Informatik Spektrum (2020) 43:129–136
https://doi.org/10.1007/s00287-020-01253-8

HAUPTBEITRAG

Data Science für alle: Grundlagen der Datenprogrammierung

Ein Data-Science-Kurs für alle Studierenden der TU Berlin

Ziawasch Abedjan[1] · Hagen Anuth[1] · Mahdi Esmailoghli[1] · Mohammad Mahdavi[1] · Felix Neutatz[1] · Binger Chen[1]

Online publiziert: 27. Februar 2020

Zusammenfassung

Die Nachfrage nach Data Scientists in den verschiedensten Bereichen der Industrie, Gesellschaft und Forschung stellt Universitäten vor die Frage, in welcher Form eine Data-Science-Ausbildung ermöglicht werden soll. Neben dem traditionellen Ansatz, Data Science als Studienfach anzubieten, gibt es auch Forderungen nach Einbettung von Data-Science-Veranstaltungen in informatik- und mathematikfremden Fächern, um die gesteigerte Nachfrage nach Datenkompetenzen in diesen Bereichen abzudecken. Dies wird auch durch die erst kürzlich von der GI geförderten Initiative für „Data Literacy“ unterstützt. Vor diesem Hintergrund haben wir an der TU Berlin einen Data-Science-Kurs auf Bachelorniveau nach dem Vorbild des Data8-Kurses an der Berkeley-Universität in Kalifornien konzipiert und erfolgreich durchgeführt. In dem Kurs „Data Science 1: Essentials of Data Programming“ werden Grundlagen der Programmierung, statistische Datenanalyse, maschinelles Lernen und ethische Fragen bei der Anwendung dieser Methoden vermittelt. Das Angebot stieß auf ein sehr starkes Interesse seitens der Studierenden verschiedenster Studiengänge der TU Berlin einschließlich Kunstgeschichte und Philosophie. Zur erfolgreichen Durchführung des Kurses gehörte nicht nur die entsprechend entworfene integrierte Synopsis, die orientiert an Fallbeispielen mathematische Konzepte und Programmiertechniken vermittelt, sondern auch regelmäßige Übungsstunden und Hausaufgaben sowie eine zentralverwaltete JupyterHub-Infrastruktur, die sowohl die Nicht-Informatikstudierenden vor jeglicher Installation von unbekannter Software behütete als auch die Automatisierung der Korrektur der Programmierhausaufgaben ermöglichte. In diesem Beitrag möchten wir über unsere Erkenntnisse berichten, wie es uns gelungen ist, Studierende mit sehr unterschiedlichen Informatikkenntnissen für Data Science zu begeistern. Dabei gehen wir auf die praktische Durchführung des Kurses und der abschließenden Leistungsüberprüfung ein. Zuletzt zeigen wir die Vorteile eines solchen Kurses auf. Dazu zählt die skalierbare Möglichkeit, weiten Teilen der Studierenden Datenkompetenzen zu vermitteln und den Quereinstieg in die Informatik zu verschaffen.

Von Data Literacy zu Data Science

Auf Berufsportalen wie LinkedIn sind inzwischen mehr Stellen für Data Scientists als für klassische Informatiker ausgeschrieben, noch dazu bleiben diese Stellen eine Woche länger auf dem Markt als der Durchschnitt[1]. Trotz der derzeitigen Popularität ist der Begriff „Data Science“ nicht präzise definiert. Je nach wissenschaftlicher Perspektive werden unterschiedliche Schwerpunkte im Hinblick auf die Teilthemengebiete Statistik, Informatik und Wirtschaftswissenschaften gesetzt. Entsprechend sind die dazugehörigen Grundlagen noch nicht eindeutig festgelegt. Üblicherweise suchen Unternehmen nach Universitätsabsolvent*innen, die Kernfähigkeiten aus der Informatik, insbesondere der Programmierung, der skalierbaren Datenverarbeitung und des maschinellen Lernens mit mathematischen Grundlagen, insbesondere zur statistischen Datenanalyse und Modellentwicklung, sowie Domänenwissen in einer Anwendung und Kommunikations- und Präsentationsfähigkeiten vorzeigen können. Zusätzlich gibt es ein steigendes Interesse, auch Studierende von informatik- und mathematikfernen Studiengängen mit grundlegenden Methoden der Data Science vertraut zu machen. In diesem Zusammenhang wurden bereits sogenannte Informationskompetenzen definiert [1]. Auch die GI hat einen neuen Schwerpunkt auf die

[1] https://data-science-blog.com/blog/2019/06/16/the-data-scientist-job-and-the-future/

✉ Ziawasch Abedjan
abedjan@tu-berlin.de

[1] TU Berlin, Berlin, Deutschland

sogenannte „Data Literacy“[2] gesetzt, welche digitale Kompetenzen zusammenfasst.

Um diese Initiative zu unterstützen, haben wir an der TU Berlin im Rahmen eines geplanten „Data Literacy“-Zertifikates, welches Studierende aus informatik- und mathematik*fremden* Studiengängen zusätzlich zu ihrem Studienabschluss erlangen können, einen neuen Grundlagenkurs angeboten und erfolgreich durchgeführt.

Inspiriert durch den „Data8“-Kurs, der am Data Science Institute der kalifornischen Berkeley-Universität für alle Studierenden angeboten wird und in enger Kollaboration mit ihren Dozenten haben wir den Kurs „Data Science 1: Essentials of Data Programming“ konzipiert, welcher sich an Studierende ohne jegliche Programmiererfahrung richtet und Grundlagen des algorithmischen Umgangs mit Daten auf einem geeigneten Niveau für Studierende verschiedenster Studienrichtungen behandelt. Unsere Initiative hierzu wurde zu Beginn mit großer Skepsis betrachtet, da ein solcher Kurs mehrere Herausforderungen überwinden muss. Insbesondere wurde es als unrealistisch angesehen, Programmierung und Mathematik innerhalb eines Kurses zu vermitteln. Weiterhin ist es eine große Herausforderung, ein Abstraktionsniveau auszuwählen, für das lediglich die Hochschulreife ausreicht. Schließlich erfordert eine solche Initiative, die sich an die Großzahl der Studierenden richtet, ein sinnvolles Skalierungskonzept. Es geht nicht nur um die frontale Vermittlung von Konzepten, sondern diese auch praktisch durch regelmäßige Hands-on-Übungen anzuwenden, welche wiederum eine intensive Betreuungsanstrengung erfordern.

