

Business Intelligence

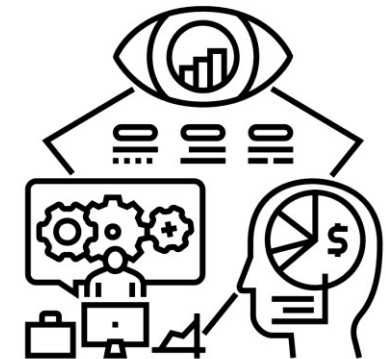
Vert. Prof. Dr. Aikaterini Nakou

Was ist Business Intelligence?

Definition: Business Intelligence (BI) bezeichnet Prozesse, Technologien und Werkzeuge zur Sammlung, Speicherung, Analyse und Bereitstellung von Daten, um bessere geschäftliche Entscheidungen zu treffen.

- **Datengetriebene Entscheidungsfindung:** BI ermöglicht es Unternehmen, auf Grundlage gesammelter Daten fundierte Entscheidungen zu treffen.
- **Synonyme Begriffe:** Datenanalyse, Entscheidungsunterstützungssysteme, Data-Driven Management.

BUSINESS INTELLIGENCE

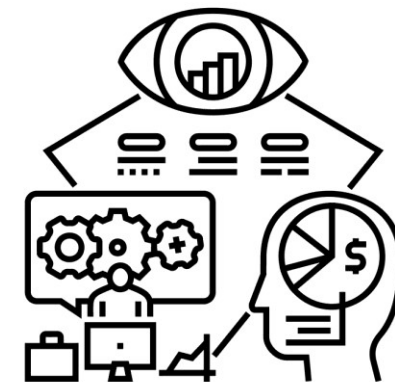


Was ist Business Intelligence?

Abgrenzung zu verwandten Begriffen:

- **Data Warehouse (DWH):** Eine zentrale Datenbank, die Daten aus verschiedenen Quellen integriert und speichert.
- **Data Mining:** Techniken zur Identifikation von Mustern und Zusammenhängen in großen Datensätzen.
- **Big Data:** Verarbeitung und Analyse sehr großer und vielfältiger Datenmengen, oft unstrukturiert.

BUSINESS INTELLIGENCE



Ziele und Nutzen von Business Intelligence

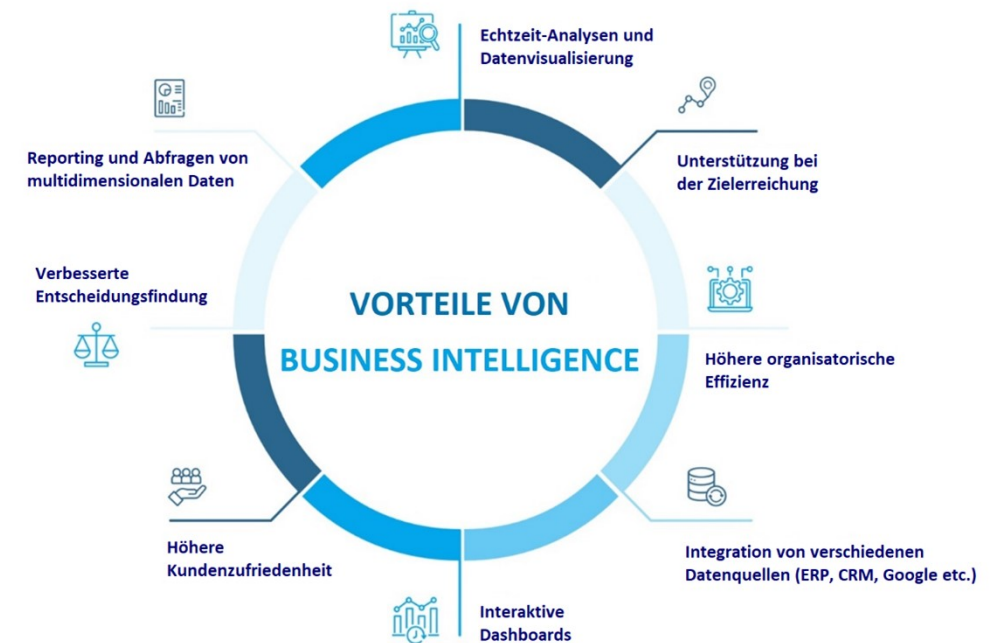
Ziele von BI:

- **Bessere Entscheidungsfindung:** Datenbasierte Einsichten führen zu fundierten strategischen und operativen Entscheidungen.
 - **Optimierung von Geschäftsprozessen:** Identifikation von Ineffizienzen und Verbesserungspotenzialen durch detaillierte Datenanalysen.
 - **Steigerung der Wettbewerbsfähigkeit:** Schnelles Reagieren auf Marktveränderungen durch Echtzeit-Datenanalysen.
 - **Kundenzufriedenheit erhöhen:** Verstehen von Kundenbedürfnissen durch personalisierte Datenanalyse.
-

Ziele und Nutzen von Business Intelligence

Nutzen von BI:

- **Transparenz:** Klare Einblicke in Geschäftsprozesse, Leistungskennzahlen (KPIs) und Trends.
- **Risikomanagement:** Frühzeitige Erkennung von Risiken und Markttrends.
- **Effizienz:** Automatisierung von Berichten und Analysen spart Zeit und Kosten.
- **Strategische Ausrichtung:** Ausrichtung von Unternehmen auf langfristige, datenbasierte Ziele.



