Angleichungsleistung
Data Science und Engineering mit Python

Dozent / Prüfer: Prof. Dr. Christian Decker, christian.decker@reutlingen-university.de

Ziele des Moduls

Datenprodukte sind softwarebasierte Services, die Verfahren des Machinenlernens nutzen. ML basierte SW Systeme bilden den Kern dieser Services. Studierende lernen und nutzen in diesen Modul neuste Technologien für ML basierte SW Systeme. Auszeichnendes Merkmal ist, dass Kenntnisse und Fertigkeiten über diese Technologien flexibel durch Online-Kurse gelernt werden. Der Kompetenznachweis erfolgt durch darauf aufbauende Aufgaben, in denen Studierende über das Wissen der Onlinekurse hinausgehen.

Ziele im Einzelnen

- Praktikabilität / Nutzbarkeit der Einbeziehung von externen Ausbildungsinhalten in den curricularen Modulen des Studiengangs Digital Business Engineering (DBE) evaluieren
- Studierende können individuell flexibel studieren und werden kompetent betreut
- Studierende sind Know-How Träger und können das Wissen aus Onlinekursen selbständig für die Entwicklung und Betrieb von Datenprodukten erweitern.

Kenntnisse, Fertigkeiten und Kompetenzen

Nach der Durchführung des Kurses haben die Studierenden die folgenden Kenntnisse, Fertigkeiten und Kompetenzen.

Kenntnisse

- Programmierkenntnisse in Python
- Wissen in statistischer Datenanalyse und Umsetzung der Verfahren in Software
- Wissen in machine learning Verfahren, insb. aktueller Deep Learning Verfahren
- Umsetzung

Fertigkeiten

- Aufbau einfacher Systeme
- Durchführung nachvollziehbaren SW Entwicklung mit Versionskontrollsystemen
- Quantifizierung und Verbesserung der Qualität von ML basierten SW Systemen durch systematisches Testen

Kompetenz

- Verstehen alle Bestandteile und Funktionen eine ML basierten SW Systems
- Können in einem systematischen Prozesse, ML basierte SW Systeme für konkrete Fragestellungen entwerfen und technisch umsetzen
- Können nach Zielvorgaben ML basierte SW Systeme aufbauen und betreiben
- Sind in der Lage die Qualität von ML basierte SW Systemen sicherzustellen

Inhalte

Die Inhalte sind aufgeteilt in

1. Spezielle Online Kurse, die technische Kenntnisse und Fertigkeiten vermitteln.
2. Systementwicklungen von ML basierten SW Systemen mit Online Services, die den Kompetenzerwerb fördern und nachweisen.

Im ersten Teil nehmen die Studierenden an den folgenden externen Kursen teil:

- Python – Einführung für absolute Anfänger (engl.)
- Versionskontrolle mit git und github (engl.)
- Udemy Kurs:
    - „Python für Data Science, Maschinelles Lernen & Visualization
      Data Science Grundlagen mit Python! Von Daten Analysen bis zum Machine Learning.“

Im zweiten Teil entwickeln Studierenden konkrete ML basierte SW Systeme. Die Inhalte sind

- Prozesse für den Aufbau und Betrieb von ML basierten SW Systemen
- Reproduzierbarkeit von Entwurf und Implementierung m.H. Versionierung
- Implementierung von ML basierten SW Systemen
- Betrieb und Ablauf von ML basierter Software transparent gestalten
- Automatisches Testen von ML basierten SW Systemen

Prüfungsleistung

Der Nachweis der erfolgreichen Teilnahme am Kurs erfolgt durch die Erstellung und Betrieb funktionsfähiger ML basierter SW Systeme. Die Prüfung umfasst mehrere Aufgaben, die unmittelbar auf dem Udemy Online Kurs aufbauen.

Umfang und Arbeitsaufwand

Annahme ist, dass Studierende bereits ein Grundwissen in der Programmierung haben. Daher wird der Aufwand für den Einstieg in Python nicht berücksichtigt. Der Udemy Kurs enthält ebenfalls eine Python Einführung. Der Umfang wird dort eingeordnet. Der Udacity Kurs in Git ist optional. Erfahrungswerte zeigen, dass Studierende ein Grundverständnis von Versionsverwaltung haben.

