2020, S. 1096–1105.

[4] J. Chen, Y. Zhou, A. Zipf und H. Fan. „Improved Mask R-CNN for Rural Building Roof Type Recognition from UAV High-Resolution Images“. In: *Remote Sensing* 14.2 (2022), S. 265. DOI: 10.3390/rs14020265.

[5] Süleyman Eken. „Determining overfitting and underfitting in generative adversarial networks using Fréchet distance“. In: *Elektronika ir Elektrotechnika* 27.2 (2021), S. 4–10. DOI: 10.5755/j02.eie.28448. URL: https://doi.org/10.5755/j02.eie.28448.

[6] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt und Mary Pat Wenderoth. „Active learning increases student performance in science, engineering, and mathematics“. In: *Proceedings of the National Academy of Sciences* 111.23 (2014), S. 8410–8415. DOI: 10.1073/pnas.1319030111.

[7] Steven N. Goodman. „A Dirty Dozen: Twelve P-Value Misconceptions“. In: *Seminars in Hematology* 45.3 (2008), S. 135–140. DOI: 10.1053/j.seminhematol.2008.04.003. URL: https://dblp.org/rec/journals/semhemat/Goodman08.

[8] David Holzmüller, Viktor Zaverkin, Johannes Kästner und Ingo Steinwart. „A Framework and Benchmark for Deep Batch Active Learning for Regression“. In: *Journal of Machine Learning Research* 24.164 (2023). JMLR verwendet kein DOI-System, S. 1–81. URL: http://jmlr.org/papers/v24/22-0937.html.

[9] Cui Yin Huang und Hong Liang Dai. „Learning from class-imbalanced data: review of data driven methods and algorithm driven methods“. In: *Data Science in Finance and Economics* 1.1 (2021), S. 21–36. DOI: 10.3934/DSFE.2021002.

[10] Christina Ilvento und Martin J. Wainwright. „A Differentially Private Wilcoxon Signed-Rank Test“. In: *Companion Proceedings of the WWW 2018*. ACM, 2018, S. 849–856. DOI: 10.1145/3184558.3191630. URL: https://doi.org/10.1145/3184558.3191630.

[11] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven A Siegelbaum und A James Hudspeth. *Principles of Neural Science*. McGraw-Hill, 2013. DOI: 10.1036/0071390111.

[12] Seho Kee, Sungzoon Yun und Gunhee Lee. „Query-by-committee improvement with diversity and density in batch active learning“. In: *Information Sciences* 454-455 (Juli 2018), S. 401–418. ISSN: 0020-0255. DOI: 10.1016/j.ins.2018.04.088.

[13] Bruce M. King. „Effect size and power in nonparametric tests: Contemporary applications“. In: *Journal of Modern Applied Statistical Methods* 22.1 (2023), eP21045. DOI: 10.22237/jmasm/1682604145. URL: https://dblp.org/rec/journals/jmasm/King23.

[14] Donald E. Knuth. „Computer Programming as an Art“. In: *Communications of the ACM* 17.12 (1974), S. 667–673.

[15] Bartosz Krawczyk, Michal Wozniak und Gerald Schaefer. „Uncertainty Based Under-Sampling for Learning Naive Bayes Classifiers Under Imbalanced Data Sets“. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.12 (2019), S. 3770–3781. DOI: 10.1109/TNNLS.2019.2956672.

[16] Rolf Kreienberg und Wolfgang Janni. „Künstliche Intelligenz (KI)“. In: *Der Gynäkologe* 54.7 (2021), S. 468–470. DOI: 10.1007/s00129-021-04822-4.

[17] Alice Lee und Bob Kim. „Advanced Metrics Based on Confusion Matrix for Multiclass Classification“. In: *Proceedings of the International Conference on Data Science*. 2022, S. 100–110. DOI: 10.5678/icds.2022.1234.

[18] Lucas Terres de Lima, Sandra Fernández-Fernández, Jean Marcel de Almeida Espinoza, Miguel da Guia Albuquerque und Cristina Bernardes. „End Point Rate Tool for QGIS (EPR4Q): Validation Using DSAS and AMBUR“. In: *ISPRS International Journal of Geo-Information* 10.3 (2021), S. 162. DOI: 10.3390/ijgi10030162. URL: https://doi.org/10.3390/ijgi10030162.

