Einführung in das maschinelle Lernen

Kursankündigung

M.Sc. Wirtschaftsinformatik

Technische Hochschule Brandenburg

Prof. Dr. Artur Tarassow

Stand: März 2025

Kursinformationen

Vorlesung: 2 SWS
Übung: 2 SWS
Abschlussprüfung: Projektarbeiten
Kreditpunkte: 6 ECTS für 4 SWS

Kursleiter

Büro: WWZ A 3.04
tarassow@th-brandenburg.de
Sprechstunde: auf Anfrage.

Kursüberblick

Der Kurs vermittelt grundlegende Konzepte und Methoden des maschinellen Lernens mit praktischer Anwendung im Bereich der Wirtschaftsinformatik. Der Schwerpunkt liegt auf der Implementierung moderner Verfahren aus dem Bereich des überwachten als auch unüberwachten Lernens im ökonomischen Kontext. Besonderes Augenmerk wird auf das Domänenverständnis (*data literacy*), die praktische Umsetzung mit Python, Datenvisualisierung und Ergebnispräsentation gelegt. In den Übungen werden nach Möglichkeit Datensätze verwendet, die im betriebswirtschaftlichen Kontext relevant sind, wie beispielsweise für die Kundenklassifikation, die Vorhersage von Geschäftskennzahlen oder die Analyse von Geschäftsprozessen.

Dieser Kurs richtet sich an Masterstudierende der Wirtschaftsinformatik, die sich für die Anwendung moderner Methoden des maschinellen Lernens auf ökonomische Fragestellungen interessieren und diese im Rahmen von Projekt- und Abschlussarbeiten verwenden wollen. Die

unterrichteten Inhalte eröffnen berufliche Perspektiven als Data Scientist oder Business Analyst.

1 Lernziele

Die Studierenden beherrschen nach dem Kurs die Programmiersprache Python zur Durchführung komplexer Datenanalysen und können verschiedene maschinelle Lernverfahren selbstständig implementieren und evaluieren. Sie sind in der Lage, grundlegende und fortgeschrittene Methoden wie K-Means, baumbasierte Modelle, Regressionsanalysen und Shrinkage-Verfahren auf strukturierte Datensätze anzuwenden und die Ergebnisse professionell zu visualisieren sowie im jeweiligen ökonomischen Kontext zu interpretieren. Darüber hinaus können sie die Vor- und Nachteile verschiedener Methoden kritisch bewerten, Modelle mittels Kreuzvalidierung und Hyperparameter-Tuning optimieren und ihre Analysen mittels moderner Dashboards präsentieren.

2 Methodik

Die Lehrveranstaltung ist als Vorlesung mit einer begleitenden Übung (insgesamt 4 SWS) konzipiert, wobei die Übungen im PC-Labor stattfinden. Übungsmaterialien und Datensätze werden zur Verfügung gestellt.

Das Modul wird mit einer oder mehreren Projektarbeiten abgeschlossen, in denen die Studierenden die erlernten Methoden anwenden und die Ergebnisse in einem Bericht dokumentieren. Die Projektarbeiten können in Gruppen oder einzeln durchgeführt werden. Die Betreuung erfolgt durch den Kursleiter in regelmäßigen Konsultationsterminen.

3 Kursvoraussetzungen

- Grundkenntnisse in Statistik und Datenanalyse
- Grundlegende Programmierkenntnisse
- Regelmäßige Anwesenheit in Vorlesung und Übung
- Grundlegende Bereitschaft, sich Kenntnisse in Python anzueignen und praktische Übungen damit durchzuführen

