import random

import twml

get\_time\_based\_dataset\_files = twml.util.list\_files\_by\_datetime

def resolve\_train\_and\_eval\_files\_overlap(

train\_files, eval\_files, fraction\_kept\_for\_eval, seed=None

):

"""Resolve any overlap between train and eval files.

Specifically, if there's an overlap between `train\_files` and `eval\_files`, then a fraction of

the overlap (i.e. `fraction\_kept\_for\_eval`) will be randomly assigned (exclusively) to the

`eval\_files`.

The following example demonstrates its usage:

>>> orig\_train\_files = ['f1', 'f2', 'f3', 'f4']

>>> orig\_eval\_files = ['f1', 'f2', 'f3']

>>> resolved\_train\_files, resolved\_eval\_files = resolve\_train\_and\_eval\_files\_overlap(

... orig\_train\_files, orig\_eval\_files, 0.5

... )

>>> set(resolved\_train\_files) & set(resolved\_eval\_files) == set()

True

>>> len(resolved\_train\_files) == 3

True

>>> len(resolved\_eval\_files) == 2

True

Args:

train\_files: A list of the files used for training.

eval\_files: A list of the files used for validation.

fraction\_kept\_for\_eval: A fraction of files in the intersection between `train\_files` and

`eval\_files` exclusively kept for evaluation.

seed: A seed for generating random numbers.

Returns:

A tuple `(new\_train\_files, new\_eval\_files)` with the overlapping resolved.

"""

rng = random.Random(seed)

train\_files = set(train\_files)

eval\_files = set(eval\_files)

overlapping\_files = train\_files & eval\_files

train\_files\_selected\_for\_eval = set(rng.sample(

overlapping\_files,

int(len(overlapping\_files) \* fraction\_kept\_for\_eval)

))

train\_files = train\_files - train\_files\_selected\_for\_eval

eval\_files = (eval\_files - overlapping\_files) | train\_files\_selected\_for\_eval

return list(train\_files), list(eval\_files)

def get\_time\_based\_dataset\_files\_for\_train\_and\_eval(

base\_path,

train\_start\_datetime,

train\_end\_datetime,

eval\_start\_datetime,

eval\_end\_datetime,

fraction\_kept\_for\_eval,

datetime\_prefix\_format='%Y/%m/%d/%H',

extension='lzo',

parallelism=1

):

"""Get train/eval dataset files organized with a time-based prefix.

This is just a convenience built around `get\_dataset\_files\_prefixed\_by\_time` and

`resolve\_train\_and\_eval\_files\_overlap`. Please refer to these functions for documentation.

"""

train\_files = get\_time\_based\_dataset\_files(

base\_path=base\_path,

start\_datetime=train\_start\_datetime,

end\_datetime=train\_end\_datetime,

datetime\_prefix\_format=datetime\_prefix\_format,

extension=extension,

parallelism=parallelism

)

eval\_files = get\_time\_based\_dataset\_files(

base\_path=base\_path,

start\_datetime=eval\_start\_datetime,

end\_datetime=eval\_end\_datetime,

datetime\_prefix\_format=datetime\_prefix\_format,

extension=extension,

parallelism=parallelism

)

return resolve\_train\_and\_eval\_files\_overlap(

train\_files=train\_files,

eval\_files=eval\_files,

fraction\_kept\_for\_eval=fraction\_kept\_for\_eval

)