

Modulhandbuch

für den Studiengang

Data Science (B.Sc.)

Sommersemester 2022

Hinweise:

- Weitere Informationen zu den einzelnen Studiengängen (Studien- und Prüfungsordnungen, Studienberatung, etc.) finden Sie auf www.math.fau.de/studium
- Semesteraktuelle Informationen zu den angebotenen Lehrveranstaltungen finden Sie im UnivIS-Vorlesungsverzeichnis.
- Module eines Studiengangs sind in der jeweiligen Prüfungsordnung festgelegt. Diese Sammlung umfasst die Module, die vom Department Mathematik in den jeweiligen Studiengängen verwendet werden.

Inhaltsverzeichnis

Algorithmik kontinuierlicher Systeme (AlgoKS)	4
Diskretisierung und numerische Optimierung (DnO)	6
Einführung in die mathematische Datenanalyse (EmDA)	8
Informationsvisualisierung (InfoVIS).....	10
Introduction to Machine Learning (IntroML)	12
Introduction to Statistics and Statistical Programming (Stat)	14
Knowledge Discovery in Databases mit Übung (KDDmUe).....	16
Konzeptionelle Modellierung (KonzMod).....	19
Lineare und nichtlineare Systeme (LNS).....	21
Machine Learning for Engineers I: Introduction to Methods and Tools (MLE1).....	23
Mathematik für Data Science 2 (MDS2).....	25
Parallele und Funktionale Programmierung (PFD)	27
SWAT: Praktikum (SWAT)	29

1	Modulbezeichnung	Algorithmik kontinuierlicher Systeme (AlgoKS) (englische Bezeichnung: Algorithms and Data Structures for Processing Continuous Data)	ECTS 7,5
2	Lehrveranstaltungen	Vorlesung (4 SWS) Übung (2 SWS) Rechnerübung (2 SWS)	
3	Lehrende	Prof. Dr. Ulrich Rüde ulrich.ruede@fau.de	
4	Modulverantwortung	Prof. Dr. Ulrich Rüde ulrich.ruede@fau.de	
5	Inhalt	<ul style="list-style-type: none"> • Grundlagen kontinuierlicher Datenstrukturen (Gleitpunktzahlen, Rundungsfehleranalyse und Kondition, Diskretisierung und Quantisierung, Abtasttheorem, FFT) • Algorithmische Lineare Algebra (direkte und iterative Verfahren für lin. Gleichungssysteme, Ausgleichsprobleme) • Datenstrukturen für geometrische Objekte, Interpolation, Approximation, Grundlagen geometrischer Modellierung, Volumen- und Flächenberechnung. • Kontinuierliche und diskrete Optimierung, nichtlineare Probleme. • Grundlagen der Simulation: Algorithmen zur Lösung von Differentialgleichungen 	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • erwerben fundierte Kenntnisse über Datenstrukturen und Algorithmen zur Behandlung kontinuierlicher Probleme. Die erworbenen Kompetenzen sind sowohl theoretisch-analytischer Art (Analyse von Komplexität, Konvergenz, Fehlerentwicklung) als auch von praktischer Natur (Implementierung der Algorithmen in einer objekt-orientierten Programmiersprache). • planen und bearbeiten kleine Programmierprojekte so, dass sie zeitgerecht fertig gestellt werden. Sie erwerben damit insbesondere die Grundlagen, die für ein vertieftes Studium in den Bereichen Systemsimulation, Mustererkennung, Graphischer Datenverarbeitung unabdingbar sind. 	
7	Voraussetzungen für die Teilnahme	empfohlen: Algorithmen und Datenstrukturen	
8	Einpassung in Musterstudienplan	im 4. Semester	
9	Verwendbarkeit des Moduls	<p>Pflichtmodul in:</p> <ul style="list-style-type: none"> • B.Sc. Informatik • B.Sc. Medizintechnik • B.Sc. Informations- und Kommunikationstechnik <p>Wahlpflichtmodul in</p> <ul style="list-style-type: none"> • B.Sc. Data Science 	
10	Studien- und Prüfungsleistung	<ul style="list-style-type: none"> • Übungsleistung (wöchentliche Hausaufgaben, unbenotet) • Klausur (benotet, 90 min.) 	
11	Berechnung Modulnote	Klausur (100%)	
12	Turnus des Angebots	jährlich im Sommersemester	

13	Arbeitsaufwand	Workload: 225 h davon: <ul style="list-style-type: none"> • Vorlesung: 4 SWS x 15 = 60 h • Übung: 2 SWS x 15 = 30h • Selbststudium 135 h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	

