

# Relatório

Repositório GIT: [https://github.com/adielwes/ml\\_oops](https://github.com/adielwes/ml_oops)

## Problema

A empresa de adoção de pets, PetCare Solutions, conecta abrigos de animais com tutores de pets. A empresa desenvolveu um modelo de classificação de pets em cachorros e gatos, e agora necessita de uma plataforma de MLOps que permita a evolução contínua do modelo classificador e colaboração entre o time desenvolvedor.

## Desenvolvimento da Solução

Criamos um repositório GIT, mantido no Github, para que o time possa ter acesso ao código implementado e atualizado do algoritmo de treinamento do modelo classificador.

Usamos o Anaconda para criar um ambiente de execução para o treinamento do modelo classificador na máquina local.

```
$ conda create --name mlops python=3.9  
$ conda activate mlops
```

Versionamos os dados com DVC, rodando os seguintes comandos na pasta raiz do projeto. Instalamos o DVC e iniciamos o repositório para versionar os dados com ele.

```
$ pip install dvc  
$ dvc init  
$ git add .  
$ git commit -m "DVC Init"  
$ git push origin main
```

Copiamos a pasta “ds” com os dados de treinamento e validação para a pasta de repositório GIT e adicionamos a pasta ao versionamento do DVC.

```
$ dvc add ds  
$ git add .gitignore ds.dvc  
$ git commit "Init ds"  
$ git commit -m "Init ds"  
$ git push origin main
```

Definimos o repositório de dados remoto para uma pasta no Google Drive

```
$ pip install dvc-gdrive  
$ dvc remote add -d mlops-t1 gdrive://1cE-_V3ssk6yRHQHp07MW0OAuwHZdiUDf  
$ dvc push  
$ git add .dvc/config  
$ git commit -m "Added dvc remote"
```

```
$ git push origin main
```

Salvamos o código do modelo classificador no repositório Git.

```
$ git add .
```

```
$ git commit -m "Added Pet Classifier"
```

```
$ git push origin main
```

Por fim, executamos o código versionado.

```
$ pip install tensorflow
```

```
$ pip install Pillow
```

```
$ pip install scipy
```

```
$ python3.9 PetCare.py
```

## Resultado do Treinamento

```
adielews@POAGES3950GV2: ~/pers/ml_oops
(mlops) adielews@POAGES3950GV2:~/pers/ml_oops$ python3.9 PetCare.py
2023-12-03 00:44:14.530683: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU will not be used
.
2023-12-03 00:44:14.570851: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attempting to re
gister factory for plugin cuDNN when one has already been registered
2023-12-03 00:44:14.570885: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempting to reg
ister factory for plugin cuFFT when one has already been registered
2023-12-03 00:44:14.572083: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Attempting to
register factory for plugin cuBLAS when one has already been registered
2023-12-03 00:44:14.578328: I external/local_tsl/tsl/cuda/cudart_stub.cc:31] Could not find cuda drivers on your machine, GPU will not be used
.
2023-12-03 00:44:14.578510: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available CPU inst
ructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2023-12-03 00:44:15.390145: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Found 8000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
Epoch 1/10
125/125 [=====] - 169s 1s/step - loss: 0.6964 - accuracy: 0.5692 - val_loss: 0.6595 - val_accuracy: 0.5786
Epoch 2/10
125/125 [=====] - 174s 1s/step - loss: 0.6513 - accuracy: 0.6177 - val_loss: 0.6175 - val_accuracy: 0.6754
Epoch 3/10
125/125 [=====] - 186s 1s/step - loss: 0.6389 - accuracy: 0.6392 - val_loss: 0.6394 - val_accuracy: 0.6573
Epoch 4/10
125/125 [=====] - 192s 2s/step - loss: 0.6413 - accuracy: 0.6381 - val_loss: 0.6002 - val_accuracy: 0.6815
Epoch 5/10
125/125 [=====] - 191s 2s/step - loss: 0.6221 - accuracy: 0.6538 - val_loss: 0.5811 - val_accuracy: 0.6885
Epoch 6/10
125/125 [=====] - 190s 2s/step - loss: 0.6023 - accuracy: 0.6721 - val_loss: 0.5725 - val_accuracy: 0.6915
Epoch 7/10
125/125 [=====] - 188s 2s/step - loss: 0.5917 - accuracy: 0.6817 - val_loss: 0.5840 - val_accuracy: 0.6835
Epoch 8/10
125/125 [=====] - 192s 2s/step - loss: 0.5862 - accuracy: 0.6852 - val_loss: 0.5498 - val_accuracy: 0.7248
Epoch 9/10
125/125 [=====] - 187s 1s/step - loss: 0.5772 - accuracy: 0.6976 - val_loss: 0.5216 - val_accuracy: 0.7455
Epoch 10/10
125/125 [=====] - 177s 1s/step - loss: 0.5735 - accuracy: 0.7009 - val_loss: 0.5222 - val_accuracy: 0.7465
(mlops) adielews@POAGES3950GV2:~/pers/ml_oops$
```

## Conclusão

Oferecemos para a PetCare Solutions um repositório versionado com GIT e DVC. O repositório remoto de dados está no Google Drive e o código de treinamento do modelo e metadados de versionamento dos dados ficam disponíveis no GitHub. O sistema de versionamento com repositórios remotos compartilhados permite que o time de desenvolvimento de Machine Learning possa colaborar e trabalhar na evolução contínua do modelo classificador de pets.