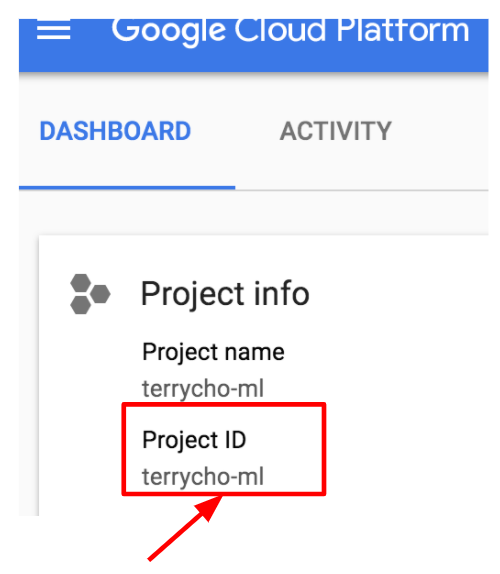
Machine Learning workshop

Hands on lab guide (Day #2-2)  
End 2 End Machine Learning Pipeline

This material is developed to help developer to understand end to end pipeline machine learning process. The source code is based on <https://github.com/GoogleCloudPlatform/training-data-analyst/blob/master/blogs/babyweight/babyweight.ipynb>

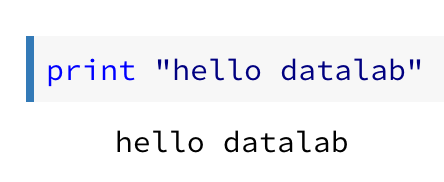
# Prerequisite

* Register google cloud (<http://cloud.google.com>)   
  Credit card (VISA or Master) is required
* Create a project
* After you create project you can see project id in the dashboard  
    
  Please remeber this “project ID” it will be used in LAB

# 

# 

# LAB 1. Prepare development environment

1. Run cloud shell in cloud console  
   
2. Run command “datalab create datalab”  
   (If you already created datalab instance, run command “datalab connect datalab”)
3. Click “Web preview” button in cloud shell console screen  
   
4. Select “+Notebook” button in datalab web browser
5. Type the following code in the code block in datalab and press Run!!  
   

# 

# 

# LAB 2. Exploring Data

1. Create new notebook with name “LAB2-Exploring-data”
2. Add the following code in code block and run the code

query="""

SELECT

weight\_pounds,

is\_male,

mother\_age,

mother\_race,

plurality,

gestation\_weeks,

mother\_married,

ever\_born,

cigarette\_use,

alcohol\_use,

FARM\_FINGERPRINT(CONCAT(CAST(YEAR AS STRING), CAST(month AS STRING))) AS hashmonth

FROM

publicdata.samples.natality

WHERE year > 2000

"""

import google.datalab.bigquery as bq

df = bq.Query(query + " LIMIT 100").execute().result().to\_dataframe()

df.head()

1. Check the result
2. Investigate relationship between mother race and baby weight. Add this code to code block

def get\_distinct\_values(column\_name):

sql = """

SELECT

{0},

COUNT(1) AS num\_babies,

AVG(weight\_pounds) AS avg\_wt

FROM

publicdata.samples.natality

WHERE

year > 2000

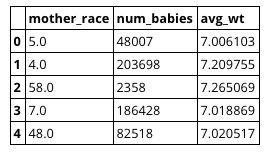
GROUP BY

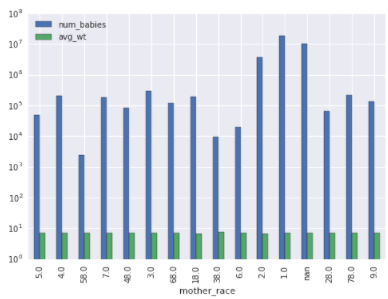
{0}

""".format(column\_name)

return bq.Query(sql).execute().result().to\_dataframe()

df = get\_distinct\_values('mother\_race')

df.head()  
  


