Vorhersagen in der Energiewirtschaft – Welche Fehlermetriken eignen sich?

Präzise Vorhersagen spielen eine zunehmend wichtige Rolle im Energiesektor: Um beispielsweise Engpässe und Überlastungen im Netz zu vermeiden, müssen die Flexibilitäten im Energiesystem effektiv genutzt werden. Dazu sind nicht nur genaue Vorhersagen für Energieproduktion, -bedarf und -preis erforderlich, sondern auch für die Flexibilitäten selbst. Konkret kann dies bedeuten, vorherzusagen, wie lange ein Elektrofahrzeug noch angeschlossen bleibt, um dessen Ladevorgang zu verschieben, also die Flexibilität optimal zu nutzen, oder wann die eine Wärmepumpe anspringt, um die benötigte Wärme zu liefern. Hierzu kann man supervised Machine-Learning-Methoden verwenden (weitere Anwendungsmöglichkeiten in der Energiewirtschaft).

Die Bewertung der Vorhersagegenauigkeit ist notwendig, um Modelle beurteilen zu können und untereinander zu vergleichen. Um die Abweichungen zwischen Vorhersagen und tatsächlichen Ereignissen zu beschreiben und zukünftige Abweichungen abschätzen zu können, werden spezifische Metriken verwendet. Unterschiedliche Metriken eignen sich für verschiedene Bewertungen der Genauigkeit. Die folgende Datei bietet einen Überblick über einige Metriken, ihre Vor- und Nachteile sowie ihre typischen Anwendungsbereiche. Man unterscheidet zwischen zwei Arten von Aufgaben, die eine Vorhersage erfüllen kann: Regression und Klassifikation. Bei der Regression werden stetige Werte vorhergesagt, während es bei der Klassifikation um die Vorhersage von Gruppenzugehörigkeiten geht. In einem weiteren Artikel werden wir näher auf die Auswahl der passenden Methode zur Vorhersage von Zeitreihen eingehen.

Metriken für Regression

Typische Regressionsaufgaben der Energiewirtschaft sind die Vorhersage der PV- oder Windenergieerzeugung, des Energiepreises oder eines Lastgangs. Im Folgenden werden die Zielvariable mit y_i bezeichnet und ihre Vorhersage mit $\hat{y}_i$.

Root-Mean-Square Error (RMSE)

Der Root-Mean-Square Error (RMSE) ist die am häufigsten verwendete Metrik für Regressionsprobleme. Der RMSE ist die Wurzel des mittleren quadrierten Fehlers der Vorhersagen. Also ist der RMSE immer positiv und je kleiner er ist, desto besser ist dir Vorhersage. In Formeln bei N Vorhersagen:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\hat{y}_i - y_i)^2}\ .$$

Der RMSE die gleiche Einheit wie die Zielvariable y_i. Der Mean-Square Error (MSE) wird häufig als die zu minimierende Loss-Funktion während des Trainings von Machine Learning Modellen verwendet, da seine Ableitung bei seinem Minimum Null ist und es somit einfacher ist, lokale Minima zu finden. Der RMSE hängt stark von der Skalierung und dem Datensatz ab, kann aber zur Vergleichbarkeit von Vorhersagen mit unterschiedlichen Skalen normalisiert werden. Eine Möglichkeit, um den RMSE zu normalisieren ist

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{y_{max} - y_{min}}.$$

Sowohl der RMSE als auch der NRMSE stark beeinflusst von Ausreisern, da sie durch das Quadrieren besonders stark in die Summe mit eingehen. Andererseits kann das Quadrieren bei Fehlern kleiner eins eine Unterschätzung des Fehlers bewirken. Der RMSE sollte also als Metrik verwendet werden, wenn Ausreiser stark bestraft werden sollen. [1,2]

Mean Absolute Error (MAE)

Der Mean Absolute Error (MAE) ist der mittlere Betrag des Fehlers der Vorhersage. Bei N Vorhersagen lässt sich der MAE berechnen durch:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=0}^{N} |\hat{y}_i - y_i| \,.$$

