Evaluationsmetriken von ML Algorithmen für Clustering, Klassifikation und Regression

Ausarbeitung

von

Jens Feser

Matrikelnummer, Kurs: 8658477, WWI22SEB

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Ausarbeitung mit dem Thema:

Evaluationsmetriken von ML Algorithmen für Clustering, Klassifikation und Regression

gemäß § 5 der „Studien- und Prüfungsordnung DHBW Technik“ vom 29. September 2017 selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Mannheim, den 1. Juli 2025

Feser, Jens

Disclaimer

Der Umfang dieser Arbeit von 14 Seiten ergibt sich aus der Vielzahl an umfangreichen mathematischen LaTeX-Darstellungen sowie den zahlreichen Beispielen, die zur Veranschaulichung der behandelten Konzepte dienen.
Der Quellcode für das Projekt sowie die zugehörige Demo sind auf GitHub zu finden:
https://github.com/Jens011203/Evaluierung_Integrationsseminar

Inhaltsverzeichnis

Abkürzungsverzeichnis

DBI	Davies-Bouldin-Index
ARI	Adjusted Rand Index
AMI	Adjusted Mutal Information
MI	Mutal Information
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ROC	Receiver Operating Characteristic
AUC	Area under the Curve
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error

Abbildungsverzeichnis

1. Einleitung

Die Evaluierung von Machine-Learning-Modellen erfolgt anhand geeigneter Metriken. Dabei hängt die Wahl der Metriken vom spezifischen Aufgabentyp ab (Clustering, Klassifikation oder Regression). Gute Metriken legen fest, wie Qualität im jeweiligen Kontext gemessen wird. So strebt man beim Clustering dichte, gut getrennte Cluster an, bei der Klassifikation eine hohe Übereinstimmung mit den Klassenlabels, und bei der Regression geringe Abweichungen der Vorhersagen vom wahren Wert. [1] legt dar, dass das Verständnis und die richtige Auswahl von Metriken entscheidend sind, um die Modellleistung objektiv zu beurteilen und Fehlinterpretationen zu vermeiden.

Die vorliegende Arbeit soll einen Überblick über die wichtigsten Evaluationsmetriken in diesen Zusammenhang liefern. Sie soll dabei als Grundlage dienen, welche Evaluationsmetriken für die verschiedenen ML-Algorithmen verwendet werden können.

Im Folgenden werden die wichtigsten Metriken für Clustering, Klassifikation und Regression vorgestellt und hinsichtlich ihrer Stärken und Schwächen diskutiert.

2. Clustering

Clustering bezeichnet Algorithmen im Bereich des *unsupervised Learnings*, welche Datenpunkte anhand ihrer Ähnlichkeit zueinander in Gruppen bzw. Cluster ordnen. Gute Cluster zeichnen sich durch eine hohe Kohäsion innerhalb eines Clusters und gute Separation zwischen den Clustern aus. Typische Evaluationsziele sind daher, kompakte, dicht gepackte Cluster zu bilden, die gleichzeitig weit voneinander getrennt sind (geringe Intra-Cluster-Distanz, hohe Inter-Cluster-Distanz). Bei der Evaluierung von Clustering-Algorithmen gibt es zwei Möglichkeiten (vgl. [1]):

- **extrinsische Metriken:** Ist eine sogenannte *Ground Truth* (also die wahre Zuordnung der Daten in Cluster) verfügbar, kann diese zu Evaluierung der Algorithmen verwendet werden. Extrinsische Metriken erzielen gute Werte, wenn die vorhergesagten Cluster sehr ähnlich zu den tatsächlichen Clustern sind.
- **intrinsische Metriken:** Diese werden verwendet, wenn keine *Ground Truth* in den Daten vorhanden ist. Zur Bewertung der Algorithmen wird die Ähnlichkeit von Punkten im selben Cluster mit der Ähnlichkeit zu anderen Clustern verglichen. Intrinsische Metriken liefern gut Werte, wenn die Intra-Cluster-Ähnlichkeit größer ist als Inter-Cluster-Ähnlichkeit

Das Beispiel mit komplexen halbmondförmigen Clustern zeigt die Schwäche von intrinsischen Metriken auf. Die intrinsischen Metriken bewerten den Clustering-Algorithmus der die Cluster möglichst kugelförmig vorhersagt besser, obwohl der andere Algorithmus die Cluster perfekt vorhersagt. Deshalb wenn eine *Ground Truth* vorhanden ist, sollten eher extrinsische Metriken verwendet werden.