Komponenten eines BI-Systems

Ein BI-System besteht aus mehreren Schichten und Komponenten:

1. **Datenquellen:** Interne und externe Quellen (ERP-Systeme, CRM, Social Media, Marktberichte etc.).
2. **Datenintegration:**
 - **ETL-Prozess (Extract, Transform, Load):** Extraktion der Daten aus verschiedenen Quellen, Transformation in ein einheitliches Format und Laden ins Data Warehouse.
3. **Data Warehouse (DWH):** Zentraler Speicher für alle integrierten Daten, der die Grundlage für BI-Analysen bildet.



Komponenten eines BI-Systems

4. Datenanalyse und Reporting:

- **OLAP (Online Analytical Processing):** Mehrdimensionale Analyse von Daten, häufig zur Untersuchung von KPIs.
- **Data Mining:** Entdeckung von Mustern und Zusammenhängen.
- **Berichte und Dashboards:** Visualisierung der Ergebnisse.

5. Präsentationsschicht:

- Benutzeroberflächen für den Zugriff auf Berichte und Dashboards, meist angepasst an die verschiedenen Zielgruppen (Management, Fachabteilungen).



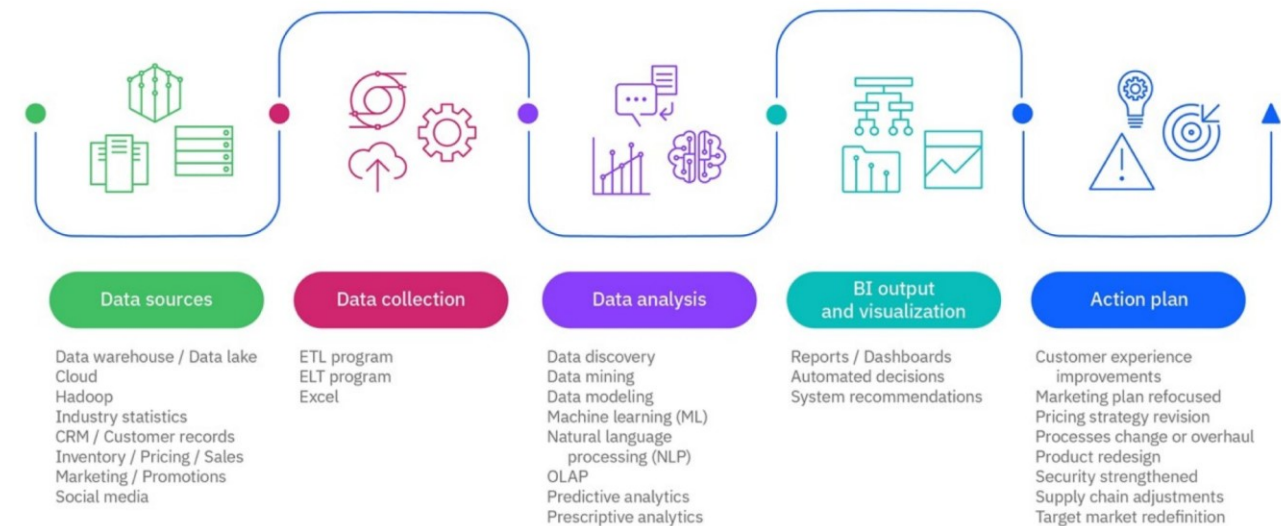
BI-Prozesse im Detail

1. Datenextraktion:

- Extrahieren von Daten aus verschiedenen operativen Systemen (ERP, CRM, Buchhaltung, Logistik usw.).
- Herausforderung: Datenqualität und -konsistenz sicherstellen.

2. Datenbereinigung und -transformation:

- Bereinigen von Datenfehlern, Duplikaten und Inkonsistenzen.
- Transformation der Daten in ein einheitliches Format, damit sie analysiert werden können.



