Der Code hinter KI

Dr. Alexander Rachmann

Fachbereich Technology
Professur für Anwendungsorientierte Informatik

E-Mail: a.rachmann@cbs.de

Kurzvorstellung

- Ausbildung zum Industriekaufmann in einer Krawattenfabrik
- Studium der Wirtschaftsinformatik, Informatik & Informationswissenschaft
- Promotion zur Entwicklung von IT-Dienstleistung zur Betreuung von alleinlebenden älteren Personen
- Beratertätigkeit in der Softwareentwicklung, oft im E-Commerce
- Verschiedene Positionen im Handel bei real.digital & Fressnapf
- Seit 2021 an der CBS / EUFH an der Professur für anwendungsorientierte Informatik

Agenda „Der Code hinter KI“

Was ist KI?

Codebeispiel „Weizenbaum“

Übung

Abschluss

Agenda „Der Code hinter KI“

Was ist KI?

Codebeispiel „Weizenbaum“

Übung

Abschluss

Für die Übung benötigt ihr gleich einen Google-Zugang (https://accounts.google.com/) und Leap.ai-Zugang (https://www.tryleap.ai). Legt ihn jetzt schon mal an, damit wir nachher gut in die Übung starten können.

Was ist KI?

Was ist KI?

Definition nach Zielsetzung

- Das Deutsche Forschungszentrum für künstliche Intelligenz schreibt: „Künstliche Intelligenz ist die Eigenschaft eines IT-Systems, »menschenähnliche«, intelligente Verhaltensweisen zu zeigen.“

Was ist KI?

Definition nach Funktionsbereichen

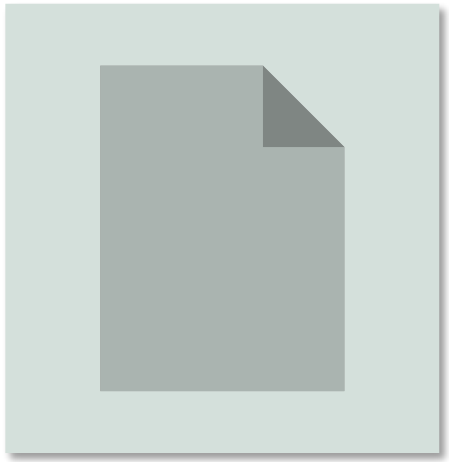
- **Verarbeitung natürlicher Sprache**, damit er einer natürlichen Sprache kommunizieren kann
- **Wissensrepräsentation**, damit er speichern kann, was er weiß und hört
- **Automatisches logisches Schließen**, um anhand der gespeicherten Informationen Fragen zu beantworten
- **Maschinenlernen**, um sich an neue Umstände anzupassen

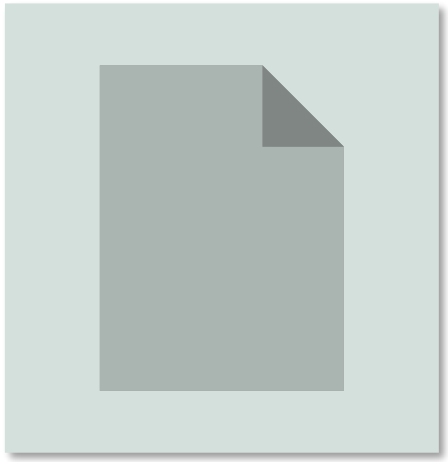
- **Sensorik**, um Objekte wahrzunehmen
- **Aktorik**, um Objekte zu manipulieren

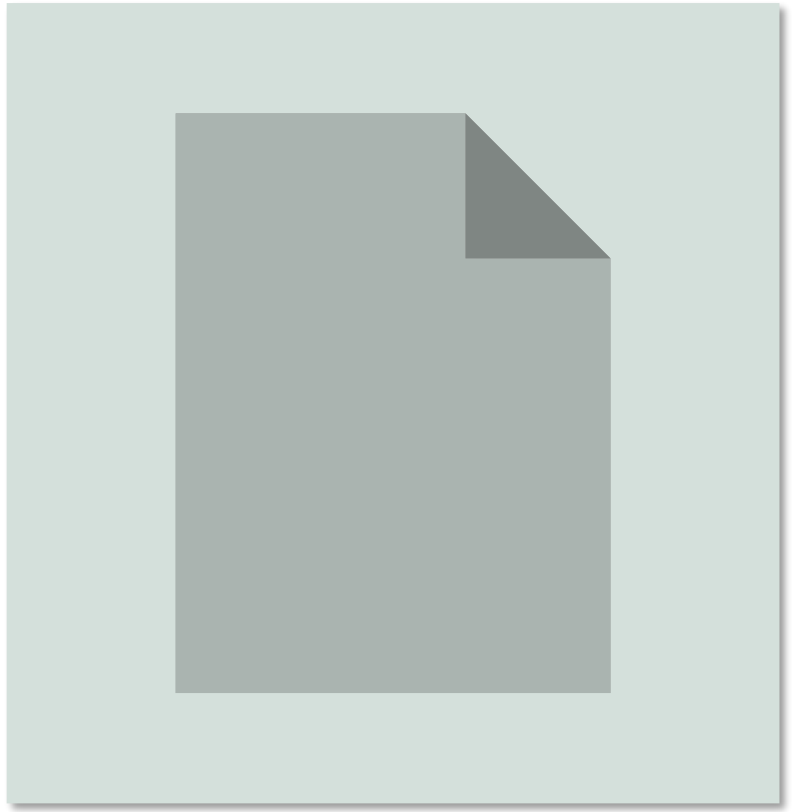
In Anlehnung an Russel, Norvig 2012

Datengetriebene KI

- Unter „datengetriebener KI“ verstehen wir solche Ansätze der KI, die aus sehr großen Datenmenge Informationen ziehen.