Aktivität	**Umfang und Arbeitsaufwand**
Udemy Kurs: Python für Data Science, Maschinelles Lernen & Visualization	3 SWS, 6 ECTS
Prüfungsaufgabe 1	0,5 SWS, 2 ECTS
Prüfungsaufgabe 2	0,5 SWS, 2 ECTS
Summe:	**4 SWS, 10 ECTS**
Optional: Udacity Kurs: https://www.udacity.com/course/version-control-with-git--ud123	1 ECTS

Konkrete Aufgaben

Nehmen Sie an den folgenden Online Kursen teil.

Python

Wenn Sie kein Vorwissen in Python haben, können Sie einen der beiden Kurse für eine Einführung in die Python Programmierung besuchen

- Python tutorial for beginners
  "Learn Python for machine learning and web development."
  https://www.youtube.com/watch?v=_uQrJ0TkZlc
- Python in 4 hours
  "This course will give you a full introduction into all of the core concepts in python. Follow along with the videos and you'll be a python programmer in no time!"
  https://www.youtube.com/watch?v=rfscVS0vtbw

Als Entwicklungsumgebung können Sie ohne Installation folgende Services nutzen:

- Online compiler:
  "Online Python compiler, Online Python IDE, and online Python REPL. Code Python, compile Python, run Python, and host your programs and apps online for free."
  https://repl.it/languages/python3
- Jupyter: "*JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data."*
  Run JupyterLab Online: https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/try.jupyter.org?urlpath=lab

Es wird empfohlen die JupyterLab Notebooks zu nutzen, da diese Notebooks auch die Grundlage für den Udemy Machine Learning Kurs sind.

Git

Git ist eine freie Software zur verteilten Versionsverwaltung von Dateien.

Absolvieren Sie den folgenden (freien) Kurs über git bei udacity.

- Version Control with Git
  https://www.udacity.com/course/version-control-with-git--ud123

Machine Learning

Absolvieren Sie folgenden Online-Kurs bei Udemy. Sie erhalten vom Dozenten eine Einladung via E-Mail. Legen Sie sich einen Account bei Udemy an, um mit dem Link aus der E-Mail den Kurs zu starten.

- Python für Data Science, Maschinelles Lernen & Visualization
  https://www.udemy.com/course/python-data-science-machine-learning/

Prüfungsaufgabe 1

Eine Aufgabe gliedert sich in nachzuweisende Lernziele, Teilaufgaben und den Erfolgsnachweis.

Ziele

- Aufbau und Betrieb beispielhafter ML basierter SW Systeme
- Reproduzierbarkeit von Entwurf und Implementierung m.H. Versionierung

Teilaufgaben

Nutzen Sie die Plattform myBinder (https://mybinder.org/), um jeweils eine Übungsaufgabe / Projekt aus den folgenden Bereichen des Kurses „Python für Data Science, Maschinelles Lernen & Visualization“ zu reproduzieren.

- Logistic Regression
- Decision Tree
- K Means Clustering
- Recomender Systems
- Natural Language Processing
- Deep Learning

myBinder nutzt GitHub. Legen Sie einen Account an und erstellen Sie für jede Übungsaufgabe ein Repository. Sie sollten insgesamt 6 Repositories, jeweils eins pro Bereich aus der obigen Liste, angelegt haben. Für die Reproduktion können Sie das jupyter Notebook der Musterlösung der jeweiligen Übungsaufgabe aus dem Kurs in Ihr jeweiliges Repository kopieren.

Jedes Repository enthält

- Übungsaufgabe, z.B. die Kopie des jupyter Notebooks mit der Musterlösung
- Daten für die Ausführung der Übungsaufgabe
- Notwendige Dateien für myBinder
- Ihre Dokumentation

Jedes Repository soll in der Datei **README.md** wie folgt dokumentiert werden

- Name des Projektes
- Binder Badge zum Starten der Binder Umgebung mit der Übungsaufgabe
- Kurze Doku, wie ein Beispiel der Übungsaufgabe auszuführen ist und was als Ergebnis zu erwarten ist.