https://github.com/Jens011203/Evaluierung_Integrationsseminar/blob/main/src/Clustering.ipynb

Im Folgenden werden relevante Metriken vorgestellt:

2.1. Intrinsische Metriken

2.1.1. Silhouette-Koeffizient

Der Silhouette-Koeffizient bewertet für jeden Datenpunkt i dessen Zugehörigkeit zu einem Cluster. Sei $a(i)$ der durchschnittliche Abstand des Objekts i zu allen anderen Punkten desselben Clusters, und $b(i)$ der kleinste durchschnittliche Abstand zu Punkten zum nächstgelegenen Fremdcluster. Dann ist der Silhouette-Wert definiert als

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

wobei $s(i) \in [-1, 1]$. Werte nahe 1 bedeuten, dass i gut zum eigenen Cluster passt und weit von anderen Clustern entfernt ist, Werte nahe 0 deuten auf überlappende Cluster hin und negative Werte darauf, dass i eher falsch zugeordnet. Der Gesamtsilhouette-Index wird als Mittelwert $\overline{s} = \frac{1}{N} \sum_i s(i)$ über alle Objekte berechnet (vgl. [2] S.55f).

Beispiel: Betrachten wir eine stark vereinfachte Situation mit zwei Clustern A und B (siehe 2.1):

- Cluster A enthält die beiden Punkte x_1 und x_2.
- Cluster B enthält die beiden Punkte y_1 und y_2.
- Punkt i ist aktuell Cluster A zugeordnet.

Die Abstände betragen:

$$d_1 = \|i - x_1\| = 1.3, \quad d_2 = \|i - x_2\| = 0.7, \quad d_3 = \|i - y_1\| = 2.8, \quad d_4 = \|i - y_2\| = 3.2.$$

Somit ergeben sich folgende mittlere Abstände:

$$a(i) = \frac{d_1 + d_2}{2} = \frac{1.3 + 0.7}{2} = 1 b(i) = \frac{d_3 + d_4}{2} = \frac{2,8 + 3.2}{2} = 3.$$

Somit ergibt sich

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} = \frac{3 - 1}{3} \approx 0.66.$$

Ein Wert $s(i) \approx 0.66$ zeigt, dass i relativ gut in Cluster A integriert ist, da $a(i) \ll b(i)$.

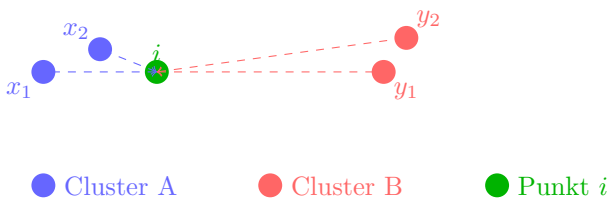

Abbildung 2.1.: Beispiel einer einfachen Clusterzuordnung

Kritische Einordnung: Ein Nachteil des Silhouette-Index ist, dass er runde und gleichmäßig dichte Cluster und runde Cluster annimmt. Bei stark ungleich geformten oder verrauschten Clustern kann er daher irreführende Werte liefern (Siehe komplexeres Code-beispiel im verlinkten Beispiel mit halbmondförmigen Cluster: https://github.com/Jens011203/Evaluierung_Integrationsseminar/blob/main/src/Clustering.ipynb). Außerdem reagiert er nicht auf Merkmals-Bias: Trennt ein Algorithmus auf Basis irrelevanter Variablen, bleibt dies im Silhouette-Score unentdeckt (vgl. [1]). Deswegen wird diese Metrik vor allem genutzt wenn keine wahren labels in den Daten vorhanden sind.

