ANGEWANDTE PROGRAMMIERUNG

Einführung und Data Analysis Lifecycle Modelle

Vorlesung 01

Dennis Glüsenkamp

3. März 2023

- Vorstellung & Einführung in das Modul
- Prüfungsleistungen
- Datenquellen, Tools und Programmiersprache
  - git
  - GitHub
  - Kaggle
  - Python
  - Anaconda
- Data Analysis Lifecycle Modelle

Vorstellung & Einführung in das Modul

Persönliche Vorstellung

Berufliche Erfahrungen

- Aktuell seit Juli 2022 als Lead Data Strategist bei qdive GmbH
- Seit 2010 bei verschiedenen Unternehmen als Data Scientist und in weiteren Rollen tätig
- Seit Januar 2020 nebenberuflicher Lehrbeauftragter der FOM dem Schwerpunkt Data Science, Machine Learning, Business Analytics

Akademischer Hintergrund:

- Studium der Physik (Diplom) in Osnabrück und Bonn
- Berufbegleitendes Studium Business Intelligence Systems & Data Mining (MSc) in Leicester, UK

Kontaktdaten

Wenn Sie mich dringend/schnell erreichen möchten, rufen Sie mich am besten an. Bei E-Mails geben Sie mir bitte 7 Tage Antwortzeit. Sollte ich bis dahin nicht zurückgeschrieben haben, erinnern Sie mich bitte!

Telefon	+49 (0) 176 73900073
E-Mail	data@gluesenkamp.info

GitHub-Repo	https://github.com/dgluesen/ss23-applied-programming

Modulziele (1/2)

- Den für Big-Data-Analysen typischen **Anwendungszyklus beschreiben** und in der Praxis begleiten
- Im Anwendungszyklus häufig eingesetzte Systeme, **Programmiersprachen und Programmierumgebungen** benennen
- Relevante **Programmiermodelle beschreiben**

Modulziele (2/2)

- In einer typischen Systemumgebung mithilfe ausgewählter Programmierwerkzeuge strukturierte, semistrukturierte und unstrukturierte **Daten**
  - für die Analyse **aufbereiten**,
  - in Analysesysteme **integrieren**,
  - automatisch und manuell **analysieren**,
  - **visualisieren** sowie
  - **Ergebnisse** für weitere Verarbeitungen **bereitstellen**
- Die eingesetzten **Methoden und Werkzeuge** im Rahmen von umfangreichen Analyse- und Consultingprojekten effektiv und programmgesteuert **anwenden**

Planung der Inhalte

Nr.	Datum	Inhalte
01	03.03.2023	Einführung und Data Analysis Lifecycle Modelle
02	24.03.2023	SQL
03	25.03.2023	Einführung Python 1
04	12.04.2023	Einführung Python 2
05	22.04.2023	Wichtige Packages für Python
06	22.04.2023	Einführung Machine Learning
07	29.04.2023	Machine Learning 1 [ONLINE]
08	12.05.2023	Machine Learning 2, Wiederholung und Fragen zur Klausur
09	17.05.2023	Machine Learning Beispiele (mit Gastdozenten)
10	02.06.2023	Präsentationen der Jupyter Notebooks
11	17.06.2023	Klausur

Änderungen vorbehalten

Prüfungsleistungen

Prüfungsleistungen

- Für das Bestehen und die Benotung müssen **zwei Prüfungsleistungen** erbracht werden
- Leistungen müssen jeweils **unabhängig voneinander mindestens ausreichend** sein um das Modul zu bestehen
- **Teilleistungen** sind:
  - **Jupyter Notebook und Präsentation**, 25% der Gesamtnote, Termin ist 02.06.2023
  - **Klausur**, 75% der Gesamtnote, Termin ist 17.06.2023

Notebook-Präsentation (1/2)

- Vorstellung findet **im Rahmen einer Vorlesung** statt
- Zeit für Präsentation ohne Zwischenfragen ist **7 Minuten** - ein Überziehen der Zeit führt zu **Punktabzug**!
- **Fragen, Diskussion und Austausch** mit dem Kurs direkt im Anschluss ohne Zeitbegrenzung
- Juypter Notebook, Präsentation und Diskussion nach eigener Wahl in **deutsch oder englisch**
- **Thema und Datensatz** für Ausarbeitung soll **selbstständig gewählt** werden (z.B. von Kaggle)

Notebook-Präsentation (2/2)

Aufgabenstellung, Zielsetzung und Bewertungsmaßstäbe:

- Formulierung einer **zentralen Forschungsfrage** und ggf. untergeordneter bzw. angegliederter Nebenfragestellungen
- Auswahl eines **geeigneten Datensatzes** aus beliebiger Quelle
- **Erschließung, Exploration, Prädiktion** etc. der Daten und entsprechend der Fragestellung **im Jupyter Notebook**
- **Gestaltung des Notebooks** in für ein Fachpublikum geeigneter Weise, d.h. angemessene Kommentierung, grafische Gestaltung, Nutzung von Interaktivität etc.
- Fachkundige und verständliche **Präsentation bei Einhaltung des Zeitlimits**
- **Kompentente Beantwortung** der in der Diskussion aufgeworfenen Fragen

Datenquellen, Tools und Programmiersprache

- Versionsverwaltung von Dateien
- Ähnliche Werkzeuge: CVS, BitKeeper
- konsistente Fortentwicklung von Programmcode

GitHub

- Dateien mit Versionskontrolle online verwalten
- Ähnliche Werkzeuge: Bitbucket, GitLab
- Setzt auf Versionsverwaltungssoftware git auf
- Möglichkeit Projekte über eigenen Websites zu präsentieren
- Einfache, agile Tools inkludiert

Kaggle

kaggle

- Online-Community für Data Scientists und verwandte Berufsgruppen
- Datensätze und Beispielcodes sind frei verfügbar
- Foren für datenbezogene Diskussionen
- Anwendung und Vertiefung der eigenen Kenntnisse
- Modul nutzt vielfach die dort verfügbaren Daten

Python

- Höhere Programmiersprache, die in diesem Modul eingesetzt wird
- Ziel der Entwickler:innen war möglichst hoher Grad an Einfachheit sowie Übersichtlichkeit
- Standardsprache für viele Data Science und Machine Learning Entwicklungen/Anwendungen

Anaconda

- Python arbeitet sehr stark mit verschiedenen Paketen
- Pakete müssen installiert und eingebunden werden
- Paketkombinationen und verschiedene -versionen können Konflikte hervorrufen
- Anaconda als Distribution für Python adressiert dieses Problem
- Jupyter Notebooks für interaktive Entwicklung und explorative Datenanalyse sind inkludiert

Data Analysis Lifecycle Modelle

Verbindlichkeit durch Prozessmodell

- Prozessbeschreibung von Schritten, die bei der Durchführung von datengetriebenen Aktivitäten erforderlich sind, erzeugt
  - Vollständigkeit, da Schritte systematisch abgearbeitet werden
  - Iterationsfähigkeit, da in bestimmten Zyklen Fortentwicklung und Ergebnisbereitstellung erfolgt
  - Nachvollziehbarkeit, da eine logische und sinnvolle Struktur abgearbeitet wird
  - Transparenz, da Rollen und Zuständigkeiten geklärt sind
  - Sicherheit, da Schutzmechnismen integriert werden können
- Vermeidung von Fehlern oder nicht notwendiger Ineffizienz durch mangelnde Organisation
- Steigender Komplexitätsgrad bei Daten-Projekten erfordert höheres Maß an Struktur um Sackgassen oder Chaos zu verhindern

CRISP-DM Life Cycle

- **CR**oss-**I**ndustry **S**tandard **P**rocess for Data Mining [1]
- 1996 im Rahmen von EU-Förderprojekt entwickelt
- Offenes, freies Prozessmodell zur Durchführung von Data Mining Vorhaben
- Prozess kann flexibel und unabhängig von Branche, Toolset und Anwendung verwendet werden

