

ABSCHNITT 7

DIE HÄNDE SCHMUTZIG MACHEN

EINE PROGRAMMIERÜBUNG ZU ML IN PYTHON

FÜR DEN REST DES NACHMITTAGES

Wählt eine von drei Programmierübungen und fangt an, euch mit Machine Learning in Python vertraut zu machen. Ihr könnt in Kleingruppen (zwei bis drei Personen) zusammenarbeiten, um euch gegenseitig zu unterstützen.

ÜBUNG 1

Survival on the Titanic

ML Basics
die klassische Data
Science Pipeline
Logistische Regression &
Random Forest

ÜBUNG 2

Bags of Popcorn

Kaggle Tutorial
klassisches NLP
Word Embeddings
(Word2Vec)
Sentiment Analysis

ÜBUNG 3

Hugging Transformers

Hugging Face Tutorial
Transformers
Transformers verwenden
und fine-tunen

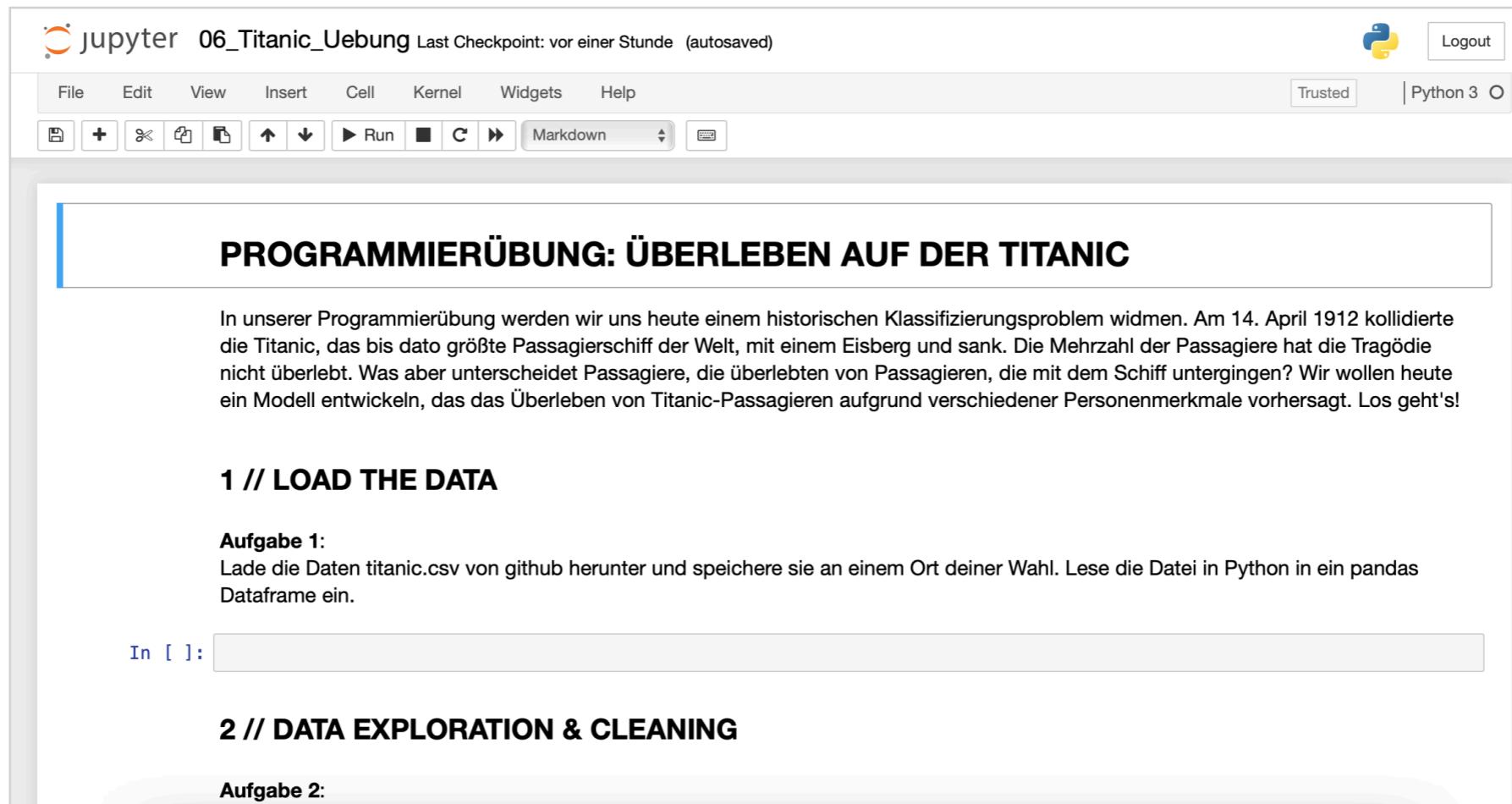


ÜBUNG 1: SURVIVAL ON THE TITANIC

WELCHE MERKMALE BESTIMMEN DAS ÜBERLEBEN VON TITANIC PASSAGIEREN?

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

WELCHE MERKMALE BESTIMMEN DAS ÜBERLEBEN VON TITANIC PASSAGIEREN?



jupyter 06_Titanic_Uebung Last Checkpoint: vor einer Stunde (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

PROGRAMMIERÜBUNG: ÜBERLEBEN AUF DER TITANIC

In unserer Programmierübung werden wir uns heute einem historischen Klassifizierungsproblem widmen. Am 14. April 1912 kollidierte die Titanic, das bis dato größte Passagierschiff der Welt, mit einem Eisberg