2.1.2. Davies-Bouldin-Index

Der Davies-Bouldin-Index (DBI) misst die Ähnlichkeit zwischen jedem Cluster und dem ähnlichsten Fremdcluster. Für jeden Cluster i definiert man s_i als den durchschnittlichen Abstand der Punkte von ihrem Cluster-Zentroid (also die Varianz im Cluster), und für Clusterpaare (i, j) den Abstand d_{ij} zwischen den Zentroiden. Dann berechnet man für jedes Paar $R_{ij} = (s_i + s_j)/d_{ij}$ und wählt für jeden Cluster i den größten Wert $\max_{j \neq i} R_{ij}$. Der Davies-Bouldin-Index ist das Mittel dieser Maxima:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \frac{s_i + s_j}{d_{ij}},$$

mit k der Anzahl der Cluster. Ein niedriger DBI bedeutet bessere Cluster (maximale Separation und geringe Streuung). Der Wert kann theoretisch bis 0 reichen (perfekte Trennung) (vgl. [3] S.224f).

Beispiel: Nehmen wir zwei Cluster mit Zentroiden c_1 und c_2, deren Streuungen $s_1 = 1$ und $s_2 = 0{,}5$ sind, und Abstand $d_{12} = 2$. Dann ist

$$R_{12} = \frac{s_1 + s_2}{d_{12}} = \frac{1 + 0.5}{2} = 0.75,$$

was für dieses Paar den hohen Ähnlichkeitswert beiträgt und den DBI erhöht. Befänden sich die Zentroiden jedoch bei $d_{12} = 5$, so wäre

$$R_{12} = \frac{1 + 0.5}{5} = 0.3,$$

und der DBI bliebe gering (gutes Clustering).

Kritische Einordnung: Der DBI ist ein effizientes, intrinsisches Maß, bei dem kleinere Werte eine bessere Clusterstruktur anzeigen (Minimum = 0). Im Vergleich zum Silhouette-Index ist die Berechnung deutlich schneller und daher für großvolumige Datensätze sehr gut geeignet. Nachteil: Er setzt euklidische Abstände zwischen Clusterzentroiden voraus und kann bei spärlichen oder nicht-euklidischen Daten irreführende Ergebnisse liefern, wenn alternative Distanzmaße besser zur Struktur der Daten passen. (vgl. [1]). Zudem kann er wie der Silhoutte-Koeffizient bei komplexeren Cluster-Strukturen verfälschte Werte liefern.

2.1.3. Calinski-Harabasz-Index

Der Calinski-Harabasz-Index (auch Variance Ratio Criterion) setzt die Streuung zwischen den Clustern ins Verhältnis zur Streuung innerhalb der Cluster (vgl. [4]). Formal sei N die Gesamtzahl der Punkte,

k die Anzahl der Cluster, B die zwischen-Cluster-Varianz (Summe der quadrierten Abstände der Cluster-Zentren zum Gesamtdurchschnitt) und W die intra-Cluster-Varianz (Summe der quadrierten Abstände der Punkte zu ihren jeweiligen Zentren). Dann ist der CH-Index definiert als

$$CH = \frac{B/(k-1)}{W/(N-k)} = \frac{B}{W} \cdot \frac{N-k}{k-1}$$

Ein höherer Wert zeigt klarere Trennung und dichtere Cluster gute Klassifikation, während ein kleinerer Wert auf schlechtere Partitionen (vgl. [4]).

Beispiel: Sind die Cluster perfekt kompakt ($W \to 0$), wächst CH ins Unendliche. Bei nur einem Cluster ($k = 1$) definiert man meist $CH = 0$. Üblicherweise wählt man das k, für das CH am größten ist.

Kritische Einordnung: Der CH-Index ist schnell berechenbar und eignet sich für große Datensätze. Wie auch die anderen intrinsischen Metriken belohnt er kompakte, klar getrennte Cluster, kann aber bei sehr vielen kleinen Clustern hohe Werten liefern und ist nur im Vergleich unterschiedlicher k-Lösungen sinnvoll (vgl. [1]).

2.2. Extrinsische Metriken

2.2.1. Purity

Die Purity eines Clusters C_i mit n_i Datenpunkten und $n_h^{(i)}$ Datenpunkten der dominanten (wahrheitsgemäßen) Klasse ist definiert als vgl. [5] S.54:

$$P(C_i) = \frac{1}{n_i} \max_h n_h^{(i)}.$$

Der Gesamt-Purity-Score eines Clusterings ist das gewichtete Mittel über alle k Cluster:

$$\text{Purity} = \sum_{i=1}^{k} \frac{n_i}{N} P(C_i).$$

Beispiel: Ein Vorgeschlagenes Cluster C_1 enthält 10 Punkte, davon 7 aus Klasse A. Dann gilt

$$P(C_1) = \frac{7}{10} = 0{,}7.$$

Kritische Einordnung: Purity ist sehr einfach und anschaulich, aber durch Einpunkt-Cluster trivialisierbar und ignoriert Minderheitsklassen. Zudem ist ein fairer Vergleich nur bei gleicher Anzahl k möglich.