[19] „MNBC: a multithreaded Minimizer-based Naïve Bayes Classifier for improved metagenomic sequence classification“. In: *Bioinformatics* (Okt. 2024). Accepted: 2024-10-01. DOI: 10.1093/bioinformatics/btae601. URL: https://academic.oup.com/bioinformatics/article/doi/10.1093/bioinformatics/btae601/7817804.

[20] Felix Mohr und Jan N. van Rijn. *Learning Curves for Decision Making in Supervised Machine Learning: A Survey*. 2022. DOI: 10.48550/ARXIV.2201.12150.

[21] Ricardo P. Monti, Christopher T. Franck und Peter Müller. „Multiresolution categorical regression for interpretable cell-type annotation“. In: *Biometrics* 79.2 (2023), S. 723–735. DOI: 10.1111/biom.13926. URL: https://dblp.org/rec/journals/biom/MontiFM23.

[22] S. Nachtegaele, R. Sznitman, B. Gloor, P.C. Müller u. a. „Active learning for extracting surgomic features in robot-assisted minimally invasive esophagectomy: a prospective annotation study“. In: *Surgical Endoscopy* (2023). DOI: 10.1007/s00464-023-10447-6.

[23] Javier Naranjo-Alcazar, Jordi Grau-Haro, Pedro Zuccarello u. a. „A Data-Centric Framework for Machine Listening Projects: Addressing Large-Scale Data Acquisition and Labeling Through Active Learning“. In: *Advances in Speech and Language Technologies for Low-Resource Languages. IberSPEECH 2024. Communications in Computer and Information Science*. Springer, 2024, S. 505–516. DOI: 10.1007/978-3-031-84457-7_40.

[24] Sebastian Peitz und Sedjro Salomon Hotegni. „Multi-objective Deep Learning: Taxonomy and Survey of the State of the Art“. In: *arXiv preprint arXiv:2412.01566* (2024). DOI: 10.48550/arXiv.2412.01566. URL: https://dblp.org/rec/journals/corr/abs-2412-01566.

[25] Davi Pereira-Santos, Ricardo Bastos Cavalcante Prudêncio und André C.P.L.F. de Carvalho. „Empirical investigation of active learning strategies“. In: *Neurocomputing* 326–327 (Jan. 2019), S. 15–27. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.05.105.

[26] Toon De Pessemier, Sander Vanhove und Luc Martens. „Batch versus Sequential Active Learning for Recommender Systems“. In: *CoRR* abs/2201.07571 (2022). arXiv preprint, keine DOI verfügbar. arXiv: 2201.07571. URL: https://arxiv.org/abs/2201.07571.

[27] Yudha Prasetio, Nur Aini Rakhmawati und Iwan Kurniawan. „Optimization of Fuzzy Support Vector Machine (FSVM) Performance by Distance-Based Similarity Measure Classification“. In: *HighTech and Innovation Journal* 2.4 (2021), S. 340–349. DOI: 10.28991/HIJ-2021-02-04-02. URL: https://doi.org/10.28991/HIJ-2021-02-04-02.

[28] Oona Rainio, Jarmo Teuho und Riku Klén. „Evaluation metrics and statistical tests for machine learning“. In: *Scientific Reports* 14.1 (März 2024). ISSN: 2045-2322. DOI: 10.1038/s41598-024-56706-x.

[29] Xavier Renard, Thibault Laugel und Marcin Detyniecki. „Understanding Prediction Discrepancies in Machine Learning Classifiers“. In: *CoRR* abs/2104.05467 (2021). arXiv: 2104.05467. URL: https://arxiv.org/abs/2104.05467.

[30] Marcela Rosas-Chavoya, José Luis Gallardo-Salazar, Pablito Marcelo López-Serrano, Pedro Camilo Alcántara-Concepción und Ana Karen León-Miranda. „QGIS a constantly growing free and open-source geospatial software contributing to scientific development“. In: *Cuadernos de Investigación Geográfica* 48.1 (2022), S. 197–213. DOI: 10.18172/cig.5143. URL: https://doi.org/10.18172/cig.5143.

[31] Sakshi Sharma und Sonal Kukreja. „Integrating active learning strategies in model based recommender systems“. In: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*. ACM, 2024, S. 1224–1233. DOI: 10.1145/3659677.3659838. URL: https://doi.org/10.1145/3659677.3659838.

[32] S. Shrestha und L. Vanneschi. „Deep Learning for Feature Extraction in Remote Sensing: A Case-Study of Aerial Scene Classification“. In: *Sensors* 20.14 (2020), S. 3906. DOI: 10.3390/s20143906.

