|  |
| --- |
| **Next steps for Doggy Detector**   * Continue working through “Train at scale” * Troubleshoot this error:   google.auth.exceptions.DefaultCredentialsError: File /Users/josephgulay/wagon-bootcamp-321000-ad8abda7e95e.json was not found.  I have a short term fix, but need to work out the long term fix. Need to figure out why it is trying to load that .json file as a default, and change it to the doggy detector one.  Create tests for:   * utils.array\_to\_tensor() * model.init\_model() * data.model\_to\_pickle() * data.model\_from\_pickle() * utils.predict\_breed() * trainer.py * data\_from\_pickle   Continue with the implementation of “Your first CD”- Deploy on a free Heroku dyno when building the front end.  Update readme file on github |
| **Today**  **05-02-2022**   * Continue with Run the code on the AI Platform   **Issue:** Not able to run the make file. File "/Users/joe/.pyenv/versions/lewagon/lib/python3.8/site-packages/google/cloud/storage/blob.py", line 1282, in download\_to\_filename  with open(filename, "wb") as file\_obj:  FileNotFoundError: [Errno 2] No such file or directory: './data/Pickle Files/y.pickle'  What I think is happening is similar to what happened on 25-02-2022, when running the make file, the trainer.py file is running as if it is from the same location as the make file. I need to find a way to make it work such that it always works no matter where it is being run from. It needs to run according to an absolute path.  **Solution:**  I decided to implement the following pseudo code:   1. Find the current location, and return it as a string 2. Shorten the string to the first instance of “Doggy Detector”. This will be the single point of reference no matter where the code is being run from 3. Make changes to the individual functions to reference this single point of reference   The code itself is:  #Convert the current working directory into a string  cwd = str(os.getcwd())  #Find the first occurance of DoggyDetector, and add 13 to create slicer value  slicer = cwd.index("DoggyDetector") + 13  #create absolute working directory  awd = cwd[0:slicer] |
| **History**  **04-02-2022**  - Create ‘pickle from gcp’ function, modify trainer.py to be able to download pickle from gcp  -create model locally following directions on Kitt  - Update “file\_to\_gcp”  **03-02-2022**  **-** Continue to work on pickle to GCP. Note that the pickle file will first have to be saved locally before it is uploaded to GCP  **Learning:**   * I first had to Google Cloud Platform -> Credentials -> Under the heading “Service Accounts”, click the first project -> Navigate to the “Keys” tab -> Click “Add Key” * Next I had to download the key. I saved it to “/Users/joe/code/keys/doggy-detector-2022-c42f18ed1a2f.json” * Finally, I ran this code in terminal:   export GOOGLE\_APPLICATION\_CREDENTIALS=/Users/joe/code/keys/doggy-detector-2022-c42f18ed1a2f.json  **02-03-2022**  Started to create py files to upload into GCP  Continue with creating a bucket in GCP  **Issue:** Not able to run the make file, the following error pops up:  gcloud: No such file or directory  **Solution:** It looks like I had to do some of the initial set up to be able to run in GCP. This included setting up the CLI, authentication and setting the project name.  **01-03-2022**  Started looking at GCP setup.  **Learning:** When I remove the machine learning requirements of the package, I am able to deploy to Heroku. So it looks like the machine learning requirements are what is taking up space. I will need to deploy Heroku only for the front end it seems. For now I will delete the Heroku file.  **Update:** It looks like the issue isn’t with the pickle file or the images file. It has something to do with the packages being installed. Looking at rebeccas code, it looks like only the front end is uploaded, not the back end. This could be the root cause.  Troubleshoot this step- slug size still too large, investigate what else is taking up space  Add back in the image folder and the X.pickle file, to see which one is pushing it over the limit.  Figure out why .slugignore isn’t working.  ## Warning - The same version of this code has already been built: 553cdf8d698cfb45cae0cf39df1b5c44fe20325c  remote: !  remote: ! We have detected that you have triggered a build from source code with version 553cdf8d698cfb45cae0cf39df1b5c44fe20325c  remote: ! at least twice. One common cause of this behavior is attempting to deploy code from a different branch.  remote: !  remote: ! If you are developing on a branch and deploying via git you must run  remote: !  remote: ! git push heroku <branchname>:main  remote: !  remote: ! This article goes into details on the behavior:  remote: ! https://devcenter.heroku.com/articles/duplicate-build-version  **28-02-2022**  Troubleshoot this step- slug size still too large, investigate what else is taking up space  Update: When I remeove both the Image folder and the X.pickle file, the slug error disappears, but another error appears.  Compiled slug size: 746.1M is too large (max is 500M).  See: <http://devcenter.heroku.com/articles/slug-size>  .slugignore doesn’t seem to be having an impact  *Learning: Add files to .slugignore that you don’t need, as the max size of the slug package is 500 MB*  **27-02-2022**  Start to Implement CI and CD  *(Learning: Need to run git push Heroku master from the same location as git push origin master)*  Build predictor for a single image  Run model and predict locally  **26-02-2022**  Continue to build trainer.py  **25-02-22**  **Start making trainer.py**  ***Issue:*** *I’m having trouble creating a test file for the pickle tests- it seems to to be able to see the test folder. Does a makefile run the tests from the makefile location or the test folder?*  ***Learning:*** *I changed the pickle\_path. It looks like the file is run from the makefile location* |