2.2.2. Adjusted Rand Index (ARI)

Der Adjusted Rand Index (ARI) beschreibt die Übereinstimmung zweier Cluster X (Vorhersage), Y (*Ground Truth*) und korrigiert um die erwartete Übereinstimmung bei zufälliger Zuordnung (vgl. [6] S.194ff und [1]):

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{1 - \mathbb{E}[\mathrm{RI}]},$$

wobei der *Rand-Index*

$$\mathrm{RI} = \frac{a + d}{\binom{n}{2}}$$

den Anteil aller Paare misst, die in beiden Partitionen entweder im gleichen Cluster (a) oder in verschiedenen Clustern (d) liegen. Die Erwartung $\mathbb{E}[\mathrm{RI}]$ unter zufälliger Clusterbildung berechnet sich aus den Clustergrößen in X und Y:

$$\mathbb{E}[\mathrm{RI}] = \frac{\sum_i \binom{n_i^{(X)}}{2} \times \sum_j \binom{n_j^{(Y)}}{2}}{\binom{n}{2}^2},$$

wobei $n_i^{(X)}$ und $n_j^{(Y)}$ die Größen der Cluster i in X bzw. j in Y sind.

Wertebereich: $\mathrm{ARI} \in [-1, 1]$, wobei $\mathrm{ARI} = 1$ perfekte Übereinstimmung, $\mathrm{ARI} = 0$ Zufallsniveau und $\mathrm{ARI} < 0$ schlechter als Zufall bedeutet.

Beispiel: Gegeben sind drei Objekte $\{1, 2, 3\}$ mit

$$X : \{1, 2\}, \{3\}, \quad Y : \{1\}, \{2, 3\}.$$

Es existieren $\binom{3}{2} = 3$ Paare. Wir ermitteln

$$a = 0 \quad \text{(kein Paar in beiden Partitionen im selben Cluster)},$$

$$d = 1 \quad \text{(z. B. Paar } (1, 3) \text{ in beiden Partitionen getrennt)}.$$

Damit ist:

$$\mathrm{RI} = \frac{a + d}{3} = \frac{0 + 1}{3} = \tfrac{1}{3}.$$

Für die Erwartung unter Zufall:

$$\sum_i \binom{n_i^{(X)}}{2} = 1, \quad \sum_j \binom{n_j^{(Y)}}{2} = 1, \quad \mathbb{E}[\mathrm{RI}] = \frac{1 \cdot 1}{3^2} = \tfrac{1}{9}.$$

Folglich

$$\mathrm{ARI} = \frac{\frac{1}{3} - \frac{1}{9}}{1 - \frac{1}{9}} = \frac{\frac{2}{9}}{\frac{8}{9}} = \tfrac{1}{4} = 0{,}25.$$

Kritische Einordnung: ARI ist eine sehr häufig verwendete und zuverlässige Metrik. Wie bei allen extrinsischen Metriken ist hier zur Bewertung eine *Ground Truth* notwendig. Eine weitere Einschränkung beim ARI ist die Verzerrung bezüglich der Clustergröße. Wenn eine Clustering-Lösung eine Mischung aus großen und kleinen Clustern enthält, wird ARI vorwiegend von den großen Clustern beeinflusst (vgl. [1]).

2.2.3. Adjusted Mutual Information (AMI)

Die Adjusted Mutal Information (AMI) beschreibt wie viel Information zwischen der dem vorhergesagten Cluster X und den tatsächlichen Cluster Y geteilt wird und korrigiert für zufällige Übereinstimmungen. Die Metrik hat denselben Wertebereich wie ARI und wird wie folgt bestimmt (vgl. [1]):

$$\mathrm{AMI}(X,Y) = \frac{\mathrm{MI}(X,Y) - \mathbb{E}[\mathrm{MI}(X,Y)]}{\mathrm{avg}(H(X),H(Y)) - \mathbb{E}[\mathrm{MI}(X,Y)]},$$

wobei:

- H = individuelle Entropie – ein Maß für die erwartete Unsicherheit
- MI = Der Mutual-Information-Algorithmus
- $\mathbb{E}$ = der Erwartungswert basierend auf Zufall

Beispiel: Gegeben sind die Clusterings $X : \{A,B\},\{C\}$ und $Y : \{A\},\{B,C\}$. Würde man hiervon die Mutal Information (MI) berechnen erhält man $MI = 0{,}251$. Nach Korrektur um den Zufallswert erhält man $\mathrm{AMI}(X,Y) = 0{,}136$, was etwas besser als 0 (zufällig) ist.