und sank. Die Mehrzahl der Passagiere hat die Tragödie nicht überlebt. Was aber unterscheidet Passagiere, die überlebten von Passagieren, die mit dem Schiff untergingen? Wir wollen heute ein Modell entwickeln, das das Überleben von Titanic-Passagieren aufgrund verschiedener Personenmerkmale vorhersagt. Los geht's!

1 // LOAD THE DATA

Aufgabe 1:
Lade die Daten titanic.csv von github herunter und speichere sie an einem Ort deiner Wahl. Lese die Datei in Python in ein pandas Dataframe ein.

In []:

2 // DATA EXPLORATION & CLEANING

Aufgabe 2:

WELCHE MERKMALE BESTIMMEN DAS ÜBERLEBEN VON TITANIC PASSAGIEREN?

Bestandteile der Übung:

explorative Datenanalyse & Plotten

fehlende Werte ersetzen

feature engineering

train-test split & cross validation

logistische Regression & Random Forest

Hyperparameter tunen

Modellevaluierung & Vergleich

WELCHE MERKMALE BESTIMMEN DAS ÜBERLEBEN VON TITANIC PASSAGIEREN?

Benötigte Packages:

pandas

matplotlib

seaborn

sklearn

A large pile of popped popcorn is shown in a shallow, round wooden bowl. The popcorn is a mix of white and yellow pieces, some with visible orange centers. The bowl sits on a dark, textured surface, possibly a wooden board or a textured cloth. A few popcorn pieces have spilled out onto the surface around the bowl.