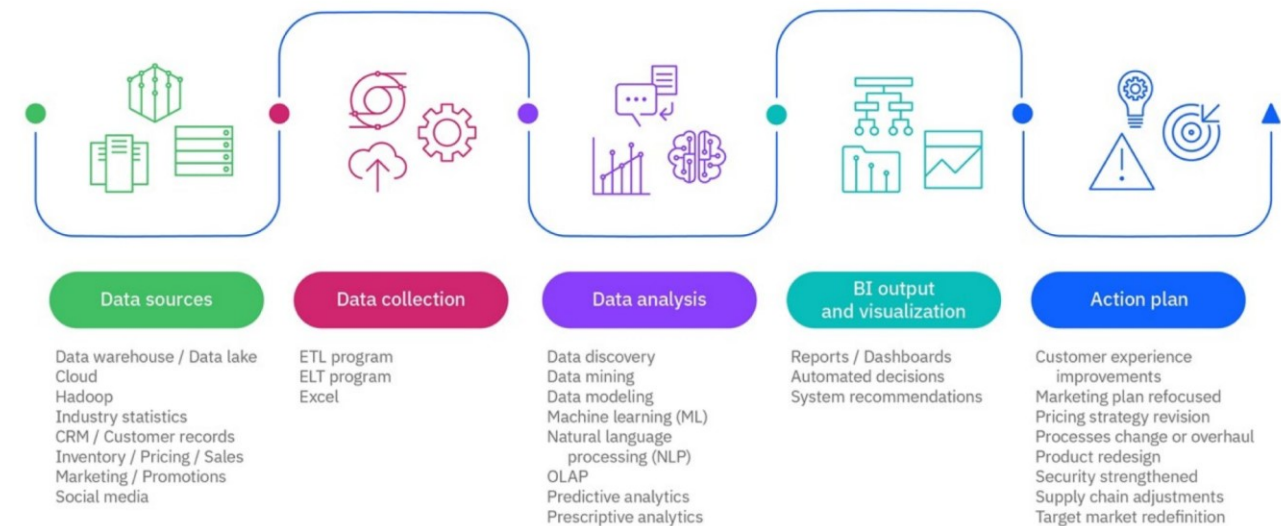
BI-Prozesse im Detail

3. Datenintegration:

- Zusammenführen von Daten aus verschiedenen Quellen und Formaten.
- Hier kommt oft das Data Warehouse ins Spiel, um eine zentrale Quelle für alle analysierten Daten zu bieten.

4. Datenanalyse:

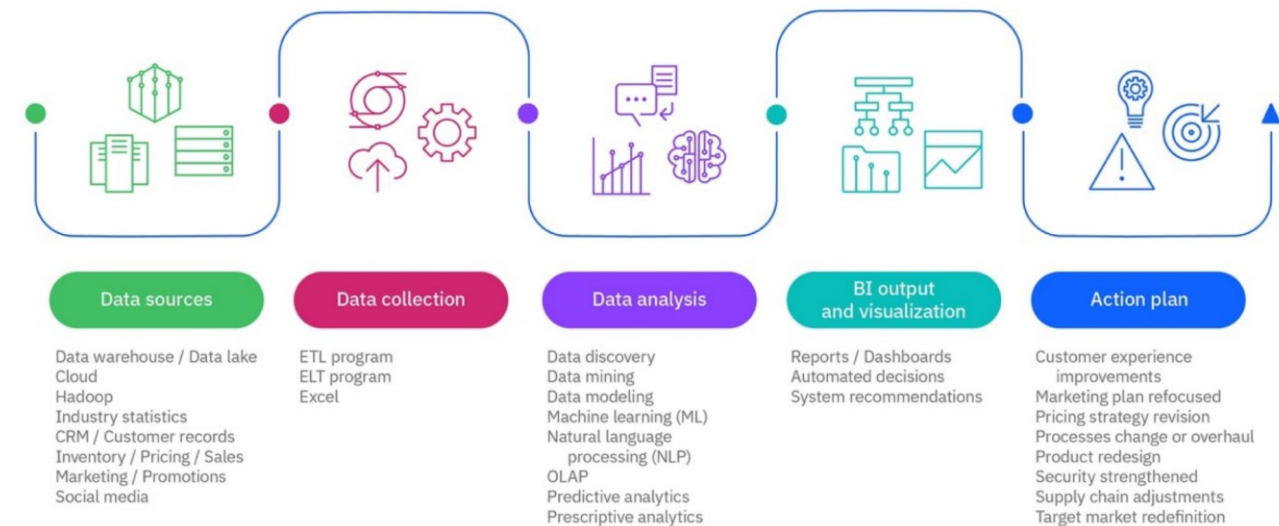
- Methoden wie OLAP, Data Mining, und statistische Analysen werden verwendet, um Einsichten zu gewinnen.



BI-Prozesse im Detail

5. Berichtserstellung und Dashboards:

- Visualisierung der Analyseergebnisse in Form von Berichten, Diagrammen und interaktiven Dashboards.
- Wichtig: Anpassung der Berichte an die Bedürfnisse der Zielgruppe (z.B. Managementberichte vs. operative Berichte).



6. Entscheidungsfindung:

- Action Plan

Tools in der BI

BI-Tools:

- **Power BI** (Microsoft): Visualisierung und Self-Service-BI.
 - **Tableau**: Starke Visualisierungsfähigkeiten und einfache Bedienung.
 - **Qlik Sense**: Data Discovery und interaktive Dashboards.
 - **SAS**: Starke analytische Fähigkeiten für komplexe Datenanalyse.
 - **SAP BusinessObjects**: Unternehmensweite BI-Lösungen mit integriertem Reporting
-

Anwendungsbeispiele aus der Praxis

1. Einzelhandel:

- **Lagerbestandsoptimierung:** Nutzung von BI, um Lagerbestände zu minimieren und gleichzeitig die Lieferfähigkeit zu verbessern.
- **Kundenanalyse:** Analyse von Kaufmustern und Personalisierung von Angeboten.

2. Gesundheitswesen:

- **Patientenanalyse:** Verwendung von BI zur Analyse von Patientendaten, um Behandlungspläne zu optimieren und Kosten zu senken.
- **Kostenmanagement:** Identifizierung von Kostentreibern und Einsparpotenzialen.

