Download and preprocess Criteo TB dataset

Apache Beam enables distributed preprocessing of the dataset and can be run on Google Cloud Dataflow. The preprocessing scripts can be run locally via DirectRunner provided that the local host has enough CPU/Memory/Storage.

Install required packages.

```
python3 setup.py install
```

Set up the following environment variables, replacing bucket-name with the name of your Cloud Storage bucket and project name with your GCP project name.

```
export STORAGE_BUCKET=gs://bucket-name
export PROJECT=my-gcp-project
export REGION=us-central1
```

Note: If running locally above environment variables won't be needed and instead of gs://bucket-name a local path can be used, also consider passing smaller `max_vocab_size` argument.

1. Download raw Criteo TB dataset to a GCS bucket.

Organize the data in the following way:

- The files day_0.gz, day_1.gz, . . . , day_22.gz in ${STORAGE_BUCKET}/criteo_raw/train/
- The file day_23.gz in ${STORAGE_BUCKET}/criteo_raw/test/

2. Shard the raw training/test data into multiple files.

```
python3 shard_rebalancer.py \
  --input_path "${STORAGE_BUCKET}/criteo_raw/train/*" \
  --output_path "${STORAGE_BUCKET}/criteo_raw_sharded/train/train" \
  --num_output_files 1024 --filetype csv --runner DataflowRunner \
  --project ${PROJECT} --region ${REGION}
```

```
python3 shard_rebalancer.py \
  --input_path "${STORAGE_BUCKET}/criteo_raw/test/*" \
  --output_path "${STORAGE_BUCKET}/criteo_raw_sharded/test/test" \
  --num_output_files 64 --filetype csv --runner DataflowRunner \
  --project ${PROJECT} --region ${REGION}
```

3. Generate vocabulary and preprocess the data.

Generate vocabulary:

```
python3 criteo_preprocess.py \
  --input_path "${STORAGE_BUCKET}/criteo_raw_sharded/*/*" \
  --output_path "${STORAGE_BUCKET}/criteo/" \
  --temp_dir "${STORAGE_BUCKET}/criteo_vocab/" \
  --vocab_gen_mode --runner DataflowRunner --max_vocab_size 5000000 \
  --project ${PROJECT} --region ${REGION}
```

Vocabulary for each feature is going to be generated to `${STORAGE_BUCKET}/criteo_vocab/tftransform_tmp/f` files. Vocabulary size can be found as `wc -l <feature_vocab_file>`.

Preprocess training and test data:

```
python3 criteo_preprocess.py \
  --input_path "${STORAGE_BUCKET}/criteo_raw_sharded/train/*" \
  --output_path "${STORAGE_BUCKET}/criteo/train/train" \
  --temp_dir "${STORAGE_BUCKET}/criteo_vocab/" \
  --runner DataflowRunner --max_vocab_size 5000000 \
  --project ${PROJECT} --region ${REGION}
```

```
python3 criteo_preprocess.py \
  --input_path "${STORAGE_BUCKET}/criteo_raw_sharded/test/*" \
  --output_path "${STORAGE_BUCKET}/criteo/test/test" \
  --temp_dir "${STORAGE_BUCKET}/criteo_vocab/" \
  --runner DataflowRunner --max_vocab_size 5000000 \
  --project ${PROJECT} --region ${REGION}
```

4. (Optional) Re-balance the dataset.

```
python3 shard_rebalancer.py \
  --input_path "${STORAGE_BUCKET}/criteo/train/*" \
  --output_path "${STORAGE_BUCKET}/criteo_balanced/train/train" \
  --num_output_files 8192 --filetype csv --runner DataflowRunner \
  --project ${PROJECT} --region ${REGION}
```

```
python3 shard_rebalancer.py \
  --input_path "${STORAGE_BUCKET}/criteo/test/*" \
  --output_path "${STORAGE_BUCKET}/criteo_balanced/test/test" \
  --num_output_files 1024 --filetype csv --runner DataflowRunner \
  --project ${PROJECT} --region ${REGION}
```

At this point training and test data are in the buckets:

- `${STORAGE_BUCKET}/criteo_balanced/train/`
- `${STORAGE_BUCKET}/criteo_balanced/test/`

All other buckets can be removed.