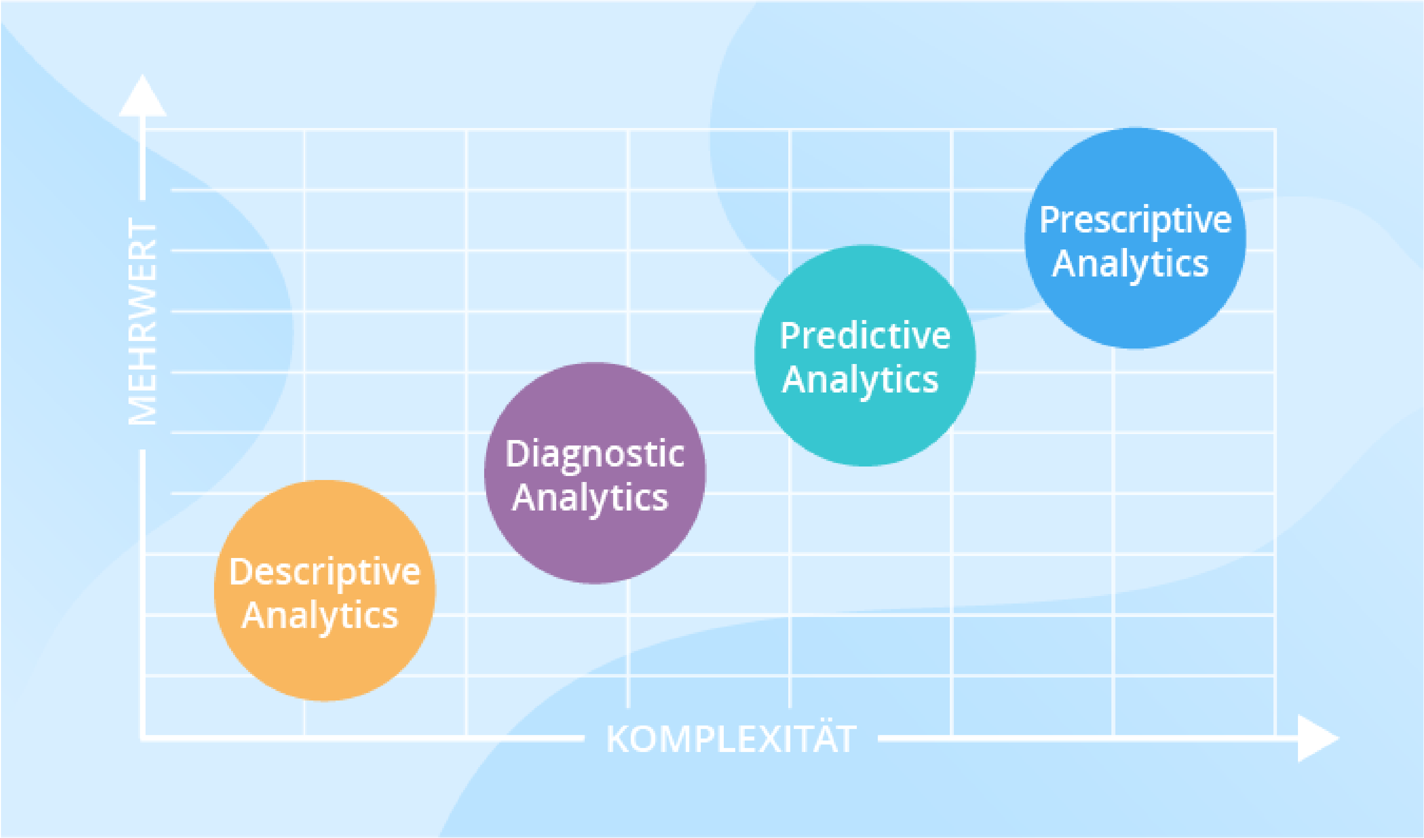
Business Intelligence

Vert. Prof. Dr. Aikaterini Nakou

Datenanalyse

Methoden der Datenanalyse



Künstliche Intelligenz

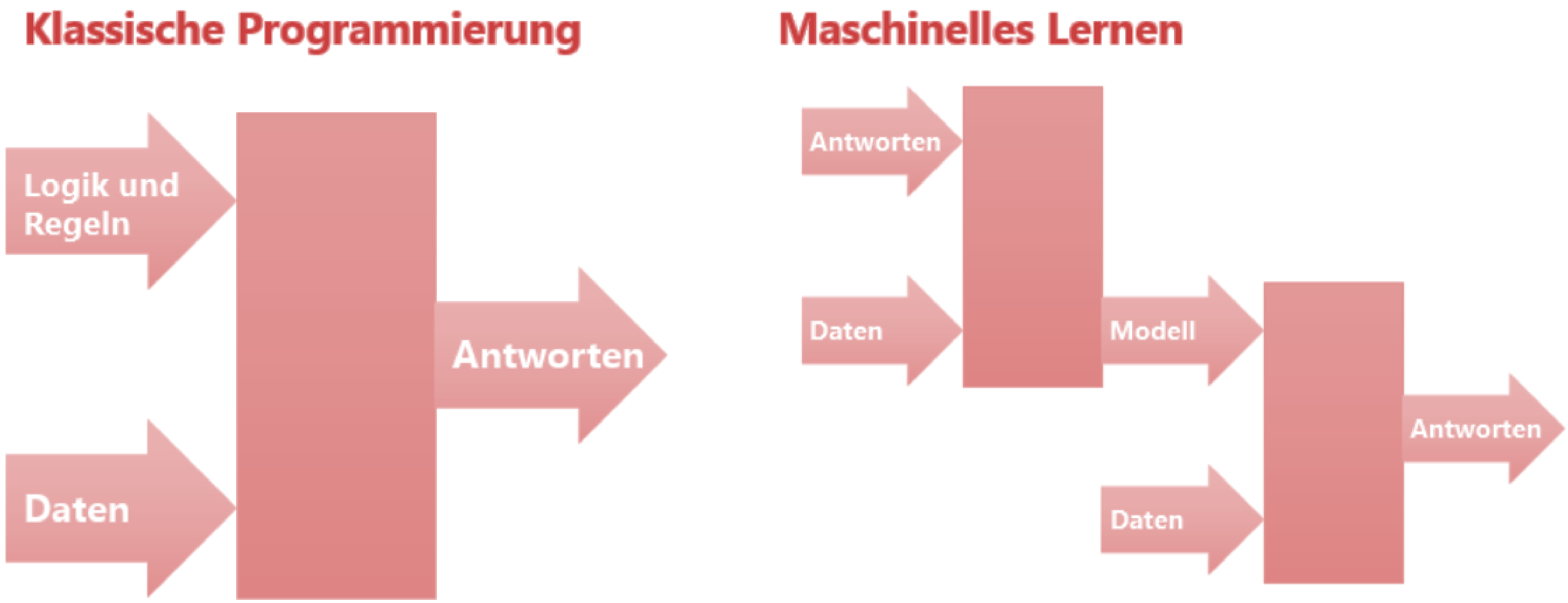
KI ist ein Mix aus vielen **verschiedenen Technologien**. Sie **befähigt Maschinen** dazu, mit menschenähnlicher Intelligenz zu **verstehen, zu handeln und zu lernen**.