Wir haben diesen Kurs bereits einmal erfolgreich durchgeführt und möchten in diesem Artikel über dessen Aufbau und Durchführung berichten. Der Erfolg dieses Kurses ist dem effektiv ausgewählten Abstraktionslevel und der aufgebauten technischen Infrastruktur für praktische Programmierübungen und Tutorien zuzuschreiben. Dabei wird die übliche Lehre invertiert. Studierende lernen zunächst den praktischen Umgang mit Werkzeugen und Methoden, wohingegen theoretische Konzepte erst im Nachgang vertieft werden. Konkret lernen Studierende mit praktischen Fragestellungen umzugehen und nutzen die zentralgewartete Jupyter-Notebook-Umgebung, um sich auf die eigentlichen Programmieraufgaben zu konzentrieren.

Im Folgenden werden wir zunächst die Kursinhalte und den Kursaufbau beschreiben. Darauffolgend werden wir anhand von Statistiken beleuchten, inwieweit der Kurs den selbstgesetzten Zielen gerecht wurde, Data Literacy effektiv an eine diverse Gruppe von Studierenden zu vermitteln und zukünftige Quereinstiegsmöglichkeiten zur Data Science zu ermöglichen.

Kursinhalte

Die Synopsis des Moduls „Data Science 1: Essentials of Data Programming“ umfasst zentrale Data-Science-Grundlagen: Kausalität und Korrelation, Big Data, Datenextraktion, Datenvisualisierung, Zufallsvariablen, Vergleich von Stichproben, Hypothesentests, Schätzung und Vorhersage von Prüfgrößen, Klassifizierung und ethische Fragen in Data Science. Zeitgleich sollten die Studierenden, viele zum ersten Mal, die Konzepte der Programmierung erlernen. Das heißt, dass wir parallel zu den jeweiligen mathematischen Konzepten schrittweise Programmierkonzepte wie Datentypen, Variablen, Zuweisungen, Kontrollstrukturen und Datenverarbeitungsfunktionen vermitteln mussten. Wir haben uns bei den Kursinhalten grob an dem Onlinebuch „Computational and Inferential Thinking“[3] orientiert, diese aber durch Themenblöcke zu Big Data und Ethik erweitert.

Zunächst war es uns wichtig die Studierenden des Kurses für das Thema Data Science als solches zu begeistern.

Hierbei ist es wichtig zu erwähnen, dass dies bei einer diversen Studierendenlandschaft nicht immer mit Hinblick auf die hervorragenden beruflichen Aussichten für Datenwissenschaftler*innen geschehen kann. Obgleich eines der Ziele die Motivation zum Quereinstieg in Data Science und Informatik darstellt, war es uns wichtig, dass die Studierenden die Relevanz der Methoden für ihre eigenen Studiengebiete entdecken. Deshalb war ein zentrales Element jeder Vorlesung der Bezug zur Praxis und die Anwendung der vermittelten Konzepte auf echten Datensätzen. Wir haben hierbei Datensätze über unterschiedlichste Themen aus Leben und Wissenschaft wie z. B. Daten über den Verkehr um die TU Berlin, historische Begebenheiten, amerikanischer Zensus und Gesundheitsfragen zur Hilfe genommen. So vertieften die Studierenden zum Beispiel die Konzepte der Vorhersage, indem sie anhand von Bitcoin-Kursdaten den Ethereum-Wert vorhersagten oder das Alter des Universums abschätzten. In einem anderen Beispiel wendeten sie die Bootstrapping-Methode an, um die Verteilung der Gehälter von Profisportler*innen unter die Lupe zu nehmen.

In den ersten Kurswochen wurden die nötigen Grundlagen in Mathematik und Informatik vermittelt, um alle Studierenden auf einem gemeinsamen Niveau abzuholen. Insbesondere wurde die notwendige Terminologie vermittelt. Es erfolgte eine Einführung in die Programmierung mit Python. Selbst diese Schritte wurden praxisorientiert vermittelt. Zunächst lernten die Studierenden einfache Operationen auf Tabellen auszuführen. Die Programmieraufgaben zu diesen Operationen umfassten Datenextraktion, -verarbeitung und -visualisierung. Dabei lernten die Studierenden auch die richtigen Diagrammarten für kategorische und nu-

[2] https://gi.de/dataliteracy/

[3] https://www.inferentialthinking.com

merische Daten auszuwählen und deren Inhalte zu interpretieren.

Ausgerüstet mit diesen Grundlagen beschäftigten wir uns dann mit den grundlegenden Begrifflichkeiten und Themen der Statistik wie Wahrscheinlichkeitsbegriff, Zufallsvariablen und dem Rechnen mit Wahrscheinlichkeiten. Daraufhin erörterten und verglichen wir verschiedene Methoden, um Stichproben von großen Gesamtheiten zu erlangen. Erst hier lernten Studierende die Anwendung von Schleifen in Python.

Aufbauend auf Grundlagen der Statistik und mächtigen Kontrollstrukturen der Programmiersprache Python behandelten wir das nächste Themengebiet der Simulationen, statistischen Tests und Signifikanzwerte für anschauliche Anwendungsfälle.