1	Modulbezeichnung	Diskretisierung und numerische Optimierung (DnO) (englische Bezeichnung: Discretization and Numerical Optimization)	ECTS 10
2	Lehrveranstaltungen	Vorlesung (4 SWS) Übung (2 SWS) Tutorium (2 SWS)	
3	Lehrende	Prof. Dr. Martin Burger martin.burger@fau.de	
4	Modulverantwortung	Prof. Dr. Eberhard Bänsch baensch@math.fau.de	
5	Inhalt	<p>Teil 1: Diskretisierung Ein- und Mehrschrittverfahren für Anfangswertaufgaben gewöhnlicher Differentialgleichungen:</p> <ul style="list-style-type: none"> • explizite und implizite Runge-Kutta-Verfahren, BDF, Extrapolation • asymptotische Stabilität (Nullstabilität), Konsistenz, Konvergenz • Steifheit und Stabilität bei fester Schrittweite • Schrittweiten- und Ordnungsadaptivität • Randwertaufgaben für gewöhnliche Differentialgleichungen • Einführung in Finite-Element-Verfahren <p>Teil 2: Unrestringierte Optimierung</p> <ul style="list-style-type: none"> • Abstiegsverfahren • CG-Verfahren (mit Vorkonditionierung, CG-Newton) • Quadratische Optimierungsprobleme • Penalty- und Barriereverfahren <p>Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • verwenden algorithmische Zugänge zu Problemen, die mittels gewöhnlicher Differentialgleichungen beschrieben werden können oder von unrestringierten, endlichdimensionalen Optimierungsproblemen herkommen, und erklären und bewerten diese; • urteilen über die Stabilität und Effizienz eines numerischen Verfahrens; • setzen mit eigener oder gegebener Software Verfahren um und bewerten deren Ergebnisse kritisch; • erläutern und verwenden ein breites Problem- und Verfahrensspektrum: Differenzenverfahren für Anfangs- und Randwertaufgaben, Finite-Element-Verfahren für 2-Punkt-Randwertaufgaben • übertragen die erlangten Fachkompetenzen auf die Behandlung partieller Differentialgleichungen, Abstiegs- und CG-Verfahren bis zum Barriereverfahren; • sammeln und bewerten relevante Informationen und erkennen Zusammenhänge. 	

7	Voraussetzungen für die Teilnahme	empfohlen: <ul style="list-style-type: none"> • Analysis • Lineare Algebra • Programmierung • Einführung Numerik
8	Einpassung in Musterstudienplan	im 4. Semester
9	Verwendbarkeit des Moduls	Wahlpflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Data Science • B.Sc. Mathematik • B.Sc. Technomathematik • B.Sc. Wirtschaftsmathematik
10	Studien- und Prüfungsleistung	<ul style="list-style-type: none"> • Übungsleistung (wöchentliche Hausaufgaben, unbenotet) • Klausur (90 Min., benotet)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	Workload: 300h davon: <ul style="list-style-type: none"> • Vorlesung: 4 SWS x 15 = 60h • Übung: 2 SWS x 15 = 30h • Tutorium: 2 SWS x 15 = 30h • Selbststudium: 180h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	<ul style="list-style-type: none"> • P. Deuffhard und F. Bornemann: Numerische Mathematik II; de Gruyter, Berlin 2002 • J. Stoer und R. Bulirsch: Numerische Mathematik II; Springer, Berlin, 2005 • K. Strehmel und R. Weiner: Numerik gewöhnlicher Differentialgleichungen; Teubner, Stuttgart 1995 • A. Quarteroni, R. Sacco und F. Saleri: Numerische Mathematik I, II; Springer, Berlin 2002 • Vorlesungsskriptum auf der Homepage des Bereichs Modellierung, Simulation und Optimierung des Departments Mathematik (laufend aktualisiert)

	Modulbezeichnung	Einführung in die mathematische Datenanalyse (EmDA) (englische Übersetzung: Introduction in Mathematical Data Analysis)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (1 SWS)	
3	Lehrende	Prof. Dr. Jan Heiland jan.heiland@fau.de	
4	Modulverantwortung	Prof. Dr. Martin Burger martin.burger@fau.de	
5	Inhalt	<ul style="list-style-type: none"> • Fragestellungen der mathematischen Datenanalyse • Datentypen • Clustering • Lineare Regression • fortgeschrittene Inhalte zu Eigenwerten • Hauptachsentransformation • Singulärwertzerlegung (SVD) • Hauptkomponentenanalyse (PCA) • graphbasierte Daten • grundlegende Graphenalgorithmen (Spanning Trees, Dijkstra, Graph Cut) • analytische und numerische Verfahren (z.B. Gradientenabstieg) zur Lösung von Optimierungsproblemen im Kontext der mathematischen Datenanalyse 	
6	Lernziele und Kompetenzen	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • kennen die grundlegenden Fragestellungen und Methoden der mathematischen Datenanalyse, • können die Rolle der Singulärwertzerlegung im Kontext der Analyse von Daten mittels PCA erklären, • sind in der Lage, Lineare Regression und einfache Clusteringverfahren auf Datensätze anzuwenden, • verstehen graphenbasierte Daten und grundlegende Graphenalgorithmen • lösen einfache, datenbasierte Optimierungsprobleme mittels analytischer oder numerischer Verfahren • sind in der Lage, einfache im Modul behandelte Algorithmen auf Datensätze anzuwenden 	
7	Voraussetzungen für die Teilnahme	empfohlen: Mathematik für Data Science 1	
8	Einpassung in Musterstudienplan	im 2. Semester	
9	Verwendbarkeit des Moduls	Pflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Data Science 	
10	Studien- und Prüfungsleistung	Klausur (benotet, 60 Min.)	
11	Berechnung Modulnote	Klausur (100%)	
12	Turnus des Angebots	jährlich im Sommersemester	

13	Arbeitsaufwand	Workload: 150h davon: <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Übung: 1 SWS x 15 = 15h • Selbststudium: 105h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	

