Vorbereitung für das Seminar „Einführung ins Maschinelle Lernen: Hype oder Hybris“

Marvin Kastner <marvin.kastner@tuhh.de>

28. März 2021

In diesem Dokument werden die Schritte aufgezählt, die von den Seminarteilnehmern selbständig *vor* Beginn des Seminars auf dem eigenen Laptop durchgeführt werden müssen. Dies ist notwendig, weil zur Einrichtung der Arbeitsumgebung mehrere Gigabyte Daten heruntergeladen und verarbeitet werden. Falls Sie während des Seminars keinen eigenen leistungsfähigen Laptop 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. Der Text drumherum beschreibt, warum diese Schritte gemacht werden und zeigt ein paar Alternativen auf.

Installation von JupyterLab

Gehen Sie auf die Webseite `https://www.anaconda.com/distribution/` und laden Sie die Anaconda-Version für Ihr Betriebssystem herunter. Anaconda ist eine Python-Distribution, die die Entwicklungsumgebung *JupyterLab*, welche im Workshop verwendet wird, bereits mit ausliefert. Der Download-Bereich der Webseite sollte ungefähr wie in Abbildung 1 aussehen.

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

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 *admin*) 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.

Folgen Sie den Installationsschritten und konsultieren Sie im Fehlerfall offizielle Quellen (wie z. B. `https://docs.anaconda.com/anaconda/install/`) oder Foren.

Warum werden Jupyter Notebooks eingesetzt? Im Bereich Maschinelles Lernen und Data Science spielen Jupyter Notebooks eine immer größere Rolle[1]. Hierbei bezeichnen Jupyter Notebooks

[1] Jeffrey M. Perkel. Why jupyter is data scientists' computational notebook of choice, 2018. URL `https://www.nature.com/articles/d41586-018-07196-1`; and Or Hiltch. Jupyter notebook is the cancer of ML engineering, 2019. URL `https://medium.com/@_orcaman/jupyter-notebook-is-the-cancer-of-ml-engineering-70b98685ee71`

gleichzeitig das Medienformat als auch die Standard-IDE. Das Medienformat ist auf JSON-Basis und enthält u. a. Text (mit Unterstützung für HMTL), Bilder, ausführbaren Code und den persistierten Outpt ebendieses. Die gleichnamige IDE kann Jupyter Notebooks öffnen, den enthaltenen Code ausführen, die vom Code erzeugten Visualisierungen nah an diesem platziert anzeigen und das Ergebnis auch wieder speichern. Die Seminarleitung empfiehlt neuen Benutzern die Verwendung der Standard-IDE Jupyter Notebook für das Bearbeiten der Aufgaben. 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`).

Bezug der Seminar-Materialien

Klonen Sie das git-Repository `https://github.com/1kastner/machine-learning-hype-or-hybris`, damit Sie die Seminar-Materialien lokal haben – dies umfasst nur den Programmierteil des Seminars. Am einfachsten ist es, wenn Sie die Dateien lokal unterhalb des Ordners `Eigene Dateien` ablegen. Denn in diesem Ordner öffnet sich standardmäßig JupyterLab. 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 sie ursprünglich stammt, eingesetzt. Deswegen lohnt es sich für (fast) jeden, sich Fähigkeiten mit diesem Tool anzueignen. Das Original-Tool ist ein Kommandozeilentool, welches über `https://git-scm.com/` heruntergeladen werden kann. Wer lieber grafische Oberflächen mag, kann sich eine von vielen GUI ClientsDas Versionierungssystem `git` ist ursprünglich 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-Clients[2] 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 Workshops 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 Button „Pull“ im GUI-Client Ihrer Wahl.

[2] Eine Liste ist auf `https://git-scm.com/download/gui/win` zu finden.

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.

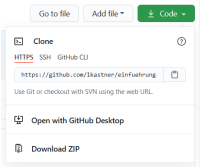

Abbildung 2: Ein GitHub-Repository bietet verschiedene Möglichkeiten zum Bezug der Inhalte an.

Installation der Bibliotheken

Mit dem Anaconda Navigator können die im Workshop später 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 new environment“. Klicken Sie hier auf den Ordner in der Zeile, die mit „Specification File“ beginnt (mit einer (2) markiert). Danach öffnet sich ein Fenster (mit einer (3) markiert), in dem Sie dann zu den heruntergeladenen Dateien navigieren können. Wählen Sie die `environment.yml` aus den soeben bezogenen Dateien aus (sie liegt auf der obersten Ebene des Projektordners) und schließen den Vorgang ab. Das Erstellen der Umgebung nimmt für gewöhnlich einige Zeit in Anspruch.