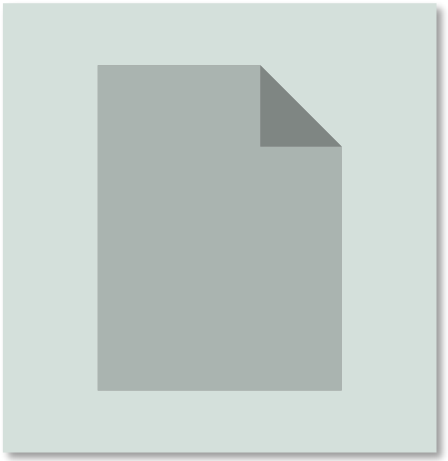




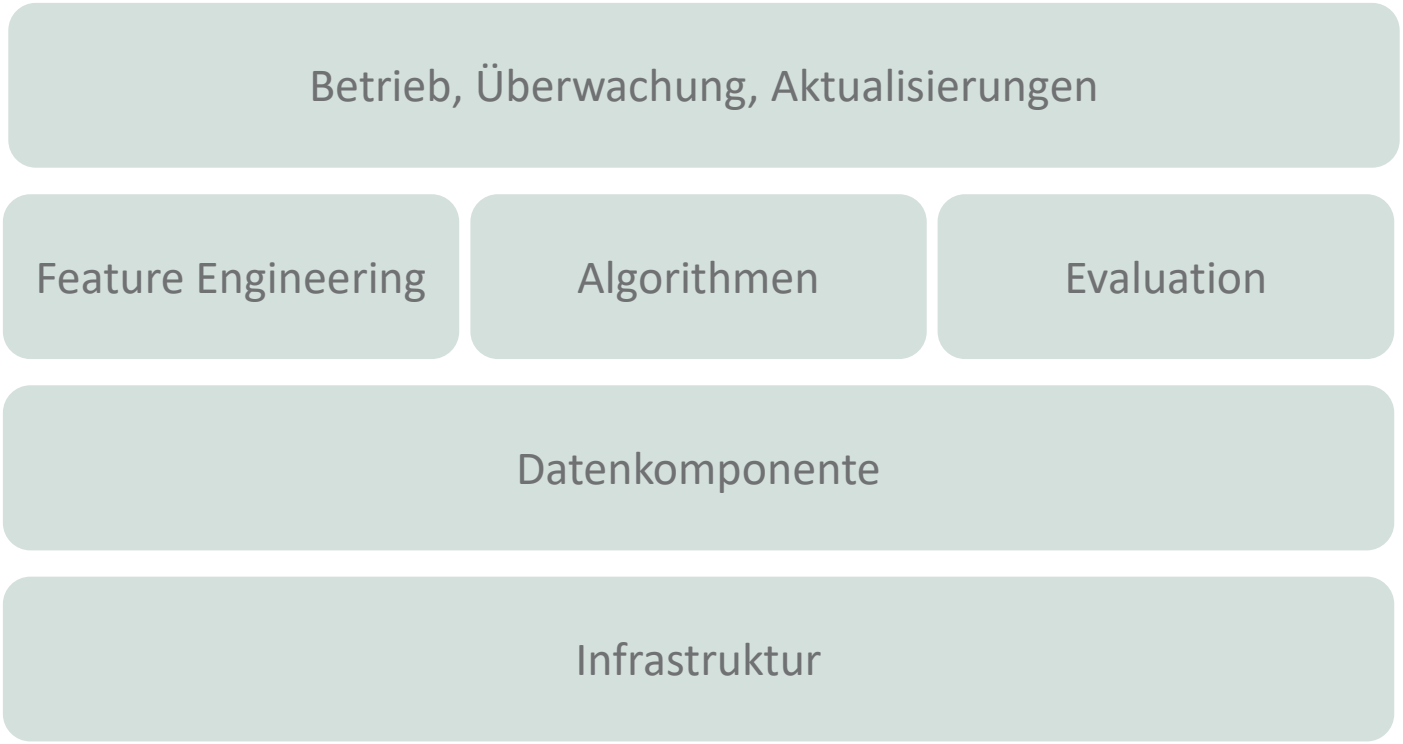


Generative KI

- Mit „generativer KI“ meinen wir solche (datengetriebenen) KI-Systeme, die aus den vorhandenen Daten „neue“ Informationen generieren
  - Texte (Text-to-text, Image-to-text etc.)
  - Bilder (Text-to-Image, Image-to-Image etc.)
  - Ton (Text-to-speech etc.)
- Bedenken:
  - Urheberschaft: Wem gehören die „neu“ generierten Informationen?
  - Deep Fakes: Wie können Fälschung erkannt werden?



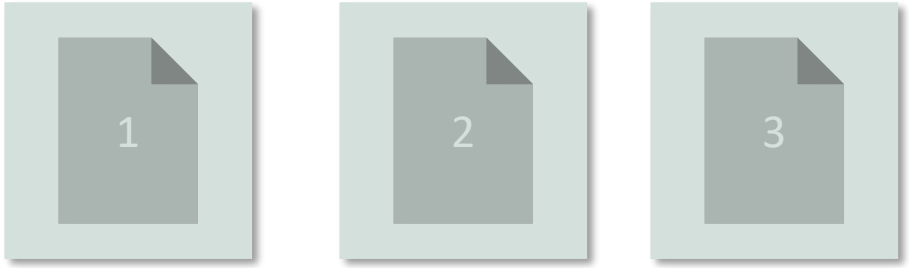
Software für moderne Machine Learning Systeme



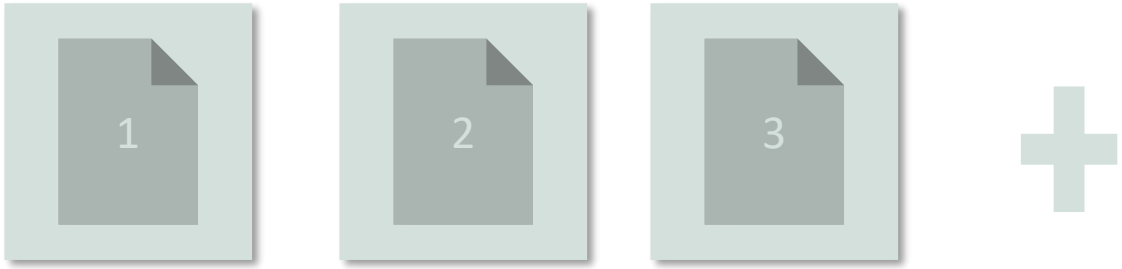
In Anlehnung an Huyen (2022)

Codebeispiel „Weizenbaum“

Was machen wir?

