

# Zusammenfassung

Florian Fink

CIS, LMU

`finkf@cis.lmu.de`

February 09, 2021

# Klausur

- ▶ Die Klausuren (Vorlesung und Übung) finden *beide* am Donnerstag 18.02.2021 statt
- ▶ Beginn ist 16:00 Uhr s.t.
- ▶ Die Bearbeitungszeit beträgt 45 Minuten (pro Klausur)
- ▶ Zusätzlich jeweils 10 Minuten Puffer für die finale Abgabe der jeweiligen Klausur(en)
- ▶ FAQ Eintrag auf der Webseite des Kurses:  
`https://cis-sp2021.github.io/faq.html`

# Eigenständigkeitserklärung

- ▶ Laden Sie die Eigenständigkeitserklärung  
`https://cis-sp2021.github.io/erklaerung.pdf`  
von der Kurs-Webseite herunter.
- ▶ Entweder
  1. Ausdrucken der Eigenständigkeitserklärung
  2. Ausfüllen (Vor- und Nachname, Datum und Unterschrift)
  3. Abfotografieren / einscannen
- ▶ Oder:
  1. Ausfüllen im pdf (Vor- und Nachname, Datum und Unterschrift)
  2. Speichern
- ▶ Zusammen mit der Klausur (nächste Folie) zurückschicken.

# Vorgehen

- ▶ E-Mail mit der Klausur (an die @campus.lmu.de Adresse)
- ▶ E-Mail enthält *zwei* Klausuren (pdf) im Anhang:
  1. Die Klausur zur Vorlesung: `vorlesung.pdf`
  2. Die Klausur zur Übung: `uebung.pdf` [sic]
- ▶ Bearbeitete Klausuren via E-Mail an `finkf@cis.lmu.de`:
  1. Die Klausur zur Vorlesung:  
`vorlesung_MATRIKELNUMMER_NAME_VORNAME.pdf`
  2. Die Klausur zur Übung:  
`uebung_MATRIKELNUMMER_NAME_VORNAME.pdf`
  3. Die Eigenständigkeitserklärung (vorherige Folie):  
`erklaerung_MATRIKELNUMMER_NAME_VORNAME.pdf`

# Bearbeitungszeit

- ▶ Die Bearbeitungszeit beträgt 45 Minuten (pro Klausur)
- ▶ Falls *nur eine* Klausur bearbeiten werden soll/muss (Vorlesung *oder* Übung): Abgabe spätestens  $45 + 10 = 55$  Minuten nach Erhalt der Klausuren
- ▶ Falls *beide* Klausur bearbeiten werden sollen/müssen (Vorlesung *und* Übung): Abgabe spätestens  $45 + 45 + 10 + 10 = 110$  Minuten nach Erhalt der Klausuren
- ▶ Schicken Sie nur die relevanten Dokumente zurück:
  - ▶ Eigenständigkeitserklärung und Klausur zur Vorlesung *oder*
  - ▶ Eigenständigkeitserklärung und Klausur zur Übung *oder*
  - ▶ Eigenständigkeitserklärung, Klausur zur Vorlesung und Klausur zur Übung

# Bearbeitung

- ▶ Die Bearbeitungszeit beträgt 45 Minuten (pro Klausur)
- ▶ Handschriftliche Bearbeitung auf Ausdruck:
  1. Ausdrucken der Klausurangabe
  2. Handschriftliche Bearbeitung der Klausur
  3. Einscannen oder Abfotografieren der bearbeiteten Blätter
- ▶ Handschriftliche Bearbeitung auf separaten Blättern:
  1. Handschriftliche Bearbeitung der Klausur auf separaten Blättern
  2. Einscannen oder Abfotografieren der bearbeiteten Blätter
- ▶ Digitale Bearbeitung:
  1. Bearbeitung direkt im pdf der Klausur oder in einer separaten Textdatei.
- ▶ E-Mail der entsprechenden Daten an `finkf@cis.lmu.de`
  - ▶ Benennung der Dateien
  - ▶ Dateien als Anhang

# Testdurchlauf

- ▶ Mittwoch 10.02.2021 Testdokument an die @campus.lmu.de Adresse
- ▶ Bearbeitung (ausdrucken, ausfüllen, einscannen)
- ▶ Antwort der Testdatei als  
test\_MATRIKELNUMMER\_NAME\_VORNAME als E-Mail an  
finkf@cis.lmu.de
- ▶ Individuelle E-Mails bei Problemen / fehlender Antwort etc.