3. Finanzwesen:

- **Risikomanagement:** Vorhersage von Kreditrisiken durch prädiktive Analysen.
 - **Betrugsbekämpfung:** Data-Mining-Techniken zur Identifizierung von betrügerischem Verhalten in Transaktionsdaten.
-

Herausforderungen und Trends in Business Intelligence

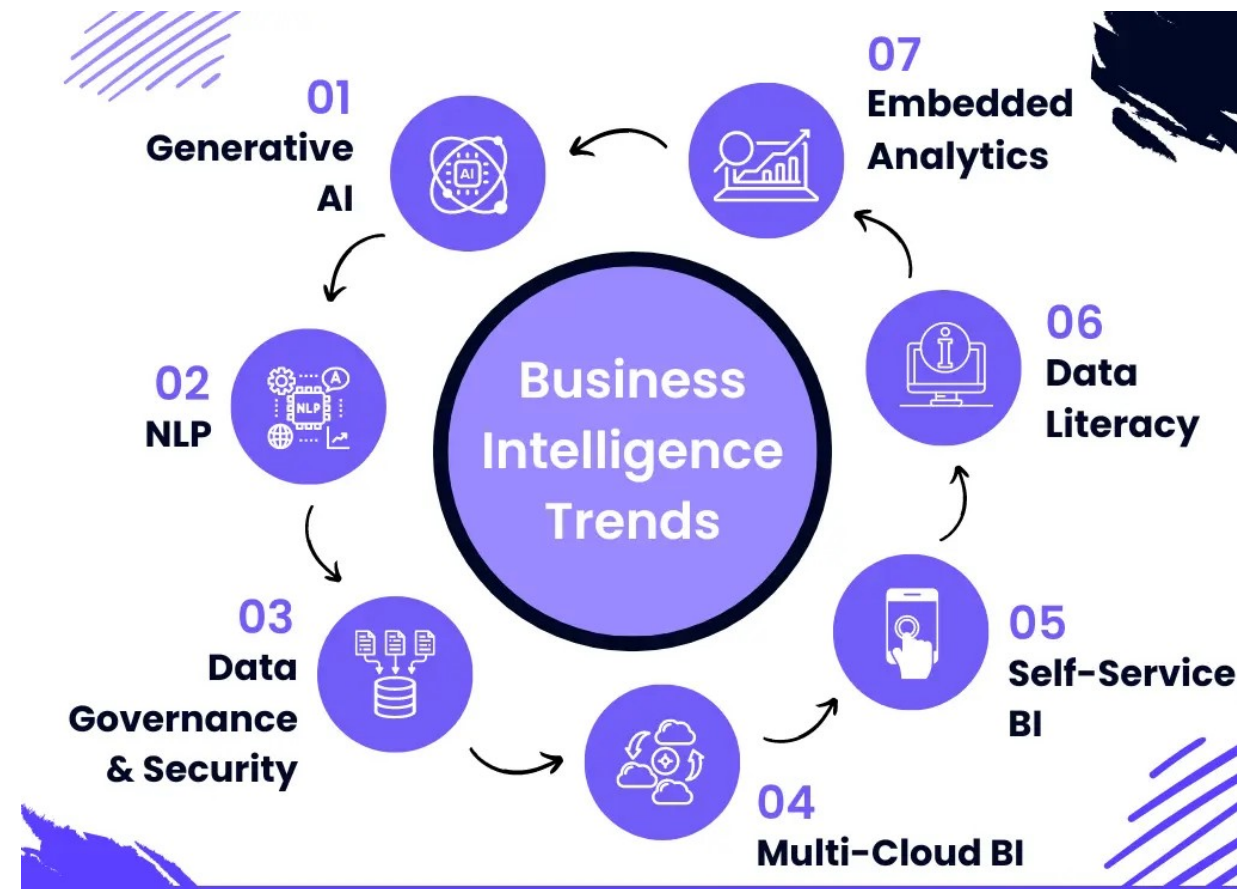
Herausforderungen:

- **Datenqualität:** Schlechte Datenqualität kann zu falschen Analysen und Entscheidungen führen.
- **Datenschutz und Sicherheit:** Datenschutzvorschriften (z.B. DSGVO) müssen bei der Nutzung von BI beachtet werden.
- **Komplexität:** Integration und Verwaltung großer, unstrukturierter Datenmengen kann anspruchsvoll sein.