Themengebiete

1. Einführung Maschinelles Lernen und Data Science (**1 SWS**)
   (a) Was sind Maschinelles Lernen und Data Science?
   (b) Anforderungsprofil
   (c) Daten als Grundlage für Wertschöpfung
   (d) Der Data Science Workflow
   (e) **Literatur:** Breiman [2001], Yu and Barter [2024, Kap. 1–2]
2. Datentypen, -formate und -beschaffung (**1 SWS**)
   (a) Strukturierte und unstrukturierte Daten
   (b) Datenquellen und -beschaffung
   (c) Einführung in Python
3. Datenaufbereitung (**1 SWS**)
   (a) Datenprüfung
   (b) Fehlende und fehlerhafte Werte identifizieren
   (c) Imputationsverfahren
   (d) Datenformate
   (e) Regeln für den Umgang mit Datensätzen
   (f) Pre-processing Schritte: Standardisierung, Enkodierungsverfahren
   (g) Feature-Building
   (h) **Literatur:** Wickham [2014], Yu and Barter [2024, Kap. 4–5]
4. Wiederholung deskriptive Statistiken (**1 SWS**)
   (a) Lagemaße und Streuungsmaße
   (b) Korrelationsanalyse
   (c) Explorative Datenanalyse mit Python
5. Visualisierung (**1 SWS**)
   (a) Geschichte der Datenvisualisierung
   (b) Elemente von Grafiken
   (c) Symbole, Beschriftung, Ausrichtung
   (d) Typen der Visualisierung
   (e) Best Practices

(f) **Literatur:** Yu and Barter [2024, Kap. 5], Schwabish [2014], Krause and Rennie, FlowingData

6. KNN für Klassifikation und Regression (**1 SWS**)
   (a) Grundlagen des KNN-Modells
   (b) Distanzmaße
   (c) Anwendung für Klassifikation
   (d) Anwendung für Regression
   (e) Implementierung in Python
   (f) **Literatur**: James et al. [2023, Kap. 3.5, 4.7.6]
7. Baumbasierte Modelle für Klassifikation (**2 SWS**)
   (a) Grundlagen von Entscheidungsbäumen
   (b) Entscheidungsbäume für Klassifikation
   (c) Feature-Importance
   (d) Implementierung in Python
   (e) **Literatur**: James et al. [2023, Kap. 8.1], Yu and Barter [2024, Kap. 12.2]
8. CRISP-DM und Evaluation von Klassifikationsmodellen (**1 SWS**)
   (a) CRISP-DM als Standard für Data Mining Prozesse
   (b) Evaluationsmetriken für Klassifikation
   (c) Confusion Matrix, Precision, Recall, F1-Score
   (d) ROC-Kurve und AUC
   (e) **Literatur**: https://datasolut.com/crisp-dm-standard/, Yu and Barter [2024, Kap. 11.4.1]
9. Kreuzvalidierung und Hyperparameter-Tuning (**1 SWS**)
   (a) Training- und Testset
   (b) Übersicht Typen von Kreuzvalidierung
   (c) Grid Search und Random Search
   (d) Hyperparameter-Optimierung in Python
   (e) **Literatur**: James et al. [2023, Kap. 5], Yu and Barter [2024]
10. Regressionsanalyse (**1 SWS**)
    (a) Grundlagen der linearen Regression
    (b) KQ-Schätzer

(c) Shrinkage-Estimators: Ridge & Lasso

(d) Implementierung in Python

(e) **Literatur**: James et al. [2023, Kap. 3.1–3.4], Yu and Barter [2024, Kap. 8–10]

11. Regression-Trees und Ensembles (**1 SWS**)

    (a) Regression-Trees

    (b) Ensemble-Methoden

    (c) Boosted-Trees

    (d) Implementierung in Python

    (e) **Literatur**: James et al. [2023, Kap. 8.1–8.3], Yu and Barter [2024, Kap. 12.3]

12. Unsupervised Learning (**1 SWS**)

    (a) Dimensionsreduktion mittels PCA

    (b) K-means Clustering

    (c) Silhouette-Plot und andere Evaluationsmetriken

    (d) Implementierung in Python

    (e) **Literatur**: James et al. [2023, Kap. 6.3, 12.1–12.4], Yu and Barter [2024, Kap. 5–7], Greenacre et al. [2022]