Machine Learning ist ein Teilbereich von KI, der es Computern ermöglicht, **aus Erfahrungen zu lernen**, ohne explizit programmiert zu werden.

Deep Learning ist eine fortgeschrittene Form des Machine Learning, die auf **künstlichen neuronalen Netzen** basiert.

Quelle: IONOS: Deep learning vs. machine learning – what are the differences?

Machine Learning



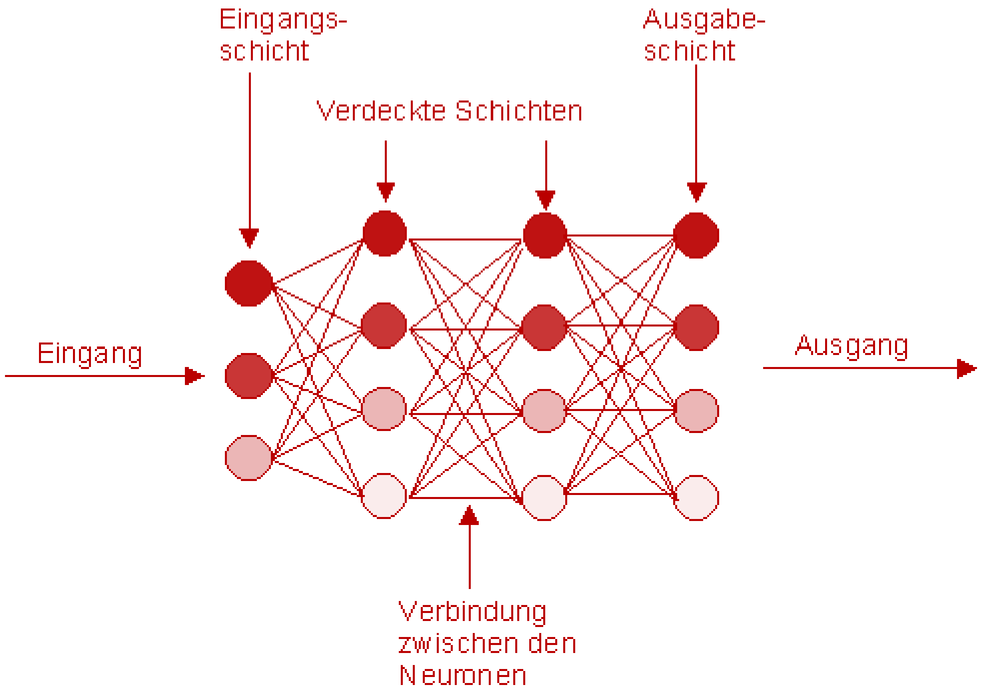
Typen von Lernalgorithmen (ML)

Überwachtes Lernen	Unüberwachtes Lernen	Bestärkendes Lernen
• Werden mit sowohl Eingabe- als auch Ausgabedaten trainiert • **Ziel**: Zuordnung oder Beziehung zwischen den Eingaben und den entsprechenden Ausgaben zu lernen • **Anwendungen**: Vorhersage, Klassifizierung	• Werden nur mit Eingabedaten trainiert. • **Ziel**: Das Modell sucht nach Mustern und Strukturen in den Daten, ohne auf eine spezifische Ausgabe hingewiesen zu werden. • **Anwendungen**: Clusteranalyse, Dimensionsreduktion	• Lernt durch Belohnung/Bestrafung →Belohnt werden die guten Aktionen, bestraft werden die schlechte Aktionen • **Ziel**: Durch das Optimieren der Belohnungsfunktion lernt das Modell, die besten Aktionen für bestimmte Situationen zu wählen. • **Anwendungen**: Prozessoptimierung, Robotik