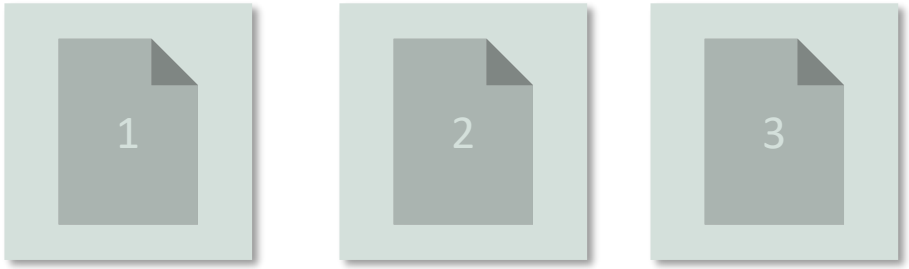


Was machen wir?



„A photo of @me with a tall black hat and sunglasses“

Was machen wir?



+ „A photo of @me with a tall black hat and sunglasses“

=

Vorlage des Beispiels

- Das hier gezeigte Beispiel stammt von „Santiago“
  - Geschäftsführung ml.school (Lehre von Machine Learning)
  - Autor bei Substack (https://substack.com/profile/27043436-santiago)
  - Autor bei Twitter (https://twitter.com/0xbnomial)
- Das Beispiel wurde unter https://twitter.com/svpino/status/1635611742008147968 veröffentlicht.

Vier Schritte des Beispiels

- Schritt 1: Rahmenbedingungen
- Schritt 2: Unsere Funktionen definieren
- Schritt 3: Model bauen
- Schritt 4: Model anwenden & Bild generieren

Vier Schritte des Beispiels

- Schritt 1: Rahmenbedingungen
- Schritt 3: Unsere Funktionen definieren
- Schritt 3: Model bauen
- Schritt 4: Model anwenden & Bild generieren

1

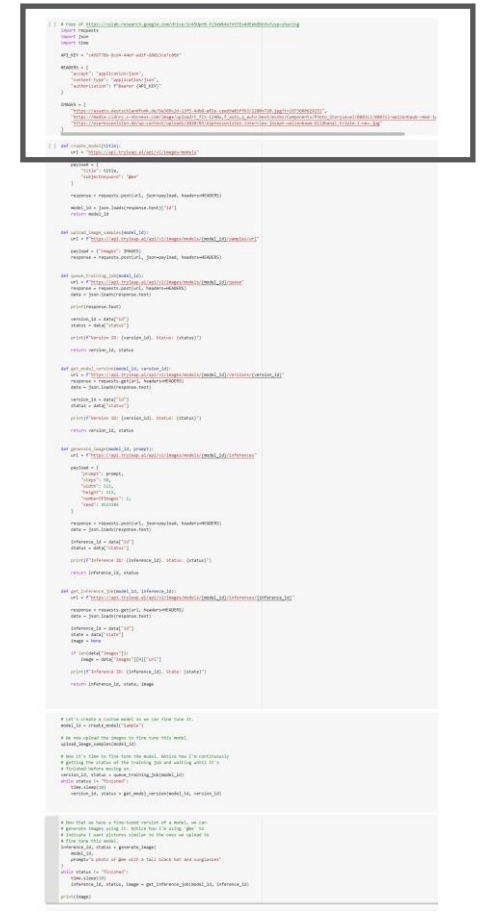
2

3

4

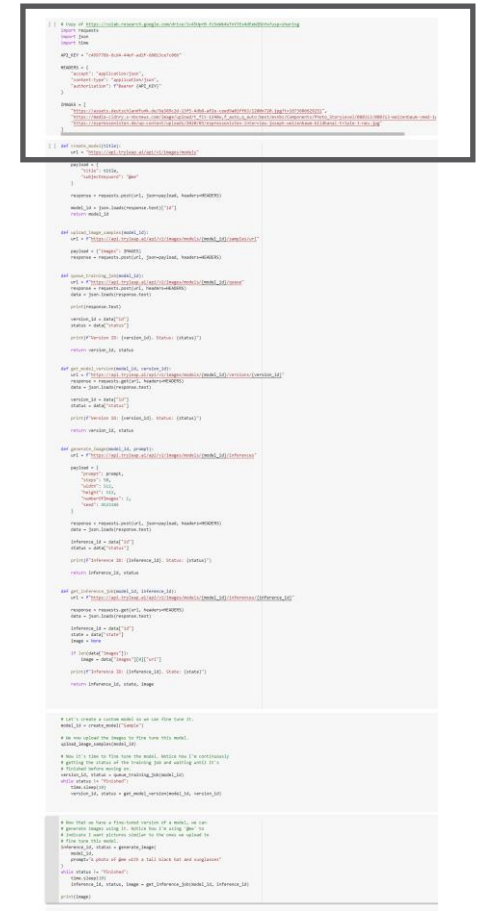
Schritt 1: Rahmenbedingungen

- Wir nutzen Bibliotheken, d.h. Programme, die andere Personen geschrieben und veröffenlicht haben.
- Wir legen Parameter fest, die die Rahmenbedingungen für unser Programm festlegen:
  - Welchen KI-Dienst wollen wir nutzen? Wie authentifizieren wir uns gegen diesen Dienst?
  - Welche Bilder wollen wir als Grundlagen nutzen?



Schritt 1: Rahmenbedingungen

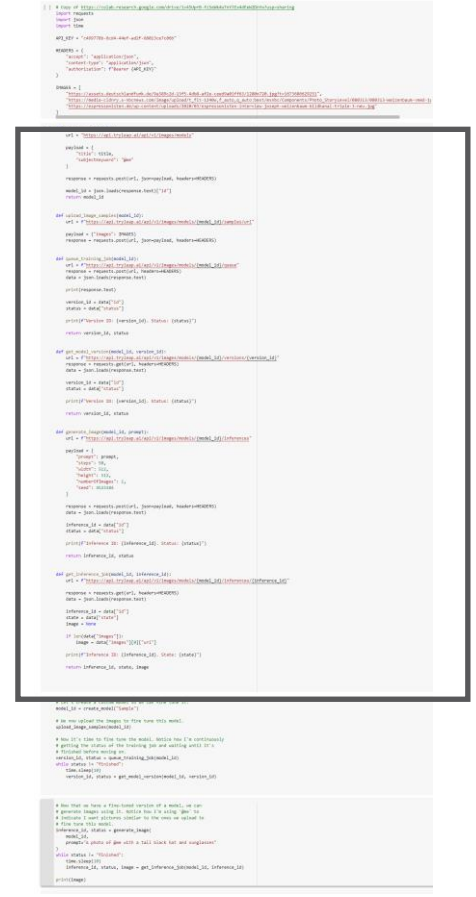
```
import requests
import json
import time

API_KEY = "c499778b-6cd4-44ef-ad2f-68013ce7c06b"

HEADERS = {
    "accept": "application/json",
    "content-type": "application/json",
    "authorization": f"Bearer {API_KEY}"
}

IMAGES = [
    "https://assets.deutschlandfunk.de/9a389c2d-13f5-4db8-af2e-
ceed9a01ff63/1280x720.jpg?t=1673686629251",
    "https://media-cldnry.s-nbcnews.com/image/upload/t_fit-
1240w,f_auto,q_auto:best/msnbc/Components/Photo_StoryLevel/080313/080313-Weizenbaum-
vmed-1p.jpg",
    "https://espressonisten.de/wp-content/uploads/2020/03/espressonisten-interview-
joseph-weizenbaum-bildkanal-triple-3-neu.jpg"
]
```