1. Let’s draw graph. Add code in new code block  
   df.plot(x='mother\_race', logy='num\_babies', kind='bar');  
   
2. Lets investigate relationship between gender of baby and baby weight. Add code  
   df = get\_distinct\_values('is\_male')  
   df.plot(x='is\_male', logy='num\_babies', kind='bar');  
   df.plot(x='is\_male', y='avg\_wt', kind='bar');
3. Lets investigate relationship between mother’s age and baby weight. Add code and run  
   df = get\_distinct\_values('mother\_age')

df = df.sort\_values('mother\_age')

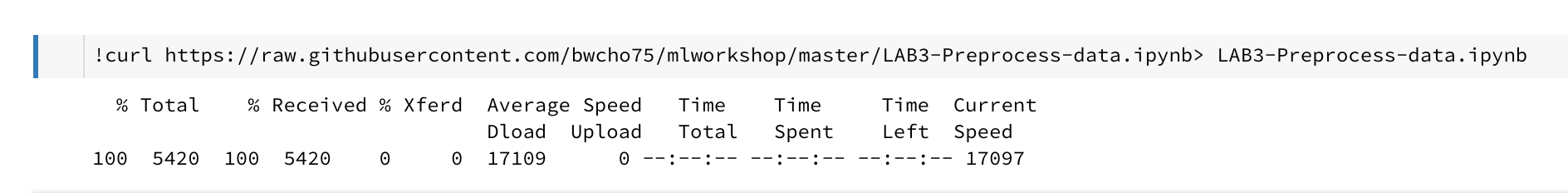
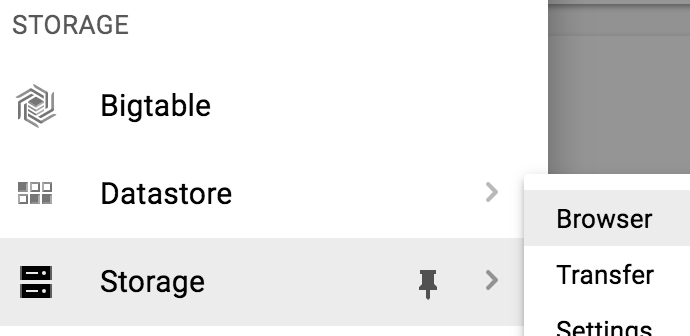
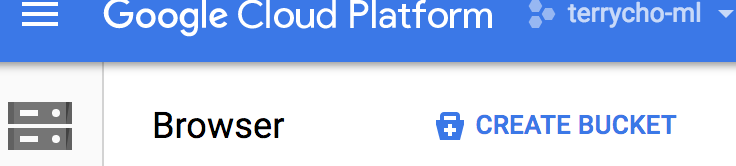
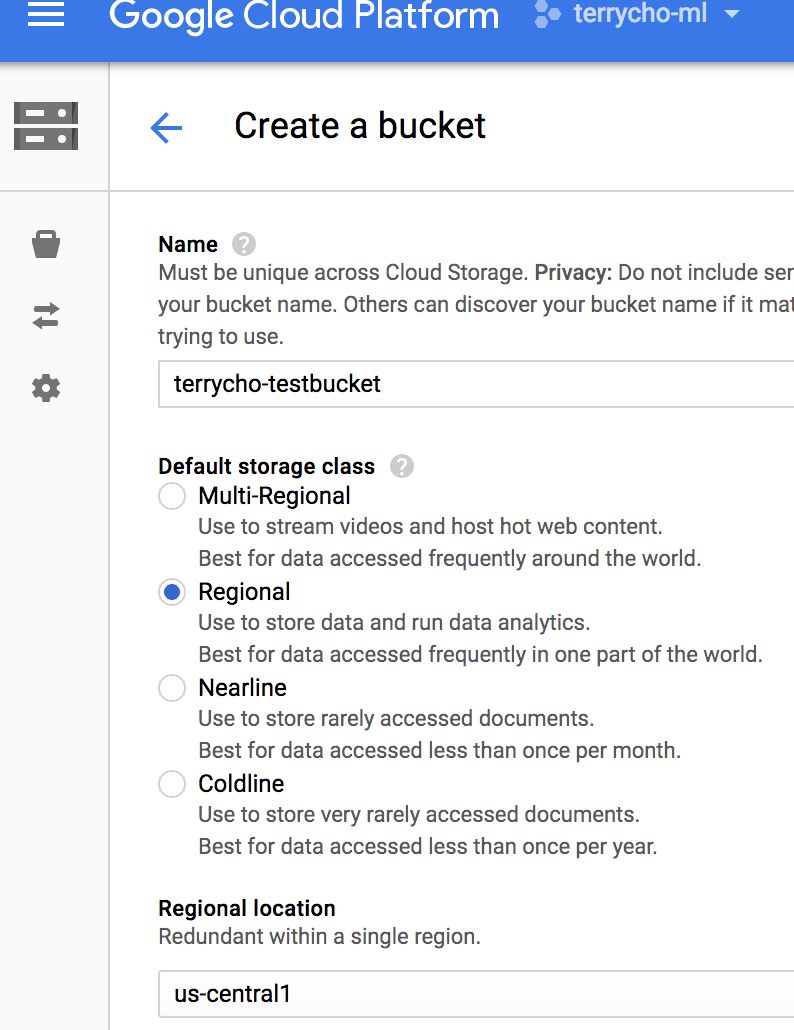
df.plot(x='mother\_age', y='num\_babies');