1	Modulbezeichnung	Informationsvisualisierung (InfoVIS) (englische Bezeichnung: Information Visualization)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (2 SWS)	
3	Lehrende	Dr. Roberto Grosso Roberto.Grosso@fau.de	
4	Modulverantwortung	Dr. Roberto Grosso Roberto.Grosso@fau.de	
5	Inhalt	In diesem Modul werden folgende Techniken der Informationsvisualisierung behandelt: <ul style="list-style-type: none"> • Graphen und Netzwerke • Dynamische Graphen • Hierarchien und Bäume • Multivariate Daten • Time-Series Daten • Textvisualisierung 	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <p>Wissen</p> <ul style="list-style-type: none"> • zählen Datentypen der Informationsvisualisierung auf • nennen Techniken zur Visualisierung unterschiedlicher Datentypen der Informationsvisualisierung • beschreiben Anwendungsfällen für die unterschiedlichen Datentypen der Informationsvisualisierung <p>Verstehen</p> <ul style="list-style-type: none"> • stellen Algorithmen der Informationsvisualisierung dar und erläutern ihre Eigenschaften, Vorteile und Nachteile • illustrieren Techniken zu Auswertung und Analyse von Daten der Informationsvisualisierung • implementieren die vorgestellten Algorithmen in JavaScript <p>Anwenden</p> <ul style="list-style-type: none"> • wenden Algorithmen zur Visualisierung unterschiedlichen Daten an • erklären und charakterisieren Techniken der Informationsvisualisierung <p>Analysieren</p> <ul style="list-style-type: none"> • klassifizieren Algorithmen zur Visualisierung multivariater Daten, Netzwerke, Hierarchien und Text und erklären ihrer Funktionsweise • erkunden die Effizienz der vorgestellten Algorithmen für unterschiedlichen Anwendungsmöglichkeiten <p>Evaluieren</p> <ul style="list-style-type: none"> • bewerten Anwendbarkeit und Performance spezieller Algorithmen der Informationsvisualisierung • vergleichen Methoden zur Analyse und Auswertung von Daten der Informationsvisualisierung • überprüfen die Anwendbarkeit der diskutierten Techniken für unterschiedliche, speziell ausgewählten Fälle 	
7	Voraussetzungen für die Teilnahme	empfohlen: Algorithmik kontinuierlicher Systeme Die Programmieraufgaben werden in JavaScript implementiert.	
8	Einpassung in Musterstudienplan	im 4. Semester	

9	Verwendbarkeit des Moduls	Wahlpflichtmodul in: <ul style="list-style-type: none"> • M.Sc. Artificial Intelligence • B.Sc. Computational Engineering • M.A. Digital Humanities • B.Sc. Data Science • M.Sc. Data Science • B.Sc. Informatik • M.Sc. Informatik • M.Sc. International Information Systems • M.Sc. Mathematik
10	Studien- und Prüfungsleistung	elektronische Prüfung mit Multiple Choice (benotet, 90 min.)
11	Berechnung Modulnote	elektronische Prüfung (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	Workload: 150 h davon: <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Übung: 2 SWS x 15 = 30h • Selbststudium: 90h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	englisch
16	Literaturhinweise	<ul style="list-style-type: none"> • Robert Spence: Information Visualization: Design for Interaction • Stuart K. Card, Jock Mackinlay, Ben Shneiderman: Readings in Information Visualization – Using Vision to Think • Benjamin B. Bederson, Ben Shneiderman: The Craft of Information Visualization – Readings and Reflections • Tamara Munzner: Visualization Analysis and Design

1	Modulbezeichnung	Introduction to Machine Learning (IntroML)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (2 SWS) Tutorium (2 SWS)	
3	Lehrende	Dr.-Ing. Vincent Christlein vincent.christlein@fau.de	
4	Modulverantwortung	Prof. Dr.-Ing. Andreas Maier andreas.maier@fau.de	
5	Inhalt	<p>Die Vorlesung hat zum Ziel, die Studierenden mit dem prinzipiellen Aufbau eines Mustererkennungssystems vertraut zu machen. Es werden die einzelnen Schritte von der Aufnahme der Daten bis hin zur Klassifikation von Mustern erläutert.</p> <p>Die Vorlesung beginnt dabei mit einer kurzen Einführung, bei der auch die verwendete Nomenklatur eingeführt wird. Die Analog-Digital-Wandlung wird vorgestellt, wobei der Schwerpunkt auf deren Auswirkungen auf die weitere Signalanalyse liegt.</p> <p>Im Anschluss werden gebräuchliche Methoden der Vorverarbeitung beschrieben. Ein wesentlicher Bestandteil eines Mustererkennungssystems ist die Merkmalsextraktion. Verschiedene Ansätze zur Merkmalsberechnung/-transformation werden gezeigt, darunter Momente, Hauptkomponentenanalyse und Lineare Diskriminanzanalyse.</p> <p>Darüber hinaus werden Möglichkeiten vorgestellt, Merkmalsrepräsentationen direkt aus den Daten zu lernen.</p> <p>Die Vorlesung schließt mit einer Einführung in die maschinelle Klassifikation. In diesem Kontext wird der Bayes- und der Gauss-Klassifikator besprochen.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • erklären die Stufen eines allgemeinen Mustererkennungssystems • verstehen Abtastung, das Abtasttheorem und Quantisierung • verstehen und implementieren Histogrammequalisierung und -dehnung • vergleichen verschiedene Schwellwertmethoden • verstehen lineare, verschiebungsinvariante Filter und Faltung • wenden verschiedene Tief- und Hochpassfilter sowie nichtlineare Filter an • wenden verschiedene Normierungsmethoden an • verstehen den Fluch der Dimensionalität • erklären verschiedene heuristische Merkmalsberechnungsmethoden, z.B. Projektion auf einen orthogonalen Basisraum, geometrische Momente, Merkmale basierend auf Filterung • verstehen analytische Merkmalsberechnungsmethoden, z.B. Hauptkomponentenanalyse, Lineare Diskriminanzanalyse • verstehen die Basis von Repräsentationslernen • erläutern die Grundlagen der statistischen Klassifikation (Bayes-Klassifikator) • benutzen die Programmiersprache Python, um die vorgestellten Verfahren der Mustererkennung anzuwenden • lernen praktische Anwendungen kennen und wenden die vorgestellten Algorithmen auf konkrete Probleme an 	
7	Voraussetzungen für die Teilnahme	Ein Mustererkennungssystem besteht aus den folgenden Stufen: Aufnahme von Sensordaten, Vorverarbeitung, Merkmalsextraktion und maschinelle Klassifikation. Diese Vorlesung beschäftigt sich in erster Linie mit den ersten drei Stufen und schafft damit die Grundlage für weiterführende Lehrveranstaltungen (Pattern Recognition und Pattern Analysis).	