13. Dashboards & Story-Telling (**1 SWS**)

    (a) Grundlagen des Story-Tellings mit Daten

    (b) Dashboard-Erstellung mit Python (Dash, Streamlit)

    (c) Best Practices für Präsentationen

    (d) Interaktive Visualisierungen

Software

Für die statistisch empirische Analyse wird die Programmiersprache Python mit verschiedenen Bibliotheken verwendet. Folgende Bibliotheken werden im Kurs u.a. eingesetzt:

- NumPy und Pandas für Datenmanipulation
- Matplotlib und Seaborn für Visualisierung
- Scikit-learn für maschinelles Lernen
- Dash oder Streamlit für Dashboards

Es wird empfohlen, eine Python-Distribution wie Anaconda zu installieren, die alle benötigten Bibliotheken enthält.

Installation:

Windows, MAC OS X, LINUX: https://www.anaconda.com/download

Im PC-Labor finden Sie darüber hinaus einen Jupyter-Server mit allen benötigten Bibliotheken vor.

Alternativ können Sie auch Google Colab für die Übungen nutzen: https://colab.research.google.com/.

.

Literatur

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. doi: 10.1.1.125.5395.

FlowingData. Defense Against Dishonest Charts. URL https://flowingdata.com/projects/dishonest-charts/.

Michael Greenacre, Patrick J. F. Groenen, Trevor Hastie, Alfonso Iodice D'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2(1):1–21, December 2022. ISSN 2662-8449. doi: 10.1038/s43586-022-00184-w. URL https://www.nature.com/articles/s43586-022-00184-w.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning: With Applications in Python*. Springer, Cham, Switzerland, 1st ed. 2023 edition edition, July 2023. ISBN 978-3-031-38746-3. URL https://hastie.su.domains/ISLP/ISLP_website.pdf.download.html.

Andreas Krause and Nicola Rennie. Best Practices for Data Visualisation. URL https://royal-statistical-society.github.io/datavisguide/.

Jonathan A. Schwabish. An Economist's Guide to Visualizing Data. *Journal of Economic Perspectives*, 28(1):209–234, February 2014. ISSN 0895-3309. doi: 10.1257/jep.28.1.209. URL https://www.aeaweb.org/articles?id=10.1257/jep.28.1.209.

Hadley Wickham. Tidy Data. *Journal of Statistical Software*, 59:1–23, September 2014. ISSN 1548-7660. doi: 10.18637/jss.v059.i10. URL https://doi.org/10.18637/jss.v059.i10.

Bin Yu and Rebecca L. Barter. *Veridical Data Science: The Practice of Responsible Data Analysis and Decision Making*. The MIT Press, Cambridge, Massachusetts, October 2024. ISBN 978-0-262-04919-1. URL https://vdsbook.com/.

Weitere Materialien

- Python Data Science Handbook: https://jakevdp.github.io/PythonDataScienceHandbook/
- Anaconda + Vscode (10 Min. Video): https://www.youtube.com/watch?v=sts3CFewvkY
- Umgang mit Jupyter Notebooks in Vscode (4 Min. Video): https://youtu.be/h1sAzPojKMg?feature=shared
- Jupyter Notebooks in VS Code Walkthrough (10 Min. Video):https://youtu.be/DA6ZAHBPF1U?feature=shared
- Mit Markdown vertraut machen: https://www.markdownguide.org/cheat-sheet
- Basis-Kurs Python (ohne Anmeldung):https://www.learnpython.org/en/Welcome
- Scikit-learn Dokumentation: https://scikit-learn.org/stable/
- StanfordOnline: Statistical Learning with Python https://www.edx.org/learn/python/stanford-university-statistical-learning-with-python
- Kaggle Learn - Python und Machine Learning: https://www.kaggle.com/learn
- DataCamp - Python für Data Science: https://www.datacamp.com/courses/intro-to-python-for-data-science
- Übersicht unterschiedlicher Notebooks https://datasciencenotebook.org/