Durch die Nutzung des Betrags ist der MAE im Gegensatz zum RMSE robust gegen Ausreiser. Ähnlich wie der RMSE hat er dieselbe Einheit wie die Variable y_i und ist noch leichter zu interpretieren, da nicht quadriert wird. Den MAE als Loss Funktion zu verwenden, ist möglich, allerdings ist er bei Null nicht differenzierbar und gegen Null bleibt er konstant, was das Finden der Minima schwieriger gestaltet. Ähnlich wie den RMSE kann man auch den MAE normalisieren und eine bessere Vergleichbarkeit bei unterschiedlich Skalen oder Datensets zu gewährleisten. Eine Möglichkeit ist es den MAE der Vorhersage mit dem MAE eines Naiven Vorhersage zu vergleichen und so den Mean Absolute Scaled-Error (MASE) zu berechnen:

$$\mathrm{MASE} = \frac{\mathrm{MAE}}{\mathrm{MAE}_{naive}}.$$

Je besser die Vorhersagen sind, desto kleiner ist der MASE. Vorhersagen, die besser sind als die naive Methode, haben einen MASE-Wert kleiner als 1. Ein Beispiel für eine naive Vorhersage ist, den Wert des vorherigen Tages zu verwenden. Weitere Möglichkeiten sind die Verwendung von Durchschnittswerten oder eine lineare Regression als einfaches Vergleichsmodell. Eine alternative Methode zur Skalierung des MAE besteht darin, den prozentualen Abweichungsfehler zu verwenden, was als Mean Absolute Percentage Error (MAPE) bezeichnet wird

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=0}^{N} \left|\frac{\hat{y}_i - y_i}{y_i}\right|.$$

Der MAPE ist unabhängig von der Skala und leicht zu interpretieren. Allerdings ergeben sich sehr große Werte für y_i gegen Null und er ist asymmetrisch, da Fehler mit $\hat{y}_i > y_i$ stärker bestraft werden als für $\hat{y}_i < y_i$. [1 3,4,5]

Huber-Loss

Als Loss-Funktion eignet sich der MAE gut, falls Ausreiser in den Daten vorkommen diese aber nicht entscheidend für die Vorhersage sein sollen. Andererseits hat der RMSE den Vorteil differenzierbar bei Null zu sein und sich damit besser eignet Minima zu finden. Um die Vorteile von RMSE und MAE zu kombinieren kann man den Huber Loss HL_δ nutzen. Dieser ist für Abweichungen kleiner δ linear zum MSE und für größere zum MAE:

$$\mathrm{HL}_\delta(y_i, \hat{y}_i) = \begin{cases} \frac{1}{2}(\hat{y}_i - y_i)^2, & \text{falls } |\hat{y}_i - y_i| \le \delta \\ \delta|\hat{y}_i - y_i| - \frac{1}{2}\delta^2, & \text{sonst.} \end{cases}$$

Ein Problem des Huber Loss ist allerdings der Hyperparameter δ welcher zusätzlich optimiert werden kann. [6]