8	Einpassung in Musterstudienplan	im 4. Semester
9	Verwendbarkeit des Moduls	Wahlpflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Computational Engineering • B.Sc. Data Science • B.Sc. Elektrotechnik • B.Sc. Informatik • M.Sc. Informatik • B.Sc. Informations- und Kommunikationstechnik • M.Sc. International Information Systems • M.Sc. Maschinenbau • B.Sc. Mathematik • B.Sc. Mechatronik • M.Sc. Mechatronik • B.Sc. Medizintechnik • B.Sc. Wirtschaftsinformatik
10	Studien- und Prüfungsleistung	Klausur (benotet, 60 Min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	Workload: 150 h davon <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30 h • Übung: 2 SWS x 15h = 30 h • Selbststudium: 90 h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	englisch
16	Literaturhinweise	<ul style="list-style-type: none"> • Vorlesungsfolien • Heinrich Niemann: Klassifikation von Mustern, 2. überarbeitete Auflage, 2003 • Sergios Theodoridis, Konstantinos Koutroumbas: Pattern Recognition, 4. Auflage, Academic Press, Burlington, 2009 • Richard O. Duda, Peter E. Hart, David G. Stock: Pattern Classification, 2. Auflage, John Wiley & Sons, New York, 2001

1	Modulbezeichnung	Introduction to Statistics and Statistical Programming (Stat)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Tafelübungen (1 SWS) Rechnerübungen (1 SWS) optionale Wiederholungsstunde (1 SWS)	
3	Lehrende	Prof. Dr. Christoph Richard christoph.richard@fau.de	
4	Modulverantwortung	Prof. Dr. Christoph Richard christoph.richard@fau.de	
5	Inhalt	<ul style="list-style-type: none"> • Einführung in die Statistik-Software R und elementares Programmieren • Beschreibende Statistik: Visualisierung und Kenngrößen kategorieller und metrischer Daten, qq-Plots, Kurvenanpassung, log- und loglog-Plots, robuste Verfahren • Schließende Statistik: Schätz- und Testverfahren: parametrische Tests, ausgewählte nichtparametrische Tests, exakte und asymptotische Konfidenzintervalle • Simulation: Zufallszahlen, Monte-Carlo <p>Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden können</p> <ul style="list-style-type: none"> • die grundlegenden Verfahren der beschreibenden und schließenden Statistik beschreiben und erläutern; • in einer Gruppe ihren Lösungsvorschlag für ein nicht triviales statistisches Problem angemessen erklären und alternative Lösungsvorschläge kritisch vergleichen; • statistische Standard-Auswertungen innerhalb eines vorgegebenen Zeitrahmens mit dem Computer bearbeiten und dessen Ausgaben richtig interpretieren; • einfache statistische Simulationen durchführen; • zu einem Datensatz angemessene Fragen formulieren, adäquate statistische Verfahren zur Beantwortung dieser Fragen wählen und solche Verfahren am Computer umsetzen. 	
7	Voraussetzungen für die Teilnahme	empfohlen: Stochastische Modellbildung	
8	Einpassung in Musterstudienplan	im 4. Semester	
9	Verwendbarkeit des Moduls	<p>Wahlpflichtmodul in:</p> <ul style="list-style-type: none"> • M.Sc. Artificial Intelligence • B.Sc. Data Science • B.Sc. Informatik • M.Sc. Informatik • M.Sc. Integrated Life Sciences • B.Sc. Mathematik (Angewandte Mathematik) • B.Sc. Wirtschaftsmathematik 	

		Pflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Wirtschaftsmathematik Schlüsselqualifikation in: <ul style="list-style-type: none"> • B.Sc. Mathematik • B.Sc. Wirtschaftsmathematik
10	Studien- und Prüfungsleistung	<ul style="list-style-type: none"> • Übungsleistung (wöchentliche Hausaufgaben, unbenotet) • Klausur (benotet, 90 min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	Workload: 150h davon: <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Tafelübung: 1 SWS x 15 = 15h • Rechnerübung: 1 SWS x 15 = 15h • Selbststudium: 90h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	englisch
16	Literaturhinweise	<ul style="list-style-type: none"> • Rice: Mathematical Statistics and Data Analysis; Thomson, 2007 • www.cran.r-project.org

1	Modulbezeichnung	Knowledge Discovery in Databases mit Übung (KDDmUe) (englische Bezeichnung: Knowledge Discovery in Databases with Exercise)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (2 SWS)	
3	Lehrende	M.Sc. Dominik Probst, M.Sc. Melanie Bianca Sigl dominik.probst@fau.de , melanie.sigl@fau.de	
4	Modulverantwortung	Prof. Dr. Richard Lenz richard.lenz@fau.de	
5	Inhalt	<p>Theoretisches Wissen zu:</p> <ul style="list-style-type: none"> • Warum Data Mining? • Was ist Data Mining? • Eine multidimensionale Sicht auf Data Mining • Welche Arten von Daten können gewonnen werden? • Welche Arten von Mustern können gewonnen werden? • Welche Techniken werden benutzt? • Welche Anwendungen werden anvisiert? • Größere Probleme im Data Mining • Ein kurzer geschichtlicher Hintergrund zu Data Mining <p>Praktische Übungen zu:</p> <ul style="list-style-type: none"> • Einführung in Pandas & scikit-learn • Datenanalyse und -verarbeitung • Häufige Muster • Klassifikation • Clustering • Ausreißer 	

