

Vorbereitung für das Seminar „Maschinelles Lernen: Potentiale und Risiken“

Marvin Kastner <marvin.kastner@tuhh.de>

8. November 2022

In diesem Dokument werden die Schritte aufgezählt, die von den Seminarteilnehmenden selbständig *vor* Beginn des Seminars auf dem eigenen Endgerät (PC, Laptop, Apple, ...) durchgeführt werden müssen. Dies ist notwendig, weil zur Einrichtung der Arbeitsumgebung große Datenmengen heruntergeladen und verarbeitet werden müssen. Bitte planen Sie genügend Zeit hierfür ein. Falls Sie während des Seminars kein eigenes leistungsfähiges Endgerät (z. B. nur einen alten Laptop oder ein Tablet) zur Verfügung haben sollten, sprechen Sie bitte frühzeitig die Seminarleitung an.

Im Folgenden wird aufgezählt, welche Software und welche Dateien benötigt werden. Hier gibt es nur manchmal eine zeitliche Abhängigkeit, schauen Sie also gerne schon etwas voraus und parallelisieren Sie Aufgaben da, wo dies den Prozess beschleunigt. Die Absätze, die mit einer Frage beginnen, beschreiben, warum diese Schritte gemacht werden und zeigt manchmal Alternativen auf.

Installation von JupyterLab

Gehen Sie auf <https://www.anaconda.com/distribution/> und laden Sie die Anaconda-Version für Ihr Betriebssystem herunter. Anaconda ist eine Python-Distribution, die vorkompilierte Bibliotheken ausliefert und damit den Aufwand beim Installieren von Bibliotheken minimiert. Der Download-Bereich der Webseite sollte ungefähr wie in Abbildung 1 aussehen.

Falls Sie auf Ihrem Endgerät zwei Accounts, einen Administrator- bzw. root-Account und einen Account fürs alltägliche Arbeiten, verwenden, seien Sie bitte vorsichtig. Die Installation von Anaconda erfordert keine erhöhten Rechte. Eine Installation mit einem Nutzer mit erhöhten Rechten (z. B. *root* oder einem dedizierten Admin-Account unter Windows) kann u. U. dazu führen, dass Anaconda nur für diesen einen Nutzer mit erhöhten Rechten installiert ist bzw. dass es zu Problemen mit der Rechteverwaltung kommt. In der Vergangenheit hat ein erneutes Installieren von Anaconda dieses Problem dann *nicht* beheben können. Lösungen hierfür lassen sich unter <https://docs.anaconda.com/anaconda/install/multi-user/> finden.

Rufen Sie nach dem Download den Installer auf und folgen Sie den Installationsschritten. Konsultieren Sie im Fehlerfall offizielle Quellen des Herstellers (wie z. B. <https://docs.anaconda.com/anaconda/install/>) oder Foren (wie z. B. <https://stackoverflow.com>).



Abbildung 1: Der Download-Bereich von Anaconda (Ausschnitt).

WARUM WERDEN JUPYTER NOTEBOOKS EINGESETZT? Im Bereich Maschinelles Lernen und Data Science spielen Jupyter Notebooks eine immer größere Rolle. Das Medienformat ist auf JSON-Basis und kann u. a. Text (d. h. Plaintext, Markdown, LaTeX und HTML), Bilder und ausführbaren Code enthalten. So können die Arbeitsschritte für Dritte nachvollziehbar dokumentiert und die dazugehörigen Konzepte in einem Dokument zentral erklärt werden.

Bezug der Seminar-Materialien

Klonen Sie das git-Repository <https://github.com/1kastner/ml-potentials-and-risks>, damit Sie die Seminar-Materialien lokal haben – dies umfasst nur den Programmiereteil des Seminars. Am einfachsten ist es, wenn Sie die Dateien lokal unterhalb des benutzer-eigenen Ordners Dokumente ablegen (unter Windows typischerweise `C:\Users\BENUTZERNAME\Documents`). Denn dieser Ordner ist, wenn JupyterLab startet, direkt verfügbar. Falls Sie noch nie mit git gearbeitet haben, lesen Sie bitte die nächsten Absätze.

