Set region environment variable

```
REGION=us-central1
```

Download and unzip ML github data for demo

```
wget https://github.com/GoogleCloudPlatform/cloudml-samples/archive/master.zip
```

```
unzip master.zip
```

Navigate to the cloudml-samples-master > census > estimator directory.

ALL Commands must be run from this directory

```
cd ~/cloudml-samples-master/census/estimator
```

Develop and validate trainer on local machine

Get training data from public GCS bucket

```
mkdir data
```

```
gsutil -m cp gs://cloud-samples-data/ml-engine/census/data/* data/
```

Set path variables for local file paths

These will change later when we use AI Platform

```
TRAIN_DATA=$(pwd)/data/adult.data.csv
```

```
EVAL_DATA=$(pwd)/data/adult.test.csv
```

Run sample requirements.txt to ensure we're using same version of TF as sample

```
sudo pip install -r ~/cloudml-samples-master/census/requirements.txt
```

Run a local trainer

Specify output directory, set as variable

```
MODEL_DIR=output
```

Best practice is to delete contents of output directory in case

data remains from previous training run

```
rm -rf $MODEL_DIR/*
```

Run local training using gcloud

```
gcloud ai-platform local train --module-name trainer.task --package-path train
--job-dir $MODEL_DIR -- --train-files $TRAIN_DATA --eval-files $EVAL_DATA \
--train-steps 1000 --eval-steps 100
```

Run trainer on GCP AI Platform - single instance

Create regional Cloud Storage bucket used for all output and staging

```
gsutil mb -l $REGION gs://$DEVSHELL_PROJECT_ID-aip-demo
```

Upload training and test/eval data to bucket

```
cd ~/cloudml-samples-master/census/estimator

gsutil cp -r data gs://$DEVSHELL_PROJECT_ID-aip-demo/data
```

Set data variables to point to storage bucket files

```
TRAIN_DATA=gs://$DEVSHELL_PROJECT_ID-aip-demo/data/adult.data.csv

EVAL_DATA=gs://$DEVSHELL_PROJECT_ID-aip-demo/data/adult.test.csv
```

Copy test.json to storage bucket

```
gsutil cp ../test.json gs://$DEVSHELL_PROJECT_ID-aip-demo/data/test.json
```

Set TEST_JSON to point to the same storage bucket file

```
TEST_JSON=gs://$DEVSHELL_PROJECT_ID-aip-demo/data/test.json
```

Set variables for job name and output path

```
JOB_NAME=census_single_1

OUTPUT_PATH=gs://$DEVSHELL_PROJECT_ID-aip-demo/$JOB_NAME
```

Submit a single process job to AI Platform

Job name is JOB_NAME (census_single_1)

Output path is our Cloud storage bucket/job_name

Training and evaluation/test data is in our Cloud Storage bucket

```
gcloud ai-platform jobs submit training $JOB_NAME \
--job-dir $OUTPUT_PATH \
--runtime-version 1.4 \
--module-name trainer.task \
--package-path trainer/ \
--region $REGION \
-- \
--train-files $TRAIN_DATA \
--eval-files $EVAL_DATA \
--train-steps 1000 \
--eval-steps 100 \
--verbosity DEBUG
```

Can view streaming logs/output with gcloud ai-platform jobs stream-logs $JOB_NAME

When complete, inspect output path with gsutil ls -r $OUTPUT_PATH

Run distributed training on AI Platform

Create variable for distributed job name

```
cd ~/cloudml-samples-master/census/estimator
```

```
JOB_NAME=census_dist_1
```

Set new output path to Cloud Storage location using new JOB_NAME variable

```
OUTPUT_PATH=gs://$DEVSHELL_PROJECT_ID-aip-demo/$JOB_NAME
```

Submit a distributed training job

The '–scale-tier STANDARD_1' option is the new item that initiates distributed scaling

```
gcloud ai-platform jobs submit training $JOB_NAME \
--job-dir $OUTPUT_PATH \
--runtime-version 1.4 \
--module-name trainer.task \
--package-path trainer/ \
--region $REGION \
--scale-tier STANDARD_1 \
-- \
--train-files $TRAIN_DATA \
--eval-files $EVAL_DATA \
--train-steps 1000 \
--verbosity DEBUG \
--eval-steps 100
```

Prediction phase (testing it out)

Deploy a model for prediction, setting variables in the process

```
cd ~/cloudml-samples-master/census/estimator
```

```
MODEL_NAME=census
```

Create the ML Engine model

```
gcloud ai-platform models create $MODEL_NAME --regions=$REGION
```

Set the job output we want to use. This example uses census_dist_1

```
OUTPUT_PATH=gs://$DEVSHELL_PROJECT_ID-aip-demo/census_dist_1
```

CHANGE census_dist_1 to use a different output from previous

IMPORTANT - Look up and set full path for export trained model binaries

gsutil ls -r $OUTPUT_PATH/export

Look for directory $OUTPUT_PATH/export/census/ and copy/paste timestamp value (without colon) into the below command

```
MODEL_BINARIES=gs://$DEVSHELL_PROJECT_ID-aip-demo/census_dist_1/export/census/<timestamp> ###CHANGE ME!
```

Create version 1 of your model

```
gcloud ai-platform versions create v1 \
--model $MODEL_NAME \
--origin $MODEL_BINARIES \
--runtime-version 1.4
```

Send an online prediction request to our deployed model using test.json file

Results come back with a direct response

```
gcloud ai-platform predict \
--model $MODEL_NAME \
--version v1 \
--json-instances \
../test.json
```

Send a batch prediction job using same test.json file

Results are exported to a Cloud Storage bucket location

Set job name and output path variables

```
JOB_NAME=census_prediction_1

OUTPUT_PATH=gs://$DEVSHELL_PROJECT_ID-aip-demo/$JOB_NAME
```

Submit the prediction job

```
gcloud ai-platform jobs submit prediction $JOB_NAME \
--model $MODEL_NAME \
--version v1 \
--data-format TEXT \
--region $REGION \
--input-paths $TEST_JSON \
--output-path $OUTPUT_PATH/predictions
```

View results in web console at

```
gs://$DEVSHELL_PROJECT_ID-aip-demo/$JOB_NAME/predictions/
```