6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • kennen den typischen KDD-Prozess; • kennen Verfahren zur Vorbereitung von Daten für das Data Mining; • definieren Distanz- oder Ähnlichkeits-Funktionen auf einem speziellen Datenbestand; • überprüfen Attribute eines Datensatzes auf ihre Bedeutung für die Analyse hin und transformieren ggf. Attributwerte geeignet; • wissen, wie ein typisches Data Warehouse aufgebaut ist; • kennen die Definition von Distanz- bzw. Ähnlichkeitsfunktionen für die verschiedenen Typen von Attributen; • sind vertraut mit dem Prinzip des Apriori-Algorithmus zur Bestimmung von Mengen häufiger Elemente (frequent itemsets); • kennen den FP-Growth-Algorithmus zum schnellen Auffinden von Mengen häufiger Elemente; • geben die Definitionen von Support und Confidence für Assoziationsregeln wieder; • beschreiben die Ermittlung von Assoziationsregeln auf der Basis von Mengen häufiger Elemente; • sind in der Lage, die Vorgehensweise bei Klassifikationsaufgaben darzustellen; • legen dar, wie ein Entscheidungsbaum auf einem Trainingsdatensatz erzeugt wird; • stellen das Prinzip der Bayesschen Klassifikation dar; • zählen verschiedene Clustering-Verfahren auf; • beschreiben den Ablauf von k-Means-Clustering; • kennen die verschiedenen Arten von Ausreißern; • können die verschiedenen Schritte eines KDD Prozesses auch praktisch anwenden.
7	Voraussetzungen für die Teilnahme	Empfohlen: Konzeptionelle Modellierung
8	Einpassung in Musterstudienplan	im 4. Semester
9	Verwendbarkeit des Moduls	<p>Wahlpflichtfach in:</p> <ul style="list-style-type: none"> • B.Sc. Data Science • M.Sc. Informatik
10	Studien- und Prüfungsleistung	Klausur (benotet, 90 min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	<p>Workload: 150h davon:</p> <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Übung: 2 SWS x 15h = 30h • Selbststudium: 90h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	englisch

16	Literaturhinweise	<p>Die Vorlesung basiert auf dem folgenden Buch:</p> <ul style="list-style-type: none"> • J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011, ISBN: 0123814790 <p>Weitere interessante Literatur umfasst:</p> <ul style="list-style-type: none"> • A. Géron, Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems, 2nd ed. O'Reilly Media, 2017, ISBN: 978-1491962299 • H. Du, Data Mining Techniques and Applications: An Introduction. Cengage Learning EMEA, May 2010, p. 336, ISBN: 978-1844808915 • I. H. Witten, E. Frank, M. A. Hall, et al., Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques, 4th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016, ISBN: 0128042915
----	--------------------------	---

1	Modulbezeichnung	Konzeptionelle Modellierung (KonzMod) (englische Bezeichnung: Conceptual Modelling)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übungen (2 SWS)	
3	Lehrende	Prof. Dr. Viktor Leis viktor.leis@fau.de	
4	Modulverantwortung	Prof. Dr. Richard Lenz richard.lenz@fau.de	
5	Inhalt	<ul style="list-style-type: none"> • Grundlagen der Modellierung • Datenmodellierung am Beispiel Entity-Relationship-Modell • Modellierung objektorientierter Systeme am Beispiel UML • Relationale Datenmodellierung und Abfragemöglichkeiten • Grundlagen der Metamodellierung • XML • Multidimensionale Datenmodellierung • Domänenmodellierung und Ontologie 	
6	Lernziele und Kompetenzen	<p>Die Studierenden:</p> <ul style="list-style-type: none"> • definieren grundlegende Begriffe aus der Datenbankfachliteratur • erklären die Vorteile von Datenbanksystemen • erklären die verschiedenen Phasen des Datenbankentwurfs • benutzen das Entity-Relationship Modell und das erweiterte Entity-Relationship Modell zur semantischen Datenmodellierung • unterscheiden verschiedene Notationen für ER-Diagramme • erläutern die grundlegenden Konzepte des relationalen Datenmodells • bilden ein gegebenes EER-Diagramm auf ein relationales Datenbankschema ab • erklären die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF • definieren die Operationen der Relationenalgebra • erstellen Datenbanktabellen mit Hilfe von SQL • lösen Aufgaben zur Datenselektion und Datenmanipulation mit Hilfe von SQL • erklären die grundlegenden Konzepte der XML • erstellen DTDs für XML-Dokumente • benutzen XPATH zur Formulierung von Anfragen an XML-Dokumente • definieren grundlegende Strukturelemente und Operatoren des multidimensionalen Datenmodells • erklären Star- und Snowflake-Schema • benutzen einfache UML Use-Case Diagramme • benutzen einfache UML-Aktivitätsdiagramme • erstellen UML-Sequenzdiagramme • erstellen einfache UML-Klassendiagramme • erklären den Begriff Meta-Modellierung • definieren den Begriff der Ontologie in der Informatik • definieren die Begriffe RDF und OWL 	
7	Voraussetzungen für die Teilnahme	empfohlen: <ul style="list-style-type: none"> • Algorithmen und Datenstrukturen • Grundlagen der Logik und Logikprogrammierung 	
8	Einpassung in Musterstudienplan	im 2. Semester	
9	Verwendbarkeit des Moduls	Pflichtmodul: <ul style="list-style-type: none"> • B.Sc. Data Science • B.Sc. Informatik 	

