Performance-Metriken und Evaluation von Supervised Machine Learning Algorithmen

Marcel Rothering

22. Mai 2018

1 Einführung

Die Daten wurden aufbereitet, neue Features abgeleitet und verschieden Machine Learning Modelle trainiert. Doch wie bewertet man die Performance dieser Modelle? Wie entscheide ich mich z.B. zwischen einem Neuronalen Netz, Random Forest oder Gradient Boosting Classifier? Hierfür verwendet man im Machine Learning Bereich sogenannte Performance Metriken. Diese werden für einen Testdatensatz (*Hold-out Set*) evaluiert, welches nicht zum Training des Modells verwendet wurde, um die Performance für ungesehene Daten abzuschätzen. Doch welche Metriken werden für welches Problem eigentlich verwendet und wann ist die Performance ausreichend?

In diesem Blogbeitrag werden wir verschiedene Performance Metriken für Supervised Machine Learning vorstellen. Zunächst starten wir mit den Metriken für Klassifikation, gefolgt von denen für Regressionsmodelle. Anschließend stellen wir kurz das Kreuzvalidierungsverfahren vor, um einen noch besseren Schätzwert für die Performance eines Modells auf ungesehene Daten zu ermitteln.

2 Metriken für Klassifaktionsmodelle

Klassifikationsmodelle könne in verschiedene Kategorien eingeteilt werden: Binäre, Multi-class, multi-label und hierarchische Klassifikation. Die meisten Performance Metriken für die verschiedenen Kategorien lassen sich aus denen der binären Klassifikation ableiten, weshalb wir uns in diesem Blogbeitrag auf binäre Klassifikation fokussieren. Der interessierte Leser findet die Ableitungen der Performance-Metriken auf komplexere Klassifikationsmodelle in [1].

Accuracy Die wohl einfachste Performance Metrik für Klassifikationsmodelle ist die Accuarcy/Genauigkeit. Diese lässt sich wie folgt berechnen

$$\text{Acc} = \frac{\#correct\ predictions}{\#all\ predictions}. \tag{1}$$

Hat man z.B. 55 richtige Vorhersagen für 100 Beobachtungen gemacht, dann ist die Genauigkeit 55 % und nicht viel besser als der Zufall [1]. Letztere Aussage gilt nur, falls die beiden Klassen, 0 und 1, ausbalanziert sind, d.h. im Datensatz gibt es ungefähr gleich viele Beobachten der Klasse 0 und der Klasse 1. Falls eine der beiden Klassen unter- oder überrepresentiert ist, dann ist die Accuarcy keine gute Metrik. Stellen Sie sich zum Beispiel eine Datensatz vor in dem 95 Beobachtungen zur Klasse 0 gehören und nur 5 zur Klasse 1. Ein Modell, welches nun immer die Klasse 0 vorhersagen würde erzielt dann eine Accuracy von 95 %. Klingt gut, ist in diesem Fall aber schlecht und die Accuracy ist hier nicht geignet. Bei solch einem Problem ist es von Vorteil die Klassifikationsergebnisse in eine sogenannte Confusion-Matrix aufzuteilen.

Confusion-Matrix In Tabelle 1a ist die allgemeine Confusion-Matrix gezeigt. Diese beinhaltet als Elemente die Anzahl an korrekten positiven (TP), korrekten negativen (TN), an falschen positiven (FP) und falschen negativen Vorhersagen. Den beiden Klassen werden hierbei die

[1] Das gilt nur, falls es sich hier um zwei ausbalanzierte Klassen handelt.

	Predicted: 0	Predicted: 1
Actual: 0	TN	FP
Actual: 1	FN	TP

(a) Confusion-Matrix.

	Predicted: 0	Predicted: 1
Actual: 0	$\frac{\text{TN}}{\#all\ negatives}$	$\frac{\text{FP}}{\#all\ negatives}$
Actual: 1	$\frac{\text{FN}}{\#all\ positives}$	$\frac{\text{TP}}{\#all\ positives}$

(b) Normierte Confusion-Matrix.

Tabelle 1: Tabellarische Darstellung der Confusion Matrix (links) und der normierten Confusion-Matrix (rechts)

	Predicted: 0	Predicted: 1
Actual: 0	83	22
Actual: 1	17	57

(a) Confusion-Matrix.

	Predicted: 0	Predicted: 1
Actual: 0	0.79	0.21
Actual: 1	0.23	0.77

(b) Normierte Confusion-Matrix

Tabelle 2: Zahlenbeispiel für die Confusion Matrix (links) und für die normierten Confusion-Matrix (rechts)

beiden Attribute positiv für Klasse 1 und negativ für Klasse 0 zugeordnet. Oftmals wird die Confusion Matrix noch normiert (siehe Tabelle 1b), da es schwierig ist anhand der absoluten Zahlen die Performance einzuschätzen. Die Elemente spiegeln dann die Raten TNR, FPR, FNR und TPR wider. Ein Zahlenbeispiel ist in den Tabellen 2 aufgeführt. Dort können wir direkt ablesen, dass 21 % der Vorhersagen für die negative Klasse falsch sind und 23 % der Vorhersagen für die positive Klasse. Diese beiden Fehler werden oft auch als Type 2 und Type 1 Fehlerraten bezeichnet. Welchen Fehler man nun reduzieren möchte hängt stark vom Anwendungsfall des Modells ab. Soll dieses Modell beispielsweise als Spam Filter dienen (Spam ist hier die positive Klasse), so würde man gerne die Rate der als falsch positiv klassifizierten E-Mails reduzieren (geringe FNR). Denn es ist weitaus dramatischer, wenn eine wichtige Mail mal im Spamordner landet, als das eine Spam E-Mail einmal nicht als solche erkannt wurde (höhere FPR). Umgekehrte Bedürfnisse stellen wir zum Beispiel an ein Modell welches jugendfreie Filme (Klasse 1) herausfiltern soll. Lieber lassen wir unsere Kinder weniger Filme anschauen (höhere FNR), als das nicht jugendfreie Filme freigegeben werden (geringe FPR).