Schritt 2: Unsere Funktionen definieren

- Wir müssen ein neues Model bei unserem KI-Dienst anlegen (create_model).
- Wir müssen die Bilder beim KI-Dienst hochladen (upload_image_samples).
- Wir müssen unser Model in eine Warteschlange bei unserem Dienstleister einreihen können (queue_training_job).
- Wir müssen unser Model und unsere Bildgenerierungen über jeweils eine Kennziffer in der Datenbank des Dienstleisters ausfindig machen (get_model_version, get_inference_job).
- Wir müssen ein Bild aus unserem Model generieren (generate_image).

```
def create_model(title)
def upload_image_samples(model_id)
def queue_training_job(model_id)
def get_model_version(model_id, version_id)
def generate_image(model_id, prompt)
def get_inference_job(model_id, inference_id)
```



Schritt 2a: Funktion zum Bild generieren

- Wir möchten dem Dienstleister Informationen mitgeben zu dem Bild, welches wir generieren möchten, z.B.
  - Technische Informationen zur Erstellung des Bildes (wie viel Arbeit in das Bild gesteckt werden soll)
  - Anzahl der Bilder
  - Layoutinformationen zur Darstellung des Bildes (Breite, Höhe)

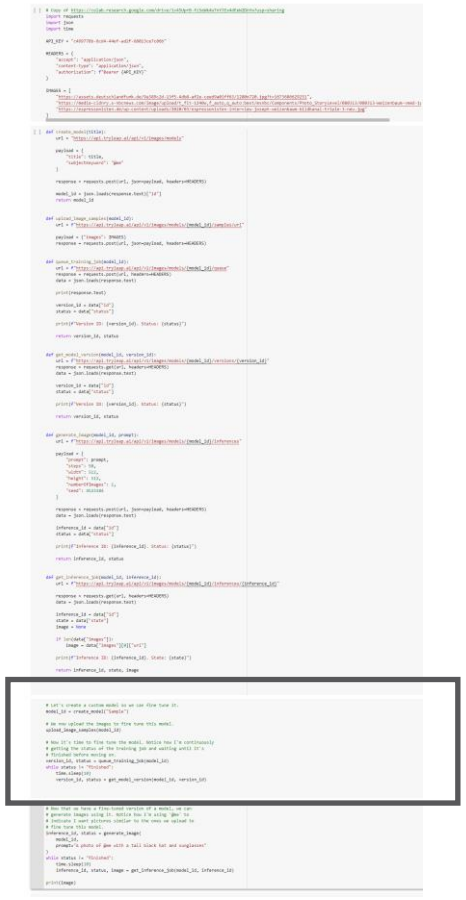
Schritt 2a: Funktion zum Bild generieren

```python
def generate_image(model_id, prompt):
    url = f"https://api.tryleap.ai/api/v1/images/models/{model_id}/inferences"

    payload = {
        "prompt": prompt,
        "steps": 50,
        "width": 512,
        "height": 512,
        "numberOfImages": 1,
        "seed": 4523184
    }

    response = requests.post(url, json=payload, headers=HEADERS)
    data = json.loads(response.text)

    inference_id = data["id"]
    status = data["status"]

    print(f"Inference ID: {inference_id}. Status: {status}")

    return inference_id, status
```

```
"prompt": prompt,
"steps": 50,
"width": 512,
"height": 512,
"numberOfImages": 1,
"seed": 4523184
```

Schritt 3: Model bauen

- Unser Dienstleister stellt uns ein **Foundation Model** zur Verfügung, welehes wir **fine-tunen** können für unsere Zwecke.

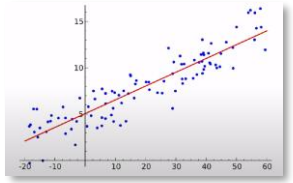


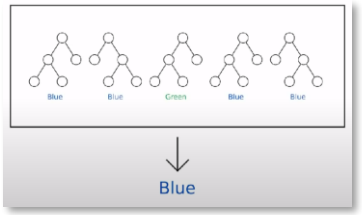
Schritt 3: Model bauen

Foundation Model

- Ein Model im Wortgebrauch der KI, ist eine Sammlung von statistischen Funktionen, anhand derer Voraussagen getroffen werden können.
- Ein Foundation Model wird genutzt, um ein möglichst breite Anwendungsvielfalt zu gewährleisten.