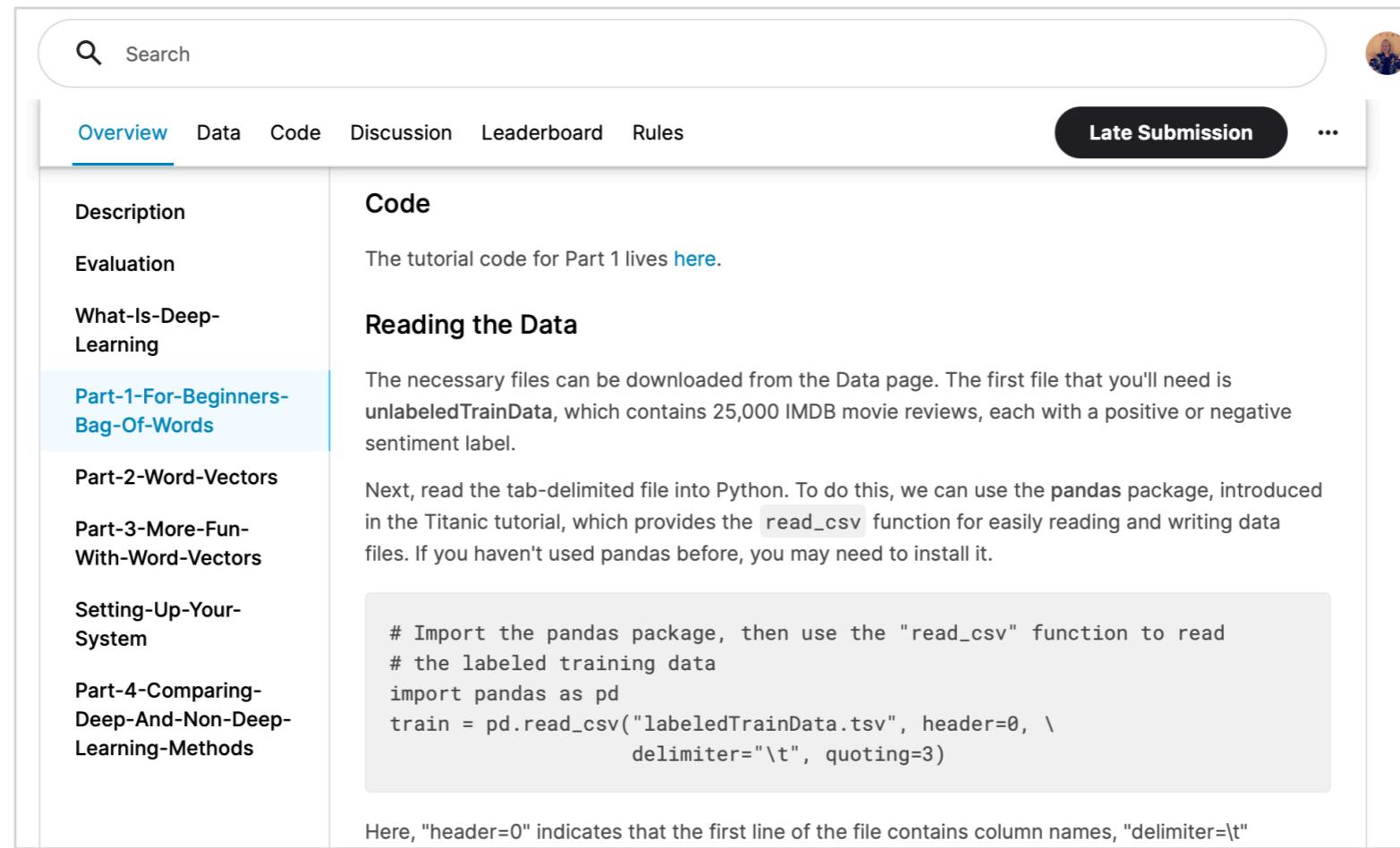
ÜBUNG 2: BAGS OF POPCORN

IST DIE VORLIEGENDE FILMBEWERTUNG POSITIV ODER NEGATIV?

Data fields

- id - Unique ID of each review
- sentiment - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- review - Text of the review

IST DIE VORLIEGENDE FILMBEWERTUNG POSITIV ODER NEGATIV?



The tutorial code for Part 1 lives [here](#).

Reading the Data

The necessary files can be downloaded from the Data page. The first file that you'll need is `unlabeledTrainData`, which contains 25,000 IMDB movie reviews, each with a positive or negative sentiment label.

Next, read the tab-delimited file into Python. To do this, we can use the `pandas` package, introduced in the Titanic tutorial, which provides the `read_csv` function for easily reading and writing data files. If you haven't used pandas before, you may need to install it.

```
# Import the pandas package, then use the "read_csv" function to read
# the labeled training data
import pandas as pd
train = pd.read_csv("labeledTrainData.tsv", header=0, \
                    delimiter="\t", quoting=3)
```

Here, "header=0" indicates that the first line of the file contains column names, "delimiter=\t"

IST DIE VORLIEGENDE FILMBEWERTUNG POSITIV ODER NEGATIV?