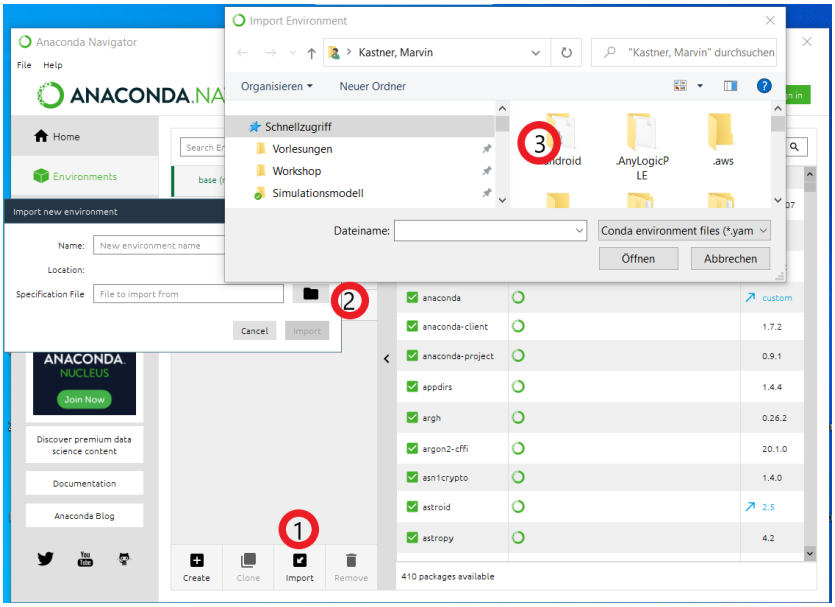

Abbildung 3: Der Anaconda Navigator erlaubt das Importieren von `environment.yml`-Dateien.

Warum ist dieser Schritt notwendig? Die Jupyter Notebooks, die Sie soeben heruntergeladen haben, benötigen spezielle Bibliotheken. Einige werden bereits bei der Installation mit ausgeliefert, ande-

re müssen aber noch nachinstalliert werden. Auf der obersten Ebene des git-Repositorys befindet sich die Datei `environment.yml`, in der alle benötigten Bibliotheken aufgelistet werden. Diese Datei sollten Sie nun bereits auf Ihrem Endgerät in einem lokalen Ordner vorliegen haben. Die Struktur der Datei `environment.yml` ist von Anaconda vorgegeben und erlaubt es, die Abhängigkeiten von Bibliotheken automatisch aufzulösen. Damit man an einem PC in verschiedenen Projekten unterschiedliche Versionen einer gleichen Bibliothek haben kann, strukturiert Anaconda die zu einem Projekt gehörenden Bibliotheken standardmäßig in Umgebungen (eng. Environments). Für das Seminar erstellen wir die Umgebung `intro-jupyter-notebooks` basierend auf der gegebenen `environment.yml`.

ABER IST ES DENN NOTWENDIG, HIER MIT EINER GUI ZU ARBEITEN? Natürlich gibt es auch ein Kommandozeilentool, das bereits mit Anaconda ausgeliefert worden ist. Es heißt `conda`. Unter Windows wird dies je nach Auswahl während der Installation nicht in die Pfad-Variable mit aufgenommen. Dann steht Ihnen immer noch die *Anaconda Prompt* zur Verfügung. Weitere Informationen gibt es auf `https://docs.anaconda.com/anaconda/install/verify-install/#conda`. 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` eingelesen werden kann. Folgen Sie der Anleitung und erstellen Sie die passende Umgebung.

Start von JupyterLab

Nun ist JupyterLab einsatzbereit. Es kann nun über den Anaconda Navigator gestartet werden. In Abbildung 5 ist dies abgebildet. Zunächst wird links Menü (im Screenshot mit einer (1) markiert) „Home“ ausgewählt. Im zweiten Schritt muss die Umgebung `intro-jupyter-notebooks` ausgewählt werden (mit einer (2) markiert). Danach startet ein Klicken auf `Launch` JupyterLab im Browser (mit einer (3) markiert). Standardmäßig öffnet sich nun der Browser und dort wird der Inhalt vom Ordner `Eigene Dateien` angezeigt.