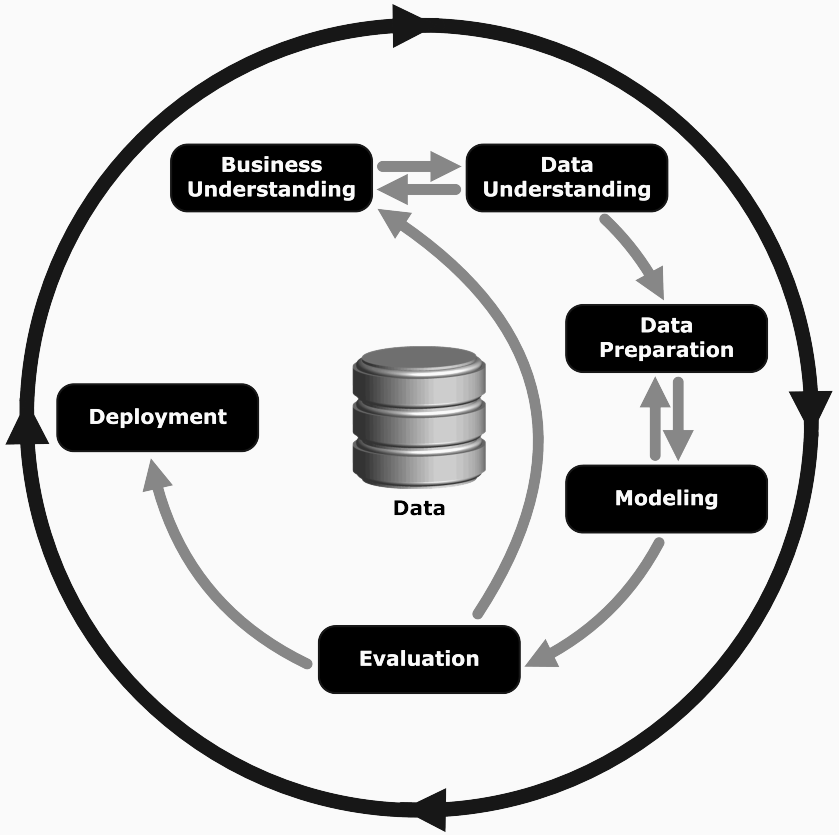

Figure 1: CRISP-DM Prozessmodelldiagramm (Quelle: Kenneth Jensen, CC BY-SA 3.0)

Phasen von CRISP-DM (1/6)

1. **Business Understanding:**
   - Formulierung von konkreten Fragestellungen und Zielen
   - Abgleich von Aufgaben und Erwartungen
   - Vereinbarung eines Vorgehens/einer Planung
   - Identifikation von wichtigen Einflussfaktoren
   - Verständnis des Geschäftsmodells
   - Definition von Erfolgskriterien

Phasen von CRISP-DM (2/6)

2. Data Understanding:

- Betrachtung des Datenbestands
- Auswertung der Datenverfügbar-keit, -reliabilität, -qualität
- (Statistische) Auffälligkeiten in den Daten
- Abstimmung zum Datenschutz

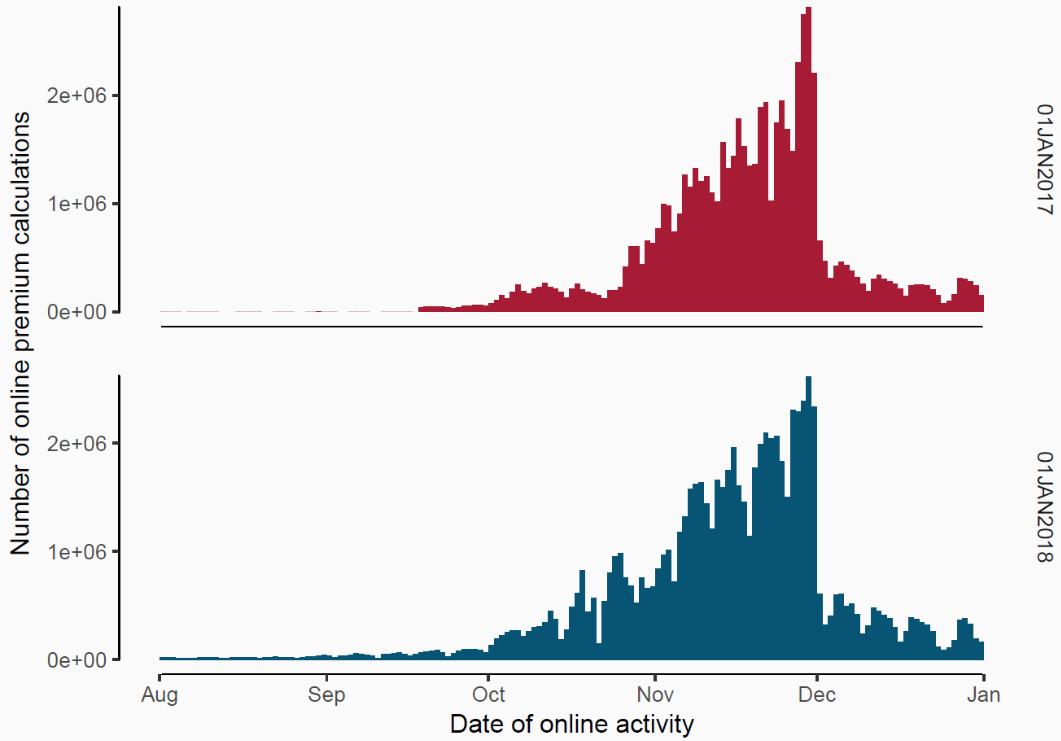

Figure 2: Anzahl von Preisanfragen für Kfz-Versicherungen über Aggregator- bzw. Vergleichswebsites bei einem deutschen Versicherer [2]

Phasen von CRISP-DM (3/6)

3. Data Preparation:

- Datenbereinigung und Transformationen
- Datenverknüpfung und -aggregation
- Feature Engineering
- Feature Selection