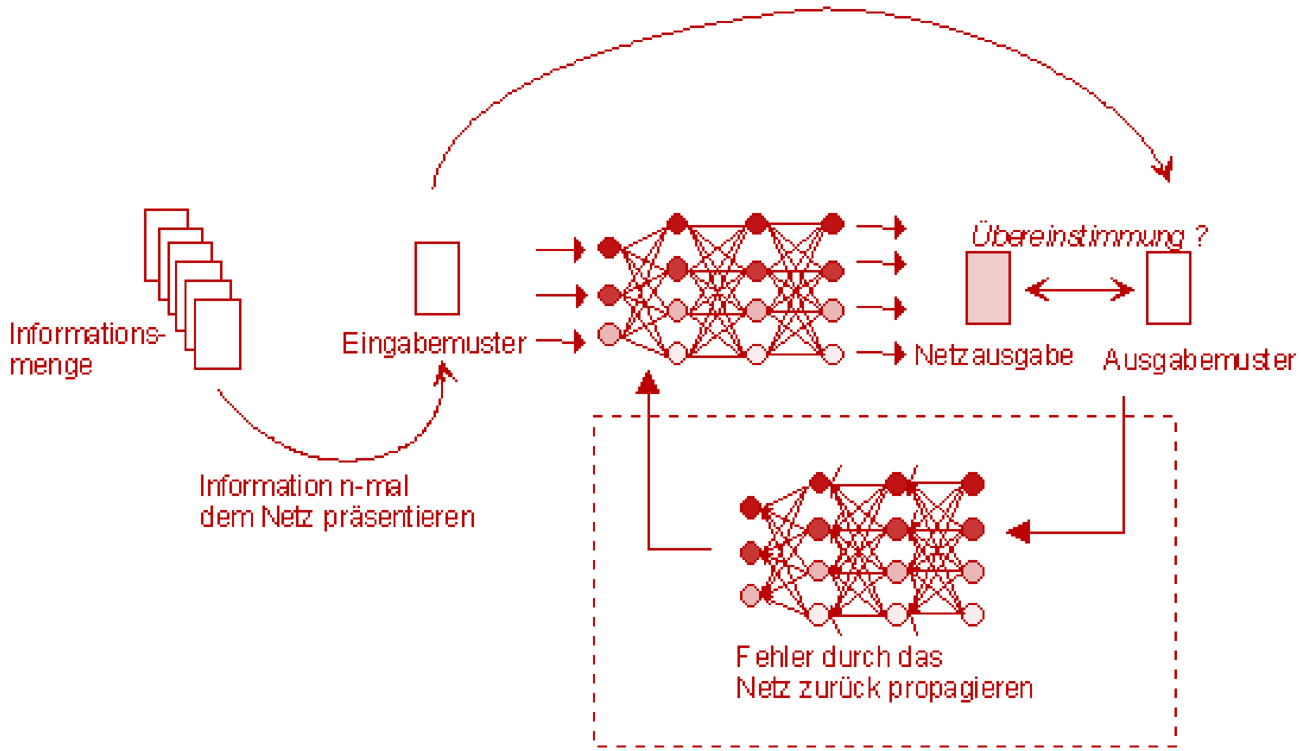
Deep Learning

Deep Learning ist ein Teilbereich des Maschinellen Lernens.

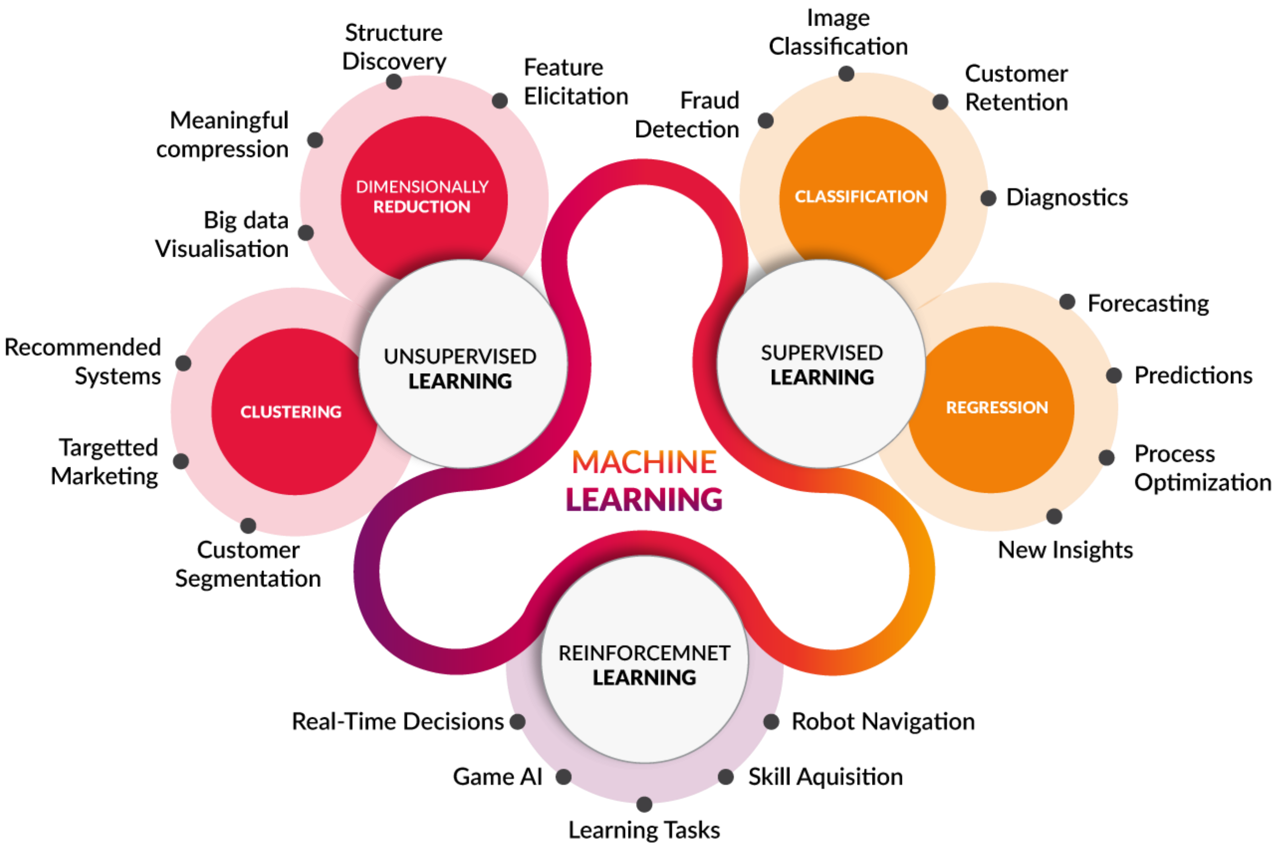
Grundlage sind Neuronale Netze mit mindestens einem Hidden-Layer