WARUM SOLLTE ICH GIT LERNEN? Für die Versionsverwaltung ist git quasi der Standard und wird immer häufiger auch außerhalb der Software-Entwicklung, aus der git ursprünglich stammt, eingesetzt. Deswegen lohnt es sich für (fast) alle, sich Fähigkeiten mit diesem Tool anzueignen. Es ist als ein Kommandozeilentool entwickelt worden, welches über <https://git-scm.com/> heruntergeladen werden kann. Wer lieber grafische Oberflächen mag, kann sich eine von vielen GUI-Clients¹ aussuchen. Hier sollte neben dem Betriebssystem auch die ggf. kostenpflichtige Lizenz beachtet werden. Einige Lizenzen unterscheiden z. B. zwischen der privaten Verwendung und der Verwendung im Arbeitskontext. Bis zum Start des Seminars werden u. U. die Materialien noch überarbeitet oder erweitert. Aktualisieren Sie also bitte regelmäßig Ihre vorliegende Version über ein `git pull` bzw. durch das Klicken auf den „Pull“-Button im GUI-Client Ihrer Wahl.

IST ES FÜR DAS SEMINAR ZWINGEND NOTWENDIG, GIT ZU LERNEN? Falls Ihnen git unbekannt ist und Sie keine Zeit dafür haben, sich mit git auseinanderzusetzen, gibt es auch die Möglichkeit, den Inhalt als ZIP-Ordner herunterzuladen. Klicken Sie dafür auf den Button, wie er in Abbildung 2 zu sehen ist. Falls Lernmaterialien später noch angepasst werden, müssen Sie diese dann allerdings erneut herunterladen und in einem neuen Ordner entpacken.

¹ Eine Liste ist auf <https://git-scm.com/download/gui/win> zu finden.

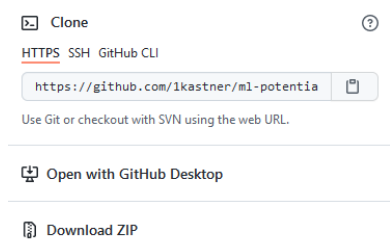


Abbildung 2: Ein GitHub-Repository bietet verschiedene Möglichkeiten zum Bezug der Inhalte an, auch den Download als ZIP-Datei.

Installation der Bibliotheken

Mit der Installation von Anaconda ist u. a. die Applikation Anaconda Navigator ausgeliefert worden. Mit dem Anaconda Navigator können die benötigten Bibliotheken automatisch installiert werden. In der Abbildung 3 sehen Sie unten den Button „Import“ (im Screenshot mit einer 1 markiert). Klicken Sie diesen an. Damit öffnet sich das Fenster „Import environment“. Klicken Sie hier auf den Ordner in der Zeile, die mit „Local drive“ bezeichnet wird (mit einer 2 markiert). Danach öffnet sich ein Fenster zur Datei-Auswahl (mit einer 3 markiert), in dem Sie dann zu den von GitHub bezogenen Dateien navigieren können. Wählen Sie die Datei `environment.yml` aus – sie liegt auf der Wurzelebene des Projektordners – und klicken Sie auf „Öffnen“. Der zu erstellenden Umgebung können Sie nun einen sprechenden Namen geben, z. B. `ml-potentials-and-risks`. Unter diesem Namen werden Sie die Umgebung später in Anaconda wiederfinden. Das Erstellen der Umgebung beinhaltet den Download und die Installation der benötigten Bibliotheken; es nimmt für gewöhnlich mehrere Minuten in Anspruch.

Falls Sie Windows 10 verwenden und kein Multi-User-Setup haben, können Sie den folgenden Schritt dadurch abkürzen, dass Sie die Batch-Datei `create-env.bat` auf der Wurzelebene des Projekts durch einen Doppelklick ausführen.

