|  |
| --- |
| Next steps for Doggy Detector 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.  **Potential Solution:**  I think the solution is fiound here: <https://cloud.google.com/docs/authentication/production#passing_variable>  I will need to go to ~/.zshrc (Note: This is a hidden file) and add this line:  export GOOGLE\_APPLICATION\_CREDENTIALS=/Users/joe/code/keys/doggy-detector-2022-c42f18ed1a2f.json  Create a params file  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 |
| Today13- 03-2022 Integrate into ML Flow, the iterate with ML flow |
| History12- 03-2022 Create starting ml\_flow file  Create a params file  Resolve git pull request conflict  Update git for the laptop 11- 03-2022 **Issue:**  Troubleshoot this. I suspect the pickle file is for some reason being loaded in as a Boolean. Need to correct this by investigating the data type.  The replica master 0 exited with a non-zero status of 1. Traceback (most recent call last): File "/usr/lib/python3.7/runpy.py", line 193, in \_run\_module\_as\_main "\_\_main\_\_", mod\_spec) File "/usr/lib/python3.7/runpy.py", line 85, in \_run\_code exec(code, run\_globals) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 214, in <module> trainer.train\_GCP\_data(n=1000, pickle=True, make\_file=True) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 105, in train\_GCP\_data y = pickle.load(y\_pickle\_in) AttributeError: 'bool' object has no attribute 'load'  **Solution:**  I needed to use “pickle.loads()” instead of “pickle.load()”. Pickle was taken as an argument (boolean). I renamed the argument pickle\_source, and also imported “pickle”. I also had to change how it saved the model (creating a model.joblib file), then uploaded this joblib file up to google. 10-03-2022 **Issue:**  GCP doesn’t seem to be able to find the pickle file locally.  **Solution:**  Have utilised  pickle\_in = blob.download\_as\_string()  to try and install the pickle file in the RAM I guess?  **Implement makefile = True argument**  **Issue:**  Troubleshoot this error:  "Traceback (most recent call last): File "/usr/lib/python3.7/runpy.py", line 193, in \_run\_module\_as\_main "\_\_main\_\_", mod\_spec) File "/usr/lib/python3.7/runpy.py", line 85, in \_run\_code exec(code, run\_globals) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 217, in <module> trainer.train\_GCP\_data( n=1000, pickle=True) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 94, in train\_GCP\_data slicer = cwd.index("DoggyDetector") + 13 ValueError: substring not found  It looks like the current working directory doesn’t have DoggyDetector in it. This may be an issue that was caused by me trying to have a root folder.  Troubleshooting option 0. Try to run it as a package locally from somewhere else in the system. I suspect that when being run as a package it doesn’t “know” that the package name is DoggyDetector, hence why the root system thing isn’t working.  Troubleshooting option 1:  Find a way to export the string to google cloud.  **Solution:**  Option 0 seems to have worked. 09-03-2022 Didn’t code 08-03-2022 Didn’t code 07-03-2022 Didn’t code 06-03-2022 **Issue:**  Troubleshoot this error:  "Traceback (most recent call last): File "/usr/lib/python3.7/runpy.py", line 193, in \_run\_module\_as\_main "\_\_main\_\_", mod\_spec) File "/usr/lib/python3.7/runpy.py", line 85, in \_run\_code exec(code, run\_globals) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 217, in <module> trainer.train\_GCP\_data( n=1000, pickle=True) File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 94, in train\_GCP\_data slicer = cwd.index("DoggyDetector") + 13 ValueError: substring not found  It looks like the current working directory doesn’t have DoggyDetector in it. This may be an issue that was caused by me trying to have a root folder.  Troubleshooting option 1:  The option might be to re-run the code as if it is running from the make file. The default location could be the makefile, and could add an argument when testing the .py files to run locally.  Troubleshooting option 2. Try to run it as a package locally from somewhere else in the system. I suspect that when being run as a package it doesn’t “know” that the package name is DoggyDetector, hence why the root system thing isn’t working.  Troubleshooting option 3:  Find a way to export the string to google cloud.  **Solution:**  **Issue:**  Troubleshoot this error:  The replica master 0 exited with a non-zero status of 1.  Traceback (most recent call last):  File "/usr/lib/python3.7/runpy.py", line 193, in \_run\_module\_as\_main  "\_\_main\_\_", mod\_spec)  File "/usr/lib/python3.7/runpy.py", line 85, in \_run\_code  exec(code, run\_globals)  File "/root/.local/lib/python3.7/site-packages/DoggyDetector/trainer.py", line 2, in <module>  from DoggyDetector.data import category\_list, create\_training\_data, data\_from\_pickle, model\_to\_pickle, data\_to\_pickle, file\_from\_gcp, file\_to\_gcp  File "/root/.local/lib/python3.7/site-packages/DoggyDetector/data.py", line 3, in <module>  import matplotlib.pyplot as plt  File "/root/.local/lib/python3.7/site-packages/matplotlib/\_\_init\_\_.py", line 208, in <module>  \_check\_versions()  File "/root/.local/lib/python3.7/site-packages/matplotlib/\_\_init\_\_.py", line 204, in \_check\_versions  raise ImportError(f"Matplotlib requires {modname}>={minver}; "  ImportError: Matplotlib requires numpy>=1.17; you have 1.16.5  To find out more about why your job exited please check the logs: https://console.cloud.google.com/logs/viewer?project=979213966914&resource=ml\_job%2Fjob\_id%2Fdoggy\_detector\_training\_pipeline\_20220305\_074612&advancedFilter=resource.type%3D%22ml\_job%22%0Aresource.labels.job\_id%3D%22doggy\_detector\_training\_pipeline\_20220305\_074612%22  **Solution:**  Changed requirements.txt to include numpy >= 1.17. Also had to include opencv-python, a pip >=22, added tesnorflow-gpu, and added “import keras” to each relevant py file.  . Note: It’s interesting that these errors are happening on google cloud and not when deploying to local file.  **05-03-2022**   * Continue with Run the code on the AI Platform   **Issue:** Error: The provided GCE region 'AUSTRALIA-SOUTHEAST2' is not available, or your project needs to be whitelisted to use it.  **Solution:**  I changed the location to ‘australia-southeast1’ (note the lowercase) as it doesn’t look like Google AI platform is available on Australia Southeast 2. I also had to re-create the bucket.  **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]  **04-03-2022**  - 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”  **03-03-2022**  **-** 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.  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>  **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. I’ll continue with building the back end an only upload the front end to Heroku  **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* |