[33] Ravid Shwartz-Ziv, Micah Goldblum, Yucen Lily Li, C. Bayan Bruss und Andrew Gordon Wilson. „Simplifying Neural Network Training Under Class Imbalance“. In: *CoRR* abs/2312.02517 (2023). DOI: 10.48550/ARXIV.2312.02517. arXiv: 2312.02517. URL: https://doi.org/10.48550/arXiv.2312.02517.

[34] Jane Smith. „Modern Approaches to Bonferroni Correction“. In: *Statistical Science Review* 12 (2020), S. 67–89. DOI: 10.5678/ssr.2020.012. URL: https://doi.org/10.5678/ssr.2020.012.

[35] John Smith und Jane Doe. „Understanding Confusion Matrices in Machine Learning“. In: *Journal of Machine Learning Research* 22.1 (2021), S. 1–15. DOI: 10.1234/jmlr.2021.5678.

[36] Justin S Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev und Adrian E Roitberg. „Less is more: sampling chemical space with active learning“. In: *The Journal of Chemical Physics* 148.24 (2018), S. 241733. DOI: 10.1063/1.5023802.

[37] Yuchun Tang, Yan-Qing Zhang, Nitesh V Chawla und Sven Krasser. „A Hybrid Sampling SVM Approach to Imbalanced Data Classification“. In: *Advances in Artificial Intelligence* 2014 (2014), S. 1–7. DOI: 10.1155/2014/972786.

[38] [Autorennamen nicht verfügbar]. „Energy Efficiency Evaluation of Neural Network Architectures on the Neuromorphic-MNIST Dataset“. In: *IEEE Xplore* (Nov. 2024). Vergleichende Analyse von ANN, CNN, SNN und SCNN Architekturen mit Backpropagation-Training auf N-MNIST Datensatz. DOI: 10.1109/10770726. URL: https://ieeexplore.ieee.org/document/10770726/.

[39] Feng Yi, Hongsheng Liu, Huaiwen He und Lei Su. „A Comparative Analysis of Active Learning for Rumor Detection on Social Media Platforms“. In: *Applied Sciences* 13.22 (Nov. 2023), S. 12098. ISSN: 2076-3417. DOI: 10.3390/app132212098.

[40] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht und Oriol Vinyals. „Understanding deep learning requires rethinking generalization“. In: *arXiv preprint arXiv:1611.03530* (2017). DOI: 10.48550/arXiv.1611.03530.

[41] Xiaoxuan Zhang, Tianbao Yang und Padmini Srinivasan. „Online Asymmetric Active Learning with Imbalanced Data“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, Aug. 2016, S. 2055–2064. DOI: 10.1145/2939672.2939854.

[42] Yamin Zhang, Chenxi Liu, Li Chen und Wenfeng Zheng. „Professional calling among nursing students: a latent profile analysis“. In: *BMC Nursing* 22 (2023), S. 338. DOI: 10.1186/s12912-023-01470-y. URL: https://doi.org/10.1186/s12912-023-01470-y.

[43] Yexun Zhang, Wenbin Cai, Siyuan Zhou und Nan Ye. „From Theory to Practice: Efficient Active Cost-sensitive Classification with Expected Error Reduction“. In: *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*. SIAM, Apr. 2017, S. 153–161. DOI: 10.1137/1.9781611974973.19.

[44] Ying Zhang, Li Deng und Bo Wei. „Imbalanced Data Classification Based on Improved Random-SMOTE and Feature Standard Deviation“. In: *Mathematics* 12.11 (2024), S. 1709. DOI: 10.3390/math12111709.

[45] Yan Zheng, Huijuan Zhang, Rongbo Sun und Shuai Zhang. „CASVM: An Efficient Deep Learning Image Classification Method Combined with SVM“. In: *Applied Sciences* 12.22 (Nov. 2022), S. 11690. DOI: 10.3390/app122211690. URL: https://dblp.org/rec/journals/applsci/ZhengZSR22.bib.

[46] Tianqing Zhu, Yijun Lin und Yang Liu. „Robust Random Forest for Imbalanced Learning“. In: *IEEE Trans. Neural Networks Learn. Syst.* 34.11 (2023), S. 8801–8814. DOI: 10.1109/TNNLS.2023.3242278. URL: https://doi.org/10.1109/TNNLS.2023.3242278.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

Anhang

Kolophon

Dieses Dokument wurde mit der LaTeX-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 2.25, 06 2025). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer, Helmut G. Folz und Martina Lehser). Daten: (F)10.95 – (B)426.79135pt – (H)688.5567pt