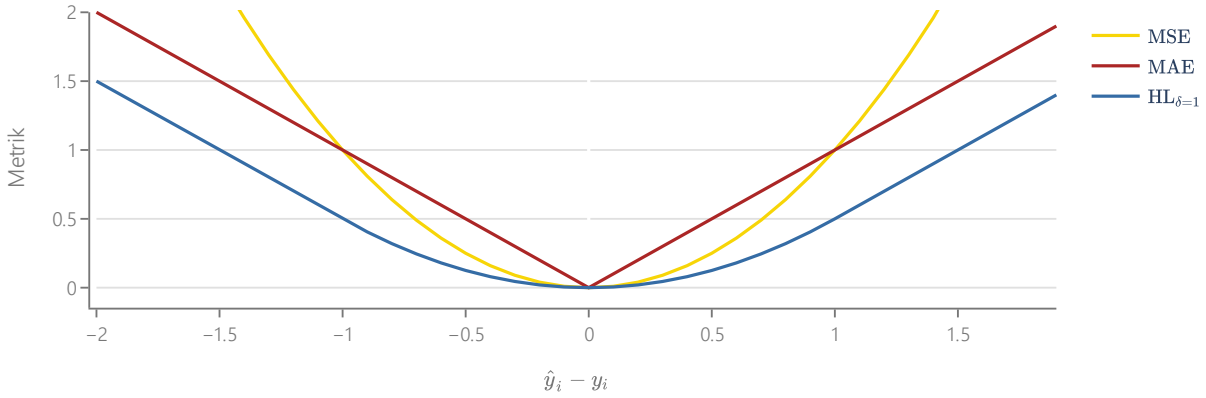

Abbildung 1: Vergleich zwischen MSE, MAE und Huber-Loss.

R^2-Score

Der R^2-Score vergleicht den MSE der Vorhersagen mit dem MSE, den man erhalten würde, wenn man einfach den Durchschnittswert $\overline{y}$ als Vorhersage verwenden würde

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=0}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=0}^{N}(\overline{y} - y_i)^2}.$$

Das Model, das immer den Mittelwert vorhersagt, wird ein "R^2-Score" von 0 zugeordnet und je besser ein Model wird, desto näher an 1 ist der „R^2-Score". Sehr schlechte Vorhersagen können hier auch Werte unter 0 erzielen.

Bis jetzt wurde in keiner der Metriken mit einbezogen wie viele zusätzliche Variablen (im folgenden Feature genannt) genutzt werden, um eine Vorhersage zu treffen. Jedes zusätzliche Feature bringt zusätzliche Komplexität in das Modell, die bestraft werden sollte, falls die Vorhersagen nicht verbessert werden. Dies wird durch den angepassten R^2-Score ($\mathrm{R}^2_{\mathrm{adj}}$) berücksichtigt

$$\mathrm{R}^2_{\mathrm{adj}} = 1 - \frac{N-1}{N-k-1}(1 - \mathrm{R}^2),$$

wobei N erneut die Anzahl der Vorhersagen ist und k die Anzahl der Feature. Falls nun also Feature hinzugefügt werden, die den R^2 nicht stark verbessern verschlechtert sich der $\mathrm{R}^2_{\mathrm{adj}}$, das ermöglicht es nur Feature einzubeziehen, die das Model tatsächlich verbessern. [1,3]

Quantil-Metriken

In vielen Fällen ist es nicht entscheidend, dass eine Vorhersage genau stimmt, sondern es ist wichtiger, Grenzen zu bestimmen, innerhalb derer sich die Zielvariable befinden sollte. Zur Bestimmung dieser Grenzen werden Quantile verwendet. Um die Genauigkeit der Quantile zu bewerten, werden geeignete Metriken benötigt. Eine Möglichkeit besteht darin, den Prozentsatz der Zielvariable zu bestimmen, der innerhalb der vorhergesagten Quantile liegt. Diese Metrik wird hier als II („inside interval") bezeichnet

$$\mathrm{II} = \frac{q_{in}}{N},$$

mit q_{in} die Anzahl der Messungen ist mit $y_i \in [\hat{q}_{i,low}, \hat{q}_{i,up}]$ mit $\hat{q}_{i,low}$ dem vorhergesagten unteren Quantil und $\hat{q}_{i,up}$ dem oberen. Die Bewertung des II ist allerdings nicht trivial, ein II von 1 kann zum Beispiel einer positiven Zielvariablen auch erreicht werden durch das Quantil-Intervall von $[0, \infty)$ ohne dass hier eine gute Einschränkung getroffen wird. Für eine große Anzahl an N sollte der II eines guten Models übereinstimmen mit dem Abstand der Quantile: Sei $q_{low=5\%}$ das 5%-Quantile und $q_{up=95\%}$ das 95%-Quantile, sollte der II einen Wert von 90% erzielen. Eine leichter zu interpretierende Metrik ist der Pinnball-Loss PL_τ, welche eine einzelne Quantil-Prognose bewertet. Für eine Quantil-Prognose $\hat{q}_{i,\tau}$ ergibt sich