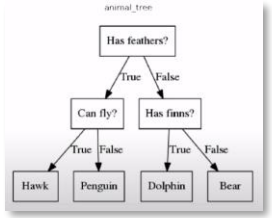
Lineare Regression

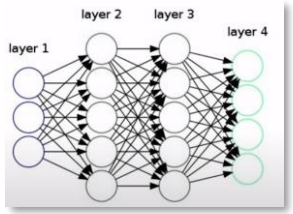




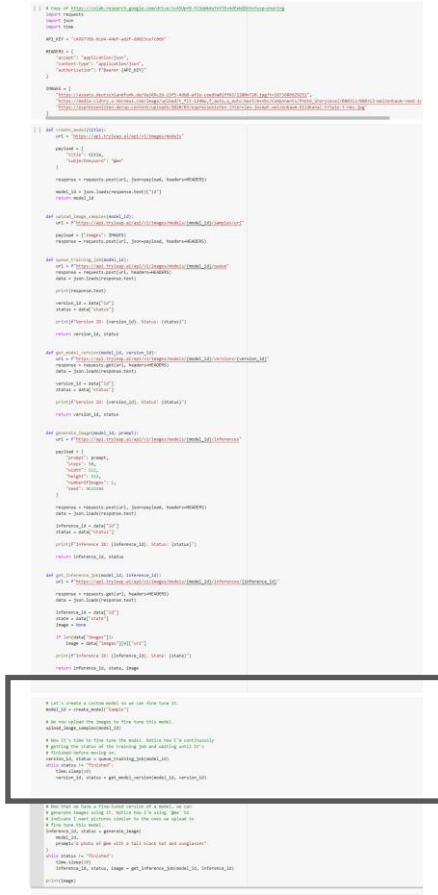
Random Forest

Decision Tree





Neuronales Netz

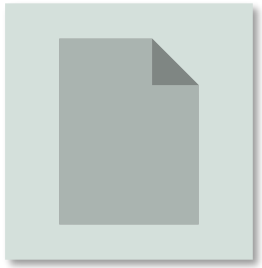


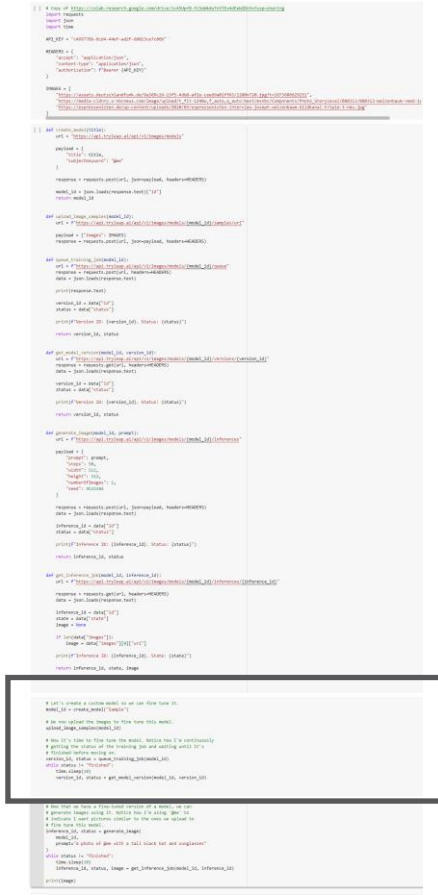
https://www.youtube.com/watch?v=yN7ypxC7838

Schritt 3: Model bauen

DALL-E mini

- Ein Model bringt z.B. einen Encoder/Decoder mit.

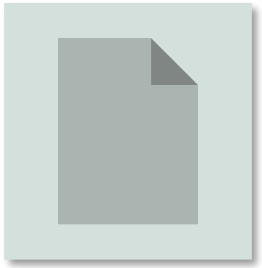


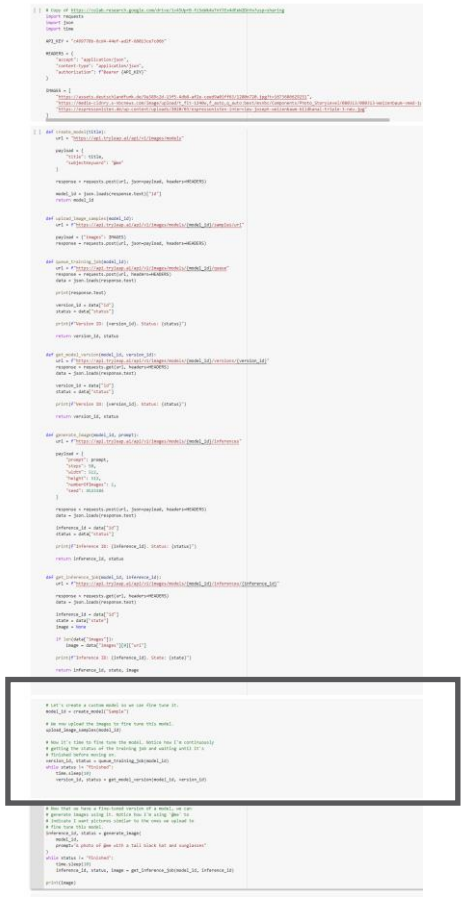


Schritt 3: Model bauen

DALL-E mini

- Ein Model bringt eine „Training“-Pipeline mit.

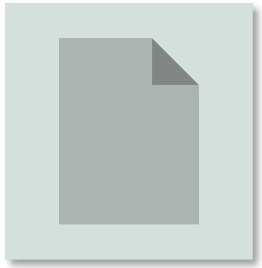


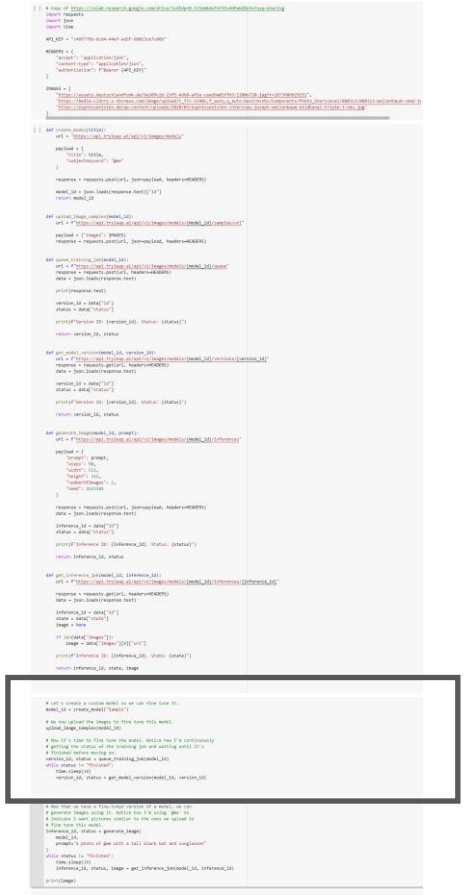


Schritt 3: Model bauen

DALL-E mini

- Ein Model bringt z.B eine Korrelationsmatrix („CLIP“) mit.

