



Ostbayerische Technische Hochschule
Amberg-Weiden

Machine Learning

Prof. Dr. Fabian Brunner

<fa.brunner@oth-aw.de>

Amberg, 31. Mai 2021

Thema heute: Logistische Regression

- Grundidee
- Kostenfunktional
- Modell-Fitting
- Modellbewertung
- Logistische Regression mit Scikit-learn (Übung)

Problemstellung

- Binäres Klassifikationsproblem
- p numerische Features („unabhängige Variablen“)
- Binäre Zielvariable („abhängige Variable“) mit den Klassen 0 und 1
- m Trainingsdatensätze

$$(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)}) ,$$

wobei $\mathbf{x}^{(i)} \in \mathbb{R}^p$ und $y^{(i)} \in \{0, 1\}$.

Modell-Training

Bestimme eine Funktion

$$f : \mathbb{R}^p \rightarrow \{0, 1\} ,$$

die möglichst gut zu den Trainingsdaten „passt“.

Modellanwendung

Für einen Query Point \mathbf{x}_q prognostiziere das Label $f(\mathbf{x}_q)$.

Idee

Fitte ein lineares Regressionsmodell der Form

$$f_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p$$

und definiere den Klassifikator wie folgt:

- Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow \text{Prognose } y = 1$
- Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow \text{Prognose } y = 0$

Idee

Fitte ein lineares Regressionsmodell der Form

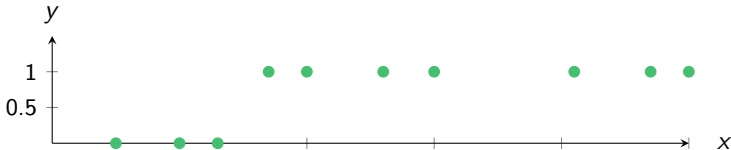
$$f_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p$$

und definiere den Klassifikator wie folgt:

- Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow \text{Prognose } y = 1$
- Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow \text{Prognose } y = 0$

Problem

- Wertebereich des linearen Regressionsmodells ist ganz \mathbb{R} .
- Im Allgemeinen keine zufriedenstellenden Ergebnisse.



Idee

Fitte ein lineares Regressionsmodell der Form

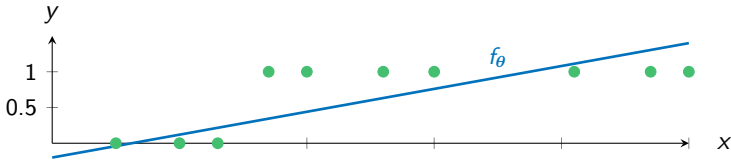
$$f_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p$$

und definiere den Klassifikator wie folgt:

- Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow \text{Prognose } y = 1$
- Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow \text{Prognose } y = 0$

Problem

- Wertebereich des linearen Regressionsmodells ist ganz \mathbb{R} .
- Im Allgemeinen keine zufriedenstellenden Ergebnisse.



Idee

Fitte ein lineares Regressionsmodell der Form

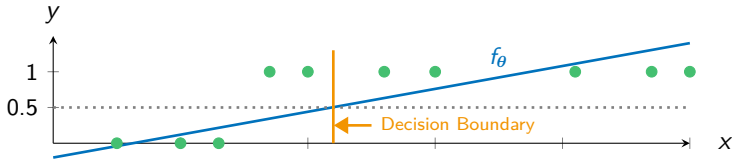
$$f_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p$$

und definiere den Klassifikator wie folgt:

- Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow \text{Prognose } y = 1$
- Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow \text{Prognose } y = 0$

Problem

- Wertebereich des linearen Regressionsmodells ist ganz \mathbb{R} .
- Im Allgemeinen keine zufriedenstellenden Ergebnisse.



Idee

Fitte ein lineares Regressionsmodell der Form

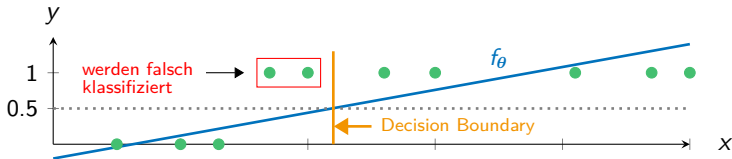
$$f_{\theta}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p$$

und definiere den Klassifikator wie folgt:

- Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow \text{Prognose } y = 1$
- Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow \text{Prognose } y = 0$

Problem

- Wertebereich des linearen Regressionsmodells ist ganz \mathbb{R} .
- Im Allgemeinen keine zufriedenstellenden Ergebnisse.