Bestandteile der Übung:

Datenreinigung (regular expressions, stop words, ...)

tokenization & bag of words

word embedding training (Word2Vec)

document representations (averaging, centroids)

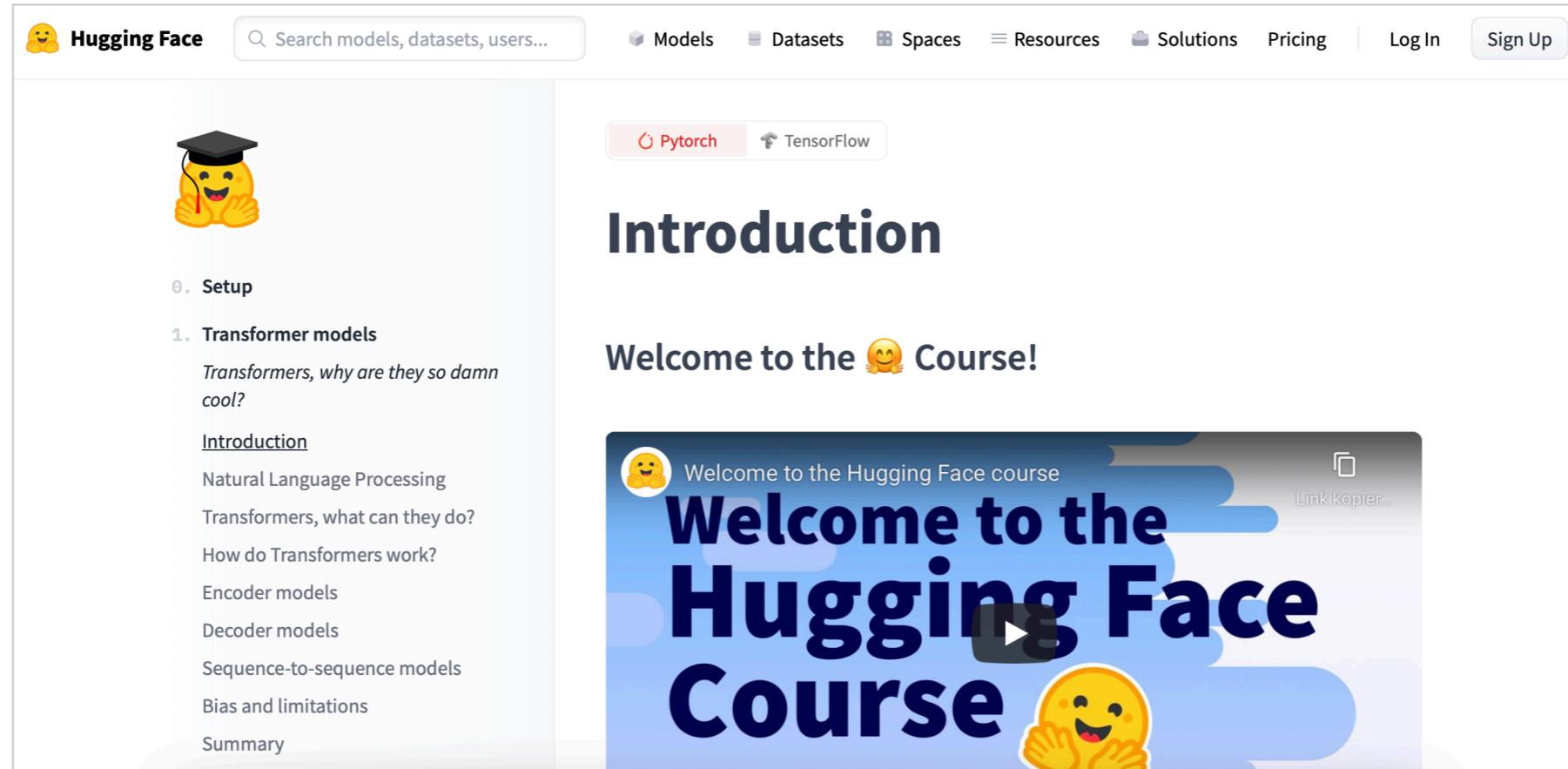
Random Forest



ÜBUNG 3: HUGGING TRANSFORMERS



HUGGING FACE: DIE TRANSFORMER LIBRARY ERKUNDEN.