Kritische Einordnung: AMI bietet durch die Zufallskorrektur eine robustere Interpretation als die einfache MI. Ein wesentlicher Unterschied zu ARI liegt in den jeweiligen Verzerrungen: Während ARI Lösungen mit ähnlich großen Clustern bevorzugt, ist AMI zu „reinen“ Clustern verzerrt, die nur einen Klassentyp enthalten und oft unausgewogen sind (vgl. [1]).

3. Klassifikation

Klassifikation bezeichnet Algorithmen im Bereich des *supervised Learnings*, welche Datenpunkte anhand bekannter Klassenlabels in vordefinierte Kategorien einordnen. Ziel ist, auf Basis eines Trainingsdatensatzes mit Features und zugehörigen Labels ein Modell zu lernen, das für neue, unbekannte Datenpunkte möglichst korrekte Vorhersagen trifft. Gute Klassifikatoren zeichnen sich durch hohe Treffergenauigkeit (Accuracy), gute Trennung zwischen den Klassen (Precision/Recall, AUC) und robuste Leistung bei unterschiedlichen Klassenhäufigkeiten aus (vgl [1]).

Viele dieser Metriken basieren im binären Fall auf den vier Kategorien einer *Konfusionsmatrix* (True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN)) (vgl. Abbildung 3.1). Im folgenden Beispiel wurden von 100 Instanzen insgesamt 30 als **True Positives**, 10 als **False Positives**, 15 als **False Negatives** und 45 als **True Negatives** klassifiziert. Dieses Beispiel wird im Folgenden zur als Beispiel der Berechnung einzelner Metriken genutzt. Ein komplexeres Beispiel, in welchen Algorithmen auf einen Brustkrebsdatensatz trainiert wurden und bestimmen sollen ob der Brustkrebs bösartig oder gutartig ist, ist im verlinkten GitHub-Repository zu finden.

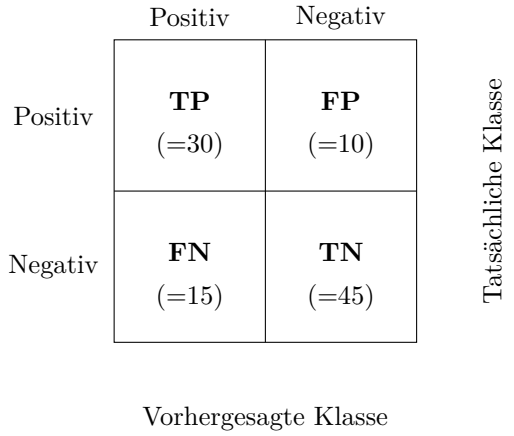

Abbildung 3.1.: Konfusionsmatrix mit Beispielwerten

3.1. Accuracy (Treffergenauigkeit)

Die Accuracy misst den Anteil korrekt klassifizierter Instanzen an allen Fällen.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} = \frac{30 + 45}{100} = 0{,}75$$

Somit misst sie wie viele Vorhersagen insgesamt richtig waren. Eine hohe Accuracy ist nur aussagekräftig, wenn die Klassenverteilung ausgeglichen ist (vgl. [1]).

3.2. Precision, Recall, F1-Score

Diese drei Metriken sind besonders wichtig, wenn die Klassenverteilung unausgeglichen ist (z. B. bei seltenen Ereignissen)(vgl. [7] S.38f).

Precision (Positiver Vorhersagewert):

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{30}{30 + 10} = 0{,}75$$

Precision beschreibt den Anteil der als positiv vorhergesagten Fälle, die tatsächlich positiv sind. Somit sollte diese Metrik verwendet werden, wenn FP teuer sind (z. B. Spam-Filter). Allerdings wird ignoriert, wie viele echte Positive verpasst werden.