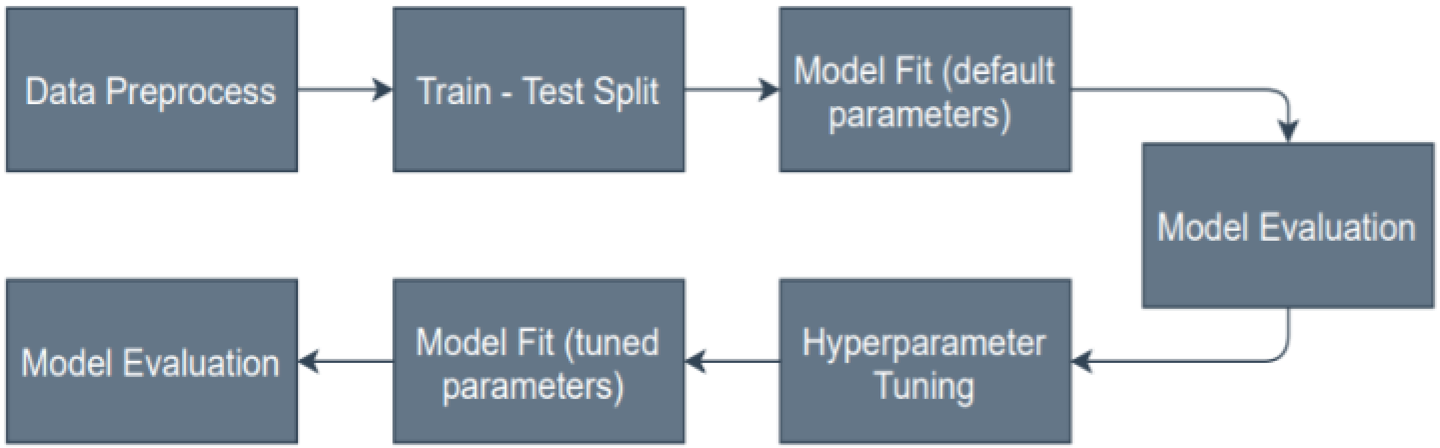
Deep Learning



Anwendungsfälle KI



Modellbildung Machine Learning



Merkmalsauswahl

Die **Merkmalsauswahl** beinhaltet die Identifizierung der **wichtigsten Merkmale** für das **Modell**.

Dies kann durch:

- statistische Methoden wie Korrelationsanalyse oder

- iteratives Vorgehen wie „forward/backward selection“ erfolgen.

Eine gute Merkmalsauswahl kann die **Modellgenauigkeit verbessern** und **Overfitting verhindern**.

Einfache Lineare Regression

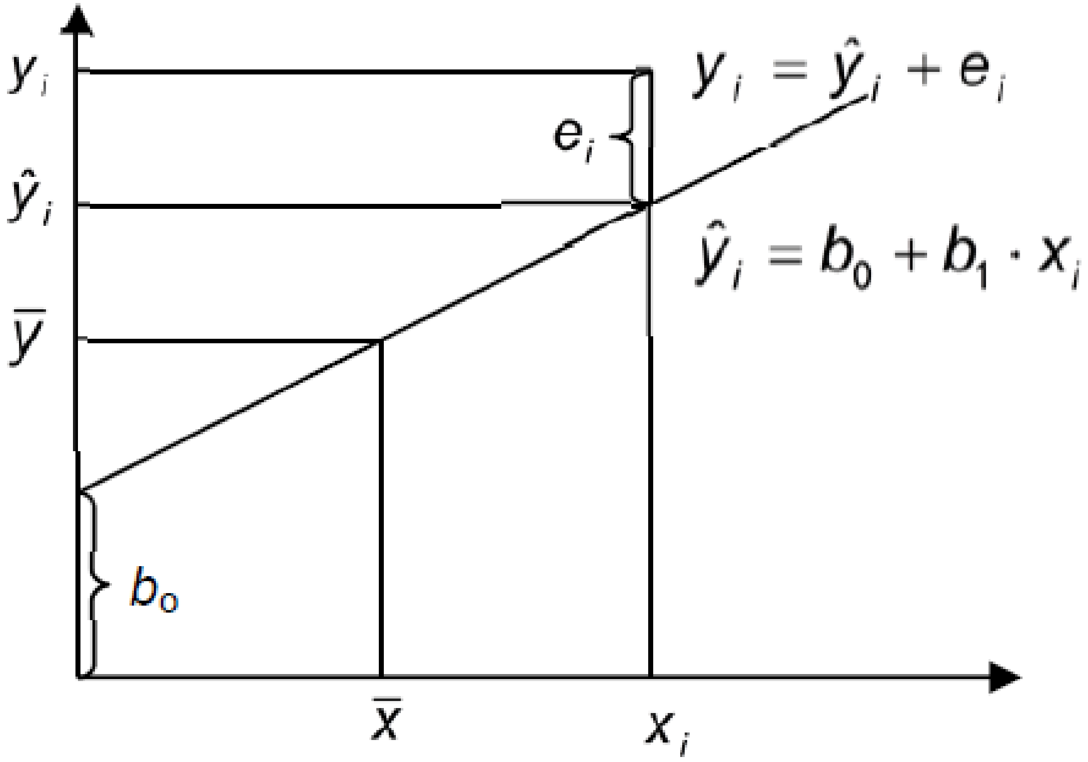
Repräsentation der Punktwolke durch eine Gerade der allgemeinen Form:

$$Y = b0 + b1 * X$$

Dabei stehen:

- *y* für die abhängige Variable,
- *x* für die unabhängige Variable,
- *b*0 für den Schnittpunkt der Geraden mit der *y*-Achse des Koordinatensystems
- *b*1 für die Steigung der Geraden, auch Regressionskoeffizient genannt

Darstellung- Regressionsgerade



Methode der kleinsten Quadrate

Die Regressionsgerade ist diejenige Gerade, die die Summe der quadrierten Residuen (Abweichungen, Vorhersagefehler) minimiert.