In den folgenden Wochen lernten die Studierenden die Theorie der Bootstrap-Methode kennen und lernten diese Methode auch anzuwenden, um Parameter einer Grundgesamtheit abzuschätzen, von der lediglich Stichproben vorliegen. Darüber hinaus behandelten wir Konfidenzintervalle, mit denen, aufbauend auf dem Wissen über den p-Wert, Hypothesentests beantwortet und begründet sowie Aussagen über die Qualität von Schätzwerten getroffen werden können. Hier tauchten wir anschließend noch tiefer in die Materie ein und stellten die Tschebyscheffsche Ungleichung und den zentralen Grenzwertsatz vor.

In der Folge lernten die Studierenden, wie sich anhand von Daten Aussagen über die Zukunft treffen lassen. So lernten die Teilnehmer den Korrelationskoeffizienten zu berechnen, um lineare Beziehungen zwischen Variablen zu messen. Allerdings beleuchteten wir hier auch die Grenzen und Voraussetzungen für eine sinnvolle Arbeit mit statistischen Größen wie dem Korrelationskoeffizienten. Darauf aufbauend erörterten wir, wie sich mithilfe des Korrelationskoeffizienten eine Regressionsgerade für die Beziehung zwischen Variablen berechnen lässt. Hierfür vollzogen wir die Experimente von Sir Francis Galton nach, welche den Grundstein für die Regressionsgleichungen legten. Weiterführend behandelten wir die Methode der kleinsten Quadrate, um eine möglichst präzise Regressionsgerade für entsprechende Szenarien zu berechnen.

Bei der Behandlung des Themas Klassifizierung stellten wir zum einen den k-nächsten-Nachbarn-Algorithmus vor und veranschaulichten hierbei auch die Effekte „Überanpassung“ und „Unteranpassung“, die bei einer unpassenden Wahl der Nachbarschaftsgröße auftreten können. Außerdem sprachen wir darüber, wie Klassifizierungsmodelle trainiert werden und ihre Präzision anhand von Testdaten gemessen werden können.

Abschließend haben wir nochmal die gesellschaftliche Relevanz und Sensibilität bezüglich der Wissensgewinnung aus Daten diskutiert. Hierbei gingen wir auf Irreführung und Täuschung, die durch mangelnde Datenqualität, Vorurteile und unbekannte Störfaktoren entstehen können, ein.

Zusammenfassend wurden in dem Kurs „Data Science 1: Essentials of Data Programming“ die zentralen statistischen Konzepte vermittelt, die für moderne Datenwissenschaften relevant sind. Zudem wurden den Studierenden die Grundlagen der Programmierung mit Python beigebracht. Abschließend gaben wir mit den Themen Klassifizierung und Vorhersagen einen Ausblick auf weitere spannende Vertiefungsrichtungen der Datenwissenschaften wie maschinelles Lernen und hoffen dadurch, nachhaltiges Interesse für diese Themen bei den teilnehmenden Studierenden geweckt zu haben.

Lehrmethoden und digitale Lehrmittel

Unser Ziel war es, die zuvor genannten Kursinhalte an ein diverses und großes Publikum zu vermitteln. Hierzu mussten wir eine technische Lösung für die Skalierung des Übungs- und Hausaufgabenbetriebes einsetzen. In dem ersten Anlauf haben wir 150 Studierende zugelassen.

Veranstaltungsangebot

Der Kurs bestand aus zwei verschiedenen Veranstaltungsformen. Zum einen gab es die Vorlesung, die von Prof. Abedjan gehalten wurde. Hier wurden in erster Linie die theoretischen Konzepte vermittelt und durch Programmierbeispiele in Jupyter Notebooks demonstriert und angewendet. Die Vorlesungsunterlagen basierten auf dem Textbuch „Inferential Thinking“ der Berkeley-Universität und wurden den Studierenden nach der Veranstaltung über die E-Learning-Plattform ISIS der TU Berlin zur Verfügung gestellt. Nach der ersten Vorlesung wurden die 150 teilnehmenden Studierenden auf die vier angebotenen Übungsstunden aufgeteilt.

Die Übungsstunden wurden somit für jeweils ca. 40 Studierende von den Doktoranden Mahdavi und Esmailoghli gehalten. Ziel der Übungsstunden war es, den vermittelten Stoff aus der Vorlesung praktisch anzuwenden und somit auch zu vertiefen. Hierzu wurden live und am Jupyter Notebook Übungsaufgaben gemeinsam bearbeitet und diskutiert. Außerdem waren die Übungsstunden besonders zu Beginn des Semesters ein gut genutzter Anlass, um etwaige Python-bezogene Fragen zu besprechen und Probleme zu lösen. Grundsätzlich waren die Übungen stets gut besucht und die Diskussionen wurden sehr gut und engagiert von den Studierenden angenommen. Da Aufgaben zentralverwaltet wurden und alle Studierenden in der gleichen Entwicklungsumgebung arbeiteten, gab es eine recht hohe Dynamik und keine technisch verursachten Unterbrechungen.

JupyterHub-Infrastruktur und Hausaufgaben

Zur Verteilung der Kursmaterialien und als zentrales Infrastrukturelement setzten wir für diesen Kurs einen JupyterHub-Server auf. JupyterHub ermöglichte es uns, den Studierenden eine einheitliche Programmierumgebung zu bieten. Somit blieben die Teilnehmer*innen von der Installation und Konfiguration jeglicher Software verschont und konnten sich über den Browser direkt bei der JupyterHub-Plattform anmelden und mit einer zur Nutzung bereitstehenden Entwicklungsumgebung arbeiten. Gerade bei informatikfremden Studierenden, welche bisher keine Programmiererfahrung sammeln konnten, erspart eine solche Umgebung Studierenden und Lehrenden viel Zeit und Frustration.

