

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

Marvin Kastner

16. Mai 2020

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 der IDE Jupyter Notebook

Gehen Sie auf die Webseite <https://www.anaconda.com/distribution/#download-section> und laden Sie die Anaconda-Version für Python 3 herunter. Anaconda ist eine Python-Distribution, die die IDE *Jupyter Notebook* bereits mit ausliefert. Der Download-Bereich der Webseite sollte ungefähr wie in Abbildung 1 aussehen.

Falls Sie auf Ihrem PC zwei Accounts, einen Administrator- bzw. root-Account und einen Account fürs alltägliche Arbeiten, verwenden, seien Sie bitte vorsichtig. Die Installation von Anaconda erfordert keine Administratorenrechte. Eine Installation mit Administratorenrechten kann u. U. dazu führen, dass Anaconda nur für den Administrator 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<sup>1</sup>. Hierbei bezeichnen Jupyter Notebooks gleichzeitig das Medienformat als auch die Standard-IDE. Das Medienformat ist auf JSON-Basis und enthält u. a. Text (inkl. Links etc.),

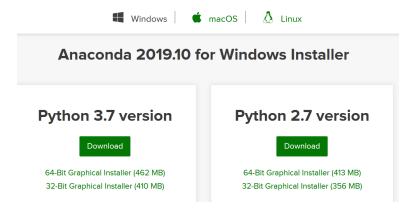


Abbildung 1: Der Download-Bereich von anaconda.

<sup>1</sup> 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>

Bilder und ausführbaren Code. 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.

### *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 Programmieranteil. Am einfachsten ist es, wenn Sie die Dateien lokal unterhalb des Ordners Eigene Dateien ablegen. Denn in diesem Ordner öffnet sich standardmäßig die IDE Jupyter Notebook. 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 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 Visualisierungen mag und klicken können möchte, kann sich eine von vielen GUI Clients<sup>2</sup> aussuchen. Hier sollte natürlich neben dem Betriebssystem auch die Lizenz beachtet werden, falls man den Client später noch mal auf der Arbeit einsetzen möchte. Bis zum Start des Seminars werden u. U. die Materialien noch überarbeitet oder erweitert. Aktualisieren Sie also bitte regelmäßig Ihre vorliegende Version.

<sup>2</sup> 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, ist das Zusammenführen der eigenen und der neuen Version allerdings schwieriger.

### *Installation der Bibliotheken*

Mit dem Anaconda Navigator können die Bibliotheken automatisch installiert werden. In der Abbildung 3 sehen Sie unten den Button „Import“. Wählen Sie die `environment.yml` aus und schließen den Vorgang ab.

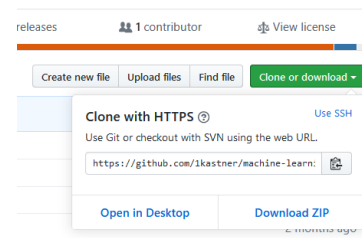


Abbildung 2: Ein github-Repository bietet verschiedene Möglichkeiten zum Bezug der Inhalte an.

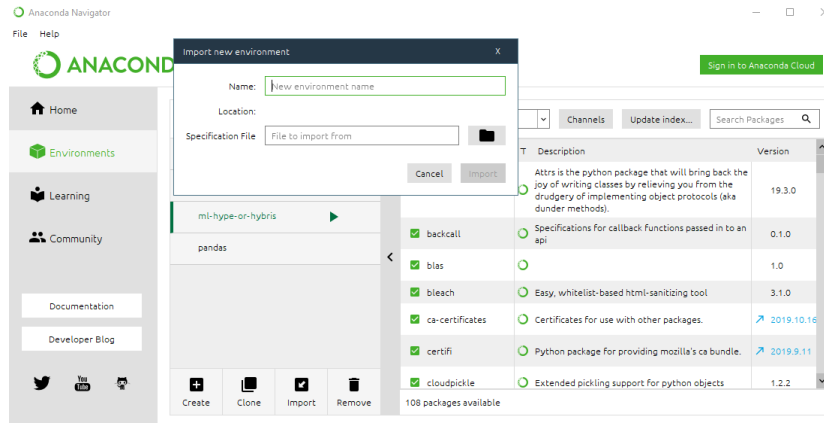


Abbildung 3: Der Anaconda Navigator erlaubt das Importieren von `environment.yml`-Dateien.

WARUM IST DIESER SCHRITT NOTWENDIG? Die Jupyter Notebooks, die Sie soeben heruntergeladen haben, benötigen jeweils spezielle Bibliotheken. Einige werden bereits bei der Installation mit Anaconda bereitgestellt, allerdings stehen für das Seminar noch Downloads von weiteren Bibliotheken aus. Auf der obersten Ebene des git-Repositorys befindet sich die Datei `environment.yml`, in der alle benötigten Bibliotheken aufgelistet werden. Diese sollten Sie nun bereits auf dem PC in einem lokalen Ordner Ihrer Wahl vorliegen haben. Das Format stammt von Anaconda 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 Bibliotheken standardmäßig in Umgebungen (eng. *environments*.) Für das Seminar erstellen wir die Umgebung `ml-hype-or-hybris` 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 installiert 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 der IDE Jupyter Notebook

Die IDE Jupyter Notebook ist nun einsatzbereit. Sie kann nun z. B. über den Anaconda Navigator gestartet werden. In Abbildung 4 ist dies abgebildet. Zunächst muss die Umgebung ml-hype-or-hybris ausgewählt werden. Danach startet ein Klicken auf Launch die IDE.

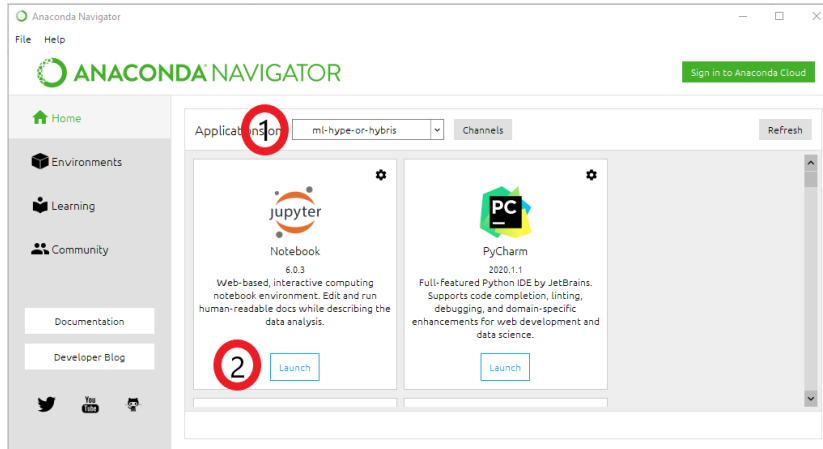


Abbildung 4: Aus dem Anaconda Navigator kann ein Jupyter Notebook gleich mit der richtigen Umgebung gestartet werden.

## Literatur

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](https://medium.com/@_orcaman/jupyter-notebook-is-the-cancer-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>.