Es gilt:

$$e_i = y_i - \hat{y}_i$$

$$e_i = y_i - (b_0 + b_1 \cdot x_i)$$

$$e_i^2 = [y - (b_0 + b_1 \cdot x_i)]^2$$

Gefordert ist:

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} [y - (b_0 + b_1 \cdot x_i)]^2 \rightarrow Min$$

Methode der kleinsten Quadrate

$$b_0 = \overline{y} - b_1 \cdot \overline{x}$$

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sum_{i=1}^{n} (x_j - \overline{x})^2} = \frac{\sum_{j=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i \cdot \sum_{i=1}^{n} y_i \right) \Big/ n}{\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2 \Big/ n}$$

$$b_1 = \frac{\textit{Summe der Abweichungsprodukte}_{xy}}{\textit{Summe der Abweichungsquadrate}_{xy}} = \frac{SP_{xy}}{SQ_{xy}}$$

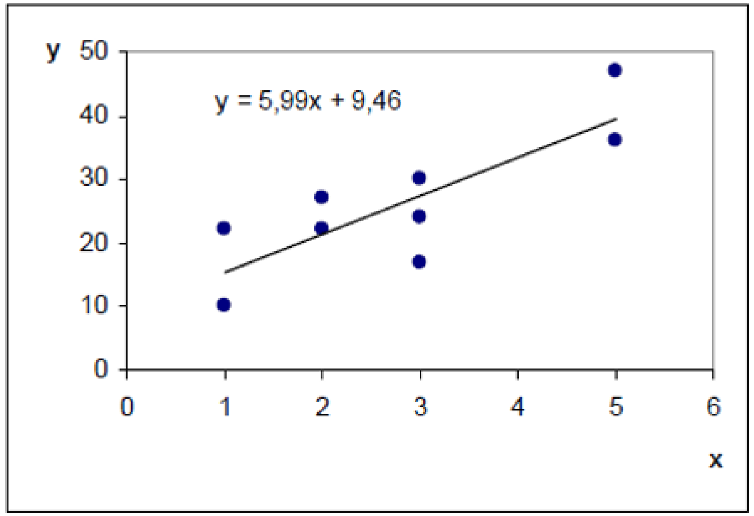
Beispiel

	Koeffizienten	*Standardfehler*	*t-Statistik*	*P-Wert*
Schnittpunkt	9,4618	4,8596	1,9471	0,0926
X Variable 1	5,9937	1,5630	3,8347	0,0064

Die Regressionsgerade lautet damit:

$$\hat{y} = b_0 + b_1 \cdot x = 9{,}4618 + 5{,}9937 \cdot x$$

In Worten: Ändert sich die Einflussgröße x um eine Einheit, so ändert sich die Zielgröße y um 5,9937 Einheiten. Ist die Einflussgröße = 0, so beträgt der Wert der Zielgröße = 9,4618.



Anpassungsgüte

Den Anteil der durch die Regression erklärten Streuung an der Gesamtstreuung bezeichnet als **Bestimmtheitsmaß** ***r*²**:

$$r^2 = \frac{SQ_{Reg}}{SQ_{Ges}} = \frac{b_1 \cdot \left[\sum_{i=1}^{n}(x_i - \overline{x}) \cdot (y_i - \overline{y})\right]}{\sum_{i=1}^{n}(y_i - \overline{y})^2} = b_1 \cdot \frac{SP_{xy}}{SQ_{xy}} = b_1^2 \cdot \frac{SQ_x}{SQ_y} \qquad ($$

Für das obige Beispiel folgt:

$$r^2 = \frac{5{,}9937 \cdot 105{,}2222}{930{,}8889} = 0{,}6775$$

Was sind Zeitreihen?

Eine **Zeitreihe** ist eine **sequenzielle** Abfolge von Datenpunkten, die in **regelmäßigen Zeitintervallen gemessen oder beobachtet** werden.

Zeitreihen finden Anwendung in verschiedenen Bereichen wie Wirtschaft, Klimatologie, Finanzwesen und mehr.

Quelle: Hayes S., 2021: Finding Seasonal Trends in Time-Series

Was ist Zeitreihenanalyse?

Die Zeitreihenanalyse beschäftigt sich mit statistischen Methoden zur **Analyse und Modellierung** einer geordneten Folge von Beobachtungen (**Zeitreihe**).

Diese Modellierung führt zu einem Prozessmodell für das System, das die Daten erzeugt hat.

Somit können anhand dieses Modells **zukünftige Ereignisse vorhergesagt** werden.

Quelle: Özen A., 2021: Seasonality Analysis and Forecast in Time Series

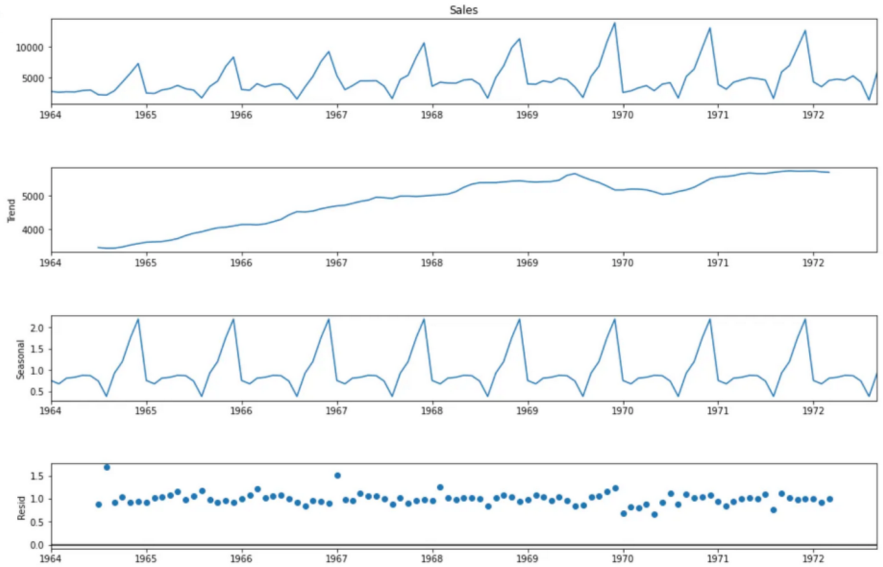
Decomposition - Zerlegung

Bei der Zeitreihenzerlegung können:

- **additive** oder
- **multiplikative**

Bei **additiven** Modellen wird die Zeitreihe als **Summe** von Trend, saisonaler Komponente und Restkomponente dargestellt.