Erfolgsnachweis und Bewertung

Für den Erfolgt der Zielerfüllung wird bewertet

- Übungsbeispiel bzw. Projekt kann erfolgreich via myBinder aus dem Repository mittels der Binder Badge gestartet werden
- Übungsbeispiel bzw. Projekt kann ausgeführt werden
- gemäß der eigenen Dokumentation kann das Übungsbeispiel bzw. Projekt mit dem erwarteten Ergebnis aus der Dokumentation reproduziert werden

Die Aufgabe ist erfolgreich gelöst, wenn mindestens 3 Übungsbeispiele die obigen Anforderungen an die Bewertung jeweils vollständig erfüllen.

Prüfungsaufgabe 2

In der vorherigen Aufgabe wurde die Erfolgsmessung durch Ausführungen und Vergleich der Ergebnisse manuell durchgeführt. In dieser Aufgabe soll dieses Vorgehen automatisiert werden.

Ziele

- Implementierung von ML basierten SW Systemen
- Betrieb und Ablauf von ML basierter Software transparent gestalten
- Automatisches Testen von ML basierten SW Systemen

Teilaufgaben

Lesen und verstehen Sie den folgenden Artikel über das Logging und einen einfachen Input – Output unit test Ansatz ML basierter SW Systeme.

- https://towardsdatascience.com/unit-testing-and-logging-for-data-science-d7fb8fd5d217

Ihre Aufgabe ist, den Ansatz auf ein Übungsbeispiel / Projekt des obigen Python Kurses zu übertragen. Gehen Sie wie folgt vor:

Wählen Sie ein Übungsbeispiel aus dem obigen Python-Kurs, z.B. das DeepLearning Übungsbeispiel mit dem MNIST Datensatz. In der vorherigen Aufgabe haben Sie ggf. dieses Beispiel auf myBinder erfolgreich ausführen können.

Legen Sie ein *neues* Git Repository für diese Aufgabe an. Prüfen Sie, dass Sie das Beispiel via myBinder erfolgreich ausführen können.

Implementieren Sie nun die my_logger und my_timer Funktionen aus dem obigen Artikel und wenden Sie sie auf von mindestens eine von Ihnen gewählte Funktion des Beispiels an. Dokumentieren Sie Ihr Ergebnis in der README.md Datei.

Mittels der Funktionen sollen nun zwei Testfälle implementiert werden; einmal für die Vorhersagefunktion des Modells, predict(), und einmal für die Trainingsfunktion des Modells, fit(). Schreiben Sie folgende zwei Testfälle.

1. Testfall: Wählen Sie geeignete Indikatoren, die Ihnen anzeigen, dass die Vorhersagefunktion predict() des Modells korrekt funktioniert. Im Artikel nutzt der Autor die Indikatoren Accuracy und Confusion Matrix. Schreiben Sie für predict() einen Testfall auf ausgewählten Testdaten. Die Testdaten legen Sie in einem Testdatenfile ab.
2. Testfall: Überprüfen Sie, dass das System innerhalb normaler Parameter läuft, indem Sie die Laufzeit der Trainingsfunktion fit() testen. Loggen Sie dazu eine repräsentative Laufzeit mit dem obigen Wrapper. In dem Testfall überprüfen Sie, dass die Laufzeit der Trainingsfunktion während der Testfallausführung einen Grenzwert, z.B. 120% der repräsentativen Laufzeit, nicht überschreitet.

Lassen Sie beide Testfälle ausführen und dokumentieren Sie die Bildschirmausgabe in der README.md. Dokumentieren Sie wie ein Nutzer die Testfälle mit dem Testdatenfile ausführen kann.

Erfolgsnachweis und Bewertung

Die Aufgabe ist erfolgreich gelöst, wenn folgende Punkte der Zielerfüllung nachgewiesen sind.

- Dokumentation der Bildschirmausgabe wie oben beschrieben
- Der Prüfer des Kurses kann beide Testfälle m.H. Ihrer Dokumentation und dem Testdatenfile erfolgreich über mybinder ausführen.