Abb. 1 zeigt die Architektur unserer Infrastruktur. Im Gegensatz zu der Installation an der UC Berkeley haben wir aus Gründen des Datenschutzes und der generellen Wartbarkeit darauf verzichtet, unser System auf Kubernetes aufzubauen, da hierbei unser Server von Google hätte gehostet werden müssen. Stattdessen haben wir den Server lokal auf einer Maschine mit 256 GB Hauptspeicher und 28 Kernen dockerbasiert installiert. Hierbei haben wir uns an Erfahrungen der Universität Versailles orientiert, die eine ähnliche Infrastruktur aufgesetzt hat. Eine ausführliche Dokumentation der Installation haben wir online zur Verfügung gestellt[4].

Auf dem JupyterHub-Server hatte jede*r Studierende Zugriff auf einen isolierten Workspace, an dem die Kursmaterialien, die zentral über ein GitHub Repository bereitgestellt wurden, stets aktualisiert verfügbar waren. Darüber hinaus konnten Studierende hier eigene Notebooks erstellen, um zu experimentieren und die Kursinhalte zu vertiefen.

Während des Semesters veröffentlichten wir außerdem insgesamt sieben digitale Hausaufgabenblätter. Pro Hausaufgabenblatt waren maximal 100 Punkte zu erreichen und um zu der schriftlichen Klausur am Ende des Semesters zugelassen zu werden, waren insgesamt mindestens 50 %, sprich 350 Hausaufgabenpunkte, notwendig. Zu der Abschlussnote des Kurses trugen die Hausaufgabenpunkte jedoch nicht bei. Die Hausaufgabenblätter wurden ebenfalls über unsere JupyterHub-Infrastruktur bereitgestellt und ausgegeben. Hierfür nutzten wir die JupyterHub-Erweiterung „nbgrader“, welche die nötigen Tools zur automatisierten Ausgabe und Kontrolle der Aufgaben ermöglicht.

Durch diese Erweiterung lassen sich Aufgabenblätter mit verschiedenen Arten von Zellen erstellen. Die gewöhnlichen Markdown- und Programmcodezellen, welche auch standardmäßig in Jupyter Notebooks verwendet werden, dienten hier als Bearbeitungsbereich für die konkreten Aufgaben. Darüber hinaus lassen sich auch schreibgeschützte Zellen erstellen, welche wir nutzten, um notwendige Bibliotheken zu importieren, Beispiele für die Studierenden anzuführen und um die Aufgabenstellungen darzustellen. Außerdem lassen sich auch Bewertungszellen einfügen. Diese Zellen sind für die Studierenden nicht sichtbar und enthalten Tests, um die Angaben der Studierenden zu überprüfen. Sie lassen sich zudem mit einer Punktzahl versehen, welche dem Studierenden zugeschrieben wird, sofern alle Tests innerhalb dieser Zelle erfolgreich waren. Nach der Bearbeitungszeit, welche meist 2 Wochen betrug, sammelten wir die Aufgabenblätter mithilfe der „nbgrader“-Erweiterung automatisiert ein und starteten die Bewertungsskripte, welche die Aufgabenblätter ausführten, die darin enthaltenen Bewertungszellen überprüften und die erreichten Punktzahlen den Studierenden gutschrieben.

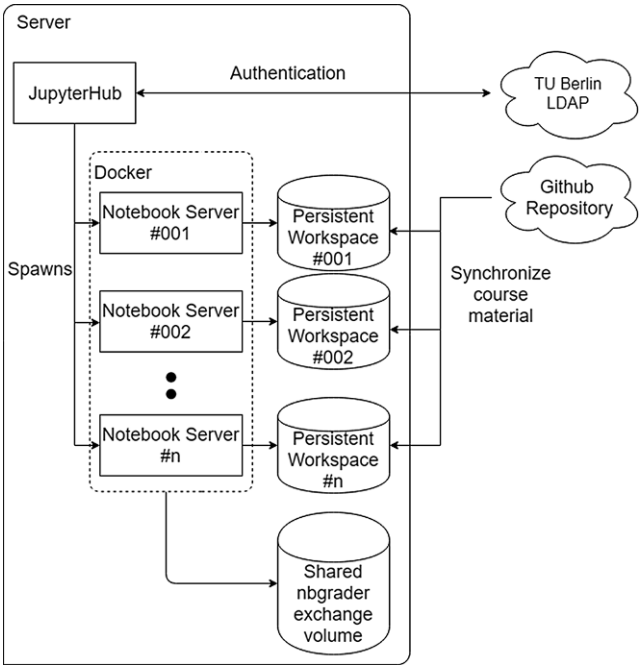


Abb. 1 TU Berlin JupyterHub-Infrastruktur

Diese Methode ersparte uns bei über 1000 Abgaben (7 Hausaufgabenblätter bei ca. 150 Studierenden) sehr viel Korrekturzeit, da die Erstellung der Hausaufgaben und der zugehörigen Tests meist an ein bis zwei Arbeitstagen erledigt waren. Die Studierenden erhielten anschließend ihre korrigierten Hausaufgaben zur Einsicht zurück und konnten diese auch mit einer bereitgestellten Musterlösung vergleichen.

Klausur

Am Ende des Semesters fand schließlich die schriftliche Klausur statt, welche den gesamten Stoff des Semesters prüfte. In der Woche vor der Klausur wurde in der Vor-

[4] https://github.com/bigdama/jupyterhub-environment

lesung zum einen eine Zusammenfassung über die behandelten Themen gegeben, zum anderen wurden auch noch einige Beispielaufgaben auf Klausurniveau vorgestellt und bearbeitet. In den Übungen wurden in dieser Woche auch noch einmal Wiederholungen angeboten und letzte Fragen geklärt. In der Klausur, welche klassisch mit Stift und Papier zu bearbeiten war, wurden meist nach kurzen Python-Programmen gefragt, welche die theoretischen Konzepte, die in dem Kurs behandelt wurden, anwendeten. Hierfür wurde den Studierenden ein „Spickzettel“ zur Verfügung gestellt, auf dem die wichtigsten Python-Funktionen dokumentiert waren. Obwohl wir von den Studierenden eine gewisse Sorgfalt erwarteten, ahndeten wir nicht jeden Syntaxfehler in den Programmen mit Punktabzug sondern legten viel mehr Wert auf Verständnis und die korrekte Anwendung der Konzepte.