$$PL_\tau(y_i, \hat{q}_{i,\tau}) = \begin{cases} |\hat{q}_{i,\tau\, y_i} - y_i|\tau, & \text{falls } y_i \geq \hat{q}_{i,\tau} \\ |\hat{q}_{i,\tau} - y_i|(1-\tau), & \text{falls } \hat{q}_{i,\tau} \geq y_i, \end{cases}$$

wobei τ das Ziel-Quantil ist. Je niedriger der Pinnball-Loss ist, desto besser ist das Quantil. Hierbei werden Abweichungen hin zum Median weniger stark bestraft. Der Pinnball-Loss wird häufig als Loss-Funktion für Quantile verwendet. [7]

Metriken für Klassifikation

Klassifikationsaufgaben in der Energiewirtschaft sind beispielsweise, ob ein Netzengpass auftritt oder, ob ein Fahrzeug an einer Ladestation angeschlossen ist. Bei der Klassifikation in zwei Kategorien ermöglicht eine Vier-Felder-Tafel (auch „Confusion-Matrix" genannt) die Unterscheidung zwischen verschiedenen Arten von Fehlern.

		Wirklichkeit	
		Positiv	Negativ
Vorhersage	Positiv	True Positive (TP)	False Positive (FP)
	Negativ	False Negative (FN)	True Negative (TN)

Tabelle 1: Vier-Felder-Tafel einer binären Klassifikation.

Accuracy

Die Genauigkeit oder auch Accuracy A gibt an zu welchem Anteil die Vorhersage richtig ist

$$\mathrm{A} = \frac{n_{TP} + n_{TN}}{N},$$

wobei N die totale Anzahl der Vorhersagen ist und n_{TP}/n_{TN} die Anzahl der True Positives / True Negatives. Allerdings können auch Vorhersagen mit einer hohen Accuracy nicht hilfreich sein. Will man beispielsweise die Anschlusszeiten eines Fahrzeugs vorhersagen welches nur 10% der Zeit angeschlossen ist, so hätte auch die Vorhersage, dass das Fahrzeug nie angesteckt ist, 90% Accuracy. Eine Möglichkeit die Einordnung der Accuracy zu verbessern ist, sie nach Klassen zu unterschieden. Damit folgt für das vorherige Beispiel 100% Accuracy für die abgesteckte Zeit aber 0% für die angesteckte Zeit. [1,8]

Sensitivität und Präzision

Eine weitere Möglichkeit sind die beiden Größen Sensitivität S (häufig Recall genannt) und Präzision P welche gegeben sind durch:

$$\mathrm{S} = \frac{n_{TP}}{n_{TP} + n_{FN}}, \mathrm{P} = \frac{n_{TP}}{n_{TP} + n_{FP}}.$$

Die Sensitivität gibt den Anteil der Positiven an, die auch als positiv vorhergesagt werden. Andererseits gibt die Präzision an, zu welchem Anteil die positiv Vorhergesagten tatsächlich positiv sind. Je nach Anwendungsfall kommt es darauf an, ob man eine besonders hohe Sensitivität oder Präzision erreichen will. Um Engpässe im Netz zu verhindern, kann es wichtiger sein eine sehr hohe Sensitivität zu erreichen, um lieber einmal zu oft einzugreifen als zu wenig. Andererseits kann eine hohe Präzision beispielswiese dann wichtiger sein, wenn man mit die Flexibilität eines Fahrzeugs nutzen will. Hier muss das Fahrzeug tatsächlich angeschlossen sein. Häufig will man einen Kompromiss aus einer hohen Sensitivität als auch Präzision erreichen. [1,8]

