

# Naive Bayes

## Grundlegendes

<https://courses.mooc.fi/org/uh-cs/courses/dap-22/chapter-6/naive-bayes-classification>

- arbeitet auf Basis von Klassifikationen
- Training mit richtig klassifizierten Daten
- berechnet Wahrscheinlichkeiten zur Zugehörigkeit zu einer Klasse anhand von Beobachtungen
- Annahme: Variablen bedingt unabhängig
- Notwendig: Kostenmaß  $\Rightarrow$  Risikofunktion
- Bayes-Klassifikator minimiert Risiko Fehlentscheidung
- Definition über **Minimum-risk-Kriterium**
- wird primitives Kostenmaß eingesetzt (verursacht bei Fehlentscheidungen Kosten)  $\Rightarrow$  Bayes-Klassifikator Wahrscheinlichkeit von Fehlentscheidungen
- $\rightarrow$  Klassifikator über **Maximum-a-posteriori-Kriterium** definiert

## Anwendung?

- Spam-Klassifizierung eingesetzt.
- Klassenvariable zeigt an, ob eine Nachricht Spam oder erwünscht ist
- Alle Wörter in Nachricht entsprechen den Variablen
- Anzahl an Variablen im Modell durch Länge der Nachricht bestimmt

## Varianten ?

- Gaussian Naive Bayes
- Multinomial Naive Bayes

- Bernoulli Naive Bayes
- Complement Naive Bayes
- Categorical Naive Bayes
- Multinomial Naive Bayes: Der 20 Newsgroups Datensatz enthält Texte von Newsgroups, die in 20 verschiedene Kategorien unterteilt sind. Jede Kategorie enthält etwa 1000 Dokumente. Der Datensatz kann hier heruntergeladen werden: <http://qwone.com/~jason/20Newsgroups/>
- Bernoulli Naive Bayes: Ein passender Datensatz für Bernoulli Naive Bayes ist der Spamassassin Public Corpus, der Spam- und Ham-E-Mails enthält. Der Datensatz kann hier heruntergeladen werden: <https://spamassassin.apache.org/publiccorpus/>
- Complement Naive Bayes: Der Reuters-21578 Datensatz ist ein umfangreicher Textdatensatz, der für Complement Naive Bayes geeignet ist. Er enthält Nachrichtenartikel, die in verschiedene Kategorien eingeteilt sind. Der Datensatz kann hier heruntergeladen werden: <https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection>
- Categorical Naive Bayes: Ein passender Datensatz für Categorical Naive Bayes ist der Mushroom Classification Datensatz, der Merkmale von Pilzen enthält und angibt, ob der Pilz essbar oder giftig ist. Der Datensatz kann hier heruntergeladen werden: <https://archive.ics.uci.edu/ml/datasets/Mushroom>

Der Multinomial Naive Bayes ist ein Naive Bayes-Modell, das für diskrete Daten ausgelegt ist, während der Gaussian Naive Bayes für kontinuierliche Daten ausgelegt ist.

Bei Multinomial Naive Bayes werden diskrete Merkmale verwendet, die normalerweise die Häufigkeit oder Anzahl der Vorkommen jedes Merkmals in einem Dokument oder Datensatz darstellen. Es wird verwendet, wenn das Ausmaß der Vorkommen der Merkmale nicht wichtig ist, sondern nur deren Häufigkeit zählt. Typische Anwendungen sind beispielsweise Text- oder Sprachdaten.

Der Gaussische Naive Bayes geht davon aus, dass die kontinuierlichen Merkmale einer Normalverteilung folgen. Es wird verwendet, wenn die Merkmale kontinuierlich sind und die Normalverteilung der Daten angenommen werden kann. Ein Beispiel dafür wäre die Vorhersage von Körpergrößen auf Basis von Gewicht, Größe und Alter

Diskrete Daten sind Daten, die nur bestimmte, getrennte Werte annehmen können. Beispielsweise kann die Anzahl der Kinder in einer Familie nur eine ganzzahlige Zahl sein, da man keine halben Kinder haben kann. Im Gegensatz dazu sind kontinuierliche Daten Werte,

die jeden beliebigen Wert in einem bestimmten Intervall annehmen können, wie zum Beispiel die Körpergröße einer Person.