The screenshot shows the Hugging Face website interface. At the top, there is a navigation bar with the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Resources, Solutions, Pricing, Log In, and Sign Up. Below the navigation bar, there is a sidebar on the left with a yellow emoji wearing a graduation cap, followed by a list of course chapters: 0. Setup, 1. Transformer models (with a sub-section about Transformers being cool), and several other sections like Introduction, Natural Language Processing, and Sequence-to-sequence models. To the right of the sidebar, the main content area features a large title "Introduction" and a welcome message "Welcome to the 😊 Course!". Below this, there is a large blue banner with the text "Welcome to the Hugging Face course" and "Welcome to the Hugging Face Course" with a video play button icon. A yellow emoji is also present on this banner. At the bottom right of the banner, there is a "Link kopier..." button.

HUGGING FACE: DIE TRANSFORMER LIBRARY ERKUNDEN.

Introduction

Transformer models

Using 🤗 Transformers

Fine-tuning a pretrained model

Sharing models and tokenizers

Diving in

The 🥰 Datasets library

The 🥰 Tokenizers library

Main NLP tasks

How to ask for help

Advanced

Specialized architectures

Speeding up training

A custom training loop

Contributing to Hugging Face

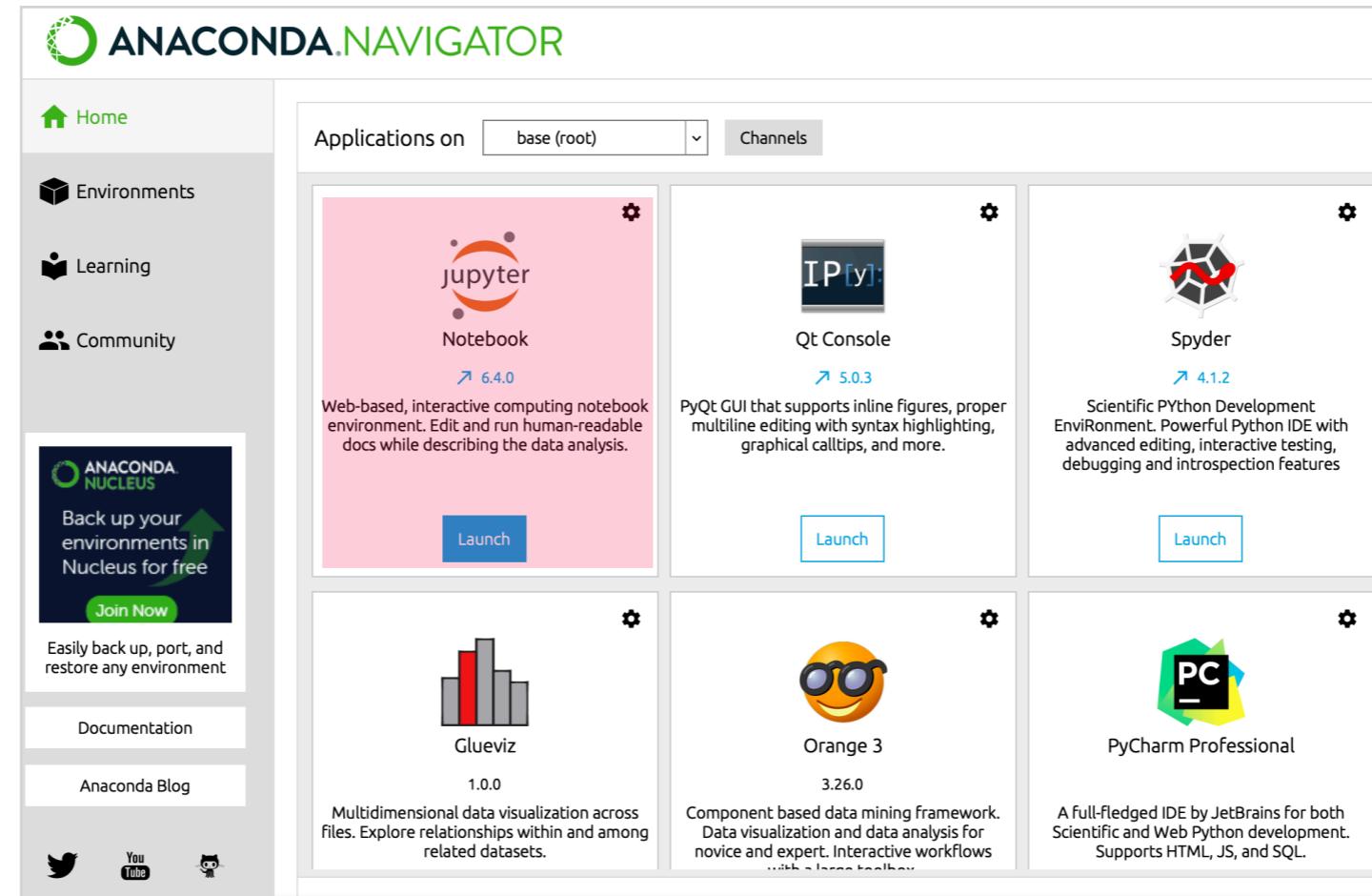
hier anfangen

sehr wahrscheinlich
nicht mehr für heute

noch nicht released

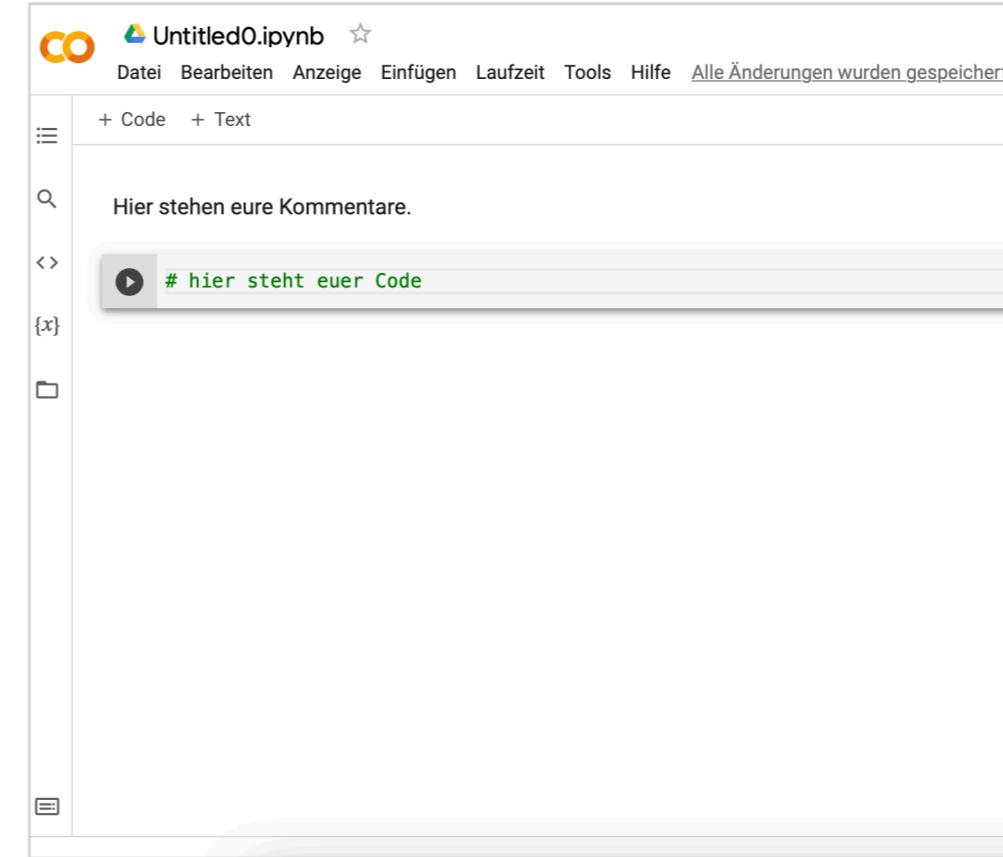
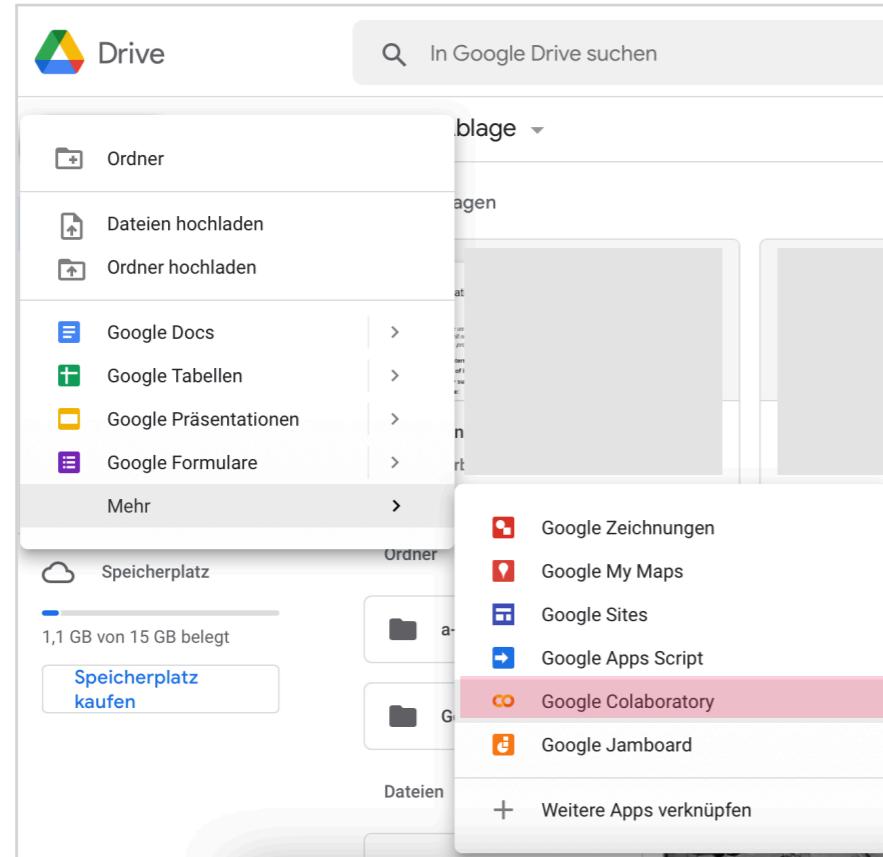
**ICH HABE NOCH KEINE
PROGRAMMIERUMGEBUNG ODER BIN
MIR MIT DEREN UMGANG UNSICHER.**