F1-Score

Die F1-Metrik beschreibt den Kompromiss zwischen Sensitivität und Präzision und ist zwischen 0 und 1, wobei 1 eine immer richtige Vorhersage ist. Die F1-Metrik ist das harmonische Mittel von Sensitivität und Präzision

$$\mathrm{F1} = 2\,\frac{\mathrm{S} * \mathrm{P}}{\mathrm{S} + \mathrm{P}}.$$

Maximiert man F1 so erreicht man einen guten Kompromiss aus einer hohen Sensitivität als auch Präzision. Um bei dem Beispiel eines an- oder abgesteckten Fahrzeugs zu bleiben, sagt eine Vorhersage mit hohem F1, sowohl die Zeiten zu denen das Fahrzeug angeschlossen ist, richtig vorher und in der Zeit das Fahrzeug als angeschlossen vorhergesagt wird, liegt die Vorhersage selten falsch. [8]

Cross-Entropy

Die meisten Machine Learning-Methoden geben für Klassifikationen Wahrscheinlichkeiten an, mit welchen die verschiedenen Ereignisse eintreffen. Bei Klassifikationen mit zwei Klassen ist das eingetroffene Ereignis $y_i \in \{0,1\}$ und $\hat{y}_i$ die prognostizierte Wahrscheinlichkeit, dass $y_i = 1$ eintritt. Eine Möglichkeit, diese zu bewerten, ist der Log-Loss, der häufig auch als Binary-Cross-Entropy-Loss bezeichnet wird

$$\mathrm{LogL} = \frac{1}{N}\sum_{i=0}^{N} -y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i),$$

mit $\hat{y}_i$ der vorhergesagten Wahrscheinlichkeit, dass 1 eintritt. Für jede einzelne Vorhersage ist der Log-Loss der negative Logarithmus der Likelihood Funktion. Der Logarithmus wird hier genutzt, um nicht mit zu kleinen Zahlen zu arbeiten. Das negative Vorzeichen wird verwendet, damit kleinere Werte eine bessere Vorhersage bedeuten, wodurch ein Minimierungsproblem entsteht. Der Log-Loss ist die meistgenutzte Loss-Funktion für das Training von Klassifikationen. Der Binary-Cross-Entropy-Loss kann auch auf mehrere Klassen verallgemeinert werden, was als Cross-Entropy-Loss bezeichnet wird. Daher kann es als Metrik für Klassifikationen mit mehr als zwei Klassen dienen. [1,9]

Brier-Score

Der Brier-Score beschreibt ebenfalls wie die Übereinstimmung der Wahrscheinlichkeit mit dem tatsächlichen Resultat, dabei ist er bei Problemen mit nur zwei Klassen ähnlich zu berechnen wie der MSE bei für die Regression:

$$\mathrm{Brier} = \frac{1}{N}\sum_{i=0}^{N} (\hat{y}_i - y_i)^2.$$

Je besser eine Vorhersage ist, desto kleiner ist ihr Brier-Score und kann ähnlich wie der Log-Loss auch als die im Training zu minimierende Loss-Funktion für eine Machine Learning Methode genutzt werden. [10]

Literaturverzeichnis

[1] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

[2] Brockwell, P. J., & Davis, R. A. (2016). Introduction to Time Series and Forecasting. Springer.

[3] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed.). OTexts.

[4] R. Hyndman (2006). Another look at forecast accuracy metrics for intermittent demand. Foresight: The International Journal of Applied Forecasting, vol. 4, pp. 43–46.

[5] Makridakis, S. (1993). Accuracy measures: theoretical and practical concerns. International Journal of Forecasting, 9(4), 527-529.

[6] Huber, P.J. (1964). Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics, 35(1), 73-101.

[7] Koenker, R., & Bassett, G. (1978). Regression quantiles. Econometrica, 46(1), 33-50.

[8] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437.

[9] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[10] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly Weather Review, 78(1), 1-3.