Ein Anwendungsbeispiel für den multinomialen Naive Bayes ist die Textklassifikation, insbesondere die Sentiment-Analyse von Texten. Dabei wird der multinomiale Naive Bayes verwendet, um automatisch zu bestimmen, ob ein Text einen positiven, negativen oder neutralen Ton hat.

Dazu wird der Text in ein Vektorformat überführt, bei dem jedes Element im Vektor die Häufigkeit eines bestimmten Wortes im Text darstellt. Diese Vektoren werden als Features verwendet, um den multinomialen Naive Bayes zu trainieren. Der Naive Bayes lernt dann, welche Wörter oder Wortgruppen in positiven, negativen oder neutralen Texten häufig auftreten, und verwendet dieses Wissen, um neue, unklassifizierte Texte zu klassifizieren.

Ein Beispiel für die Anwendung des multinomialen Naive Bayes in der Textklassifikation ist die Analyse von Produktrezensionen auf einer E-Commerce-Website. Der Naive Bayes kann verwendet werden, um automatisch die Stimmung der Rezensionen zu klassifizieren, indem er die in den Rezensionen verwendeten Wörter analysiert. Dies kann dem Unternehmen helfen, Feedback von Kunden schnell und effizient zu verarbeiten und darauf zu reagieren. Der Bernoulli Naive Bayes ist eine Variante des Naive Bayes Algorithmus, der für binäre Daten verwendet wird. Die Anwendung des Bernoulli Naive Bayes besteht darin, Textdokumente automatisch in Kategorien zu klassifizieren, indem die Wörter in den Dokumenten als Features betrachtet werden.

Das Verfahren geht davon aus, dass jedes Wort in einem Dokument unabhängig voneinander entweder vorhanden oder nicht vorhanden ist. Anhand einer Trainingsmenge wird für jedes Wort in jeder Kategorie die Wahrscheinlichkeit berechnet, mit der es in dieser Kategorie auftritt. Bei der Klassifikation eines neuen Dokuments wird für jedes darin vorkommende Wort die Wahrscheinlichkeit berechnet, dass es in jeder Kategorie auftritt. Die Wahrscheinlichkeiten werden dann miteinander multipliziert und mit der Prior-Wahrscheinlichkeit der jeweiligen Kategorie gewichtet. Die Kategorie mit der höchsten Wahrscheinlichkeit wird als Vorhersage für das Dokument ausgewählt.

Der Bernoulli Naive Bayes ist insbesondere für Textklassifikationsaufgaben geeignet, bei denen die Features binär sind, das heißt, entweder sind sie im Dokument vorhanden oder nicht. Ein Beispiel dafür ist die Klassifikation von Spam-Mails, bei der man sich auf das Vorhandensein oder Nichtvorhandensein bestimmter Wörter oder Wortkombinationen im Text konzentriert.

Complement Naive Bayes und Categorical Naive Bayes sind Varianten des Naive Bayes-Klassifikators.

Der Complement Naive Bayes ist eine Erweiterung des Multinomial Naive Bayes, der für unbalancierte Datensätze entwickelt wurde. Während der Multinomial Naive Bayes davon ausgeht, dass alle Merkmale gleich wichtig sind, berücksichtigt der Complement Naive Bayes die ungleiche Verteilung der Klassen in den Trainingsdaten. Er versucht, die Klassen, die unterrepräsentiert sind, besser zu modellieren. Der Complement Naive Bayes ist

besonders nützlich, wenn es eine große Anzahl von Merkmalen gibt und die Daten unbalanciert sind.

Der Categorical Naive Bayes ist eine Erweiterung des Bernoulli Naive Bayes, der für Daten mit kategorialen Merkmalen entwickelt wurde. Im Gegensatz zum Bernoulli Naive Bayes, der nur binäre Merkmale verarbeiten kann, kann der Categorical Naive Bayes mit kategorialen Merkmalen umgehen, die mehr als zwei Ausprägungen haben. Er modelliert die Wahrscheinlichkeit jedes Merkmals in jeder Klasse und verwendet dann Bayes' Theorem, um die Wahrscheinlichkeit jeder Klasse für einen neuen Datensatz zu berechnen.