Ansatz bei der Logistischen Regression

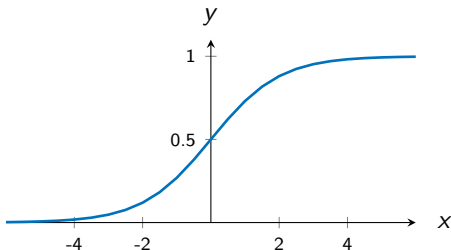
Bei der Logistischen Regression hat die Modellfunktion die folgende Gestalt:

$$f_{\theta}(\mathbf{x}) = g(\theta^T \mathbf{x}) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_p x_p) .$$

Es geht aus dem linearen Regressionsmodell durch Transformation mit der logistischen Funktion

$$g(x) = \frac{1}{1 + e^{-x}}$$

hervor. Diese nimmt nur Werte zwischen 0 und 1 an:



Interpretation

Der Wert $f_{\theta}(\mathbf{x})$ wird als Schätzwert für die bedingte Wahrscheinlichkeit

$$P(y = 1 | \mathbf{x}; \theta)$$

angesehen, d.h. die Wahrscheinlichkeit, dass $y = 1$ unter der Bedingung, dass die Eingabevariablen die Werte \mathbf{x} annehmen (unter der Parametrisierung mit θ).

Entscheidungskriterium

- Statt eine direkte Klassenzuordnung liefert das Verfahren einen Schätzwert für die (bedingte) Wahrscheinlichkeit, dass es sich um die positive Klasse handelt.
- Diese kann wie folgt für die Klassenzuordnung eines unbekannten Datenpunkts \mathbf{x}_q verwendet werden:

Klassenzuordnung bei der Logistischen Regression

Falls $f_{\theta}(\mathbf{x}_q) \geq 0.5 \rightarrow$ Zuordnung zur Klasse $y = 1$

Falls $f_{\theta}(\mathbf{x}_q) < 0.5 \rightarrow$ Zuordnung zur Klasse $y = 0$

Die Parameter θ werden beim der Logistischen Regression so bestimmt, dass die bedingte Wahrscheinlichkeit des Trainingsdatensatzes maximiert wird:

Maximum Likelihood-Methode zur Bestimmung der Modellparameter

$$\hat{\theta} = \arg \max_{\theta} \left\{ \prod_{i=1}^m P(y = y^{(i)} | \mathbf{x}^{(i)}; \theta) \right\} = \arg \min_{\theta} \underbrace{\left\{ - \sum_{i=1}^m \log (P(y = y^{(i)} | \mathbf{x}^{(i)}; \theta)) \right\}}_{=:\tilde{J}(\theta)}.$$

Das Funktional $\tilde{J}(\theta)$ kann man folgendermaßen äquivalent darstellen:

$$\begin{aligned} \tilde{J}(\theta) &= - \sum_{i=1}^m y^{(i)} \log(P(y = 1 | \mathbf{x}^{(i)}; \theta)) + (1 - y^{(i)}) \log(P(y = 0 | \mathbf{x}^{(i)}; \theta)) \\ &= - \sum_{i=1}^m y^{(i)} \log(f_{\theta}(\mathbf{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\theta}(\mathbf{x}^{(i)})) \end{aligned}$$

Die Parameter $\hat{\theta}$ erhält man durch Minimierung des folgenden Funktional:

Cross Entropy als Kostenfunktional bei der Logistischen Regression

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log \left(f_{\theta}(\mathbf{x}^{(i)}) \right) + (1 - y^{(i)}) \log \left(1 - f_{\theta}(\mathbf{x}^{(i)}) \right)$$

Interpretation

- Falls $y^{(i)} = 1$, so nimmt J große Werte für $f_{\theta}(\mathbf{x}^{(i)}) \rightarrow 0$ und kleine Werte für $f_{\theta}(\mathbf{x}^{(i)}) \rightarrow 1$ an.
- Falls $y^{(i)} = 0$, so nimmt J große Werte für $f_{\theta}(\mathbf{x}^{(i)}) \rightarrow 1$ und kleine Werte für $f_{\theta}(\mathbf{x}^{(i)}) \rightarrow 0$ an.
- J kann daher als **Straffunktion** interpretiert werden, welche die Abweichungen zwischen den Labels $y^{(i)}$ und dem Modelloutput $f_{\theta}(\mathbf{x}^{(i)})$ auf dem Trainingsdatensatz „bestraft“.
- Die Parameter θ werden durch Minimierung von J so bestimmt, dass die „Kosten“ auf dem Trainingsdatensatz möglichst gering sind.