Bei **multiplikativen** Modellen wird die Zeitreihe als **Produkt** dieser Komponenten modelliert.



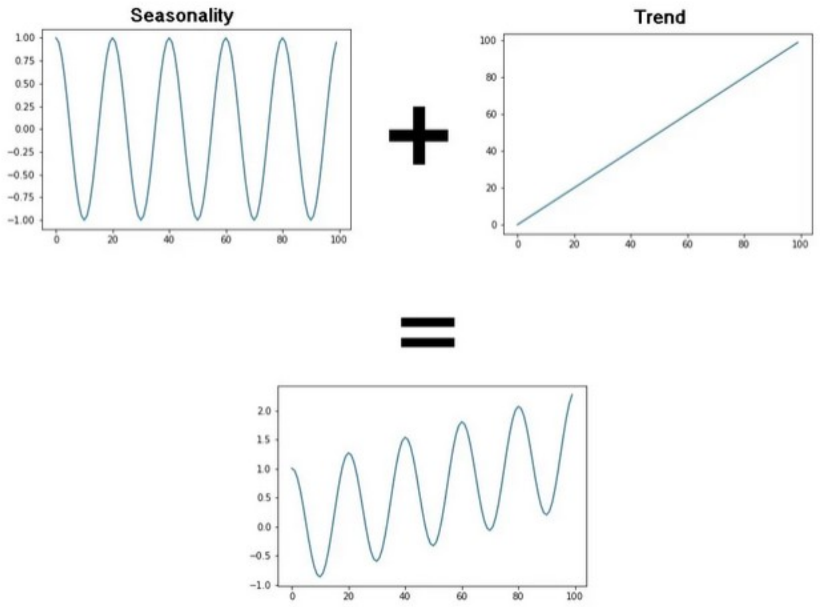
Quelle: Özen A., 2021: Seasonality Analysis and Forecast in Time Series

Additive Komponentenmodelle

Das additive Modell für eine Zeitreihe y(t) kann folgendermaßen dargestellt werden:

y(t) = Trendkomponente(t) + saisonale Komponente(t) + Restkomponente(t)

Additive Modelle eignen sich insbesondere dann gut, wenn die *saisonalen Schwankungen* in der Zeitreihe **nicht proportional zum Trend** sind und eher als additive Effekte auftreten.



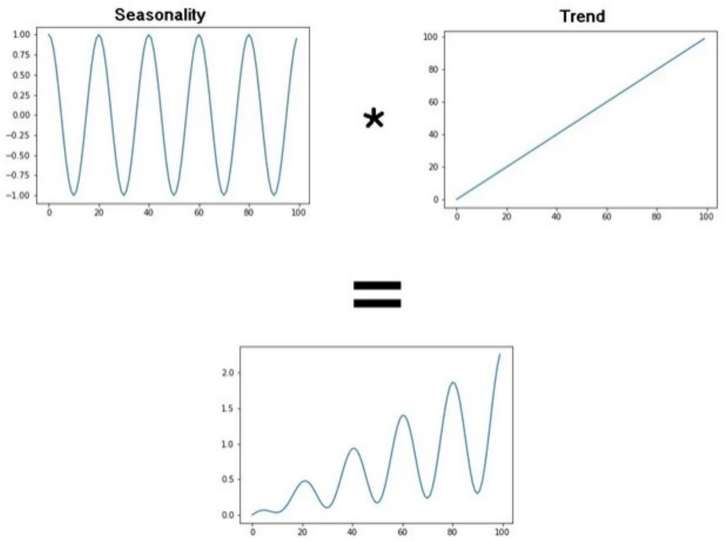
Quelle: Hayes S., 2021: Finding Seasonal Trends in Time-Series

Multiplikative Komponentenmodelle

Das multiplikative Modell für eine Zeitreihe y(t) kann folgendermaßen dargestellt werden:

y(t) = Trendkomponente(t) * saisonale Komponente(t) * Restkomponente(t)

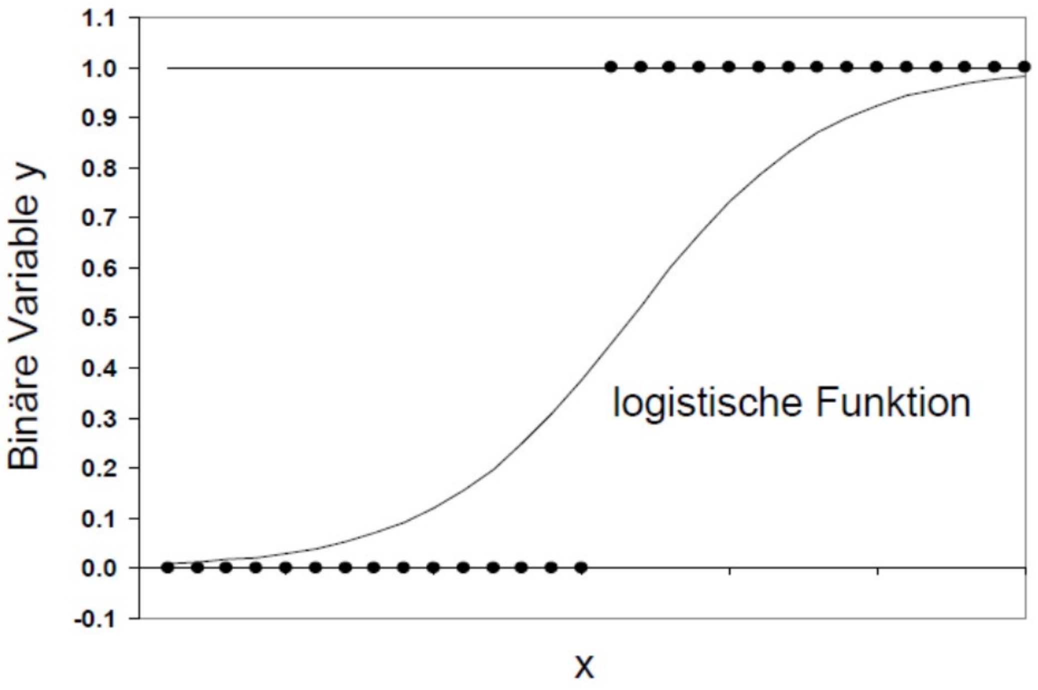
Multiplikative Modelle eignen sich besonders gut, wenn die **saisonalen Effekte relativ** zur **Trendgröße** variieren, beispielsweise wenn die saisonalen Schwankungen mit zunehmendem Trend stärker werden.