Herausforderungen und Trends in Business Intelligence

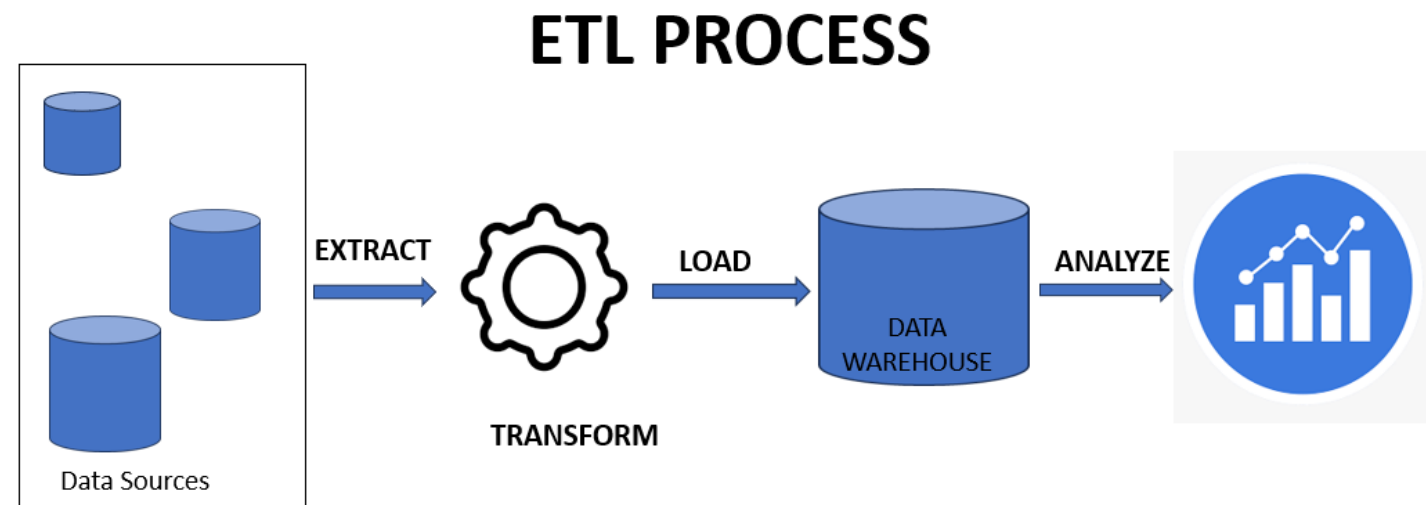
Trends:

- **Self-Service-BI:** Benutzer außerhalb der IT-Abteilungen haben direkten Zugriff auf BI-Tools und können eigenständig Analysen durchführen.
- **KI und Machine Learning in BI:** Automatisierte Mustererkennung und prädiktive Analysen werden durch den Einsatz von KI und Machine Learning weiter vorangetrieben.
- **Cloud-BI:** Zunehmender Einsatz von Cloud-basierten BI-Lösungen, die skalierbar und kosteneffizient sind.



ETL-Prozess

ETL steht für **Extract, Transform, Load** und ist ein wesentlicher Bestandteil jeder Datenverarbeitungs- und Analysepipeline. Besonders in der Business Intelligence (BI) ermöglicht der ETL-Prozess die Umwandlung von Rohdaten in nützliche Informationen, die Unternehmen helfen, fundierte Entscheidungen zu treffen.

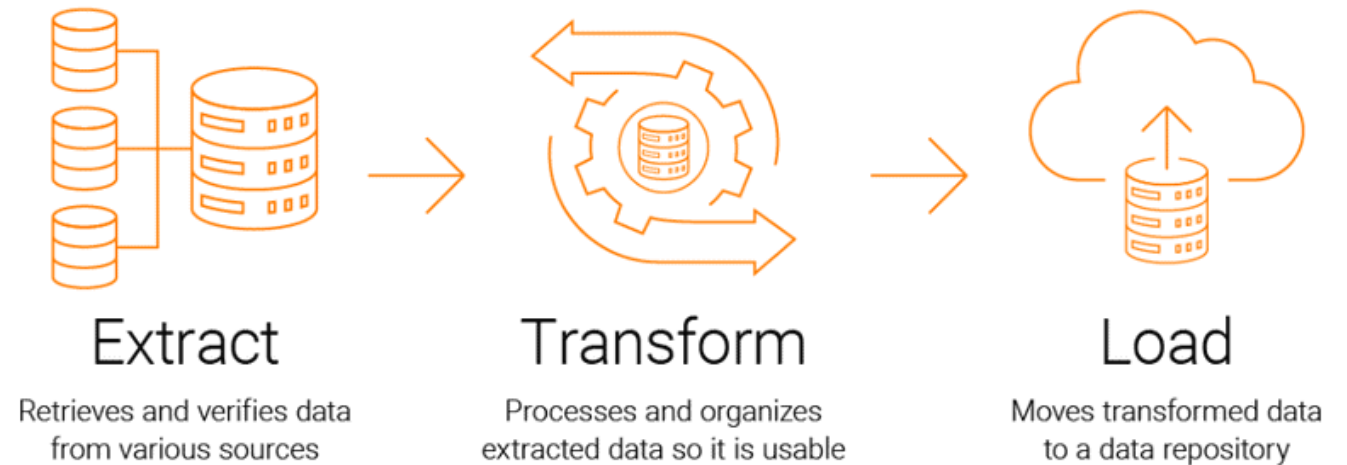


ETL-Prozess

Extract: Daten aus unterschiedlichen Quellsystemen wie Datenbanken, Cloud-Diensten, APIs, Excel-Dateien usw. extrahieren.

Transform: Daten bereinigen, anreichern, normalisieren, filtern, aggregieren und formatieren, sodass sie für analytische Zwecke geeignet sind.

Load: Die transformierten Daten in ein Zielsystem laden, üblicherweise in ein Data Warehouse oder eine Datenbank.



ETL-Prozess im Detail

Extract (Datenextraktion)

- **Ziel:** Die Extraktion von Daten aus verschiedenen Quellen.
- **Herausforderungen:**
 - Unterschiedliche Datenformate (CSV, JSON, XML, Datenbanken)
 - Unterschiedliche Struktur und Schema (relationale vs. nicht-relationale Daten)
 - Verteilte Quellen (lokal, cloudbasiert)
- **Methoden der Datenextraktion:**
 - **Vollständige Extraktion:** Alle Daten werden aus einer Quelle extrahiert. Vorteilhaft bei kleineren Datenmengen.
 - **Inkrementelle Extraktion:** Nur neue oder geänderte Daten werden extrahiert. Diese Methode reduziert die Belastung von Netzwerken und Quellsystemen.