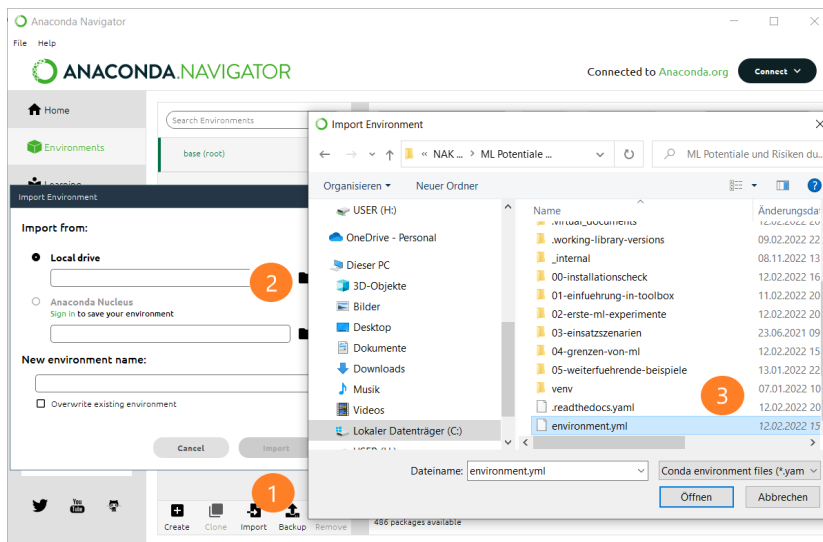


Abbildung 3: Der Anaconda Navigator erlaubt das Importieren von `environment.yml`-Dateien.

WARUM IST DIESER SCHRITT NOTWENDIG? Die Jupyter Notebooks, die Sie soeben von GitHub bezogen haben, benötigen eine bestimmte Umgebung, damit sie wie gewünscht geöffnet und ausgeführt werden können. Zu dieser Umgebung gehört z. B. ein Python-Interpreter mit einer bestimmten Python-Version ebenso wie eine Reihe ausgewählter Python-Bibliotheken. Auf der obersten Ebene des git-Repositorys befindet sich die Datei `environment.yml`, in der alle

Anforderungen an die Entwicklungsumgebung aufgeführt werden. Die Struktur der Datei `environment.yml` ist von Anaconda vorgegeben und erlaubt es, die Abhängigkeiten von Bibliotheken automatisch aufzulösen. Damit man auf einem Endgerät in verschiedenen Projekten unterschiedliche Versionen einer gleichen Bibliothek (und auch JupyterLab und andere Werkzeuge) haben kann, strukturiert Anaconda die zu einem Projekt gehörenden Bibliotheken standardmäßig in Umgebungen (eng. Environments). Für das Seminar erstellen wir die Umgebung `ml-potentials-and-risks` basierend auf der gegebenen `environment.yml`.

GEHT ES DENN NUR ÜBER DIE GUI? Natürlich gibt es auch ein Kommandozeilentool, das mit Anaconda ausgeliefert worden ist. Es heißt `conda`. Unter Windows wird dieses Tool je nach Auswahl während der Installation nicht in die Pfad-Variable aufgenommen. Es steht Ihnen aber auf jeden Fall in der *Anaconda Powershell Prompt* (basierend auf der PowerShell) zur Verfügung. Auf <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#creating-an-environment-from-an-environment-yml-file> wird erläutert, wie eine existierende `environment.yml` zur Erstellung einer Anaconda-Umgebung genutzt werden kann.

Start von JupyterLab

JupyterLab kann nun über den Anaconda Navigator gestartet werden. In Abbildung 5 ist dies abgebildet. Zunächst wird links im Menü (im Screenshot mit einer 1 markiert) „Home“ ausgewählt. Im zweiten Schritt muss die Umgebung `ml-potentials-and-risks` ausgewählt werden (mit einer 2 markiert). Danach startet ein Klick auf Launch die Anwendung JupyterLab im Browser (mit einer 3 markiert). Standardmäßig öffnet sich nun ein neuer Tab im Browser und JupyterLab zeigt zunächst den Inhalt vom Benutzer-Ordner an (unter Windows typischerweise `C:\Users\BENUTZERNAME`). Von hier können Sie in den Unterordner Dokumente navigieren und zu den heruntergeladenen Seminarunterlagen gelangen.

Falls Sie Windows 10 verwenden und kein Multi-User-Setup haben, können Sie den folgenden Schritt dadurch abkürzen, dass Sie die Batch-Datei `start-jupyterlab.bat` auf der Wurzelebene des Projekts durch einen Doppelklick ausführen. Dann ist auch die Wurzelebene in JupyterLab der entsprechende Ordner.

Erste Schritte mit JupyterLab

Wenn Sie die Seminar-Materialien wie unter Abschnitt „Bezug der Seminar-Materialien“ angegeben abgelegt haben, können Sie in JupyterLab zum Ordner `00-installationscheck` navigieren und dort das Jupyter Notebook (die Datei mit der Endung `.ipynb`) mit einem Doppelklick öffnen. Überprüfen Sie, ob Sie alle Zellen ausführen können. Falls es Fehlermeldungen gibt, melden Sie sich bitte vor Beginn der

Wenn Sie nicht `start-jupyterlab.bat` zum Starten von JupyterLab verwendet haben, können Sie den Inhalt aus dem Ordner `.user-settings` nach `%USERPROFILE%\jupyter\lab\user-settings` kopieren, um die gleichen Voreinstellungen zu verwenden. Falls Sie Apple oder Linux verwenden, ist dieser Pfad entsprechend `~/.jupyter/lab/user-settings`.