Ein Anwendungsbeispiel für den Complement Naive Bayes ist die Textklassifizierung bei unbalancierten Datensätzen, z.B. die Klassifizierung von Spam-E-mails, bei denen die Anzahl der Spam-E-mails im Vergleich zu den Nicht-Spam-E-mails sehr hoch ist. Ein Anwendungsbeispiel für den Categorical Naive Bayes ist die Klassifizierung von Produkten in verschiedene Kategorien, z.B. die Klassifizierung von Kleidungsstücken in verschiedene Größen oder Farben.

**Beim Naive Bayes bezieht sich die Vektorisierung auf die Umwandlung von Textdaten in numerische Vektoren, die von dem Algorithmus verwendet werden können, um eine Vorhersage zu treffen. Vektorisierung beinhaltet typischerweise Schritte wie Tokenisierung, Stop-Wort-Entfernung und Umwandlung der Tokens in Zahlenwerte, wie z.B. die Anzahl der Vorkommen eines bestimmten Wortes in einem Dokument. Dies wird häufig mit Hilfe von Bag-of-Words-Modellen erreicht, die jede eindeutige Texteinheit in einen Vektor umwandeln, der die Häufigkeit des Auftretens jedes Wortes im Text darstellt.**

## Wie funktioniert Naive Bayes?

- benutzt alle gegebenen Attribute
- zwei Annahmen über diese Attribute
  - alle Attribute in der Annahme gleich wichtig
  - Attribute statistisch unabhängig
    - Kenntnis eines Wertes sagt nichts über Wert anderen Attribut aussagt
    - Diese Unabhängigkeitsannahme stimmt allerdings nie
    - Dennoch funktioniert dieses Verfahren in der Praxis gut!
    - kann es gut mit fehlenden Werten arbeiten.

## Beispiel

- Trainings-Datenmenge von Wetter ↔ Möglichkeit ein Sportspiel bei schönem Wetter durchzuführen. Der erste Schritt besteht darin, die Daten in eine Häufigkeitstabelle umzuwandeln. Eine Wahrscheinlichkeitstabelle wird danach im zweiten Schritt erzeugt mit der Suche nach Wahrscheinlichkeiten wie bedecktes Wetter (0.29) und der Wahrscheinlichkeit fürs Spielen von (0.64). Im dritten Schritt wird die Naive Bayes Gleichung verwendet, um die Posterior Wahrscheinlichkeit für jede Klasse zu berechnen. Die Klasse mit der höchsten Posterior Wahrscheinlichkeit ist das Ergebnis der Vorhersage

## wie?

Der Naive-Bayes Algorithmus basiert auf dem Bayes Theorem. Es beschreibt eine Formel zur Berechnung der bedingten Wahrscheinlichkeit  $P(A|B)$  oder in Worten: Wie hoch ist die Wahrscheinlichkeit, dass das Ereignis A eintritt, wenn Ereignis B eingetreten ist? Als Beispiel: Wie wahrscheinlich ist es, dass ich an Corona erkrankt bin (= Ereignis A), wenn mein Schnelltest positiv ist (= Ereignis B)?

Diese bedingte Wahrscheinlichkeit lässt sich laut Bayes mithilfe der folgenden Formel berechnen:

$$P(A|B) = P(B|A) * P(A) / P(B)$$

- $P(B|A)$  = Wahrscheinlichkeit, dass Ereignis B eintritt, wenn Ereignis A schon eingetreten ist
- $P(A)$  = Wahrscheinlichkeit, dass Ereignis A eintritt
- $P(B)$  = Wahrscheinlichkeit, dass Ereignis B eintritt

Warum sollten wir diese Formel nutzen? Kommen wir zurück zu unserem Beispiel mit dem positiven Test und der Corona Erkrankung. Die bedingte Wahrscheinlichkeit  $P(A|B)$  kann ich nicht wissen und nur über ein aufwendiges Experiment herausfinden. Die umgedrehte Wahrscheinlichkeit  $P(B|A)$  hingegen ist einfacher herauszufinden. In Worten bedeutet sie: Wie wahrscheinlich ist es, dass eine an Corona erkrankte Person einen positiven Schnelltest vorweist.

Diese Wahrscheinlichkeit lässt sich verhältnismäßig einfach herausfinden, indem man nachweislich erkrankte Personen einen Schnelltest durchführen lässt und dann das Verhältnis errechnet, wie viele der Test auch wirklich positiv waren. Die Wahrscheinlichkeiten  $P(A)$  und  $P(B)$  lassen sich ähnlich einfach herausfinden. Die Formel ermöglicht es dann die bedingte Wahrscheinlichkeit  $P(A|B)$  einfach zu errechnen.