Erste Schritte mit JupyterLab

Wenn Sie die Workshop-Materialien wie unter Abschnitt „Bezug der Seminar-Materialien“ angegeben unterhalb des Ordners `Eigene Dateien` 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.

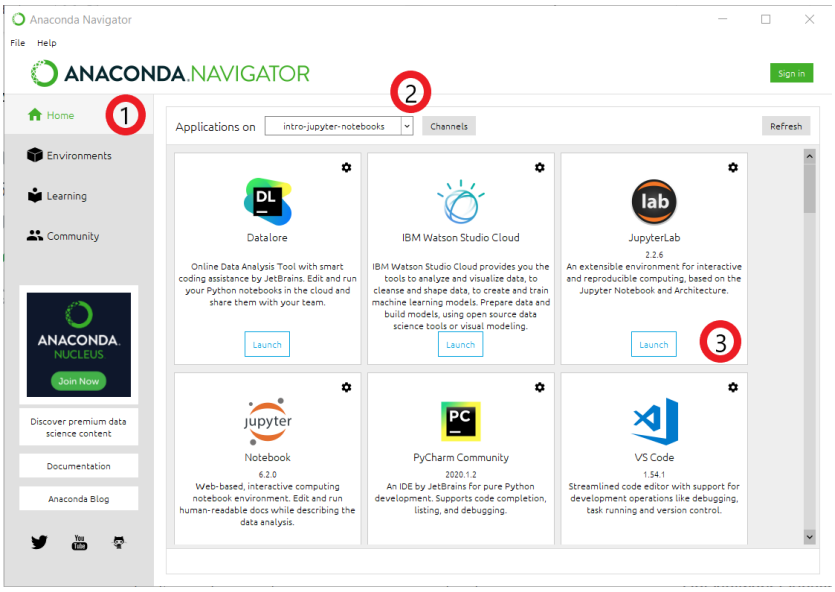

Abbildung 4: Aus dem Anaconda Navigator kann JupyterLab gleich in der richtigen Umgebung gestartet werden.

und dort überprüfen, ob Sie alle Zellen ausführen können. Falls es Fehlermeldungen gibt, melden Sie sich gerne bereits im Vorwege an die Workshop-Leitung. Falls Sie noch keine Erfahrungen mit Python haben und Sie Interesse haben, sich in diese Sprache einzudenken, können Sie sich die Einführungsmaterialien unter `01-Einfuehrung-in-Python` anschauen.

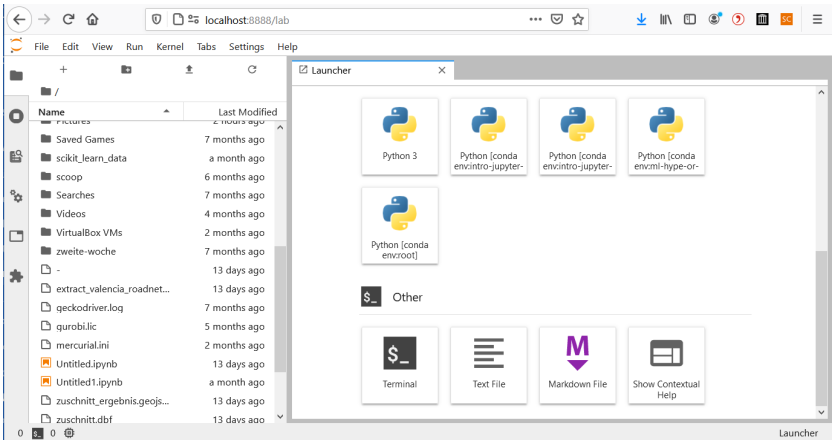

Abbildung 5: In JupyterLab können links alle Dateien und Ordner aus dem Ordner *Eigene Dateien* betrachtet werden.

Literatur

Or Hiltch. Jupyter notebook is the cancer of ML engineering, 2019. URL `https://medium.com/@_orcaman/jupyter-notebook-is-the-c`

ancer-of-ml-engineering-70b98685ee71.

Jeffrey M. Perkel. Why jupyter is data scientists' computational notebook of choice, 2018. URL https://www.nature.com/articles/d41586-018-07196-1.