Quelle: Hayes S., 2021: Finding Seasonal Trends in Time-Series

Logistische Regression

Die (binär) logistische Regressionsanalyse testet, ob ein Zusammenhang zwischen mehreren unabhängigen und einer binären abhängigen Variable besteht.



Logistische Regression - Modelgüte

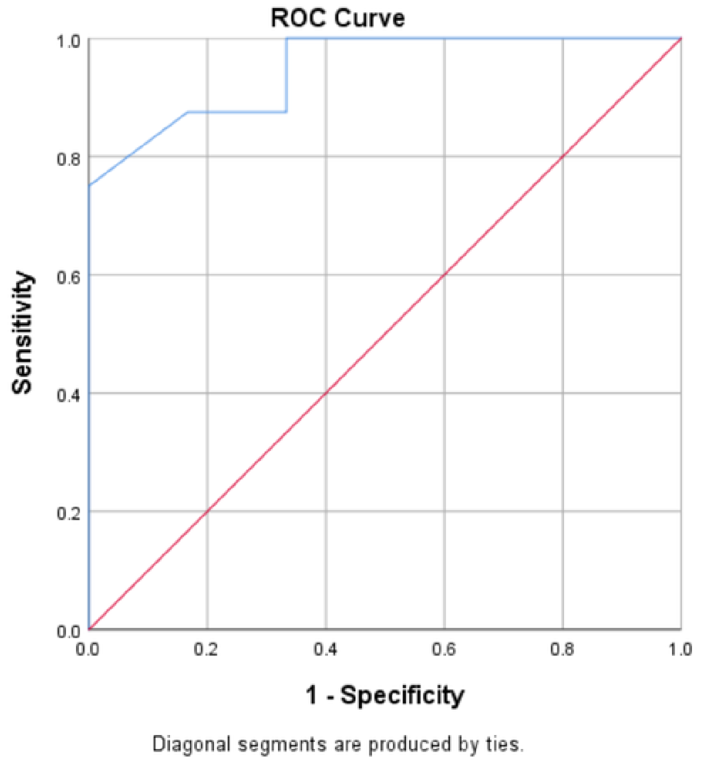
Um zu beurteilen, wie gut ein logistisches Regressionsmodell zu einem Datensatz passt, können wir die folgenden zwei Metriken betrachten:

- **Sensitivität**: Die Wahrscheinlichkeit, dass das Modell ein positives Ergebnis für eine Beobachtung vorhersagt, wenn das Ergebnis tatsächlich positiv ist.
- **Spezifität**: Die Wahrscheinlichkeit, dass das Modell ein negatives Ergebnis für eine Beobachtung vorhersagt, wenn das Ergebnis tatsächlich negativ ist.

Eine einfache Möglichkeit, diese beiden Metriken zu visualisieren, besteht darin, eine **ROC-Kurve** zu erstellen.

Logistische Regression – ROC-Kurve

ROC-Kurve ist ein Diagramm, das die Sensitivität und Spezifität eines logistischen Regressionsmodells anzeigt.



DHBW Lörrach

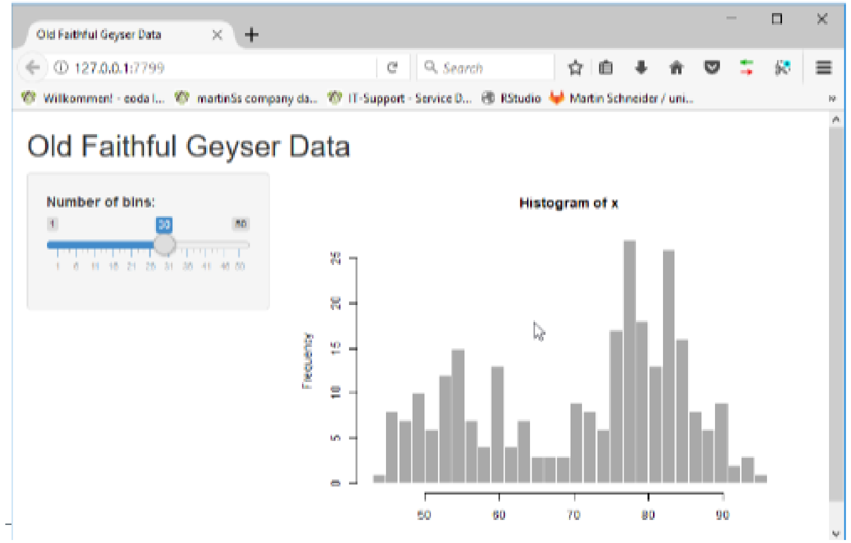
Shiny

Was ist Shiny?

Shiny bietet ein Interface, um aus R heraus Webapplikationen zu erstellen. Kenntnisse in HTML, CSS oder Javascript werden dabei in der Regel nicht benötigt.

Was kann Shiny?

Prinzipiell erstmal alles was R auch kann – also ziemlich viel!