Wenn wir nur eine Eigenschaft haben, ist damit auch schon der komplette Naive Bayes Algorithmus erklärt. Mit einem Feature für die bedingte Wahrscheinlichkeit  $P(x | K)$  für verschiedene Klassen errechnet und die Klasse mit der höchsten Wahrscheinlichkeit gewinnt. Für unser Beispiel bedeutet das, es werden die beiden bedingten Wahrscheinlichkeiten  $P(\text{Person ist krank} | \text{Test ist positiv})$  und  $P(\text{Person ist gesund} | \text{Test ist negativ})$  mithilfe des Bayes Theorem berechnet und die Klassifizierung erfolgt für die Klasse mit der höheren Wahrscheinlichkeit.

Wenn unser Datensatz aus mehr als nur einem Feature besteht gehen wir ähnlich vor und berechnen für jede Kombination aus Feature x und Klasse K die bedingte Wahrscheinlichkeit. Dann multiplizieren wir alle Wahrscheinlichkeiten für ein Feature. Die Klasse K, die dann das höchste Produkt an Wahrscheinlichkeiten hat, ist die entsprechende Klasse des Datensatzes.

## Was sind die Vor- und Nachteile des Naive Bayes Algorithmus?

Der Naive Bayes Algorithmus ist ein beliebter Einstieg in einen Klassifizierungsanwendung, da er sehr einfach und schnell zu trainieren ist und teilweise auch gute Ergebnisse liefern

kann. Sollte die Annahme der Unabhängigkeit der einzelnen Features gegeben sein, performt er sogar besser als vergleichbare Klassifizierungsmodelle, wie eine [logistische Regression](#), und benötigt weniger Daten zum Trainieren.

Obwohl der Naive Bayes Algorithmus bereits mit wenigen Daten gute Ergebnisse erzielen kann, benötigen wir so viele Daten, dass jede Klasse mindestens einmal im Trainingsdatensatz erscheint. Ansonsten wird der Naive Bayes Classifier für die Kategorie im Testdatensatz eine Wahrscheinlichkeit von 0 als Ergebnis liefern. Außerdem ist es in der Realität sehr unwahrscheinlich, dass alle Inputvariablen komplett unabhängig voneinander sind, was sich zudem auch nur sehr schwierig testen lässt.

## Wie kann man den Naive Bayes-Algorithmus verbessern?

Es gibt mehrere Möglichkeiten, die Leistung des Naive-Bayes-Algorithmus zu verbessern. Hier sind einige gängige Techniken:

- **Feature Engineering:** Die Leistung von Naive Bayes hängt von der Qualität der Eingangsmerkmale ab. Durch sorgfältige Auswahl und Umwandlung der Eingangsmerkmale können wir die Genauigkeit des Modells verbessern. Zum Beispiel können wir Techniken wie Merkmalskalierung, Merkmalsauswahl und Merkmalsextraktion verwenden, um die Qualität der Eingangsmerkmale zu verbessern.
- **Smoothing:** Naive Bayes kann unter Null-Häufigkeits-Problemen leiden, wenn eine bestimmte Merkmals- und Klassenkombination in den Trainingsdaten nicht vorhanden ist. Glättungsverfahren wie die Laplace-Glättung und die additive Glättung können helfen, dieses Problem zu lösen, indem eine kleine Konstante zur Anzahl der einzelnen Merkmale hinzugefügt wird.
- **Ensemble-Methoden:** Wir können mehrere Naive Bayes-Modelle kombinieren, um die Genauigkeit des Klassifikators zu verbessern. Eine Möglichkeit, dies zu tun, ist die Verwendung von Bagging- oder Boosting-Techniken, wie [Random Forest](#) und AdaBoost.
- **Parameter Tuning:** Naive Bayes verfügt über mehrere Hyperparameter, die abgestimmt werden können, um die Leistung des Algorithmus zu verbessern. So können wir beispielsweise den Glättungsparameter anpassen, die beste Technik für die Merkmalsauswahl auswählen oder den optimalen Satz von Merkmalen wählen.
- **Model Selection:** Neben Naive Bayes gibt es andere Klassifizierungsalgorithmen, die für bestimmte Datentypen besser geeignet sein können. Durch den Vergleich der Leistung von Naive Bayes mit anderen Algorithmen können wir das beste Modell für unsere Daten auswählen.
- **Umgang mit unausgewogenen Daten:** Wenn die Anzahl der Stichproben in jeder Klasse unausgewogen ist, kann Naive Bayes gegenüber der Mehrheitsklasse voreingenommen sein. Um dieses Problem zu lösen, können wir Techniken wie Oversampling, Undersampling oder Klassengewichtung verwenden.
- **Behandlung kontinuierlicher Merkmale:** Beim Standardalgorithmus wird davon ausgegangen, dass die Eingabemerkmale kategorisch sind. In vielen realen Anwendungen können die Eingabemerkmale jedoch kontinuierlich sein. Um kontinuierliche Merkmale zu verarbeiten, können wir Techniken wie Diskretisierung, Kernel-Dichte-Schätzung oder Gaussian Naive Bayes verwenden.