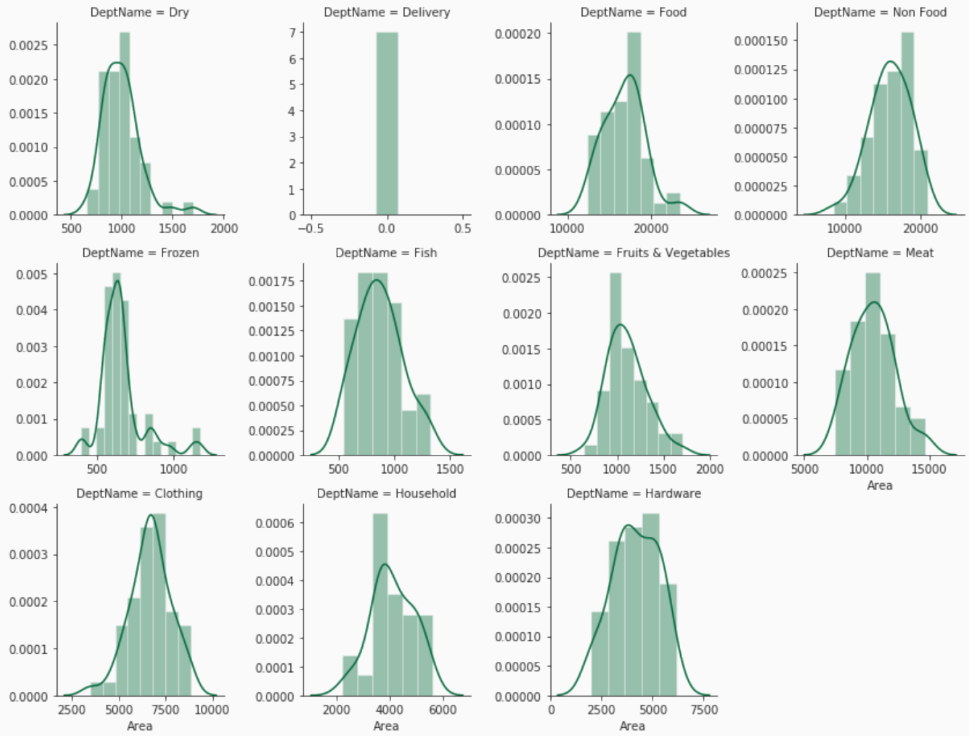

Figure 3: Verteilung von Verkaufsflächen von verschiedenen Märkten eines fiktiven Handelskonzerns, getrennt nach Fachabteilungen [3]

Phasen von CRISP-DM (4/6)

4. Modeling:

- Definition der Annahmen und Rahmenbedingungen der Modellierung
- Auswahl von geeigneten Algorithmen
- Test Design
- Training des Modells
- Tiefgreifende, zielgerichtete Datenexploration

Phasen von CRISP-DM (5/6)

5. Evaluation:

- Vergleich der verschiedenen Modelle anhand von Gütekriterien
- Betrachtung der Interpretierbarkeit des Modells
- Kritische Analyse des Modellierungsprozesses
- Abgleich mit (wirtschaftlichen) Erfolgskriterien
- Definition von Folgeaktivitäten

Phasen von CRISP-DM (6/6)

6. Deployment:

- Kommunikation der Ergebnisse
- Integration des Modells in die Systemlandschaft und Entscheidungsprozesse
- Wartung und Pflege des Modells
- Dokumentation der Erkenntnisse und Funktionsweise

Andere Lifecycle Modelle

- KDD ist Prozessmodell für **K**nowledge **D**iscovery in **D**atabases [4]
- 1996 von Fayyad, Piatetsky-Shapiro, Smyth publiziert
- **S**ample, **E**xplore, **M**odify, **M**odel, and **A**ssess ist ein von SAS vorgeschlagenes Prozessmodell [5]
- Prozess wird trotz Tool-Unabhängigkeit vornehmlich in enger Verknüpfung zu SAS-Lösungen genutzt

Referenzen

[1] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS inc, 9, 13.

[2] Gluesenkamp, D. (2018). Prediction of customer churn with premium online calculation data in insurance business. DeMontfort University, Leicester, United Kingdom.

[3] Gluesenkamp, D. (2019). Wrangling and cleansing business data. Retrieved from https://dgluesen.github.io/wrangling-sales-workload/

[4] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37.

[5] SAS Institute. SAS® Enterprise Miner. Retrieved from https://www.sas.com/content/dam/SAS/en_us/doc/factsheet/sas-enterprise-miner-101369.pdf. Publisher website: https://www.sas.com/