ETL-Prozess im Detail

Transform (Datenumwandlung)

- **Ziel:** Bereinigung und Anreicherung der extrahierten Daten für die Analyse.
- **Schritte der Transformation:**
 - **Datenbereinigung:** Duplikate, Inkonsistenzen oder fehlerhafte Daten werden entfernt.
 - **Datenanreicherung:** Zusätzliche Informationen werden hinzugefügt, z.B. durch Berechnungen oder externe Quellen.
 - **Datenaggregation:** Zusammenfassen von Daten auf ein höheres Abstraktionslevel (z.B. Summen, Mittelwerte).
 - **Datenformatierung:** Daten werden in ein einheitliches Format gebracht, um Kompatibilität mit dem Zielsystem sicherzustellen.
 - **Normierung und Denormalisierung:** Strukturelle Änderungen, um die Daten effizient für Abfragen im Data Warehouse aufzubereiten.

ETL-Prozess im Detail

Load (Datenladen)

- **Ziel:** Die transformierten Daten werden in das Zielsystem geladen.
- **Arten des Ladens:**
 - **Vollständige Ladung:** Alle Daten werden in das Zielsystem geladen, was für kleinere Systeme geeignet ist, aber bei großen Datenmengen ineffizient sein kann.
 - **Inkrementelles Laden:** Nur geänderte oder neue Daten werden geladen. Diese Methode ist effizienter und belastet das System weniger.
- **Ladeverfahren:**
 - **Batch Processing:** Große Datenmengen werden in regelmäßigen Abständen in das Zielsystem geladen (z.B. nächtliche Batch-Ladungen).
 - **Real-time Processing:** Daten werden in Echtzeit oder nahezu in Echtzeit in das Zielsystem geladen (besonders bei modernen BI-Anwendungen wichtig, die schnelle Analysen erfordern).

Herausforderungen im ETL-Prozess

- **Datenqualität:** Unterschiedliche Datenquellen können zu Inkonsistenzen führen.
- **Performance:** Besonders bei großen Datenmengen muss der ETL-Prozess optimiert werden, um Ladezeiten zu minimieren und das Zielsystem nicht zu überlasten.
- **Skalierbarkeit:** Mit wachsendem Datenvolumen und neuen Datenquellen muss der ETL-Prozess skalierbar bleiben.
- **Datenformatwechsel:** Quellsysteme ändern sich häufig, was Anpassungen am ETL-Prozess erfordert.

Best Practices für einen erfolgreichen ETL-Prozess

- **Automatisierung:** Automatisierung von ETL-Prozessen, um menschliche Fehler zu minimieren und die Effizienz zu erhöhen.
- **Datenüberwachung:** Einsetzung von Monitoring-Tools, um Datenqualitätsprobleme frühzeitig zu erkennen und zu beheben.
- **Dokumentation:** Sicherstellung, dass alle Schritte des ETL-Prozesses dokumentiert sind, um spätere Anpassungen zu erleichtern.
- **Skalierbare Infrastruktur:** Verwendung von skalierbaren Technologien und Cloud-basierten Lösungen, um mit wachsendem Datenvolumen umzugehen.

