

```
# Export the classifier to a file
model = 'model.joblib'
joblib.dump(classifier, model)
# [END train-and-save-model]

# [START upload-model]
# Upload the saved model file to Cloud Storage
model_path = os.path.join('gs://', BUCKET_ID, 'model', datetime.datetime.
now().strftime(
    'iris_%Y%m%d_%H%M%S'), model)
subprocess.check_call(['gsutil', 'cp', model, model_path], stderr=sys.
stdout)
# [END upload-model]
```

Take note of the following points in the preceding code block:

- The code uses the ‘file.io’ module from the package ‘tensorflow.python.lib.io’ to stream a file stored on Cloud Storage.
- The rest of the code runs the classifier to build the model and exports the model to a bucket location on GCS. Cloud MLE will read from this bucket when building a prediction service for online predictions.

## Execute a Scikit-learn Training Job on Cloud MLE

The bash code for executing a training job for the Scikit-learn model is presented in the following and is saved in the file ‘single-instance-training.sh’.

```
export SCALE_TIER=BASIC # BASIC | BASIC_GPU | STANDARD_1 | PREMIUM_1 |
BASIC_TPU
DATE=`date '+%Y%m%d_%H%M%S'`
export JOB_NAME=iris_sklearn_$DATE
export GCS_JOB_DIR=gs://iris-sklearn/jobs/$JOB_NAME

echo $GCS_JOB_DIR

gcloud ml-engine jobs submit training $JOB_NAME \
    --stream-logs \
    --scale-tier $SCALE_TIER \
```

```

--runtime-version 1.8 \
--job-dir $GCS_JOB_DIR \
--module-name trainer.model \
--package-path trainer/ \
--region us-central1 \
--python-version 3.5

```

The following code runs a training job to build a Scikit-learn Random forest model.

```
source ./scripts/single-instance-training.sh
```

```
gs://iris-sklearn/jobs/iris_sklearn_20181119_000349
```

```
Job [iris_sklearn_20181119_000349] submitted successfully.
```

```

INFO    2018-11-19 00:03:51 -0500    service    Validating job
                                                requirements...
INFO    2018-11-19 00:03:52 -0500    service    Job creation request
                                                has been successfully
                                                validated.
INFO    2018-11-19 00:03:52 -0500    service    Job iris_sklearn_20181119_
000349 is queued.
INFO    2018-11-19 00:03:52 -0500    service    Waiting for job to be
                                                provisioned.
INFO    2018-11-19 00:03:54 -0500    service    Waiting for training
                                                program to start.
...
INFO    2018-11-19 00:05:19 -0500    master-replica-0    Module
                                                completed;
                                                cleaning up.
INFO    2018-11-19 00:05:19 -0500    master-replica-0    Clean up
                                                finished.
INFO    2018-11-19 00:05:19 -0500    master-replica-0    Task completed
                                                successfully.

```

```
endTime: '2018-11-19T00:09:38'
```

```
jobId: iris_sklearn_20181119_000349
```

```
startTime: '2018-11-19T00:04:29'
```

```
state: SUCCEEDED
```

## Create a Scikit-learn Prediction Service on Cloud MLE

The code for creating a prediction service is shown in the following, and is saved in the file 'create-prediction-service.sh'.

```
export MODEL_VERSION=v1
export MODEL_NAME=iris_sklern
export REGION=us-central1

# Create a Cloud ML Engine model
echo "Creating model..."
gcloud ml-engine models create $MODEL_NAME --regions=$REGION

# Create a model version
echo "Creating model version..."
gcloud ml-engine versions create $MODEL_VERSION \
  --model $MODEL_NAME \
  --config config.yaml
```

The preceding code references a configuration file 'config.yaml'. This file (as shown in the following) holds the configuration for the Scikit-learn model. Let's briefly go through the attributes listed:

- **deploymentUri:** This points to the bucket location of the Scikit-learn model.
- **runtime version:** This attribute specifies the Cloud MLE runtime version.
- **framework:** This attribute is of particular importance as it specifies the model framework in use; this can be `SCIKIT_LEARN`, `XGBOOST`, or `TENSORFLOW`. For this example, it is set to `SCIKIT_LEARN`.
- **pythonVersion:** This attribute specifies the Python version in use.