Parameterbestimmung

- Das Funktional J hängt nicht-linear von θ ab, sodass die Angabe einer analytischen Lösung i. A. nicht möglich ist.
- Die Parameter müssen stattdessen durch ein iteratives Verfahren, z.B. das Gradientenverfahren, approximiert werden.
- Setzt man $x_0^{(i)} := 1$, so kann man zeigen, dass der Gradient von J durch

$$\nabla J(\theta) = \frac{1}{m} \sum_{i=1}^m \left(f_{\theta}(\mathbf{x}^{(i)}) - y^{(i)} \right) \mathbf{x}^{(i)} \in \mathbb{R}^{p+1}$$

gegeben ist. Dazu nützlich: $g'(x) = g(x)(1 - g(x))$.

Gradientenverfahren für die Logistische Regression

$$\theta^k = \theta^{k-1} - \frac{\alpha}{m} \sum_{i=1}^m \left(f_{\theta^{k-1}}(\mathbf{x}^{(i)}) - y^{(i)} \right) \mathbf{x}^{(i)} .$$

Struktur des Modells bei der Logistischen Regression

$$f_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) , \quad \text{wobei } g(x) = \frac{1}{1 + e^{-x}} .$$

Vorhersage:

$$y = 1 \quad \text{falls} \quad f_{\theta}(\mathbf{x}) \geq 0.5 \quad \Leftrightarrow \quad \boldsymbol{\theta}^T \mathbf{x} \geq 0 ,$$

$$y = 0 \quad \text{falls} \quad f_{\theta}(\mathbf{x}) < 0.5 \quad \Leftrightarrow \quad \boldsymbol{\theta}^T \mathbf{x} < 0 .$$

Verständnisfrage:

Welche Gestalt hat die Menge

$$\{\mathbf{x} \in \mathbb{R}^p : \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p = 0\}$$

geometrisch?

Struktur des Modells bei der Logistischen Regression

$$f_{\theta}(\mathbf{x}) = g(\theta^T \mathbf{x}) , \quad \text{wobei } g(x) = \frac{1}{1 + e^{-x}} .$$

Vorhersage:

$$y = 1 \quad \text{falls} \quad f_{\theta}(\mathbf{x}) \geq 0.5 \quad \Leftrightarrow \quad \theta^T \mathbf{x} \geq 0 ,$$

$$y = 0 \quad \text{falls} \quad f_{\theta}(\mathbf{x}) < 0.5 \quad \Leftrightarrow \quad \theta^T \mathbf{x} < 0 .$$

Verständnisfrage:

Welche Gestalt hat die Menge

$$\{\mathbf{x} \in \mathbb{R}^p : \theta_0 + \theta_1 x_1 + \dots + \theta_p x_p = 0\}$$

geometrisch?

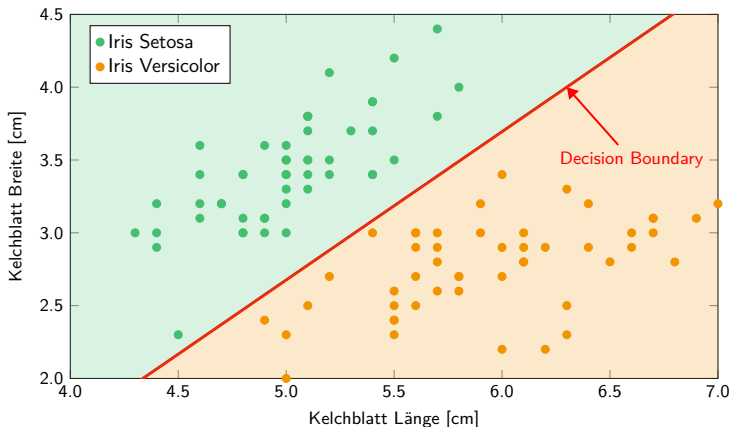
Antwort: $p - 1$ -dimensionale Hyperebene.

Decision Boundary bei der Logistischen Regression

Bei der Logistischen Regression liegt ein **linearer Decision Boundary** vor.