df.plot(x='mother\_age', y='avg\_wt');

1. Please investigate and find feature which is related with baby weight

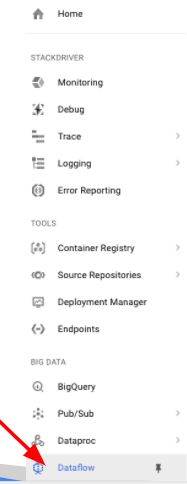
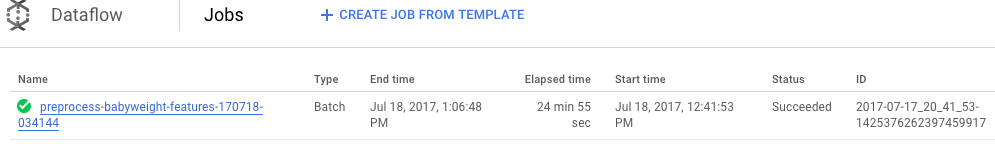
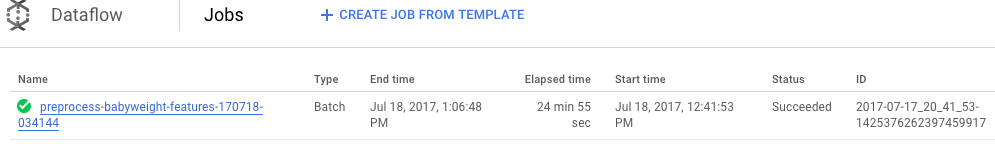
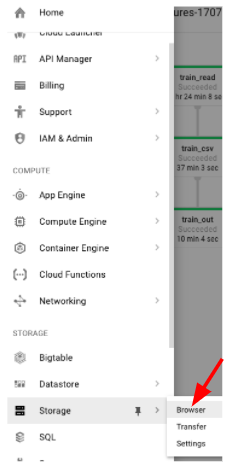
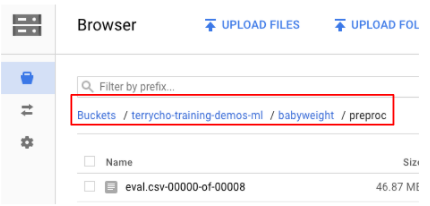
Reference

# LAB 3. Preprocess data

1. Create notebook named “workspace”
2. Add the code and run it. It will download sample code from git hub  
   “!curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/LAB3-Preprocess-data.ipynb> LAB3-Preprocess-data.ipynb”  
   
3. Move to list of notebook in datalab. You can see the notebook with name “[LAB3-Preprocess-data.ipynb](http://localhost:8081/notebooks/datalab/docs/Untitled%20Folder/LAB3-Preprocess-data.ipynb)”. And open it
4. In cloud console, please activate dateflow API
5. Create bucket to store process data  
   Move to google cloud web console and select “Storage > Browser” menu.  
     