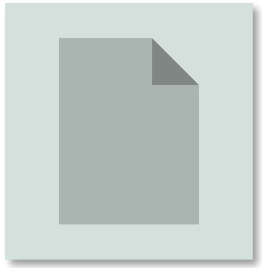


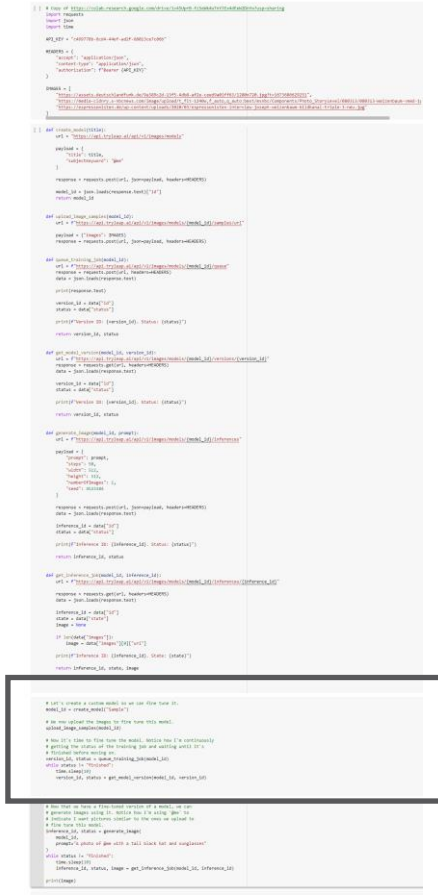


Schritt 3: Model bauen

DALL-E mini

- Ein Model bringt eine „Inferenz“-Pipeline mit.





Schritt 3: Model bauen

Fine-tuning

- Im Fine-Tuning lernt das Foundation Model Dinge, die für den spezifischen Anwendungsfall wichtig sind.

Elemente des fine-tuned Models

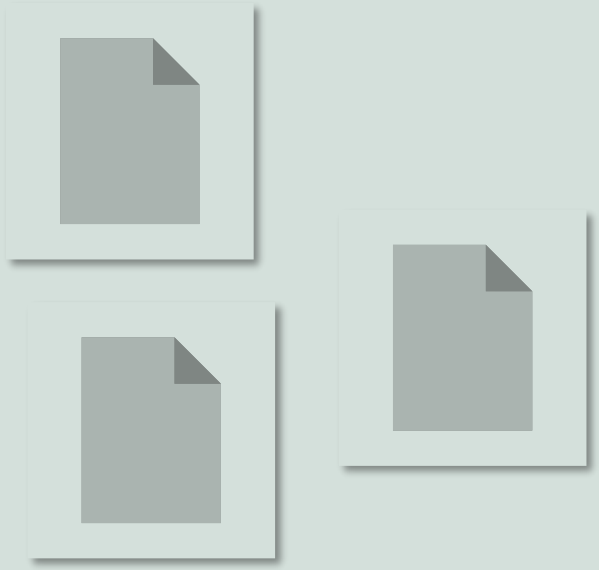
Elemente des Foundation Models

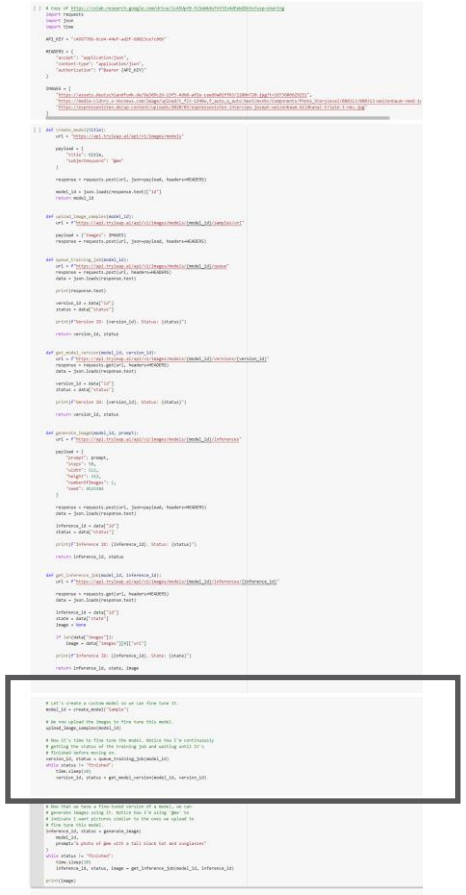
@me

„Schwarzer Hut“

„Sonnenbrille “

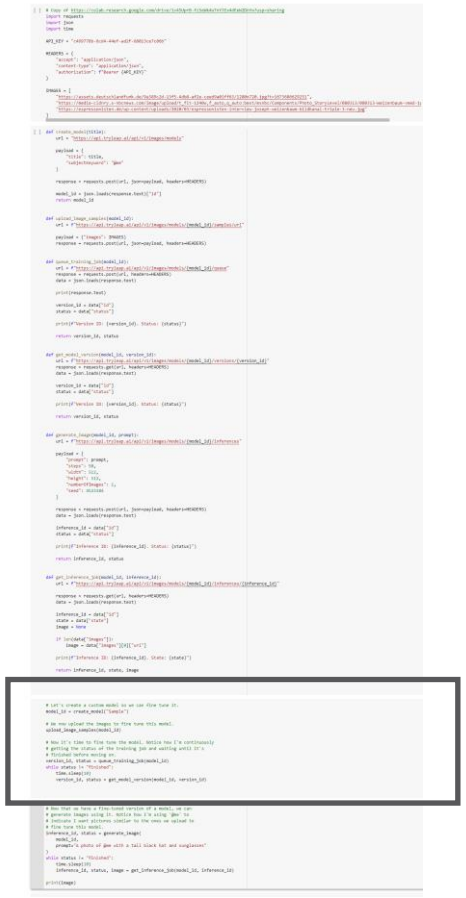
„Die Person, die auf den Bildern zu sehen ist “