Recall (Sensitivität, Trefferquote):

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{30}{30 + 15} = 0{,}67$$

Recall beschreibt den Anteil der tatsächlichen Positiven, die korrekt erkannt wurden. Die Metrik ist somit relevant, wenn FN kritisch sind (z. B. Krebsdiagnose).

F_1-Score (harmonisches Mittel):

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0{,}71$$

Der F1-Score balanciert Precision und Recall und gibt nur dann hohe Werte, wenn beide hoch sind. Beachtet allerdings TN nicht und ist daher nicht vollständig aussagekräftig, wenn TN wichtig sind (vgl. [7] S38f).

3.3. ROC-Kurve und AUC (Area Under the Curve)

Die Receiver Operating Characteristic (ROC)-Kurve ist ein Werkzeug zur Bewertung von binären Klassifikationsmodellen. Sie zeigt den Zusammenhang zwischen der *True Positive Rate* (Recall) und der *False Positive Rate* (FPR) für verschiedene Entscheidungsschwellen (Thresholds) des Modells.

$$\text{TPR} = \frac{TP}{TP + FN} \quad \text{(Recall)}$$
$$\text{FPR} = \frac{FP}{FP + TN}$$

Interpretation: Die **ROCJ!** (**ROCJ!**)-Kurve zeigt, wie gut ein Modell zwischen den Klassen unterscheidet. Sie beginnt immer bei (0,0) und endet bei (1,1). Je näher die Kurve an der oberen linken Ecke liegt, desto besser ist das Modell (vgl. [1]).

Area under the Curve (AUC) ist die Fläche unter der ROC-Kurve:

- AUC = 1: perfekter Klassifikator
- AUC = 0.5: reines Raten (Zufall)
- AUC < 0.5: systematisch falsche Klassifikation

Vorteile: AUC ist unabhängig vom gewählten Schwellenwert und eignet sich besonders bei unausgeglichenen Klassenverhältnissen.

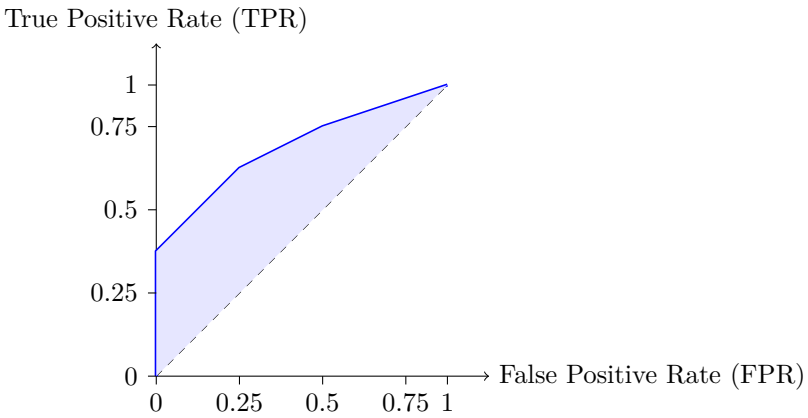

Abbildung 3.2.: Beispielhafte ROC-Kurve (blau) und zufälliger Klassifikator (gestrichelt)

Beispiel: Angenommen, ein Modell trifft mit variabler Schwelle folgende Entscheidungen (sortiert nach Modell-Score):

Score	Wahre Klasse	Klassifikation bei Schwelle
0.9	Positiv	Positiv
0.8	Negativ	Positiv
0.7	Positiv	Positiv
0.4	Positiv	Negativ
0.2	Negativ	Negativ

Bei Variation der Schwelle kann man verschiedene (FPR, TPR)-Punkte erzeugen und so die ROC-Kurve aufbauen (siehe Abbildung 3.2).