# Verschiedenes

- ▶ die richtigen Daten mit den richtigen Benennungen müssen sich im Anhang der Antwort-E-Mail befinden.
- ▶ Stabile Internetverbindung
- ▶ Genug Akku / Stromkabel
- ▶ Fragen als private Chatnachricht oder telefonisch:  
+49-89-2180-9712
- ▶ Einwilligungserklärung zur Veröffentlichung der Noten (unter Verwendung der Matrikelnummern) auf der Homepage des Kurses  
<https://cis-sp2021.github.io>



# Vorlesung vs. Übung

- ▶ In der Vorlesungsklausur stehen theoretische Aufgaben im Vordergrund
  - ▶ Beschreibung von Algorithmen
  - ▶ Verständnis von Algorithmen
  - ▶ Code-Verständnis
- ▶ In der Übungsklausur stehen praktische Aufgaben im Vordergrund
  - ▶ Implementierung von Algorithmen in Python
  - ▶ Verwendung von git
  - ▶ Tests
- ▶ Teile, die nur für die *Übungsklausur* relevant sind, sind markiert
- ▶ *Alle* Themen, in den Folien, der Vorlesung der Übung oder den Hausaufgaben besprochen wurden sind relevant.
- ▶ HMMs sind *nicht* relevant und *nicht* Thema der Klausur.
- ▶ Die folgenden Themenzusammenfassungen sind *nicht vollständig*

# Basics

Die Python-Basics umfassen unter anderem

- ▶ If-Else
- ▶ Iteration
- ▶ Arithmetik
- ▶ Strings
- ▶ Dateien
- ▶ Listen
- ▶ List-Comprehensions
- ▶ Dictionaries
- ▶ Tupel
- ▶ Regex
- ▶ Doc-Tests (Übung)

# Git (Übung)

## Grundlegendes Verständnis von git

- ▶ Commits
- ▶ Branches
- ▶ Merges
- ▶ Remotes

# Objektorientierung

## Grundlegendes Verständnis von Klassen in Python

- ▶ `self`
- ▶ Konstruktoren
- ▶ Methoden
- ▶ Attribute
- ▶ `@classmethod`
- ▶ `@static` (Übung)
- ▶ Unit-Tests (Übung)

# Dokumentensuche

## Aspekte der Dokumentensuche

- ▶ Dokumentenkollektion
- ▶ Vector Space Model (VSM)
- ▶ TF-IDF-Score
- ▶ Ähnlichkeit im VSM
- ▶ Cosinus zwischen Vektoren
- ▶ Suchanfragen

# Dokumentensuche

## Aspekte der Dokumentensuche

- ▶ Dokumentenkollektion
- ▶ Vector Space Model (VSM)
- ▶ TF-IDF-Score
- ▶ Ähnlichkeit im VSM
- ▶ Kosinus zwischen Vektoren
- ▶ Suchanfragen

# Wahrscheinlichkeitstheorie

## Grundlegende Wahrscheinlichkeitstheorie und Auswertung

- ▶ Klassifikation
- ▶ Clusteranalyse (Clustering)
- ▶ Akkuratheit (Accuracy)
- ▶ Precision und Recall
- ▶ Ereignisraum / Elementarereignisse
- ▶ Wahrscheinlichkeitsabschätzung
- ▶ Bedingte Wahrscheinlichkeit
- ▶ Theorem von Bayes

# Naive Bayes

## Naive Bayes Klassifikation

- ▶ Entscheidungskriterium
- ▶ ( $\lambda$ -)Glättung
- ▶ Logarithmen
- ▶ Odds und Log-Odds
- ▶ Multi-Klassen Klassifikation



# Supervised und unsupervised Machine Learning

## K-Nearest Neighbors (KNN) und K-Means

- ▶ Modelle
- ▶ Unsupervised vs. supervised ML
- ▶ One-hot vector
- ▶ Dokumentenrepräsentation
- ▶ Ähnlichkeitsmaße
- ▶ KNN
- ▶ K-Means

## Natural Language Toolkit

- ▶ Korpora
- ▶ `nltk.word_tokenize`
- ▶ Konkordanzen
- ▶ `nltk.FreqDist`
- ▶ `nltk.ConditionalFreqDist`
- ▶ Bi- und Trigramme
- ▶ Sprachbestimmer

## WordNet im NLTK

- ▶ Hyper(o)nym, Hyponym, Synonym ...
- ▶ Synsets
- ▶ Polysemie
- ▶ Lesk-Algorithmus
- ▶ Stemming
- ▶ Lemmatisierung

# Crawling

## Crawling und Web-Ressourcen

- ▶ `urllib.open`
- ▶ BeautifulSoup
- ▶ HTML
- ▶ Parsing
- ▶ Knoten
- ▶ Text

# POS-Tagging

## Part-of-speech-tagging mit NLTK

- ▶ `nltk.pos_tag`
- ▶ `nltk.sent_tokenize`
- ▶ Tagger
- ▶ Confusion matrix

# Fragen?