Viele weitere Metriken lassen sich aus der Confusion Matrix berechnen. Weit verbreitete Metriken sind hier die Precison, der Recall und der F1-Score, welche im folgenden kurz besprochen werden.

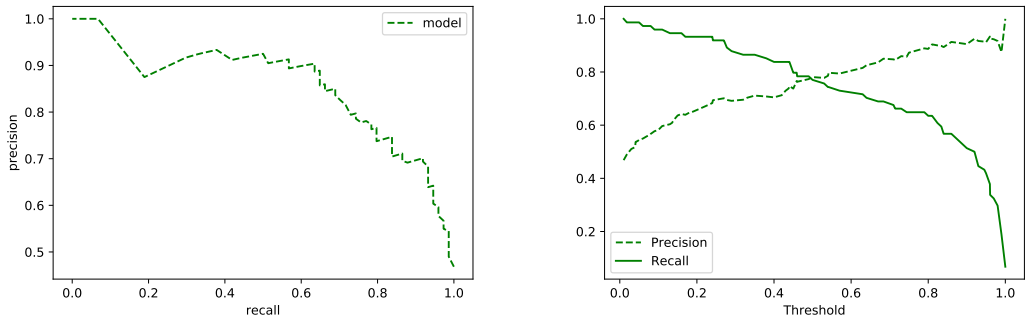

Abbildung 1: Precision-Recall Kurve (links) eines binären Klassifikationsmodells und Werte für Precision (Recall) für verschiedene Thresholds/Schwellwerte des Klassifikationsmodells (rechts).

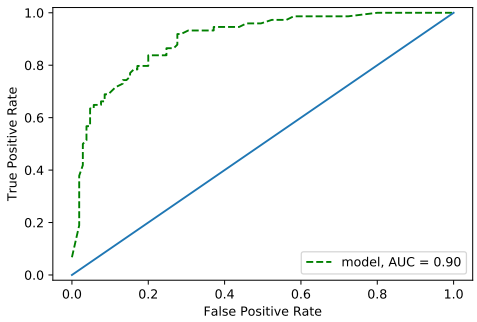

Abbildung 2: ROC Kurve für ein Klassifikationsmodell (grün) und für eine zufällige Klassifikation (blau).

Precison, Recall und F1-Score

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{TP}{P^*} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{TP}{P} \tag{3}$$

$$F1 = \left(\frac{1/Precison + 1/Recall}{2}\right)^{-1} \tag{4}$$

Receiver Opeator Charateristic Viele mehr wie sensitivity, hamming loss, etc.

Manche dieser Metriken wie Precision, Recall und F1 sind außerdem invariant unter einer bestimmten Veränderung der Confusion Matrix. Verändert man beispielsweise die Anzahl der True Negatives (TNs), so hat dies keine Auswirkung auf diese Performance Metriken. Dem sollte man sich bewusst sein, wenn man beispielsweise auch die Güte der negativen Klasse messen/beschreiben will. Die ROC Kurve hingegen beinhaltet alle Elemente der Confusion Matrix und ist daher nicht invariant unter Veränderung eines einzelnen Elements (nur FPR = FP / (TN + FP) und TPR = TP / (TP + FN).

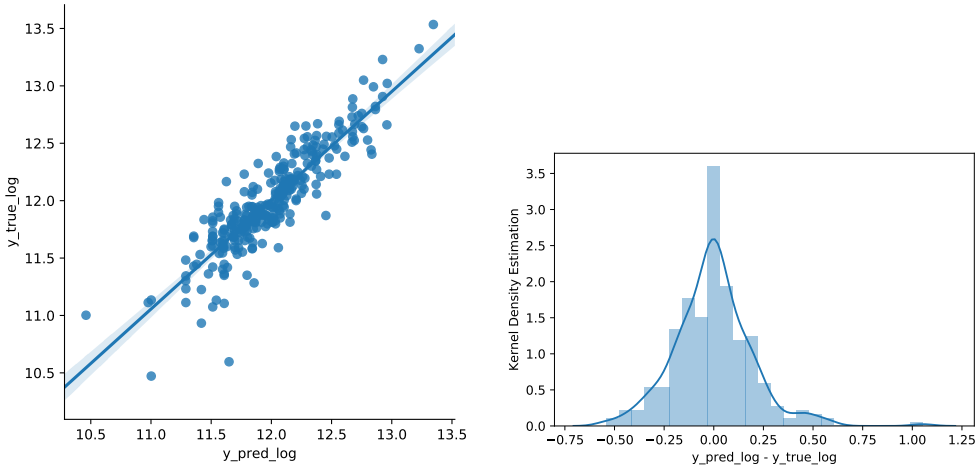

Abbildung 3: Precision-Recall Kurve (links) eines binären Klassifikationsmodells und Werte für Precision (Recall) für verschiedene Thresholds/Schwellwerte des Klassifikationsmodells (rechts).

3 Metriken für Regressionsmodelle

Bic, AIC, analysis of Variance (Anova, etc.)

4 K-Fold Kreuzvalidierung

In der Regel verwendet man K Fold Cross Validation für Modelle, welche nicht all zu lang für das Training brauchen, wie z.B. CNNs. Außerdem ausgenommen sind Forecasting Modelle für Zeitreihenanalyse, wie ARIMA, etc.

Literatur

[1] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manage.*, 45(4):427–437, July 2009.