3.4. Mehrklassige Klassifikation

Zur Evaluation nicht-binärer Klassifikationsprobleme lassen sich die Evaluationsmetriken über alle Klassen hinweg zusammenfassen. Hierbei gibt es zwei Ansätze (vgl. [8]):

- **Micro-Averaging**:
  - Man fasst alle Klassen-Confusion-Matrix-Einträge instanzübergreifend zusammen:

$$TP = \sum_{i=1}^{r} TP_i, \quad FP = \sum_{i=1}^{r} FP_i, \quad FN = \sum_{i=1}^{r} FN_i.$$

  - Anschließend werden die Metriken wie im binären Fall berechnet.
- **Macro-Averaging**:
  - Für jede Klasse i separat:

$$P_i = \frac{TP_i}{TP_i + FP_i}, \quad R_i = \frac{TP_i}{TP_i + FN_i},$$

$$F_{1,i} = 2\,\frac{P_i\,R_i}{P_i + R_i}.$$

  - Die *Macro-*F_1 ist der einfache Mittelwert:

$$\mathrm{F}_{1,\mathrm{macro}} = \frac{1}{r}\sum_{i=1}^{r} F_{1,i}.$$

Kritische Betrachtung:

- **Micro-F**$_1$: Gesamtüberblick, instanzgewichtet, robust bei Klassenungleichgewicht.
- **Macro-F**$_1$: Klassen gleichgewichtet, macht Schwächen seltener Klassen sichtbar.

4. Regression

Regression bezeichnet Algorithmen im Bereich des *supervised Learnings*, welche kardinal skalierte Zielwerte y vorhersagen. Ziel ist, auf Basis eines Trainingsdatensatzes mit Eingabevektoren $\mathbf{x}_i$ und zugehörigen Messwerten y_i ein Modell zu erlernen, das für neue Datenpunkte möglichst geringe Abweichungen $\hat{y}_i - y_i$ erzielt. Anders als bei der Klassifikation interessiert hier nicht die korrekte Kategorisierung, sondern die Minimierung des Vorhersagefehlers (vgl. [1]). Ein komplexeres Beispiel im verlinkten GitHub-Repository trainiert und evaluiert zwei einfachere Regression-Algorithmen anhand eines Diabetes-Datensatzes.

Im Folgenden werden die gebräuchlichsten Fehlermaße vorgestellt.

4.1. MSE und RMSE

Der Mean Squared Error (MSE) ist definiert als

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

der Root Mean Squared Error (RMSE) als

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}.$$

Beide Metriken messen den durchschnittlichen quadratischen Fehler und sind in der Einheit von y (RMSE) bzw. im Quadrat davon (MSE). Folglich deuten niedrigere Werte auf eine bessere Regression hin.

Beispiel: Angenommen, wir haben $N = 3$ Testwerte mit $(y_1, y_2, y_3) = (2, 5, 7)$ und Vorhersagen $(\hat{y}_1, \hat{y}_2, \hat{y}_3) = (3, 4, 10)$. Dann rechnen wir:

$$(\hat{y}_1 - y_1)^2 = (3-2)^2 = 1, \quad (\hat{y}_2 - y_2)^2 = (4-5)^2 = 1, \quad (\hat{y}_3 - y_3)^2 = (10-7)^2 = 9.$$

Somit ist

$$\mathrm{MSE} = \frac{1+1+9}{3} = \frac{11}{3} \approx 3{,}67, \quad \mathrm{RMSE} = \sqrt{3{,}67} \approx 1{,}92.$$

Somit liegt der Fehler der Vorhersagen bei etwa 1,92 Einheiten.

Kritische Einordnung MSE/RMSE bestrafen größere Fehler stärker (Quadrierung) und sind leicht zu optimieren, reagieren jedoch empfindlich auf Ausreißer (vgl. [9]).

4.2. MAE

Der MAE ist definiert als

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right|.$$

Er misst den durchschnittlichen absoluten Fehler in derselben Einheit wie y. Somit deuten auch hier niedrigere Werte auf eine bessere Regression hin.

Beispiel: Mit den gleichen Werten $(y_i) = (2, 5, 7)$ und $(\hat{y}_i) = (3, 4, 10)$ erhalten wir:

$$|3-2| = 1, \quad |4-5| = 1, \quad |10-7| = 3,$$

$$\mathrm{MAE} = \frac{1+1+3}{3} = \frac{5}{3} \approx 1{,}67.$$

Somit weichen die Vorhersagen um etwa 1,67 von den Vorhersagen ab.

Kritische Einordnung MAE ist einfacher zu interpretieren und robuster gegenüber Ausreißern, gewichtet aber alle Fehler linear (vgl. [9]).

4.3. R^2 (Bestimmtheitsmaß)

Das Bestimmtheitsmaß R^2 gibt an, welcher Anteil der Varianz der Zielwerte durch das Modell erklärt wird:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \quad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i.$$

Die Werte liegen zwischen $-\infty$ und 1.