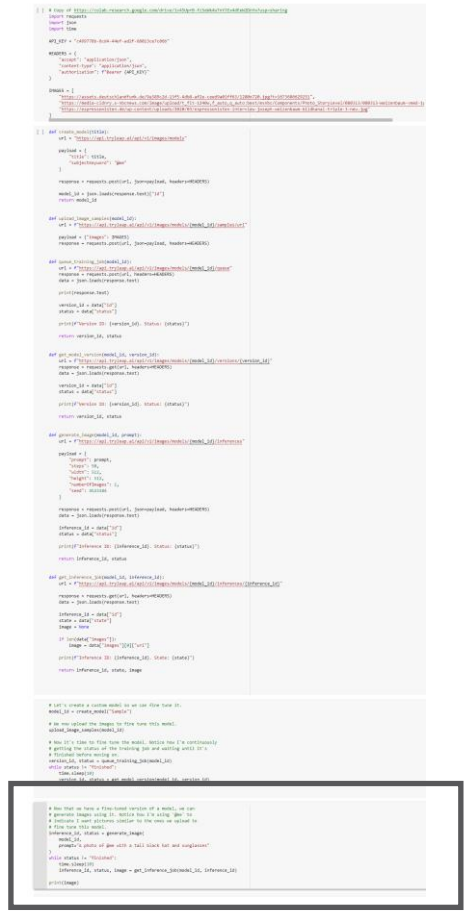
Schritt 3: Model bauen

```
# Let's create a custom model so we can fine tune it.
model_id = create_model("Sample")

# We now upload the images to fine tune this model.
upload_image_samples(model_id)

# Now it's time to fine tune the model. Notice how I'm continuously
# getting the status of the training job and waiting until it's
# finished before moving on.
version_id, status = queue_training_job(model_id)
while status != "finished":
    time.sleep(10)
    version_id, status = get_model_version(model_id, version_id)
```



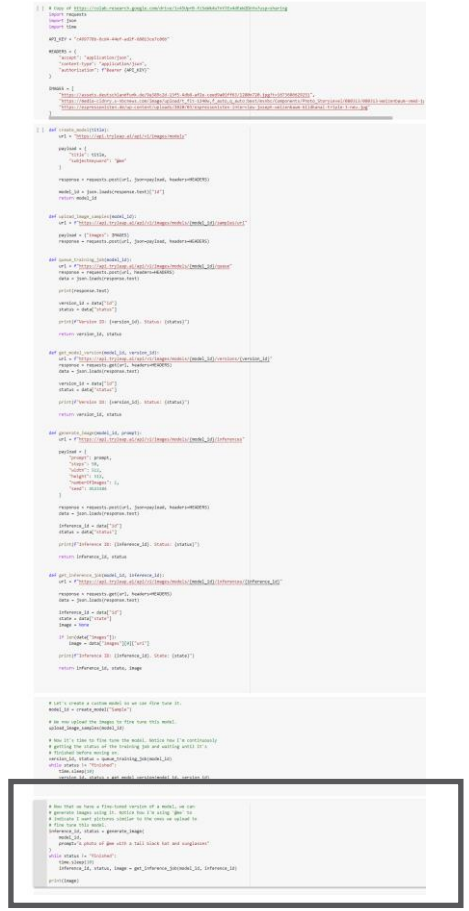
Schritt 4: Model anwenden & Bild generieren

- Wir geben dem Dienstleister einen Prompt mit, mit einer Beschreibung was wir möchten.
- Der Dienstleister soll uns regelmäßig (alle 10 Sekunden) Bescheid geben, wie weit er ist.



Schritt 4: Model anwenden & Bild generieren

```
# Now that we have a fine-tuned version of a model, we can
# generate images using it. Notice how I'm using '@me' to
# indicate I want pictures similar to the ones we upload to
# fine tune this model.
inference_id, status = generate_image(
    model_id,
    prompt="A photo of @me with a tall black hat and sunglasses"
)
while status != "finished":
    time.sleep(10)
    inference_id, status, image = get_inference_job(model_id, inferen
ce_id)

print(image)
```



Übung

Nun seid ihr dran….

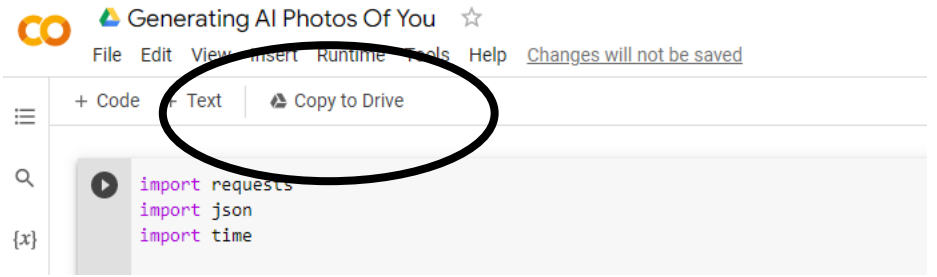
- 1. Account bei Google ist notwendig um Google Collab zu nutzen.

Nun seid ihr dran….

- 1. Account bei Google ist notwendig um Google Collab zu nutzen.
- 2. Account bei https://www.tryleap.ai/ ist notwendig um ein Model zu bauen.

Nun seid ihr dran….

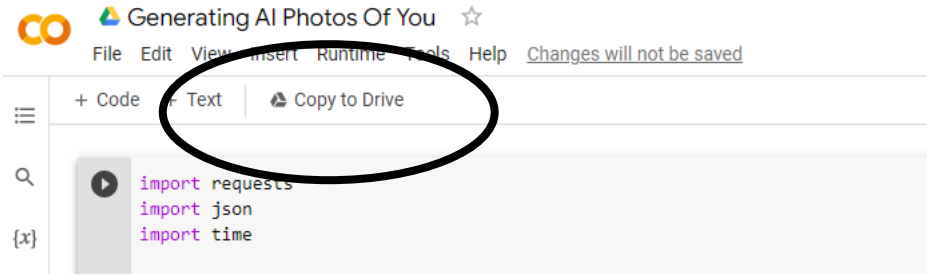
- 1. Account bei Google ist notwendig um Google Collab zu nutzen.
- 2. Account bei https://www.tryleap.ai/ ist notwendig um ein Model zu bauen.
- 3. Code kopieren von https://colab.research.google.com/drive/1v45UprB-fzSeWk4wTnYJEx4dEeW2DnYw?usp=sharing



Nun seid ihr dran….