   And then select “create bucket” button  
     
   Create bucket with name (you define the name )  
   And storage class with “Regional” and select location to “us-central1”  
   
6. Set environment variable in source code  
   BUCKET = '{Bucket name which you created in step 4}'

PROJECT = '{Your project id}’

REGION = '{Your project region}’'

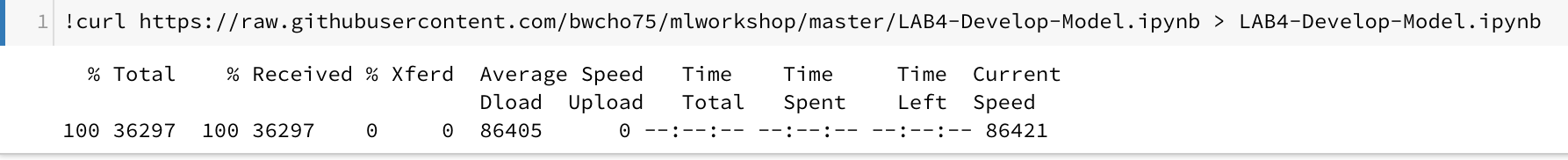
1. Run dataflow code by pressing “Run” button in datalab. It will run the dataflow process in local environment (in datalab) it is test purpose
2. Please change the code to run the code in dataflow in google cloud  
   preprocess(in\_test\_mode=True)  
   Into   
   preprocess(in\_test\_mode=False)
3. It will takes 20~30 min to process. Take a break!!
4. Check the google cloud console for dataflow running  
   
5. Click the running job  
   
6. Check the running status  
   
7. After all processings are done, check processed csv result file.  
   Go to Google Cloud Storage menu in Google Cloud Console  
   
8. Go to the bucket that you created in step 1.   
   
9. And open train.csv\* files and eval.csv\* file
10. All done, we have dataset for training and evaluation.

# 

# LAB 4. Develop Model

We will develop prediction model with Tensorflow

It will use wide & deep model and we will test the model with small # of training in datalab

1. Download notebook code  
   Open workspace notebook (which is made in LAB 3)  
   And add code “!curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/LAB4-Develop-Model.ipynb > LAB4-Develop-Model.ipynb”  
     
   And run it
2. Open downloaded notebook “LAB4-Develop-Model.ipynb”
3. Change the environment variable
4. Run code  
   BUCKET = '{Bucket name which you created in step 1}'

PROJECT = '{Your project id}’

REGION = '{Your project region}’'