Shiny

Eine Shiny-App benötigt mindestens zwei Elemente:

ui:

Layout Definition

server:

Berechnung der Ergebnisse. Dabei wird **reaktiv** auf die im User Interface vorgenommenen Einstellungen eingegangen.

```r
library(shiny)

ui <- fluidPage(
titlePanel("Old Faithful Geyser Data"),

number of bins
   sidebarLayout(
      sidebarPanel(
         sliderInput("bins",
                     "Number of bins:",
                     min = 1,
                     max = 50,
                     value = 30)
      ),
mainPanel(
         plotOutput("distPlot")
      )
   )
)

server <- function(input, output) {

   output$distPlot <- renderPlot({
      x    <- faithful[, 2]
      bins <- seq(min(x), max(x),
                  length.out = input$bins
+ 1)
hist(x, breaks = bins, col = 'darkgray',
border = 'white')
   })
}
# Run the application
shinyApp(ui = ui, server = server)
```

Shiny

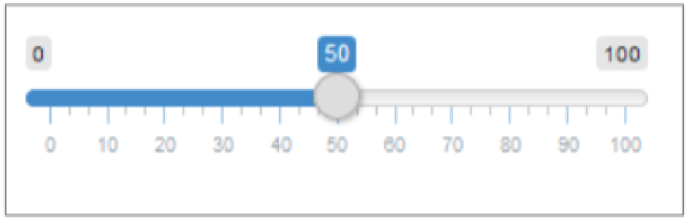
Input-Elemente lassen sich durch eine Vielzahl an **input**-Funktionen innerhalb des **ui**-Teils definieren.

Beispiel Code:

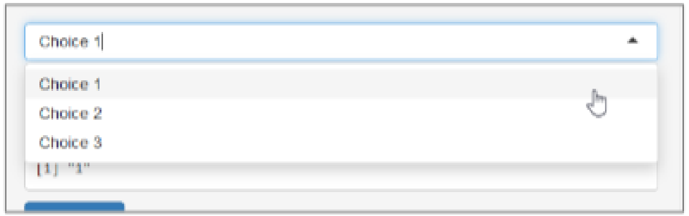
```
checkboxGroupInput("checkGroup",
  label = h3("Checkbox group"),
  choices = list("Choice 1" = 1,
                 "Choice 2" = 2,
                 "Choice 3" = 3),
  selected = 1)
```

Beispiel Input-Funktionen:

- Slider



- Select



- Text

- Checkbox

- ...

Shiny

```r
library(shiny)

ui <- fluidPage(
titlePanel("Old Faithful Geyser Data"),

number of bins
    sidebarLayout(
        sidebarPanel(
            sliderInput("bins",
                        "Number of bins:",
                        min = 1,
                        max = 50,
                        value = 30)
        ),
mainPanel(
            plotOutput("distPlot")
        )
    )
)

server <- function(input, output) {

    output$distPlot <- renderPlot({
        x    <- faithful[, 2]
        bins <- seq(min(x), max(x),
                    length.out = input$bins
+ 1)
hist(x, breaks = bins, col = 'darkgray',
border = 'white')
    })
}
# Run the application
shinyApp(ui = ui, server = server)
```

Einer ...`Output()`-Funktion im **ui** steht immer eine `render...()`-Funktion im **server** gegenüber.

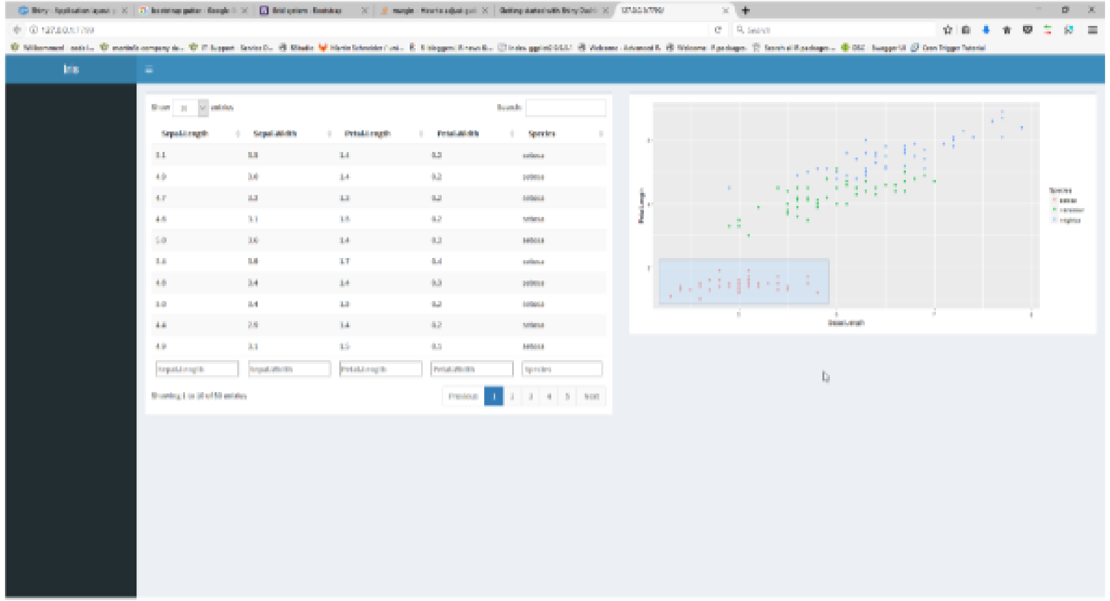
Output	Render-Funktionen
textOutput	renderText()
verbatimTextOutput()	renderPrint()
plotOutput()	renderPlot()
dataTableOutput()	renderDataTable()
leafletOutput()	renderLeaflet()
...	...

Shiny

Eine weitere Layout- und viele weitere Gestaltungsmöglichkeiten liefert das Paket `shinydashboard`.

Das Paket bietet u.a. als Gestaltungsmöglichkeit unterschiedliche Boxen an:

- tabBox
- infoBox
- valueBox



Vielen Dank!