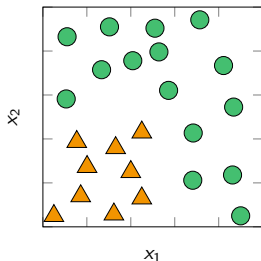
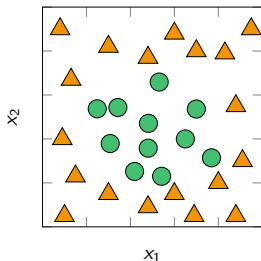
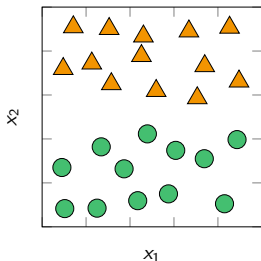
Decision Boundary der Logistischen Regression am Beispiel des Schwertlilien-Datensatzes

Wir betrachten den Iris-Datensatz mit den beiden Klassen Setosa und Versicolor und den Features „sepal length“ und „sepal width“. Die Klassen lassen sich durch ein logistisches Regressionsmodell linear separieren (Ergebnis für die Parameter: $\theta_0 = -7.3, \theta_1 = 3.08, \theta_2 = -3.022$)



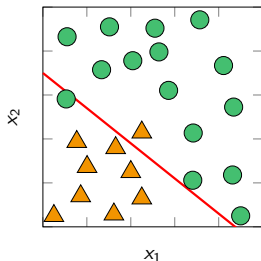
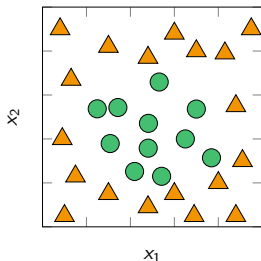
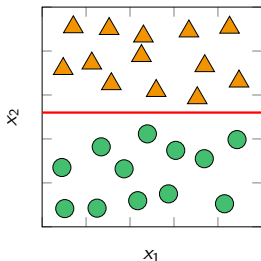
Verständnisfrage:

Für welche der folgenden binären Klassifikationsprobleme eignet sich ein Logistisches Regressionsmodell?



Verständnisfrage:

Für welche der folgenden binären Klassifikationsprobleme eignet sich ein Logistisches Regressionsmodell?



Modellgleichung in Abhängigkeit der bed. Wahrscheinlichkeiten:

Setzt man $P := f_{\theta}(\mathbf{x}) = P(y = 1|\mathbf{x}; \theta)$, so erhält man durch Auflösen der Modellgleichung die Darstellung

$$\theta_0 + \theta_1 x_1 + \dots + \theta_p x_p = \ln \left(\frac{P}{1-P} \right) .$$

Interpretation der Koeffizienten:

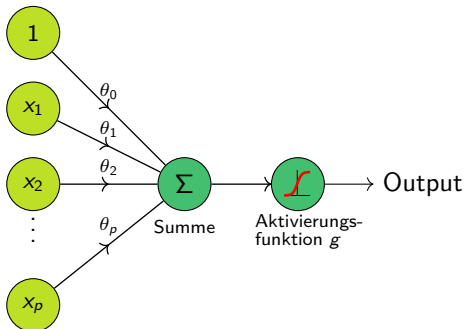
- Erhöht man x_i um eine Einheit, so erhöht sich $\ln \left(\frac{P}{1-P} \right)$ um θ_i .
- Der Ausdruck $\frac{P}{1-P}$ wird auch **Chancenverhältnis** („odds ratio“) genannt.
- Der Koeffizient θ_i , $1 \leq i \leq p$ gibt also an, um wie viele Einheiten sich das logarithmierte Chancenverhältnis ändert, wenn man x_i um eine Einheit vergrößert.

Die Logistische Regression lässt sich wie folgt kategorisieren:

Die Logistische Regression lässt sich wie folgt kategorisieren:

- Supervised Learning
- Batch Learning
- Eager Learning
- Parametrisierte Methode

Sie kann als einschichtiges künstliches Neuronales Netzwerk aufgefasst werden:



Bisher hatten wir mit dem Ansatz der Logistischen Regression binäre Klassifikationsprobleme behandelt. Im folgenden werden wir Strategien kennen lernen, um auch Probleme mit mehr als 2 möglichen Klassen zu behandeln. Folgende Begriffe sind zu unterscheiden:

Multiclass classification

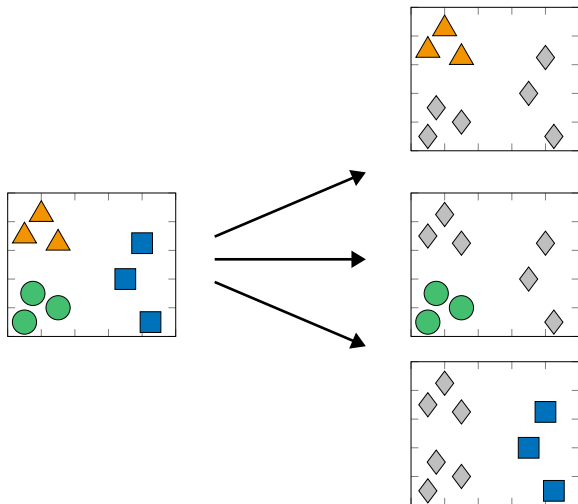
- Es gibt mehr als zwei möglichen Klassen.
- Jedes Objekt kann genau einer Klasse angehören.
- Beispiel: Schwerlinien-Klassifikation (Setosa, Versicolor, Virginica)

Multilabel classification

- Es gibt mehr als zwei mögliche Labels.
- Jedes Objekt kann mit mehr als einem Label versehen werden.
- Beispiel: Zuordnung eines Films zu einer Rubrik (z.B. Action, Comdey, Horror, Thriller etc.)

Strategie „One versus Rest“ (OvR, auch OvA)

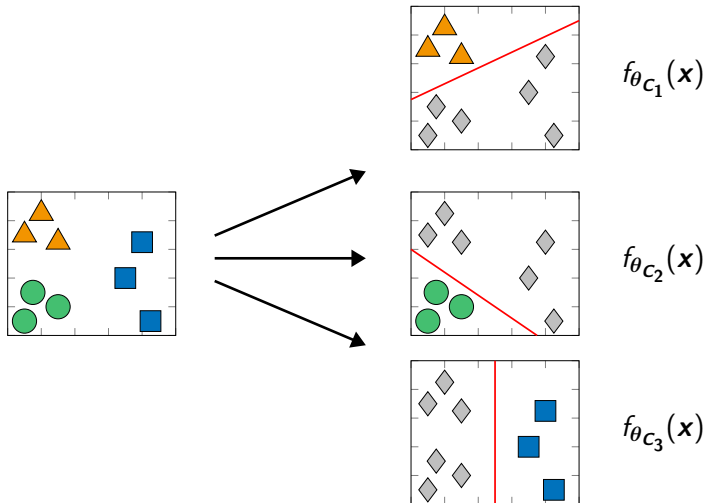
Trainiere pro Klasse C_1, \dots, C_n ein (binäres) Modell.



Logistische Regression mit mehr als 2 Klassen

Strategie „One versus Rest“ (OvR, auch OvA)

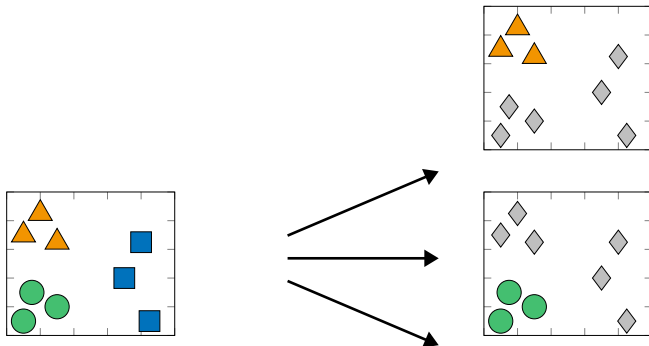
Trainiere pro Klasse C_1, \dots, C_n ein (binäres) Modell.



Logistische Regression mit mehr als 2 Klassen

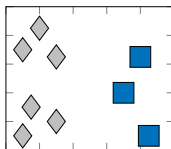
Strategie „One versus Rest“ (OvR, auch OvA)

Trainiere pro Klasse C_1, \dots, C_n ein (binäres) Modell.



Klassenzuordnung:

$$\hat{y} = \arg \max_{c \in \{C_1, \dots, C_n\}} f_{\theta_c}(\mathbf{x})$$



Multinomiale Logistische Regression

- Im Gegensatz zum Ansatz bei OvA wird bei der multinomialen Logistischen Regression ein vektorwertiges Modell erstellt, welches für ein gegebenes \mathbf{x} ein vektorwertiges Label \mathbf{y} prognostiziert.
- Dazu werden die gegebenen Labels zunächst mittels **One Hot Encoding** als binäre Vektoren kodiert:

One Hot Encoding

$$c_1 \mapsto \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n, \quad c_2 \mapsto \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n, \dots, \quad c_n \mapsto \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^n.$$

- Anschließend wird in der Definition der Regressionsgleichung die logistische Funktion g durch die vektorwertige **Softmax**-Funktion ersetzt:

$$\text{softmax}(\mathbf{z})_j := \frac{e^{z_j}}{\sum_{k=1}^n e^{z_k}}.$$

Fitting eines multinomialen Logistischen Regressionsmodells

- Beim Modell-Fitting werden Parameter $\Theta \in \mathbb{R}^{(n,p+1)}$ simultan durch Minimierung der Cross-Entropie bestimmt.