Schlussendlich führten wir den bisher ersten Kurs an der TU Berlin durch, der eine JupyterHub-Infrastruktur zur Lehre einsetzte und wir können hieraus ein positives Fazit ziehen. Die Studierenden konnten auf eine vorbereitete Programmierumgebung zugreifen und benötigten hierfür lediglich einen Browser sowie Internetzugang. Auch ermöglichte uns diese Infrastruktur, die Hausaufgaben der Studierenden automatisiert auszugeben, einzusammeln und zu bewerten. Die Infrastruktur wurde von den Studierenden schnell adaptiert und wir erhielten auch hier sehr positives Feedback.

Evaluierung des Kurses

Im Folgenden wollen wir basierend auf den erfassten Kursdaten und Evaluierungsbögen der Studierenden diskutieren, inwieweit unsere Veranstaltung Studierende dem Themengebiet Data Science nähergebracht bzw. diese motiviert hat, das Thema weiterzuführen. Zunächst werden basierend auf Statistiken zu den teilnehmenden Studierenden die Attraktivität des Kurses allgemein diskutiert. Zusätzlich nehmen wir Evaluationsbögen zur Hilfe, um die Nützlichkeit und den Schwierigkeitsgrad des Kurses für Studierende diverser Studiengebiete zu analysieren.

Teilnehmerstruktur und Kurserfolg

Aus Kapazitätsgründen und zur kontrollierten erstmaligen Durchführung des Kurses hatten wir beschlossen, den Kurs lediglich auf 120 Teilnehmer*innen zu beschränken. Die initialen Interessensbekundungen überstiegen jedoch die Zahl 300 und wir entschlossen uns, weitere 30 Studierende aufzunehmen. Tatsächlich haben wir mit einer über die Zeit abnehmenden Teilnehmerstruktur gerechnet, die sich tatsächlich bis zum Ende des Semesters auf 120 herunterregulierte.

Abb. 2 zeigt die Verteilung der Studierenden und deren Studiengangzugehörigkeit. Insgesamt nahmen Studierende aus 40 unterschiedlichen Studienprogrammen an unserem Kurs teil. Tatsächlich nahmen auch insgesamt 30 Studierende der Studiengänge Informatik, Wirtschaftsinformatik, und ICT teil, obwohl der Kurs explizit nicht für diese empfohlen war. Nichtsdestotrotz können wir behaupten, unser Ziel, Studierende verschiedener Programme anzuziehen erreicht zu haben. Interessant ist auch die Tatsache, dass der Anteil von weiblichen Studierenden mit 34 % deckungsgleich mit dem prozentualen weiblichen Anteil von Studierenden an der TU Berlin und wesentlich höher als der prozentuale Anteil in der Informatik (16 % im Bachelor und 9 % im Master) ist.

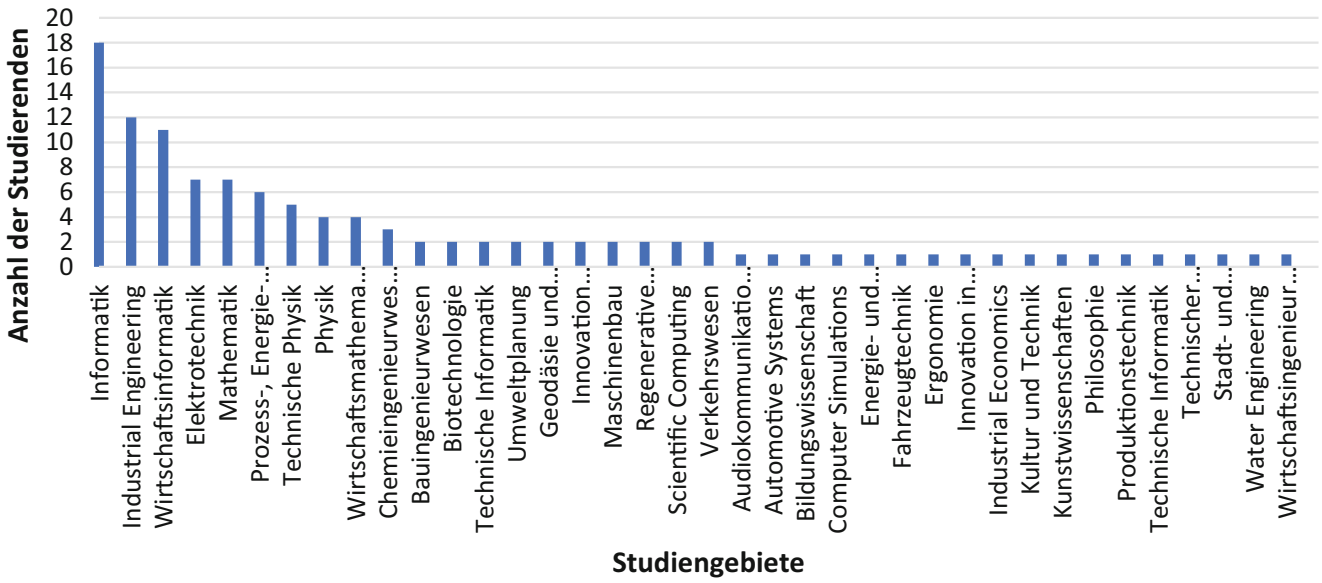


Abb. 2 Verteilung von Teilnehmer*innen nach Studienfächern

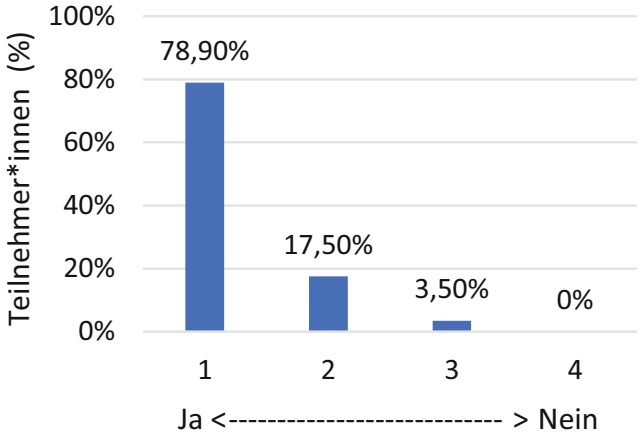


Abb. 3 Antwort auf die Frage, ob die Studierenden motiviert sind Data Science weiter zu verfolgen