Dies sind einige gängige Techniken, die zur Verbesserung der Leistung von Naive Bayes verwendet werden können. Die Wahl der Technik hängt von der jeweiligen Problemstellung und den vorliegenden Daten ab.

## **Was ist der Unterschied zwischen Multinomial Naive Bayes und Bernoulli Naive Bayes?**

Multinomial und Bernoulli Naive Bayes sind zwei beliebte Varianten des Naive Bayes Algorithmus, die häufig in der Textklassifizierung verwendet werden. Der Hauptunterschied zwischen beiden Algorithmen ist die Art und Weise, wie sie die Eingabedaten darstellen. Multinomial Naive Bayes geht davon aus, dass die Eingabedaten durch die Anzahl oder Häufigkeit von Wörtern dargestellt werden, während Bernoulli Naive Bayes davon ausgeht, dass die Eingabedaten durch binäre Merkmale dargestellt werden, d. h. durch das Vorhandensein oder Nichtvorhandensein eines Wortes in einem Dokument.

Bei Multinomial Naive Bayes werden die Eingabedaten in der Regel als Bag-of-Words dargestellt, wobei gezählt wird, wie oft jedes Wort in einem Dokument vorkommt. Der Klassifikator schätzt dann die bedingte Wahrscheinlichkeit jedes Wortes in Abhängigkeit von der Klassenvariablen unter Verwendung einer Multinomialverteilung. Bei Bernoulli Naive Bayes werden die Eingabedaten als binäre Merkmale dargestellt, wobei jedes Merkmal angibt, ob ein bestimmtes Wort in dem Dokument vorhanden ist oder nicht. Der Klassifikator schätzt dann die bedingte Wahrscheinlichkeit jedes Merkmals in Abhängigkeit von der Klassenvariablen unter Verwendung einer Bernoulli-Verteilung.

Ein weiterer wichtiger Unterschied zwischen den beiden Algorithmen ist die Art und Weise, wie sie mit fehlenden Merkmalen umgehen. Bei Multinomial Naive Bayes wird das Fehlen eines Wortes als Häufigkeitszahl Null behandelt, was zu Problemen mit Nullwahrscheinlichkeiten führen kann. Im Gegensatz dazu modelliert Bernoulli Naive Bayes das Fehlen eines Wortes ausdrücklich als separates Merkmal und behandelt es entsprechend.

Beide Algorithmen gehen davon aus, dass die Merkmale in Abhängigkeit von der Klassenvariablen bedingt unabhängig sind. Die Art und Weise, wie diese Annahme formuliert wird, ist jedoch in jedem Fall leicht unterschiedlich. Bei Multinomial Naive Bayes werden die Merkmale als aus einer Multinomialverteilung gezogen modelliert, während bei Bernoulli Naive Bayes die Merkmale als aus einer Bernoulli-Verteilung gezogen modelliert werden.

Die Wahl des Algorithmus hängt von der spezifischen Aufgabe und der Art der Eingangsmerkmale ab. Multinomial Naive Bayes wird üblicherweise für Textklassifizierungsaufgaben verwendet, bei denen die Eingangsmerkmale diskrete Wortzahlen oder Häufigkeiten sind. Bernoulli Naive Bayes wird üblicherweise für binäre oder Präsenz-Absenz-Merkmale verwendet, z. B. für die Klassifizierung von Spam oder die Stimmungsanalyse.