Cross-Entropie bei multinomialer logistischer Regression

$$J(\Theta) = - \sum_{i=1}^m \sum_{j=1}^n y_j^{(i)} \cdot \log(f_{\Theta}(\mathbf{x}^{(i)})_j) .$$

wobei die Modellfunktion f_{Θ} gegeben ist durch

$$f_{\Theta}(\mathbf{x}^{(i)})_j := \text{softmax}(\Theta \mathbf{x}^{(i)})_j .$$

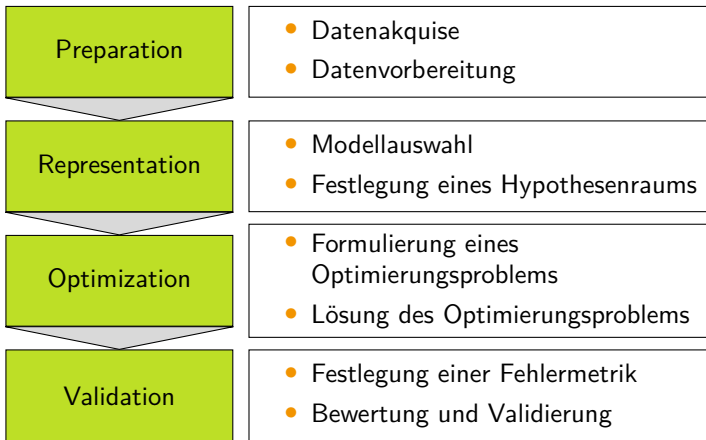
- Sind die Parameter Θ bestimmt, so erfolgt die Klassenzuordnung eines Query Points \mathbf{x}_q schließlich durch

$$\hat{y}^{(i)} := \arg \max f_{\Theta}(\mathbf{x}_q^{(i)}) ,$$

d.h. es wird diejenige Klasse zugeordnet, die dem größten Wert in $f_{\Theta}(\mathbf{x}_q)$ entspricht.

„Bausteine“ von Machine Learning-Ansätzen

Folgende Arbeitsschritte fallen typischerweise bei einer Maschine Learning-Aufgabenstellung an:



Zur Beurteilung eines binären Klassifizierer (mit positiver und negativer Klasse) kann man ihn auf einem Testdatensatz auswerten und die sog.

Konfusionsmatrix aufstellen:

| Vorhergesagte Klasse | Tatsächliche Klasse | |
|----------------------|------------------------|------------------------|
| | Klasse 1 (positiv) | Klasse 0 (negativ) |
| Klasse 1 (positiv) | TP (true positive) | FP (false positive) |
| Klasse 0 (negativ) | FN (false negative) | TN (true negative) |

Hinweis: die Definition der Konfusionsmatrix in `sklearn` lautet

$$C = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix} .$$

Precision und Recall sind zwei häufig verwendete Maße zur Beurteilung der Güte binärer Klassifikatoren (mit negativer und positiver Klasse).

Recall

$$Recall = \frac{TP}{TP + FN}$$

- Gibt den Anteil der Elemente der positiven Klasse an, die korrekt als positiv klassifiziert wurden.
- Zeigt die Fähigkeit des Klassifikators an, relevante Elemente zu finden.
- Synonyme Bezeichnungen: Sensitivität, True positive rate (TPR)

Precision


$$Precision = \frac{TP}{TP + FP}$$

- Gibt den Anteil der als positiv klassifizierten Elemente an, die tatsächlich der positiven Klasse angehören.
- Synonyme Bezeichnung: Positive predictive value (PPV)


Precision und Recall

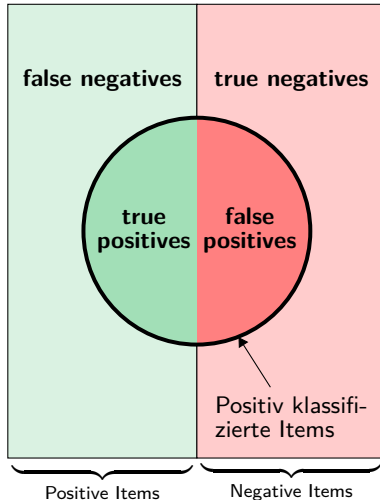
Merkhilfe zu Precision und Recall (vgl. Wikipedia):

Anteil der positiven
an den als positiv
klassifizierten Items

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$


Anteil aller positiven
Items, die als positiv
klassifiziert werden

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$




False Positive Rate

$$FPR = \frac{FP}{FP + TN}$$

Anteil der negativen Items, die irrtümlich als positiv klassifiziert werden.

Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Anteil aller Elemente der Grundgesamtheit, die korrekt klassifiziert wurden.
- Wenig aussagekräftig bei stark unbalancierten Daten

Specifity

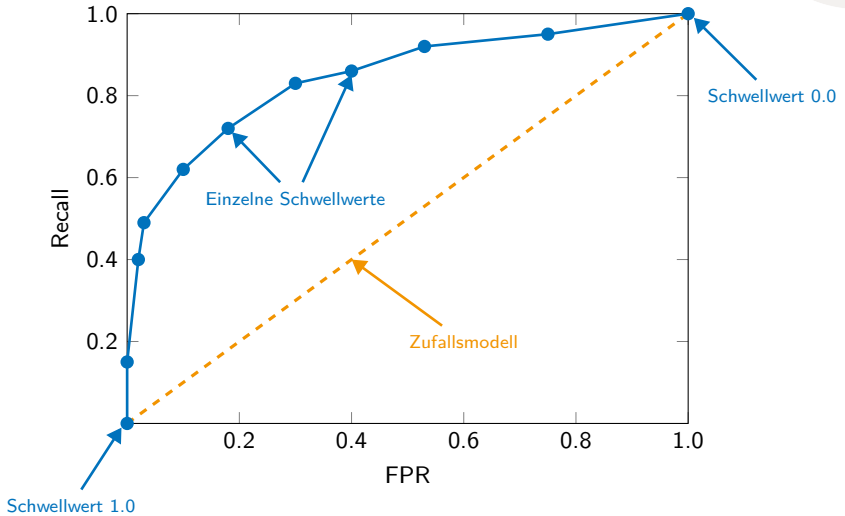
$$Specifity = \frac{TN}{TN + FP}$$

- Gibt den Anteil der negativen Items an, die als negativ klassifiziert werden.
- Misst die Fähigkeit, negative Items korrekt zu erkennen.

Idee

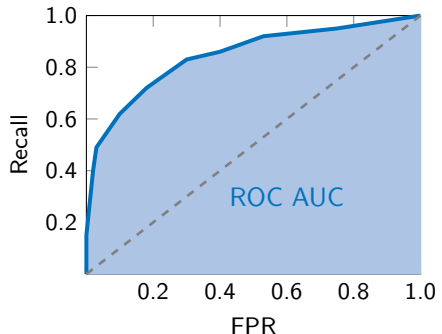
- Ein Logistisches Regressionsmodell prognostiziert nicht nur Klassen-Labels, sondern einen Score, der die Wahrscheinlichkeit der Zugehörigkeit zur positiven Klasse approximiert.
- Je größer dieser Zahlenwert ist, desto „sicherer“ ist sich das Modell, dass ein Item der positiven Klasse angehört.
- Anstatt den Schwellwert 0.5 für die Zuordnung zur positiven Klasse zu verwenden, kann man auch einen höheren oder niedrigeren Schwellwert ansetzen.
- Verständnisfrage: wie verändert sich tendenziell Precision und Recall, wenn man den Schwellwert erhöht?
- In der ROC-Kurve (**R**eciever **O**perating **C**haracteristics) werden die False Positive Rate (FPR) und der Recall, die sich auf einem Testdatensatz für verschiedenen Schwellwerte messen lassen, gegeneinander aufgetragen.
- Die resultierende Kurve erlaubt eine Beurteilung des Klassifikators unabhängig von der Wahl des Schwellwerts.