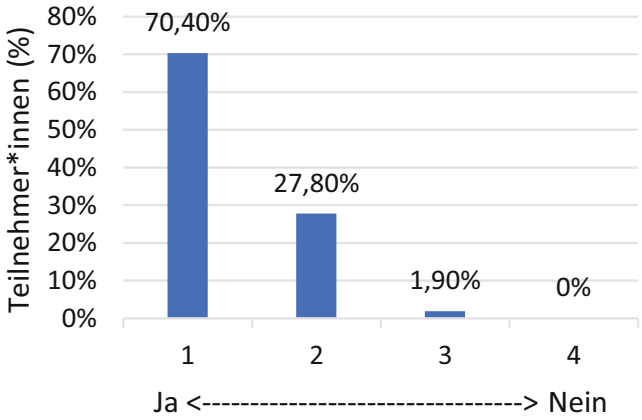


Abb. 4 Gesamtbewertung des Kurses

Zieht man noch zusätzlich Ergebnisse einer Befragung (Abb. 3) in Betracht, nach der mehr als 78 % sich sehr gut und 17 % sich gut vorstellen können, das Thema Data Science in Zukunft zu vertiefen, kann man bezüglich des Quereinstiegspotenzials optimistisch sein. Das Ergebnis einer weiteren Befragung, nach der 98 % der Studierenden die Veranstaltung mit mindestens gut (60 % sehr gut) bewertet haben (Abb. 4), kann hinsichtlich eines positiven Einflusses auf diesen Trend gedeutet werden. Dies wird auch dadurch untermauert, dass 70 % der Teilnehmer*innen den Kurs als absolut nützlich und weitere 27 % als eher nützlich bewertet haben (Abb. 5).

Bewertung des Kursaufbaus

Wie zuvor erklärt, bestand der Kurs aus einer Vorlesung und einer Übung pro Woche. Insgesamt gab es sieben Hausaufgaben im ganzen Semester. Um die Zweckmäßigkeit dieser Struktur zu zeigen, können wir die Korrelation zwischen durchschnittlichem Übungserfolg und den Ergebnissen der Endklausur in Betracht ziehen.

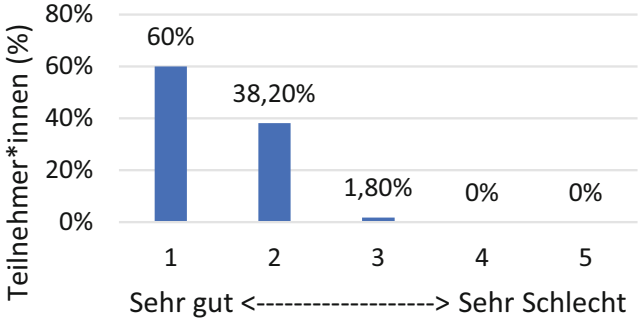


Abb. 5 Sind die Inhalte dieses Kurses insgesamt nützlich?

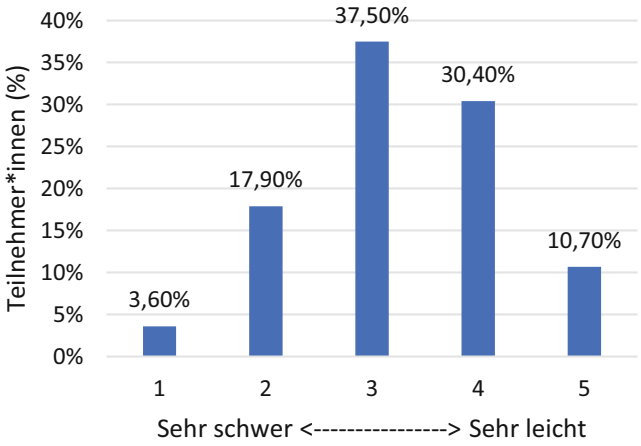


Abb. 6 Die Bewertung der Schwierigkeit der Kursinhalte durch die Teilnehmer*innen

Zuvor ist es interessant festzustellen, dass die Studierenden selbst tendenziell die Inhalte des Kurses als schwierig empfunden haben, wie man dies anhand der Antwortverteilung der Studierenden in Abb. 6 beobachten kann.

Praktisch zeigte sich jedoch, dass der Großteil der Studierenden entgegen ihrer Erwartung erfolgreich durch die Inhalte geführt wurde. Abb. 7 zeigt eine Heatmap, die Hausaufgabenpunkte und Klausurnoten zueinander in Beziehung setzt. Die x-Achse zeigt Hausaufgabenpunkte von 300 (Mindestpunktzahl für die Zulassung zur Prüfung) bis 700 (maximale Punktzahl über sieben Hausaufgaben). Die y-Achse zeigt die Noten 1 bis 5. Jede Zelle der Heatmap beziffert die Anzahl der Teilnehmer*innen mit entsprechender Gesamthausaufgabenpunktzahl und Note. Insgesamt kann eine starke Korrelation beobachtet werden, die eine sinnvolle Abstimmung von Hausaufgaben und Klausur aufzeigt.

Die ideale Anpassung von Übungen, Hausaufgaben und Klausuraufgaben war hauptsächlich durch die aufgesetzte JupyterHub-Umgebung möglich. Auch dies wird seitens der Studierenden als solches empfunden, wie in Abb. 8 zu sehen ist. Über 74 % der Teilnehmer*innen bestätigten die einfache Nutzbarkeit der Programmierumgebung. Weitere 24 % sind zumindest teilweise derselben Ansicht.