## **Welche Anwendungen nutzen den Naive Bayes Algorithmus?**

Im Bereich des Machine Learnings wird Naive Bayes als Klassifikationsmodell eingesetzt, also zur Einordnung eines Datensatzes in eine bestimmte Klasse. Für diese Modelle gibt es

verschiedene, konkrete Anwendungen für die auch Naive Bayes genutzt wird:

## Natural Language Processing

In diesem Bereich kann das Modell eingesetzt werden, um einen Textabschnitt einer bestimmten Klasse zuzuordnen. E-Mail Programme sind beispielsweise daran interessiert, eingehende Mails als „Spam“ oder „Nicht Spam“ zu klassifizieren. Dazu werden dann die bedingten Wahrscheinlichkeiten von einzelnen Wörtern berechnet und mit der Klasse abgeglichen. Das gleiche Verfahren kann auch genutzt werden, um Social Media Kommentare als „positiv“ oder „negativ“ zu klassifizieren.

Obwohl Naive Bayes für diese Anwendungen im Textbereich einen schnellen und einfachen Ansatz bietet, gibt es andere Modelle, wie beispielsweise [Transformer](#), die deutlich bessere Ergebnisse liefern. Das Naive Bayes Modell berücksichtigt nämlich keine Reihenfolge von Wörtern oder eine gewisse Anordnung. Wenn ich beispielsweise sage „Ich finde das Produkt nicht gut.“ ist das wahrscheinlich keine positive Produktrezension, nur weil das Wort „gut“ darin vorkommt.

## **Klassifizierung von Kreditrisiken**

Für Banken ist der Ausfall von Krediten ein immenses Risiko, da sie große Summen an Geld verlieren, wenn ein Kunde den Kredit nicht mehr bezahlen kann. Deshalb wird viel Arbeit in Modelle gesteckt, die je nach Kunde das individuelle Ausfallrisiko errechnen können. Das ist im Endeffekt auch eine Klassifizierung bei der der Kunde entweder der Gruppe „Kreditrückzahlung“ oder „Kreditausfall“ zugeordnet wird. Dazu werden einige spezifische Eigenschaften, wie beispielsweise Kredithöhe, Einkommen oder Anzahl vorheriger Kredite genutzt. Mithilfe von Naive Bayes kann daraus ein zuverlässiges Klassifizierungsmodell trainiert werden.

## **Vorhersage von medizinischer Behandlung**

In der Medizin muss ein Arzt entscheiden, welche Behandlung und welche Medikamente für den einzelnen Patienten und dessen Krankheitsbild am vielversprechendsten sind und die höchste Wahrscheinlichkeit besitzen den Patienten wieder gesund zu machen. Als Unterstützung kann dazu ein Naive Bayes Klassifizierungsmodell trainiert werden, das abhängig von Eigenschaften des Gesundheitszustands, wie Blutdruck, Wohlbefinden oder Symptome, sowie der möglichen Behandlung (Medikamente) eine Wahrscheinlichkeit berechnet, dass der Kunde wieder gesund wird oder eben nicht. Die Ergebnisse des Modells kann wiederum der Arzt in seine Entscheidung mit einbeziehen.

## **Das solltest Du mitnehmen**

- Der Naive Bayes Algorithmus ist eine einfache Methode, um Daten zu klassifizieren.
- Er basiert auf dem Bayes Theorem und ist naiv, da er davon ausgeht, dass alle Inputvariablen und deren Ausprägung unabhängig voneinander sind.
- Der Naive Bayes Algorithmus lässt sich relativ schnell und einfach trainieren, liefert aber in vielen Fällen keine guten Ergebnisse, da die Annahme der Unabhängigkeit der Variablen verletzt ist.



[https://scikit-learn.org/stable/modules/naive\\_bayes.html](https://scikit-learn.org/stable/modules/naive_bayes.html)

[https://www.python-kurs.eu/naive\\_bayes\\_klassifikator\\_scikit.php](https://www.python-kurs.eu/naive_bayes_klassifikator_scikit.php)

<https://www.kaggle.com/datasets/mvjimbo/naive-bayes-classifier-dataset>