1. Review source code

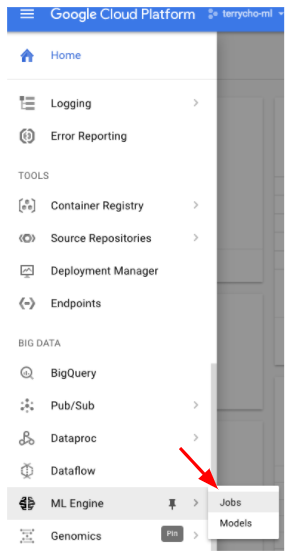
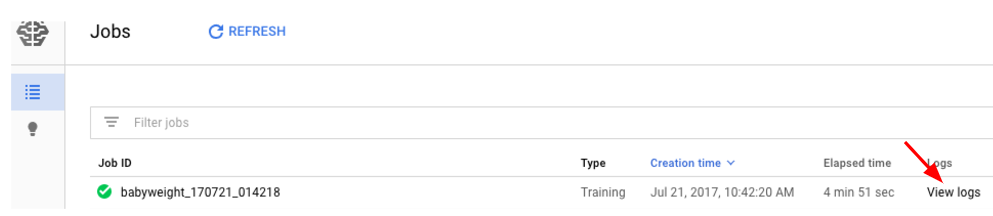
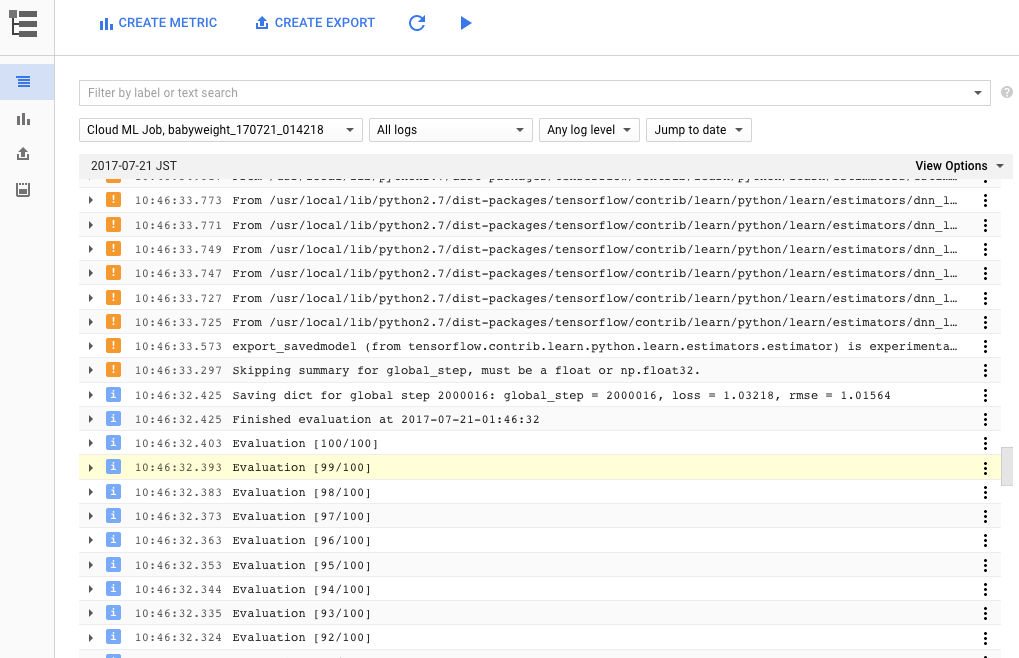
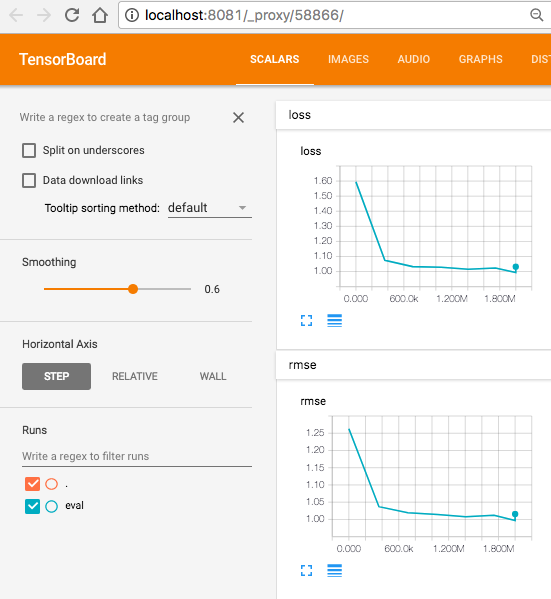
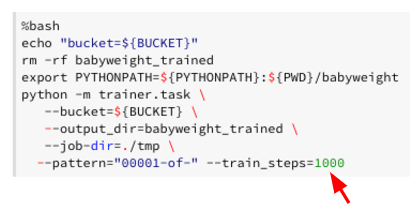
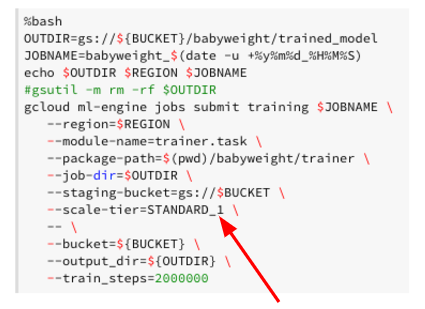
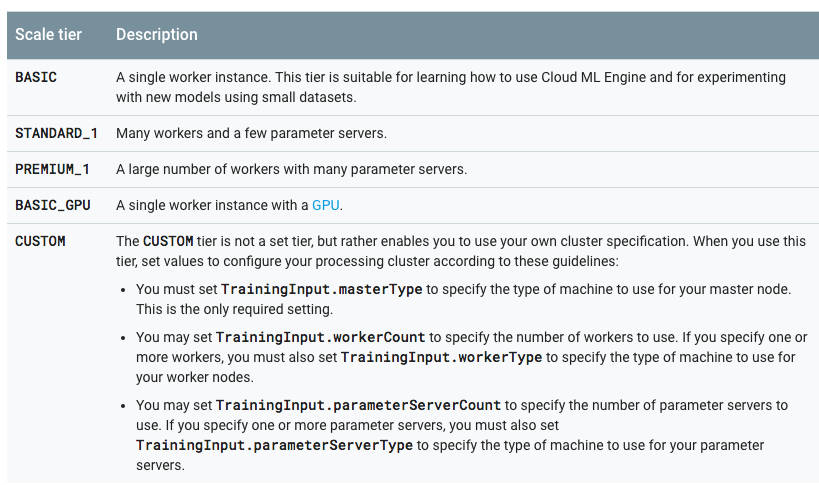
# LAB 5. Train model in CloudML