10	Studien- und Prüfungsleistung	Klausur mit Multiple Choice (benotet, 90 min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	<p>Workload: 150h davon:</p> <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Übung: 2 SWS x 15 = 30h • Eigenstudium: 90h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	<ul style="list-style-type: none"> • Alfons Kemper, Andre Eickler: Datenbanksysteme : Eine Einführung. 6., aktualis. u. erw. Aufl. Oldenbourg, März 2006. - ISBN-10: 3486576909 • Bernd Oestereich: Analyse und Design mit UML 2.1. 8. Aufl. Oldenbourg, Januar 2006. - ISBN-10: 3486579266 • Ian Sommerville: Software Engineering. 8., aktualis. Aufl. Pearson Studium, Mai 2007. - ISBN-10: 3827372577 • Horst A. Neumann: Objektorientierte Softwareentwicklung mit der Unified Modeling Language. (UML). Hanser Fachbuch, März 2002. - ISBN-10: 3446188797 • Rainer Eckstein, Silke Eckstein: XML und Datenmodellierung. Dpunkt Verlag, November 2003. - ISBN-10: 3898642224 • Elmasri, Ramez, and Sham Navathe. Grundlagen von Datenbanksystemen. Pearson Deutschland GmbH, 2009.

1	Modulbezeichnung	Lineare und nichtlineare Systeme (LNS) (englische Übersetzung: Linear and Nonlinear Systems)	ECTS 10
2	Lehrveranstaltungen	Vorlesung (4 SWS) Übung (2 SWS) Tafelübung (1 SWS)	
3	Lehrende	Dr. Dieter Weninger dieter.weninger@math.uni-erlangen.de	
4	Modulverantwortung	Prof. Dr. Alexander Martin alexander.martin@fau.de	
5	Inhalt	<p>Der Kurs beinhaltet:</p> <ul style="list-style-type: none"> • Gleichungs-/Ungleichungssysteme • Alternativsätze • Iterationsverfahren • Grundbegriffe der Optimierung • Gemischt-ganzzahlige lineare Optimierung • Gemischt-ganzzahlige nichtlineare Optimierung • Einblicke in die Optimalsteuerung von Differentialgleichungssystemen <p>Die Präsentation des Stoffes erfolgt in Vorlesungsform. Die weitere Aneignung der wesentlichen Begriffe und Techniken erfolgt durch wöchentliche Hausaufgaben.</p>	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • erkennen und analysieren selbstständig lineare und nichtlineare Systeme bzw. Optimierungsprobleme • erläutern verschiedene algorithmische Grundprinzipien und wenden diese zielorientiert an • stellen Verknüpfungen zwischen algebraischem und analytischem Wissen her 	
7	Voraussetzungen für die Teilnahme	<p>empfohlen:</p> <ul style="list-style-type: none"> • Analysis I und II • Lineare Algebra I und II 	
8	Einpassung in Musterstudienplan	im 4. Semester	
9	Verwendbarkeit des Moduls	<p>Wahlpflichtmodul in:</p> <ul style="list-style-type: none"> • B.Sc. Data Science • B.Sc. Mathematik • B.Sc. Wirtschaftsmathematik 	
10	Studien- und Prüfungsleistung	mündliche Prüfung (benotet, 20 Min.)	
11	Berechnung Modulnote	mündliche Prüfung (100%)	
12	Turnus des Angebots	jährlich im Sommersemester	

13	Arbeitsaufwand	<p>Workload: 300h davon:</p> <ul style="list-style-type: none"> • Vorlesung 4 SWS x 15 = 60h • Übung: 2 SWS x 15 = 30h • Tafelübung: 1 SWS x 15 = 15h • Selbststudium: 195h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	<ul style="list-style-type: none"> • Vorlesungsskript zu diesem Modul (siehe StudOn) • Alt: Nichtlineare Optimierung • Pochet, Wolsey: Production Planning by Mixed Integer Programming • Belotti et al.: Mixed-Integer Nonlinear Optimization • Kirk: Optimal Control Theory: An Introduction