Beispiel: Für unsere Werte $(y_i) = (2, 5, 7)$ ist der Mittelwert $\bar{y} = \frac{2+5+7}{3} = 14/3 \approx 4{,}67$. Berechne die Quadratsummen:

$$\sum(\hat{y}_i - y_i)^2 = 1+1+9 = 11, \quad \sum(y_i - \bar{y})^2 = (2-4{,}67)^2 + (5-4{,}67)^2 + (7-4{,}67)^2 \approx 7{,}11 + 0{,}11 + 5{,}44 = 12{,}67.$$

Damit

$$R^2 = 1 - \frac{11}{12{,}67} \approx 1 - 0{,}87 = 0{,}13.$$

Das Modell erklärt hier etwa 13 % der Gesamtvarianz.

Kritische Einordnung R^2 ist anschaulich und modellunabhängig, kann jedoch durch Ausreißer oder sehr homogene Daten verzerrt sein und steigt mit zunehmender Modellkomplexität (vgl. [1]).

5. Fazit

In der Arbeit sollten die Evaluationsmetriken für die ML-Algorithmen zugänglich gemacht werden. Die Auswahl geeigneter Evaluationsmetriken hängt dabei stark von der Lernaufgabe und den Zielen ab: Intrinsische Clustering-Indizes (Silhouette, Davies-Bouldin, Calinski-Harabasz) messen Kohäsion und Separation, während extrinsische Indizes (Purity, ARI, NMI) die Clusterlösung mit bekannten Klassen vergleichen. In der Klassifikation sind je nach Datenlage verschiedene Kombinationen von Accuracy, Precision, Recall, F1 und AUC sinnvoll; keine einzelne Metrik genügt im Allgemeinen. Für die Regression geben MSE/RMSE, MAE und R^2 unterschiedliche Blickwinkel auf die Fehler. Ein umfassendes Bild entsteht erst, wenn man mehrere komplementäre Metriken betrachtet.

Wichtig ist, dass man die Stärken und Schwächen jeder Metrik kennt: Manche Metriken belohnen viele kleine Cluster (Purity), andere berücksichtigen Zufallseffekte (ARI), und wieder andere gewichten Ausreißer unterschiedlich (MSE vs. MAE). Letztlich müssen Metriken im Kontext der Anwendung interpretiert werden.

Die einzelnen Modelle sollten unter verschiedenen Blickwinkeln bewertet und die Ergebnisse kritisch hinterfragt werden. Durch die Kombination mehrerer Metriken und bestmögliche Verwendung des verfügbaren Expertenwissens lässt sich die Qualität von Clustering, Klassifikation und Regression im maschinellen Lernen am aussagekräftigsten beurteilen.

Literaturverzeichnis

[1] Miller, C. u. a. „A review of model evaluation metrics for machine learning in genetics and genomics“. In: *Frontiers in Bioinformatics* 4 (2024), S. 1457619.

[2] Rousseeuw, P. J. „Silhouettes: a graphical aid to the interpretation and validation of cluster analysis“. In: *J. Comput. Appl. Math.* 20 (1987), S. 53–65.

[3] Davies, D. L./ Bouldin, D. W. „A cluster separation measure“. In: *IEEE Trans. on Pattern Anal. Mach. Intell.* 1.2 (1979), S. 224–227.

[4] Caliński, T./ Harabasz, J. „A dendrite method for cluster analysis“. In: *Communications in Statistics – Theory and Methods* 3.1 (1974), S. 1–27.

[5] Huang, A. u. a. „Similarity measures for text document clustering“. In: *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. Bd. 4. 2008, S. 9–56.

[6] Hubert, L./ Arabie, P. „Comparing Partitions“. In: *Journal of Classification* 2 (1985), S. 193–218.

[7] Powers, D. M. „Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation“. In: *arXiv preprint arXiv:2010.16061* (2020).

[8] Takahashi, K. u. a. „Confidence interval for micro-averaged F 1 and macro-averaged F 1 scores“. In: *Applied Intelligence* 52.5 (2022), S. 4961–4972.

[9] Hodson, T. O. „Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not“. In: *Geoscientific Model Development Discussions* 2022 (2022), S. 1–10.

A. Anhang