Datenanalyse

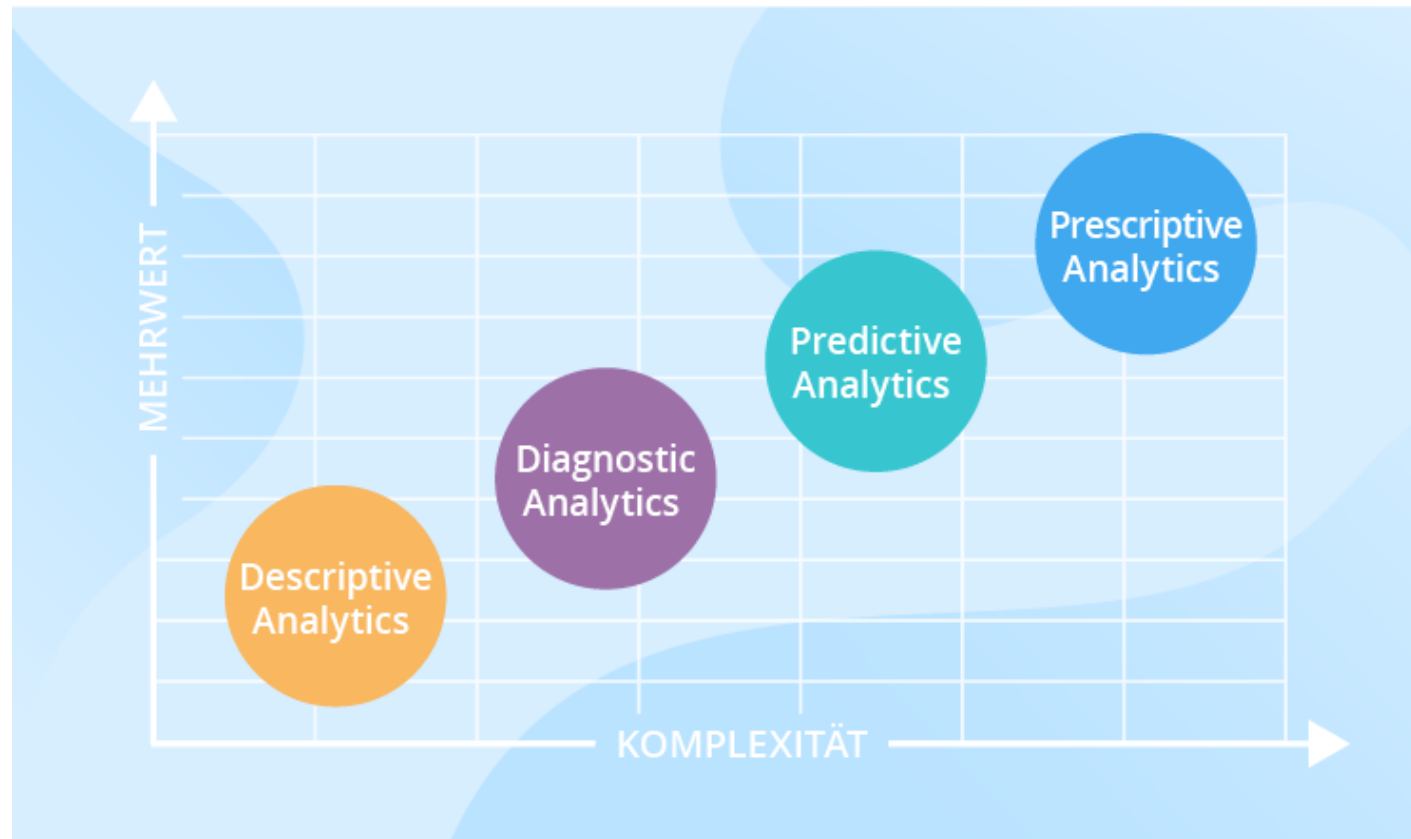
Die Datenanalyse oder Datenauswertung beschreibt den Prozess der Gewinnung von wertvollen Informationen aus Rohdaten.

- Verwendung von statistischen Methoden für die Analyse von bereits bestehenden Daten
 - Ergebnisse der Analyse werden in Form von Zahlen, Fakten, Metriken oder Datenvisualisierungen (z.B. Diagramme, Tabellen) veranschaulicht und zur Weiterverwendung bereitgestellt
 - Basis für die Entscheidungsfindung bei Unternehmen
 - Planung von zukünftigen Unternehmensschritten
 - Bewertung von bereits getätigten Entscheidungen
 - Minimierung von Risiken
-

Prozessschritte der Datenanalyse

- **Einlesen:** Die Daten in das richtige Format (z.B. CSV) bringen, um sie analysieren zu können.
 - **Explorieren:** Die Daten reinigen, prüfen und kennenlernen, fehlende Daten ersetzen.
 - **Modellieren:** Die Formel (Funktion) finden, die die Daten möglichst gut erklärt. Meist müssen mehrere Modelle verglichen werden.
 - **Validieren:** Herausfinden, ob das gewählte Modell auch neue Daten erklären kann (Validierungsstichprobe).
 - **Bewerten:** Bestimmen, ob die Ergebnisse der Datenanalyse relevant sind.
 - **Zusammenfassen:** Die Methode und die Ergebnisse so zusammenfassen, dass sie verwertbar und reproduzierbar sind.
-

Methoden der Datenanalyse



Deskriptive Analyse

Die deskriptive oder beschreibende Datenanalyse konzentriert sich auf Daten aus der Vergangenheit, um daran zu erkennen, **was passiert ist**.

Einfachste Form der Datenanalyse, weil sie Trends und Beziehungen beschreibt bzw. visualisiert, aber nicht tiefer geht.

Beispiele:

- Kundenverhalten
 - Umsatz im letzten Quartal
 - wie viele Artikel in einem Geschäft zurückgegeben wurden
 - etc...
-

Diagnostische Analyse

Bei der diagnostischen Datenanalyse werden die Daten miteinander verglichen, um Muster zu erkennen, Zusammenhänge aufzudecken und Ursachen und gegenseitige Wechselwirkungen zu identifizieren. Es geht dabei darum, herauszufinden, **warum etwas passiert ist**.

Logisches Schritt nach der deskriptiven Analyse.

Beispiele:

- Untersuchung der Marktnachfrage
 - Erklärung des Kundenverhaltens
 - Verbesserung der Unternehmenskultur
 - etc...
-

Prädiktive Analyse

Die prädiktive oder vorhersagende Analyse hilft dabei, einen Blick in die Zukunft zu werfen. Anhand dieser Analysemethode soll herausgefunden werden, **was in der Zukunft passiert bzw. passieren könnte**.

Um Vorhersagen treffen zu können, werden die Ergebnisse bereits durchgeführter deskriptiver oder diagnostischer Analysen verwendet.