MÖGLICHE UMGEBUNGEN FÜR DIE ÜBUNG: ANACONDA & JUPYTER NOTEBOOK

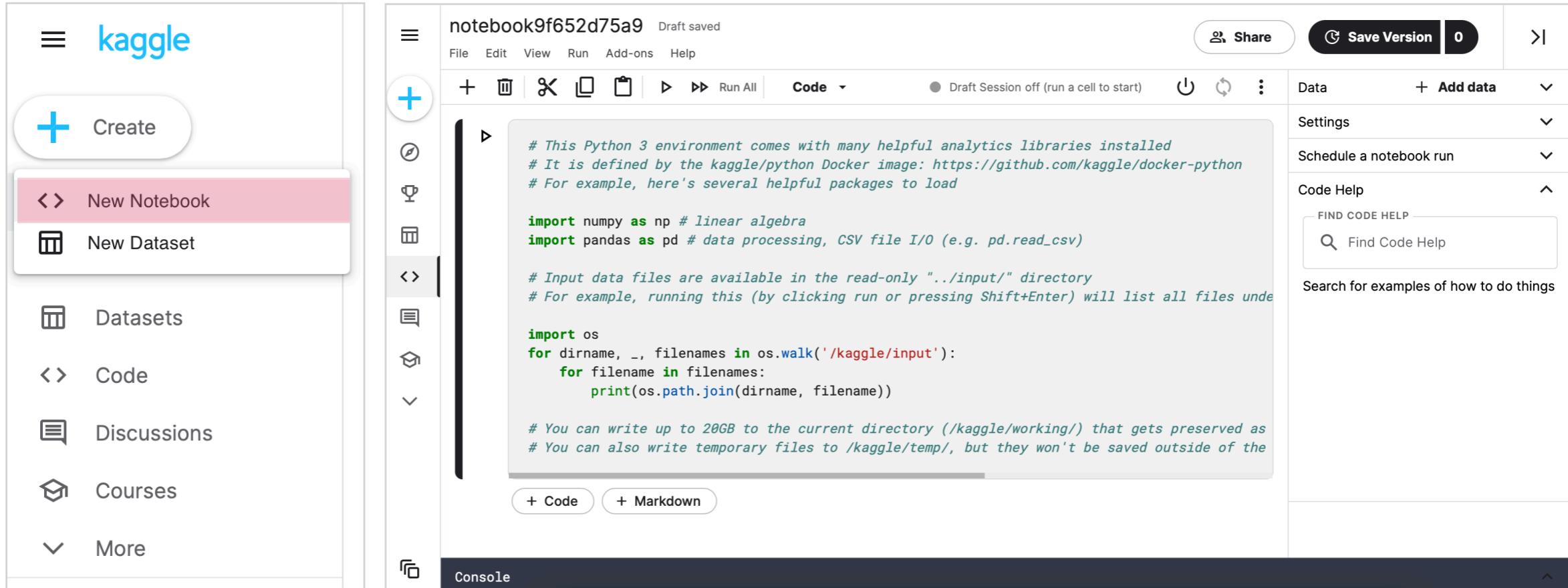


MÖGLICHE UMGEBUNGEN FÜR DIE ÜBUNG: GOOGLE COLAB

19



MÖGLICHE UMGEBUNGEN FÜR DIE ÜBUNG: KAGGLE



The screenshot shows the Kaggle web interface. On the left, there's a sidebar with options like 'Create', 'New Notebook' (which is highlighted in pink), 'New Dataset', 'Datasets', 'Code', 'Discussions', 'Courses', and 'More'. The main area is a notebook titled 'notebook9f652d75a9' with the status 'Draft saved'. The notebook interface includes a toolbar with icons for creating cells, running cells, and saving versions. A code cell contains the following Python code:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

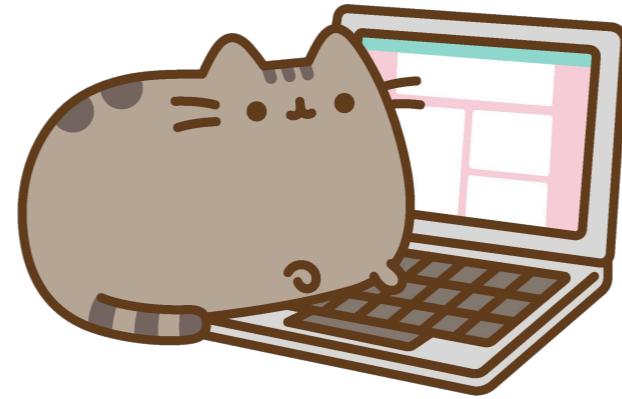
# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the notebook.
```

Below the code cell are buttons for '+ Code' and '+ Markdown'. At the bottom of the screen is a dark bar labeled 'Console'.

GUT, DANN NEHME ICH ÜBUNG ...



PAUSE NACH BEDARF.



LOS GEHT'S!