1	Modulbezeichnung	Machine Learning for Engineers I: Introduction to Methods and Tools (MLE1)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (2 SWS)	
3	Lehrende	Prof. Dr. Björn Eskofier, Prof. Dr.-Ing. Jörg Franke, Prof. Dr. Nico Hanenkamp bjoern.eskofier@fau.de , joerg.franke@faps.fau.de , nico.hanenkamp@fau.de	
4	Modulverantwortung	Prof. Dr. Björn Eskofier bjoern.eskofier@fau.de	
5	Inhalt	<p>Dieser Kurs bietet einen Überblick über einige der am häufigsten verwendeten Methoden des maschinellen Lernens. Es werden die notwendigen Grundlagen für dieses Themengebiet vermittelt und mittels Programmierübungen vertieft.</p> <p>Der Kurs beinhaltet:</p> <ul style="list-style-type: none"> • Die gängigsten Verfahren zur Datenerfassung, Anomalie Erkennung und Signalfusion • Die Vermittlung verschiedener Methodiken zum Thema Regression, Klassifikation und Dimensionsreduktion (z. B. lineare Regression, Support Vector Machines (SVM) und Deep Neural Networks (DNN)) • Die Einführung in die Python-Programmierung im Bereich Data Science • Die Anwendung von Methoden des maschinellen Lernens auf reale technische Anwendungen <p>Dieser Kurs vermittelt den Teilnehmern die theoretischen und praktischen Grundlagen um Anwendungen im Bereich der Datenverarbeitung zu entwerfen und zu entwickeln.</p>	
6	Lernziele und Kompetenzen	<p>Die Teilnehmer können nach dem Besuch des Kurses</p> <ul style="list-style-type: none"> • die gängigen Verfahren zur Datenverarbeitung, Anomalie Erkennung und Signalfusion anwenden. • die verschiedenen Aufgabenstellungen zur Regression, Klassifikation und Dimensionsreduktion unter Verwendung von Methoden wie z.B. lineare Regression und Klassifikation, Support Vector Machines (SVM) und Deep Neural Networks (DNN) lösen. • die Programmiersprache Python zur Entwicklung von "Data Science" Anwendungen anwenden. • Methoden des maschinellen Lernens auf reale technische Anwendungen anwenden 	
7	Voraussetzungen für die Teilnahme	empfohlen: Grundkenntnisse in Statistik und Wahrscheinlichkeitsrechnung	
8	Einpassung in Musterstudienplan	ab 2. Semester	

9	Verwendbarkeit des Moduls	Wahlpflichtmodu in: <ul style="list-style-type: none"> • B.Sc. Data Science • B.Sc. Informatik • M.Sc. Informatik • B.Sc. International Production Engineering and Management • B.Sc. Maschinenbau • M.Sc. Maschinenbau • B.Sc. Mechatronik • M.Sc. Mechatronik • B.Sc. Wirtschaftsingenieurwesen • M.Sc. Wirtschaftsingenieurwesen
10	Studien- und Prüfungsleistung	Klausur (benotet, 90 min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	zweimal jährlich jeweils im Sommer- und Wintersemester
13	Arbeitsaufwand	Workload: 150h davon: Selbststudium: 150h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	englisch
16	Literaturhinweise	empfohlene Literatur: <ul style="list-style-type: none"> • Machine Learning: A Probabilistic Perspective; Kevin Murphy, MIT press, 2012 • The Elements of Statistical Learning: Data Mining, Inference, and Prediction; Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009 • Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press, 2016

1	Modulbezeichnung	Mathematik für Data Science 2 (MDS2)	ECTS 10
2	Lehrveranstaltungen	Vorlesung (4 SWS) Übung (2 SWS) Tafelübung (2 SWS)	
3	Lehrende	Prof. Dr. Jan Heiland jan.heiland@fau.de	
4	Modulverantwortung	Prof. Dr. Martin Burger martin.burger@fau.de	
5	Inhalt	<p>Der Kurs beinhaltet:</p> <ul style="list-style-type: none"> • Eigenwerte • Euklidische Vektorräume (Orthonormalisierung, Orthogonalprojektion). • Diskrete Fouriertransformation als Beispiel für Orthogonalbasis, Hinführung auf Fourier-Reihen • Normierte Räume, stetige Abbildungen zwischen normierten Räumen, Kompaktheit, Vollständigkeit, Dualraum • Fixpunktsatz von Banach • Satz von Arzela-Ascoli • Bilinearformen, Skalarprodukte • Adjungierte Operatoren • Differentialrechnung in mehreren Veränderlichen: Partielle Ableitung und Jacobi-Matrix, Satz von Schwarz, • Grundlagen Lineare Differentialgleichungen mit konstanten Koeffizienten: Lösung mittels Exponentiation von Matrizen bzw. mit charakteristischem Polynom • Gewöhnliche Differentialgleichungen: Lokale und globale Existenz und Eindeutigkeit der Lösung, Phasenportrait (DGL: insgesamt 2 Wochen) • Extrema, Optimierung mit Nebenbedingungen (kurz, wird im Kernmodul vertieft) • totale Ableitung und Linearisierung, Lipschitz-Stetigkeit und Schrankensatz, Taylorformel 	
6	Lernziele und Kompetenzen	<p>Die Studierenden</p> <ul style="list-style-type: none"> • erweitern ihr Spektrum an Grundbegriffen der Analysis und erklären diese; • wenden das Grundwissen der Analysis an, reproduzieren und vertiefen grundlegende Prinzipien und ordnen diese ein; • wenden Grundtechniken der Analysis an; • sammeln und bewerten relevante Informationen und erkennen Zusammenhänge, erkennen lineare und nichtlineare Zusammenhänge und behandeln sie quantitativ und qualitativ; • verwenden und untersuchen quadratische Formen als die einfachsten nicht-linearen Funktionen; • verwenden Dualräume zur Analyse linearer Abbildungen; • erkennen die Querverbindung zur Analysis; • führen exemplarische inner- und außermathematische Anwendungen durch. 	
7	Voraussetzungen für die Teilnahme	empfohlen: Mathematik für Data Science 1	
8	Einpassung in Musterstudienplan	im 2. Semester	

9	Verwendbarkeit des Moduls	Pflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Data Science Wahlpflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Physik • Modulstudien Naturale
10	Studien- und Prüfungsleistung	<ul style="list-style-type: none"> • Übungsleistung (unbenotet) • Klausur (benotet, 120 min.)
11	Berechnung Modulnote	Klausur (100%)
12	Turnus des Angebots	jährlich im Sommersemester
13	Arbeitsaufwand	Workload: 300h davon: <ul style="list-style-type: none"> • Vorlesung: 4 SWS x 15 = 60h • Übung: 2 SWS x 15 = 30h • Tafelübung: 2 SWS x 15 = 30h • Selbststudium: 180h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	