Beispiel zur ROC-Kurve

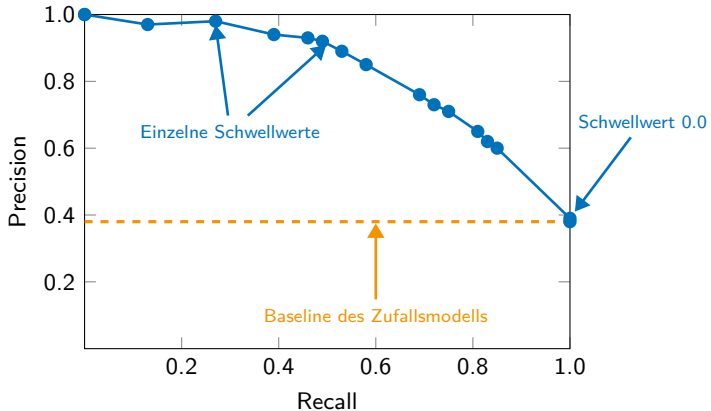


Der ROC AUC-Score

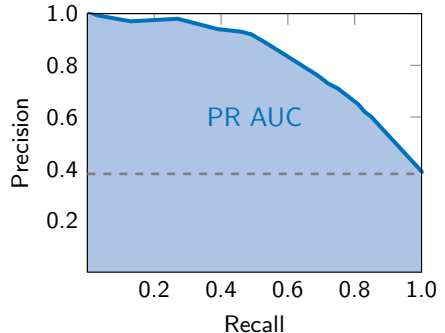
- Der ROC AUC-Score gibt die Fläche unter dem Graphen der ROC-Kurve an.
- Er dient als Maß für die Güte eines binären Klassifikators.
- Ein Zufallsmodell hat einen ROC AUC-Score von 0.5
- Ein perfektes Modell hätte einen ROC AUC-Score von 1.0



- Neben der ROC-Kurve wird auch die Precision-Recall-Kurve zur Beurteilung der Güte eines binären Klassifikators eingesetzt.
- Für verschiedene Schwellwerte werden dazu die Precision und der Recall gegeneinander aufgetragen.



- Der PR AUC-Score gibt die Fläche unter dem Graphen der Precision-Recall-Kurve an.
- Ein perfektes Modell hätte einen PR AUC-Score von 1.0
- Ein Zufallsmodell hätte einen PR AUC-Score von p , wobei p den Anteil der Datensätze mit positivem Label bezeichnet.



Die bislang diskutierten Bewertungsmethoden beziehen sich auf binäre Klassifikatoren. Diese können durch Mikro- und Makro-Methoden zur Mittelwertbildung auf den Fall mehrerer Klassen erweitert werden.

Mikro-Durchschnitt der Precisions

Seien TP_1, \dots, TP_n und FP_1, \dots, FP_n die Anzahl der true positives bzw. false positives der einzelnen Klassen C_1, \dots, C_n . Dann ist der Mikro-Durchschnitt der Precisions gegeben durch

$$Precision_{micro} = \frac{TP_1 + \dots + TP_n}{TP_1 + \dots + TP_n + FP_1 + \dots + FP_n} .$$

Makro-Durchschnitt der Precisions

Der Makro-Durchschnitt gewichtet alle Klassen gleich und berechnet sich einfach als Durchschnittswert aller Precisions für die einzelnen Klassen:

$$Precision_{macro} = \frac{Precision_1 + \dots + Precision_n}{n} .$$

Logistische Regression

- Die Logistische Regression ist ein Vertreter der parametrisierten Verfahren.
- Sie eignet sich in der klassischen Formulierung zur binären Klassifikation.
- Das Modell-Fitting erfolgt durch Lösung eines Optimierungsproblems, bei dem die Cross-Entropie minimiert wird.
- Der Decision Boundary der Logistischen Regression ist linear.
- Logistische Regression kann durch die Strategie „OvA“ oder durch Formulierung als multinomiale Logistische Regression auch zur Multiclass Classification eingesetzt werden.
- Die Methode erwartet numerische Features und setzt voraus, dass es keine fehlenden Werte gibt. Diese Voraussetzungen müssen ggf. im Rahmen der Datenvorbereitung und -transformation hergestellt werden (mehr dazu im weiteren Verlauf der Vorlesung).

Beurteilung binärer Klassifikatoren

- Maße zur Beurteilung binärer Klassifikatoren (z.B. Precision, Recall, Accuracy, False positive rate)
- Mikro- und Makro-Durchschnitte der Precision zur Beurteilung von Klassifikatoren bei der Multiclass Classification.
- Die ROC-Kurve zur grafischen Beurteilung der Güte eines binären Klassifikators hinsichtlich des Recalls und der False Positive Rate.
- Die Precision-Recall-Kurve zur grafischen Beurteilung der Güte binärer Klassifikatoren hinsichtlich Precision und Recall.
- Der ROC AUC-Score und der PR AUC-Score als Kennzahlen zur Beurteilung der Güte binärer Klassifikatoren.