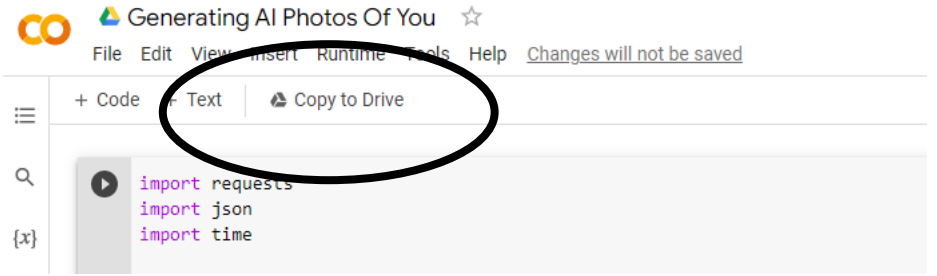
- 1. Account bei Google ist notwendig um Google Collab zu nutzen.
- 2. Account bei https://www.tryleap.ai/ ist notwendig um ein Model zu bauen.
- 3. Code kopieren von https://colab.research.google.com/drive/1v45UprB-fzSeWk4wTnYJEx4dEeW2DnYw?usp=sharing
- 4. Bilder anpassen

```
IMAGES = [
    „…",
    „…",
    „…"
]
```



Nun seid ihr dran….

- 1. Account bei Google ist notwendig um Google Collab zu nutzen.
- 2. Account bei https://www.tryleap.ai/ ist notwendig um ein Model zu bauen.
- 3. Code kopieren von https://colab.research.google.com/drive/1v45UprB-fzSeWk4wTnYJEx4dEeW2DnYw?usp=sharing



- 4. Bilder anpassen

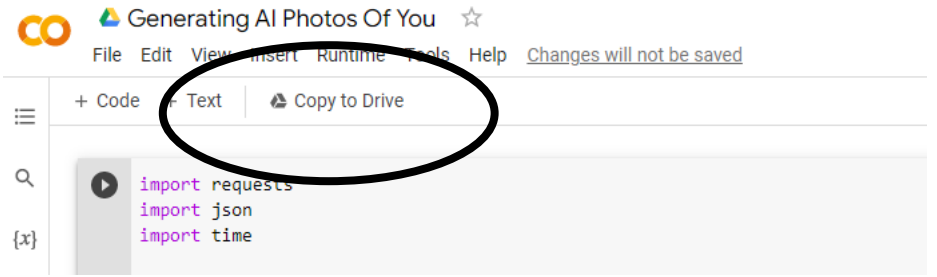
```
IMAGES = [
    „…",
    „…",
    „…"
]
```

- 5. Prompt anpassen

```
prompt="A photo of @me with a tall
 black hat and sunglasses"
```

Nun seid ihr dran….

- 1. Account bei Google ist notwendig um Google Collab zu nutzen.
- 2. Account bei https://www.tryleap.ai/ ist notwendig um ein Model zu bauen.
- 3. Code kopieren von https://colab.research.google.com/drive/1v45UprB-fzSeWk4wTnYJEx4dEeW2DnYw?usp=sharing



- 4. Bilder anpassen

```
IMAGES = [
    „…",
    „…",
    „…"
]
```

- 5. Prompt anpassen

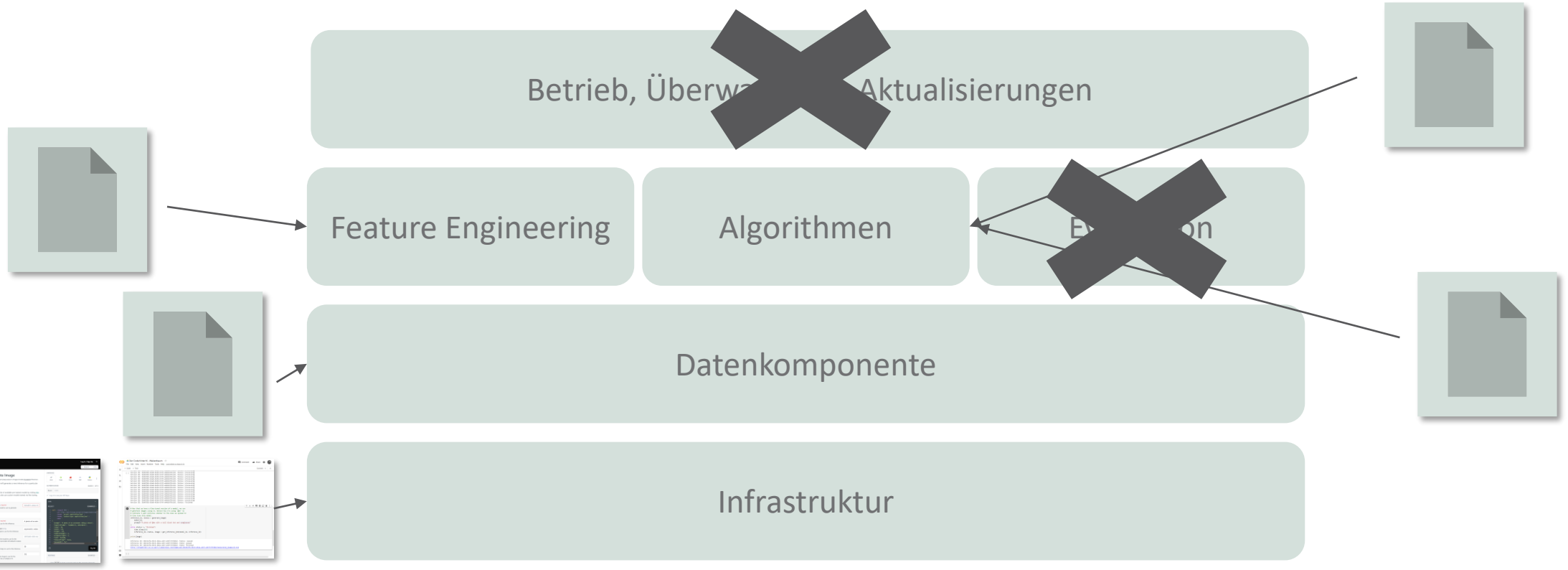
```
prompt="A photo of @me with a tall
 black hat and sunglasses"
```

- 6. Sich selbst und die Gruppe überraschen lassen ☺

Anzahl kostenloser Model-Erstellungen beachten!

Abschluss

Was haben wir heute gesehen?



In Anlehnung an Huyen (2022)

Literatur

Literatur

- Chip Huyen: Designing Machine Learning Systems. O´Reilly 2022.
- Stuart Russel, Peter Norvig: Künstlicher Intelligenz. 3. aktualisierte Auflage. Pearson Studium 2012.