Beispiele:

- Vorhersage der zukünftigen Umsätzen
 - Bestimmung des Personalbedarfs
 - Risikoanalyse
 - etc...
-

Präskriptive Analyse

Die präskriptive oder verordnende Datenanalyse ist die komplexeste und aufwendigste Analysemethode. Gleichzeitig ist sie für Unternehmen von unglaublich großem Wert, da anhand ihrer Ergebnisse entschieden werden soll, **welche Maßnahmen zu treffen sind**, um etwa ein künftiges Problem zu verhindern, zu beseitigen oder kommende Trends zu nutzen.

Beispiele:

- Marketing Maßnahmen
- Entscheidungen von Führungskräften
- etc....

Was ist R und warum R für Business Intelligence?

Was ist R?

- R ist eine Programmiersprache und Umgebung für statistische Berechnungen und Grafiken.
- Open Source: Kostenlos und ständig weiterentwickelt von einer großen Community.
- Leistungsfähig für Datenmanipulation, statistische Analyse, Visualisierung und maschinelles Lernen.

Was ist R und warum R für Business Intelligence?

Warum R für BI?

- **Vielfalt an Paketen:** Es gibt eine Vielzahl von R-Paketen (z. B. dplyr, plotly, shiny), die speziell für Datenanalyse und Visualisierung entwickelt wurden.
- **Einfache Datenintegration:** R kann mit verschiedenen Datenquellen wie SQL-Datenbanken, Excel, und CSV-Dateien arbeiten.
- **Visualisierung:** R bietet umfangreiche Tools für die Erstellung von Business-Intelligence-Dashboards und interaktiven Berichten.
- **Automatisierung:** BI-Prozesse wie Datenbereinigung und -transformation können in R automatisiert werden.

Installation und Einrichtung von R

Schritte zur Installation:

1. **R installieren:** Download der neusten Version von der offiziellen R-Website (<https://www.r-project.org>).
2. **RStudio installieren:** RStudio ist eine benutzerfreundliche IDE (Integrated Development Environment) für R. Download von <https://www.rstudio.com>.

RStudio Übersicht:

- **Script Editor:** Hier wird der Code geschrieben und ausgeführt.
 - **Konsole:** Direkte Ausführung von R-Befehlen.
 - **Environment:** Zeigt alle Variablen und Daten im aktuellen Arbeitsbereich.
 - **Plots, Files, Packages, Help:** Verwaltung von Grafiken, Dateien und Paketen.
-

Die RStudio-Umgebung

Pane Layout: Die RStudio-Umgebung ist in verschiedene Bereiche unterteilt.

- **Source Pane** (oben links): Zum Schreiben von R-Skripten.
- **Console Pane** (unten links): Direkte Interaktion mit R.
- **Environment Pane** (oben rechts): Überblick über alle geladenen Daten und Variablen.
- **Files/Plots/Packages/Help Pane** (unten rechts): Verwaltung von Dateien, Visualisierungen, Paketen und Dokumentation.

R Scripts und Workflows:

- Erstellen eines neuen Skripts: **File > New File > R Script**
 - Kommentare hinzufügen: Durch das Setzen von **#** vor einem Kommentar.
 - Code-Abschnitte ausführen: Über den Button „Run“ oder durch die Tastenkombination **Strg + Enter**.
-

Grundlegende R-Konzepte und Syntax

Datentypen und Strukturen

- **Vektoren:** Die einfachste Datenstruktur in R. Erstellen eines Vektors:

```
5 #Vektoren
6
7 zahlen <- c(1, 2, 3, 4, 5)
8 namen <- c("Anna", "Ben", "Chris")
9
```

- **Matrizen:** Zweidimensionale Anordnung von Werten (nur numerische Daten).

```
10 #Matrizen
11
12 matrix_data <- matrix(1:6, nrow=2, ncol=3)
13
14
```

- **Datenrahmen (Data Frames):** Äquivalent zu Tabellen in SQL oder Excel.

```
15 # Data Frames
16
17 daten <- data.frame(Name=c("Anna", "Ben"), Alter=c(25, 30), Beruf=c("Informatiker", "Analyst"))
18
19
```

Grundlegende R-Konzepte und Syntax

Kontrollstrukturen

- **If-Else Bedingung:**

```
19 # If-Else Bedingung:
20
21 if (Alter > 30) {
22   print("über 30")
23 } else {
24   print("unter 30")
25 }
```

- **For-Schleifen:**

```
29 # For-Schleifen:
30
31 for (i in 1:5) {
32   print(i)
33 }
```

- **Funktionen:**

```
36 # Funktionen
37
38 addiere <- function(a, b) {
39   return(a + b)
40 }
41 ergebnis <- addiere(5, 10)
42
```

- **Pakete installieren und laden:**

```
44 # Pakete installieren
45
46 install.packages("dplyr")
47
48
49 #lade notwendige Paketen
50 library(dplyr)
51
```

Datenimport und -manipulation

CSV-Dateien einlesen:

```
54 #lade die Daten
55
56 supermarket_sales <- read.csv2("supermarket_sales.csv", header = TRUE, sep= ",", dec=".")
57
```

Datenübersicht:

```
59 head(supermarket_sales) # Zeigt die ersten 6 zeilen
60 str(supermarket_sales)  # Zeigt die Struktur des Datenrahmens
61 summary(supermarket_sales) # Zusammenfassung der Daten
--
```


Vielen Dank!

