Session 02 - Homework 02

Deadline: March 24th, 09:29am

1. Goals

1.1. Code refactoring & Storing artifacts

For this exercise, you will create **three python scripts**:

- one is to preprocess data,
- one is to train the model
- and one is to infer the trained model on unseen data.

Each step will bring you closer to your ultimate goal: a functioning ML pipeline in production. That is why, each step will store either data or a model artifact into the Google Cloud Storage, so that you can retrieve it at the following step.

1.2. Docker images

For each of the steps (preprocessing, training, infering): write a Dockerfile and create a Docker image.

2. Instructions

2.1. Code refactoring

Preprocessing script: feature_engineering.py

1. On Google Cloud Storage, create a bucket and store both files: df_train.csv and df_test.csv.
2. Use **feature_engineering.py** script given in the inputs
3. Assign the name you gave to your bucket in BUCKET_NAME variable
4. Download **df_train.csv** from the Google Cloud Storage bucket where it is stored. Hint: you can use the helpers.py functions.
5. Upload **y_train.csv, X_train.npy and preprocessor.pkl** to the bucket
6. Your script needs to contain a **"___main___"** function at the end so that it can be executed from the command line

Once you have made the modifications to the script, you can test it:

- Go to Vertex AI on Google Cloud Platform and start your workbench instance ("Open Jupyter-lab")
- Within jupyter-lab, create a folder **02_homework**
- Create another folder **inside 02_homework called 'feature_engineering'**. You now have 02_homework/feature_engineering as a directory structure.
- Upload the following scripts **into feature_engineering folder**: feature_engineering.py, helpers.py, requirements.txt
- Open a terminal session within the jupyter-lab. Make sure you are located in the feature_engineering folder (tip: use `cd` bash command to navigate to the right folder)
- Run `pip install -r requirements.txt` and later `python feature_engineering.py`
- Check in your bucket whether **y_train.csv, X_train.npy and preprocessor.pkl** have been created

Tip: you don't need the files in the jupyter-lab instance, because they have been uploaded to the bucket. Actually, you could modify the feature_engineering.py script to delete the local files after you're done. It is not required for this homework though.

Training script: training.py

1. Create a **training.py** script
2. Read the **X_train.npy and y_train.csv** stored at the previous step
3. Split the training sets into train and validation sets (split 90% / 10%)

4. Train the model based on the train data sets. Bonus: do hyperparameter tuning.
5. Store the trained model in a pickle file: **trained_model_YYYYmmdd_HHMM.pkl**. The file name should contain the day and time at which the model is stored. Hint: check strftime function in python.
6. Infer the model on the validation set. If the f1 score on the validation set is higher than 0.85, store this model again under the name "accepted_model".

Once you have made the modifications to the script, you can test it:

- Go to Vertex AI on Google Cloud Platform and start your workbench instance ("Open Jupyter-lab")
- Within jupyter-lab, create another folder inside 02_homework called 'training'. You now have 02_homework/training as a directory structure.
- Upload the following input files into feature_engineering folder: training.py, helpers.py.
- Open a terminal session within the jupyter-lab. Make sure to be located in the training folder.
- Run `python training.py`
- Check in your bucket that the model has been stored.

Tip: usually, scripts are structured in a way that you don't need to upload some scripts multiple times, like we did for helpers.py. However, dealing with relative imports is not the topic of the course and we aim for straightforward solutions. The solution here is easy and straightforward but has a duplication of the helpers.py script.

Optional challenge: can you modify the scripts to have relative imports so that the helpers.py script is not duplicated?

Inference script: inference.py

1. Create an **inference.py** script
2. Read the **df_test_nolabel.csv** stored at the first step
3. Read the **preprocessor.pkl** stored during training
4. Read the **trained_model_YYYYmmdd_HHMM.pkl** stored during training
5. Preprocess the test data
6. Infer the trained model on the test set.
7. Store the results on Google Cloud Storage under the name: **predictions.csv**

2.2. Docker images

Create a folder docker, with three subfolders: feature_engineering, training, inference.

1. Create a Dockerfile to wrap the preprocessing script. Store it in docker/feature_engineering folder.
2. Create a Dockerfile to wrap the training script. Store it in docker/training folder.
3. Create a Dockerfile to wrap the inference script. Store it in docker/inference folder.

Once you have created the Dockerfiles, you made the modifications to the script, you can test it:

- Go to Vertex AI on Google Cloud Platform and start your workbench instance ("Open Jupyter-lab")
- Create a folder 02_homework
- Upload the following input files: df_train.csv, feature_engineering.py, helpers.py, requirements.txt
- Open a terminal session within the jupyter-lab
- build and run your Docker images!

Deliverables

1. requirements.txt: file containing the installations dependencies
2. feature_engineering.py
3. training.py
4. inference.py
5. docker/feature_engineering/Dockerfile

6. docker/training/Dockerfile
7. docker/inference/Dockerfile