2️⃣ Deployment

Learning steps

- ☑️ ~~yt video~~
- ☑️ ~~notebook/own implementation~~
- ☑️ ~~book chapter~~

Resources

- website 💻
    - lesson 2
- notebooks 🗒️
    - colab notebook
- book 📘
    - book chapter
    - solutions to exercises

Quick notes

YouTube video

→ link to YT video

- additional ressources with YT videos
    - follow along in the book for free
    - read the quizz (use aiquizzes.com for additional questions)
    - go to the fastai forums
- goal for today: put a model in production!
    - step 0: get data (and train model a first time)
    - step 1: data cleaning

- in the book, we'll use the bing API (not here because we need an SDK or smth) ; here: ddg
- before you clean the data, you train the data
- you can use *squishing* or *cropping* to get an image to a certain standard size
- data augmentation can also be used in the form of **randomResizedCrop** e.g.
- *confusion matrix*
  - used with categorical data
  - predicted vs actual (false/true negatives/positives)
  - useful to know which categories are difficult to identify
- **plot_top_losses**
  - shows where the loss is the highest
    - loss = measurement of how good is the model after we train it on a batch of data
    - it is really bad when the model thinks with a high prob but is wrong
    - it can also be bad if the model is not confident but was right
  - using this and the **ImageClassifierCleaner**, we can clean the data in the dataset
    - we take a look at the images for each category and <delete> the wrong ones (e.g. an image of a dog in our teddy bears images)

- step 2: put the model in production
  - tech used: gradio + huggingface spaces (link to blogpost)
  - we start from huggingface spaces on their website (link)
    - we create a new space (this is free)
    - we select gradio as a space sdk
    - spaces work through *git*
      - windows can run a Linux env using Windows Terminal (use `wsl —-install` in the windows terminal, reboot, done)
  - we clone the repo linked
  - we create an app.py file as mentioned
    - 🔥 to open a file: `code [path]` in the terminal (only in Windows)
  - on the App page, we have a (really basic) UI!
  - now, we need to do it with a deep learning model
    - we'll have to train one
    - we download one from the Kaggle or Colab notebook (.pkl file)
    - we upload it in the same folder as the app.py file
  - how do we make inference from the model?
    - items that start with `%` in Jupyter: 'magics' (e.g. `%time`)

        - we need to turn the notebook into a python script
            - we use notebook2script
        - we upload the app.py to git in another folder (/testing)
- to get everything running
    - go to github/fastai/fastsetup
    - clone it
    - mamba install > conda install
        - mamba also works to install on GPUs
- we can also use APIs and Javascript to create a custom user interface on a different URL
    - use Gradio
        - vast possibilities to create UIs for multiple types of ML models
    - we can create an interface that can combine models (e.g. recognize the dog's breed + let you ask questions about it)
- how to put the interface online (html file)?
    - use github pages
        - fastpages.fast.ai
    - use the fastai/tinypets repo and fork it

Exercises

- [x] ~~create model and export the .pkl (notebook part 1)~~
- [x] ~~create space on hugging face~~
- [x] ~~upload app.py and test hello world on UI~~
- [x] ~~create gradio interface (notebook part 2)~~
- [x] ~~test gradio UI in local~~
- [x] ~~create python script for gradio interface~~
- [x] ~~upload on 🤗 space~~
- [x] ~~test UI on 🤗~~
- [ ] modify UI using js
- [ ] put online using github pages