In previous lab, we made a model. Now we will train the model with huge size of data by using CloudML

1. Download code  
   Open “workpalce” note book  
   Add code block and add this code to the block “!curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/LAB5-Train-model-in-CloudML.ipynb > LAB5-Train-model-in-CloudML.ipynb”  
   And run it. It will download lab5 python notebook
2. Download model source code  
   Add code block and add this code to the block

|  |
| --- |
| !mkdir babyweight  !mkdir babyweight/trainer  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/setup.py > ./babyweight/setup.py  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/setup.cfg > ./babyweight/setup.cfg  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/PKG-INFO > ./babyweight/PKG-INFO  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/trainer/\_\_init\_\_.py > ./babyweight/trainer/\_\_init\_\_.py  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/trainer/model.py > ./babyweight/trainer/model.py  !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/babyweight/trainer/task.py > ./babyweight/trainer/task.py  !ls |

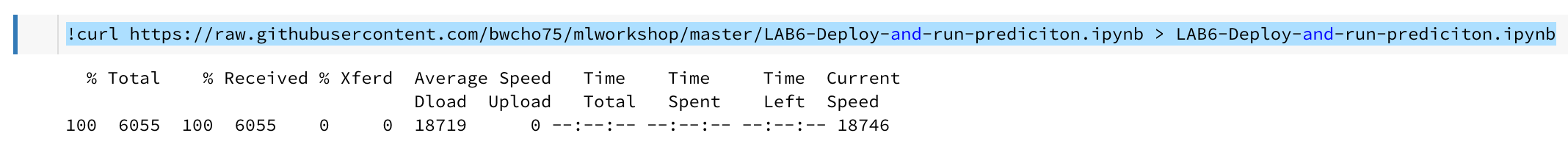
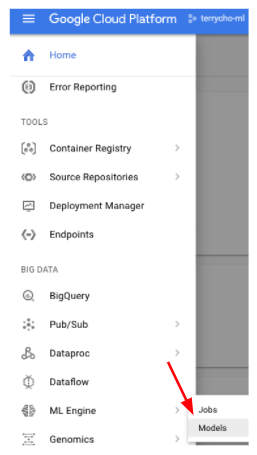
Move to downloaded “LAB5-Train-model-in-CloudML.ipynb”