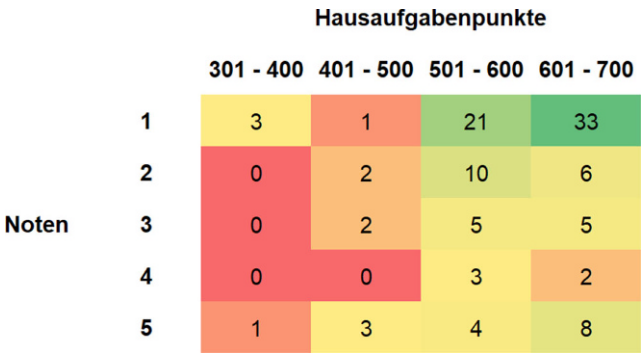


Abb. 7 Heatmap zur Darstellung der Korrelation zwischen Hausaufgabenpunkten und Klausurnoten

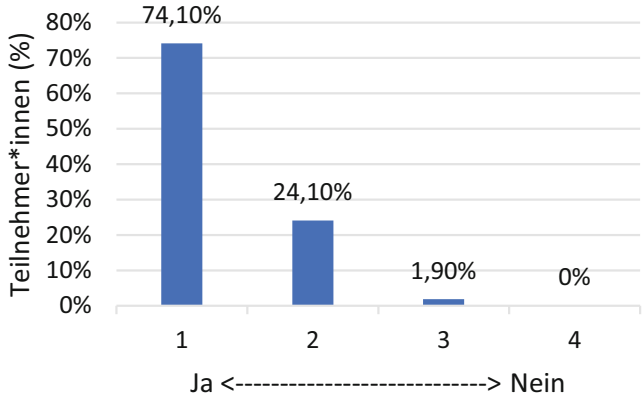


Abb. 8 Die Einschätzung der Studierenden, wie einfach die Programmierumgebung zu bedienen ist

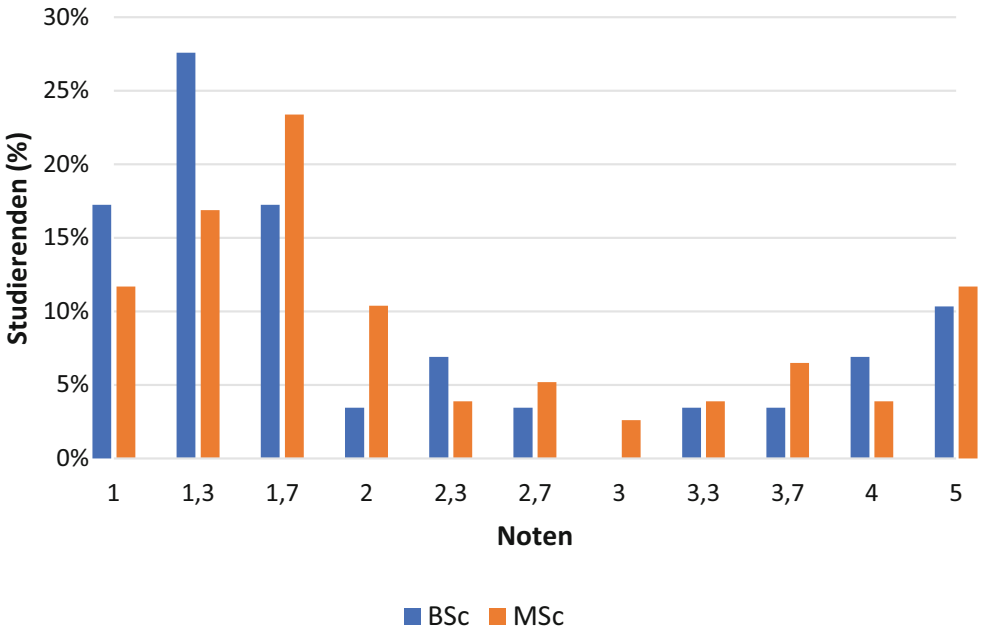
Teilnehmerstruktur und Leistung

Da die Teilnehmerstruktur des Kurses sehr divers war, wollen wir zuletzt Einblicke auf beobachtbare Tendenzen hinsichtlich der Teilnehmergruppen aufzeigen. Insbesondere wollen wir hierbei überprüfen, ob das Niveau des Kurses für diese diverse Struktur angemessen war.

Abb. 9 zeigt die Verteilung der Klausurnote, aufgeteilt nach Niveau des Studienganges. Insgesamt sind die Verteilungsbilder sehr ähnlich. Interessanterweise gab es deutlich mehr Bachelorstudierende, die Bestnoten von 1,0 und 1,3 erreichen konnten. Insofern kann man argumentieren, dass die Studienerfahrung von Masterstudierenden keinen erkennbaren Vorteil darstellt. In der Tat kann man davon ausgehen, dass die meisten Teilnehmer*innen ein genuines Interesse und den Umstand zur Weiterqualifikation mitbrachten. Dies traf auch auf eine kleine Anzahl von Promotionsstudent*innen zu, die ebenso an diesem Kurs teilnahmen.

Die nächste Aufteilung, die wir untersuchen wollen, ist die Vermittelbarkeit der inhaltlichen Konzepte mit Bezug auf den Studiengang der Teilnehmer*innen. Hierbei wollen wir auch die Vermittelbarkeit der unterschiedlichen Themen, Programmierung, Datenverarbeitung, Statistik und Maschinelles Lernen untersuchen. Abb. 10 zeigt die durchschnittliche Punktzahl zu den in der Klausur behandelten Teilgebieten aufgeteilt nach Zugehörigkeit der Studierenden zu Informatik, Mathematik und anderen Studiengebieten. Tatsächlich sind die Ergebnisse aus allen drei Kategorien sehr ähnlich. Wir können eine konsistent höhere Punktzahl bei den Mathematikstudent*innen erkennen. Interessanterweise schneiden die Informatik*innen selbst in den Programmieraufgaben durchschnittlich schlechter ab.