1	Modulbezeichnung	Parallele und Funktionale Programmierung (PFP) (Parallel and Functional Programming)	ECTS 5
2	Lehrveranstaltungen	Vorlesung (2 SWS) Übung (2 SWS)	
3	Lehrende	Prof. Dr. Michael Philippsen michael.philippsen@fau.de Dr. Norbert Oster norbert.oster@fau.de	
4	Modulverantwortung	Prof. Dr. Michael Philippsen michael.philippsen@fau.de	
5	Inhalt	Der Kurs beinhaltet: <ul style="list-style-type: none"> • Grundlagen der funktionalen Programmierung • Grundlagen der parallelen Programmierung • Datenstrukturen • Objektorientierung • Scala-Kenntnisse • Erweiterte JAVA-Kenntnisse • Aufwandsabschätzungen • Grundlegende Algorithmen 	
6	Lernziele und Kompetenzen	Die Studierenden <ul style="list-style-type: none"> • erlernen die Grundlagen der funktionalen Programmierung anhand der Programmiersprache • Scala • verstehen paralleles Programmieren mit Java • kennen fundamentale Datenstrukturen und Algorithmen • können funktionale und parallele Algorithmen entwickeln und analysieren 	
7	Voraussetzungen für die Teilnahme		
8	Einpassung in Musterstudienplan	im 2. Semester	
9	Verwendbarkeit des Moduls	Pflichtmodul in: <ul style="list-style-type: none"> • B. Sc. Data Science • B. Sc. Informatik Wahlpflichtmodul in: <ul style="list-style-type: none"> • B.Sc. Berufspädagogik Technik • B.Sc. Computational Engineering • B.Sc. Mathematik • B.Sc. Technomathematik • B.Sc. Wirtschaftsinformatik • B.Sc. Wirtschaftsmathematik 	
10	Studien- und Prüfungsleistung	Klausur (benotet, 60 min.)	
11	Berechnung Modulnote	Klausur (100%)	
12	Turnus des Angebots	jährlich im Sommersemester	

13	Arbeitsaufwand	Workload: 150h davon: <ul style="list-style-type: none"> • Vorlesung: 2 SWS x 15 = 30h • Übung: 2 SWS x 15 = 30h • Eigenstudium: 90h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	

1	Modulbezeichnung 669768	SWAT: Intensivübungen (SWAT) (englische Bezeichnung: SWAT is a Web Application Tutorial)	ECTS 5
2	Lehrveranstaltungen	Tutorium (2 SWS) Blockpraktikum (0.5 SWS) Praktikum (5 SWS)	
3	Lehrende	M.Sc. Demian Vöhringer, M.Sc. David Haller demian.voehringer@fau.de david.haller@fau.de	
4	Modulverantwortung	Prof. Dr. Richard Lenz richard.lenz@fau.de	
5	Inhalt	Der Kurs beinhaltet: <ul style="list-style-type: none"> • Entwurf und Implementierung einer typischen Web-Applikation • Kreatives Arbeiten im Team • Agile Softwareentwicklung • Verwendung von aktuellen Technologien • Moderne Programmiertechniken 	
6	Lernziele und Kompetenzen	Die Studierenden <ul style="list-style-type: none"> • konzipieren und implementieren eine mehrschichtige Web-Anwendung. • bewerten den Arbeitsaufwand von Aufgaben. • wenden agile Entwicklungsmethoden im Rahmen von Softwareentwicklung an. • arbeiten kooperativ und verantwortlich in Gruppen und können das eigene Kooperationsverhalten sowie die Zusammenarbeit in der Gruppe kritisch reflektieren und optimieren. • arbeiten sich eigenständig in Technologien ein, stellen diese Technologien in Präsentationen vor und wenden sie im Projekt an. 	
7	Voraussetzungen für die Teilnahme		
8	Einpassung in Musterstudienplan	im 4. Semester	
9	Verwendbarkeit des Moduls	Wahlpflichtfach in: <ul style="list-style-type: none"> • B.Sc. Data Science • M.Sc. Data Science • B.Sc. Informatik • M.Sc. Informatik • M.Sc. International Information Systems • B.Sc. Mathematik • M.Sc. Mathematik • B.Sc. Wirtschaftsinformatik 	
10	Studien- und Prüfungsleistung	mehrteilige Prüfung (benotet) Die Bewertung der Prüfungsleistung setzt sich zusammen aus 30% Fachvortrag (20 min), 50% Praktikum (Team-Arbeit, Arbeitsorganisation, Zeitplanung, Code, Dokumentation) und 20% mündliche Prüfung (20 min).	
11	Berechnung Modulnote	100 % (mehrteilige Prüfung)	
12	Turnus des Angebots	jährlich im Sommersemester	

13	Arbeitsaufwand	Workload: 150h davon: <ul style="list-style-type: none"> • Praktikum: 5 SWS x 12 = 60h • Übung: 1,5 SWS x 10 = 15h • Blockpraktikum: 40h • Eigenstudium: 35h
14	Dauer des Moduls	1 Semester
15	Unterrichts- und Prüfungssprache	deutsch
16	Literaturhinweise	<ul style="list-style-type: none"> • Elemental Design Patterns, Smith, 2012 • Patterns of Enterprise Application Architecture, Fowler, 2003 • Scrum mit User Stories, Wirdemann, 2011 • Agile Testing, Crispin and Gregory, 2009 • More Agile Testing, Crispin and Gregory, 2015