1. Run training
2. During training, monitor training status in google Cloud Console  
     
   Move to jobs list and click “View Logs” for the job  
     
   Check the log  
   
3. After training done, please run Tensorboard  
   In the notebook, it has code to lunch tensorboard   
     
     
   
4. After training done, trained model will be automatically exported to Google Cloud Storage automatically. The file will be used by Tensorflow Serving.  
   Run the “Check exported model” block in notebook to check the exported file   
   
5. (Optional) for hands on lab purpose, we just run the training 1000 step only. In real production, it need to run the training more than that (200,000)  
   You can change the the training step here  
   
6. (Optional) Distributed learning and change Tensor Flow Cluster.  
   CloudML provides multiple options of training runtime. We used STANDARD\_1 type of cluster which has multiple worker. You can change the cluster type by changing “--scale-teir” option  
     
   Here is scale tier detail  
     
   (<https://cloud.google.com/ml-engine/docs/concepts/training-overview#scale_tier>)

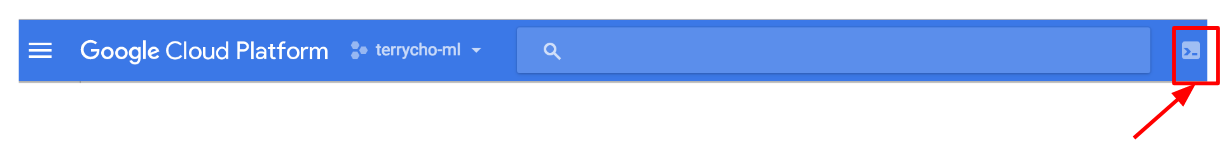
# LAB 6. Deploy and run prediction

We will deploy exported model to CloudML prediction engine.

The exported model from LAB 5 is stored in GCS.

1. Download code  
   Open “workpalce” note book  
   Add code block and add this code to the block “!curl !curl https://raw.githubusercontent.com/bwcho75/mlworkshop/master/LAB6-Deploy-and-run-prediciton.ipynb > LAB6-Deploy-and-run-prediciton.ipynb”  
     
   And run it. It will download lab5 python notebook
2. Move to downloaded “ LAB6-Deploy-and-run-prediciton.ipynb”
3. Change environment variable for your project
4. Run the codes which is deploying trained model to Cloud ML Prediction.
5. Check the cloudML/Model menu in console  
     
   Go to menu. You can find the model is deployed with name “babyweight” with version “v1”  
   
6. In note book, run “Use model to predict” part. It sends features to trained model and get prediction result.  
   This is prediction result  
   *response={u'predictions': [{u'outputs': 7.242420196533203}, {u'outputs': 6.703221321105957}, {u'outputs': 7.801552772521973}, {u'outputs': 6.150666236877441}]}*

# LAB 7. Build Prediction Web Application

1. Open cloud shell  
   
2. Download source code  
   git clone <https://github.com/GoogleCloudPlatform/training-data-analyst>  
   cd training-data-analyst/blogs/babyweight/application
3. Explore source code  
   This code is providing simple web service to predict baby weight with deployed  
     
   main.py : A Python script running on Google App Engine. It provides an API service that returns a prediction for a baby's weight. It uses the prediction API service deployed on the Cloud ML Engine in the background.  
   def predict() is function that predict with request from Web input by using deployed CloudML model  
     
   templates/form.html : A HTML file containing JavaScript code that renders the input form as in Figure 2. It sends a REST API request to the backend application running on Google App Engine and displays the result.
4. Install library  
   cd training-data-analyst/blogs/babyweight/application  
   pip install -r requirements.txt -t lib
5. Deploy application  
   gcloud app create --region=us-central  
   gcloud app deploy
6. Run application  
   https://[project id].appspot.com  
   