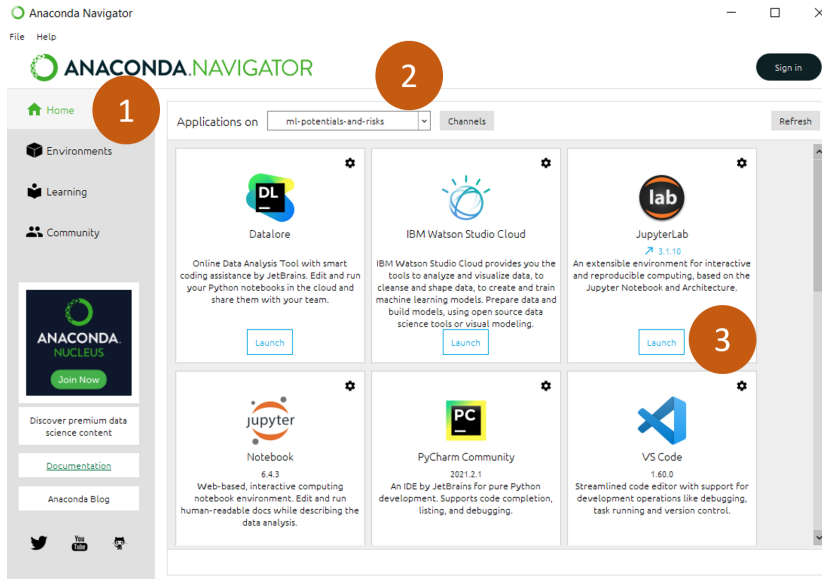


Abbildung 4: Aus dem Anaconda Navigator kann JupyterLab gleich in der richtigen Umgebung gestartet werden. Die Reihenfolge und Sortierung der Logos kann variieren.

Veranstaltung per Mail bei der Seminarleitung.

Bitte schauen Sie ebenfalls im Ordner 01-einfuehrung-in-toolbox um. Falls Sie noch keine Erfahrungen mit Python haben, arbeiten Sie bitte das Jupyter Notebook 01-Einfuehrung-in-Python.ipynb durch. Falls Ihnen das Format des Jupyter Notebooks noch unbekannt sind, schauen Sie sich bitte das Jupyter Notebook 02-Was-kann-ein-Jupyter-Notebook.ipynb an. Falls Sie noch nicht mit der Anwendung JupyterLab gearbeitet haben, könnte das Jupyter Notebook 03-Tipps-zu-JupyterLab.ipynb für Sie interessant sein.

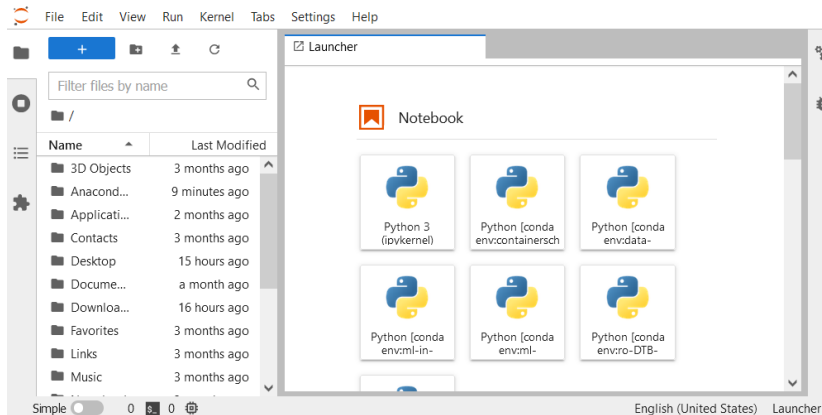


Abbildung 5: In JupyterLab können links alle Dateien und Ordner aus dem Benutzerordner betrachtet werden.

Dieses Werk ist lizenziert unter einer Creative Commons „Namensnennung 4.0 International“ Lizenz.