Abb. 9 Verteilung der Noten nach Zugehörigkeit der Studierenden zum Master- und Bachelorstudiengang



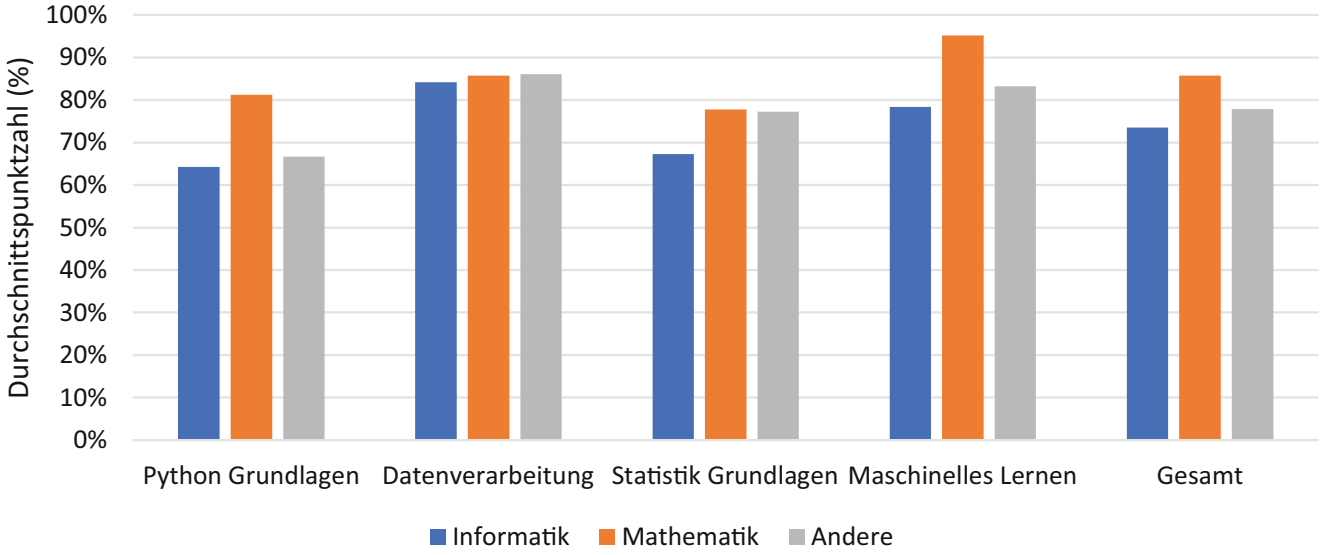


Abb. 10 Durchschnittspunktzahl der Studierenden nach Klausurthemen und Fachzugehörigkeit

Unsere Erklärung für diesen Umstand ist, dass die Teilnehmer*innen aus der Informatik schlechter vorbereitet waren, da sie ihre Fähigkeiten hier überschätzt haben könnten. Auch ist es möglich, dass sie die Teilnahme nicht so ernst wie die anderen Teilnehmer*innen genommen haben, da der Kurs sich nicht in ihrem Pflichtkatalog befindet. Die Teilnehmer*innen anderer Fachrichtungen hingegen hatten ein ernsthaftes Interesse zur Weiterqualifikation.

Fazit

Mit unserem Einsatz zur Realisierung eines fächerunabhängigen universitären Data-Science-Kurses sind wir zunächst auf sehr viel Skepsis gestoßen. Verständlicherweise erforderte die Realisierung den Einsatz und das Engagement seitens des Modulverantwortlichen und der involvierten Doktorand*innen, die jenseits ihrer Kernlehre und -forschung lag. Dies sahen wir jedoch aufgrund der heutigen Relevanz des Themas als notwendig an, um einerseits ein besseres Verständnis für das gesamte Gebiet zu erlangen und andererseits Themen der Informatik und der Datenwissenschaft an ein größeres Publikum zu vermitteln. Insbesondere die hohe Teilnahmequote von weiblichen Studierenden wirkte hier als vielversprechendes Signal. Zusammenfassend ist festzustellen, dass ein solcher Kurs drei wichtige Komponenten erfordert. Erstens ist es notwendig, eine Synopsis und ein Abstraktionslevel auszuwählen, welches für eine Breite von Studierenden zugänglich ist. Zweitens muss der Kurs interaktiv gestaltet sein und Tutorien beinhalten, um hands-on-theoretische Konzepte zu vermitteln. Drittens erfordert solch eine Interaktivität eine technische Infrastruktur, die den Teilnehmer*innen erlaubt, sich hauptsächlich auf inhaltliche Konzepte zu konzentrieren ohne Vorkenntnisse in der Programmierung und IT mitzubringen.

Danksagung Die Realisierung dieses Kurses wäre ohne die aktive Teilnahme der Studierenden der TU Berlin nicht möglich gewesen. Die regelmäßige Teilnahme und Mitarbeit der Studierenden hat uns stets motiviert. Weiterhin wäre der Kurs ohne die Vorarbeit des Data Science Instituts an der UC Berkeley nicht möglich gewesen. Auch möchten wir uns hier für den Erfahrungsaustausch bedanken. Zuletzt möchten wir uns beim Vizepräsidenten der TU Berlin für Lehre, Digitalisierung und Nachhaltigkeit, Prof. Dr. Hans-Ulrich Heiß, für die finanzielle Unterstützung des Kurses und Prof. Dr. Volker Markl für die fachliche Beratung bedanken.

Funding Open Access funding provided by Projekt DEAL.

Literatur

1. Ballod M (2005) Informationskompetenz. Dimensionen eines Begriffs. Comput Unterr 15(59):44–46