Leveraging Unstructured Data - Lab 6 : Add Machine Learning (ML) v1.3

## Overview

In this lab, you will use the Google Cloud API to incorporate several machine learning services from the Natural Language API, including sentiment analysis and entity analysis, to produce meaningful results from unstructured data.

[Cloud Machine Learning Engine](https://cloud.google.com/ml-engine/docs/) brings the power and flexibility of TensorFlow to the cloud, letting you perform large scale training on a managed cluster, and then scalably server your trained model for prediction.

## Objectives

In this lab, you will perform the following tasks:

* Add Machine Learning (ML) to a Spark application

## Task 1: Preparation

A Dataproc cluster has been prepared for you. If you login to GCP before the progress bar reports that the "Lab is Running", you may have to wait several minutes for the cluster to transition from "Provisioning" to "Running" before the cluster completes setup.

You will be performing most of the lab steps from the Master Node of the cluster in an SSH terminal window.

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **Dataproc** > **Clusters**.
2. Locate the cluster named **dataproc-cluster**. Which region and zone is it located in? The region and zone have been selected automatically for you by Qwiklabs.
3. Notice the Cloud Storage staging bucket defined for this cluster. This bucket has the same name as the project ID, which is a convenient way to make the name globally unique.
4. Click on the name **dataproc-cluster** to go to the Cluster details page.
5. The Cluster details page opens to the **Monitoring** tab. Click on the tab labeled **VM Instances**.
6. On the line for the VM named **dataproc-cluster-m** you will see that it has the Role of Master and there is an SSH link next to it. Click on **SSH** to open a terminal window to the Master Node.

## Task 2: Preparation

In this lab, you will be running **PySpark** applications. You will walk through a series of programs that gradually develops a Spark Dataproc application with Machine Learning elements.

In Lab 4, Task 2, you learned how to stage a PySpark application in Cloud Storage and submit it from the Dataproc Jobs page.

This lab uses a script to automate the customization and staging of the example Spark + Machine Learning applications. You need to set two environment variables before running the script; BUCKET and APIKEY.

### **Prepare the API Key**

The API Key in the programs is not valid. You must create a new one and replace it in the lab.

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **APIs & Services** > **Credentials**.
2. Click on **Create Credentials** and select **API Key**.
3. Copy the API Key. In the terminal, create an environment variable for easy recall of the key.

APIKEY=<your-api-key>

### **Verify the bucket to be used for Dataproc staging**

Dataproc can use a Cloud Storage bucket to stage its files during initialization. You can use this bucket to stage application programs or data for use by Dataproc. The bucket can also host Dataproc initialization scripts and output. The bucket name must be globally unique. Qwiklabs has already created a bucket for you that has the same name as the Project ID, which is already globally unique.

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **Storage** > **Browser**. Verify that the bucket exists. Notice the default storage class and the location (region) of this bucket. You will be using this region information next.

On the **dataproc-cluster-m** SSH terminal, set the BUCKET.

BUCKET=<bucket name>

### **Identify a project**

One environment variable that you will set is **$DEVSHELL\_PROJECT\_ID** that contains the Google Cloud project ID required to access billable resources.

In Cloud Shell, an environment variable is automatically created called DEVSHELL\_PROJECT\_ID that contains the GCP Project ID. This variable does not yet exist on the Master Node.

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **Home**. In the panel with Project Info, the **Project ID** is listed. You can also find this information in the Qwiklabs tab under Connection Details, where it is labeled **GCP Project ID**.

On the **dataproc-cluster-m** SSH terminal, set the DEVSHELL\_PROJECT\_ID.

DEVSHELL\_PROJECT\_ID=<project ID>

### **Verify required environment variables**

1. Verify that you have these environment variables are set. **Do not proceed until they are set**.

echo $DEVSHELL\_PROJECT\_ID, $BUCKET, $APIKEY

Export the BUCKET and APIKEY so they are available to the shell script.

export DEVSHELL\_PROJECT\_ID

export BUCKET

export APIKEY

### **Copy the application files to the dataproc-cluster-m home directory**

The sample files you need are have already been archived on dataproc-cluster-m. You will need to copy them into your user directory with the following command.

1. In the **dataproc-cluster-m** SSH terminal window.

cd

cp -r /training/training-data-analyst .

ls

### **Run the staging script**

In the terminal navigate to the source code directory for this lab.

cd ~/training-data-analyst/courses/unstructured/

Run the staging script:

./stagelabs.sh

This is what the staging script is doing:

* Edits the three python scripts **01-dataprocML.py**, **02-dataprocML.py**, **03-dataprocML.py**, and replaces the APIKEY, BUCKET, and DEVSHELL\_PROJECT\_ID with the values from the exported environment variables.
* Copies the updated files to your bucket in Cloud Storage, so that Dataproc can access them.
* Copies sample data files to your bucket.

Verify that the PySpark application files and sample data files are in the bucket.

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **Storage** > **Browser**.

Click Check my progress to verify the objective.

Copy the application files to the Master node home directory and run the staging script

Check my progress

## Task 3: Natural Language Processing

The three programs are "snapshots" from a development process. Each program builds on and enhances the one before it. Examining and running each program shows you how to progressively develop a Dataproc/Spark + Machine Learning application.

The sample data is unstructured data. That is, it either lacks structure, or it has a structure that is not suited to the intended purpose. In this lab you will use Machine Learning to identify and associate the data with values, giving it structure and making the data useful.

Navigate to this directory.

cd ~/training-data-analyst/courses/unstructured/

Examine **01-dataprocML.py** using an editor such as **nano**. Don't make any changes to the file.

1. This program is just a Python program. It will run on Dataproc, but it does not make use of any of the big data features. The program creates a sample line of text in memory and then passes it to the Natural Language Processing service for Sentiment Analysis.
2. The function SentimentAnalysis() is a wrapper around the REST API. This code creates the structured format of the request and passes the request along with the API Key.
3. Why is the output printed using a json.dumps?
4. You could do post-processing of the returned data using Python.
5. The stagelabs.sh script you ran in Task 1 should have replaced the DEVSHELL\_PROJECT\_ID, BUCKET, and APIKEY with your information from the environment variables.

### **Run the application**

1. In the Console, on **Navigation menu** ( 7a91d354499ac9f1.png) click **Dataproc** > **Jobs**. The click **SUBMIT JOB**.
2. You will need to select the region where your cluster is located, and the cluster, **dataproc-cluster**. The **Job Type** is **PySpark**.
3. In the field for **Main python file**, enter the path to the application file, which is something like this: **gs://<bucket name>/01-dataprocML.py**, where you replace <bucket name> with your bucket name.
4. Click **Submit**. View the output.

## Task 4: Load Sample Data

1. In the terminal, enter the following commands to copy sample files to the Cloud Storage bucket.

gsutil cp /training/road-not-taken.txt gs://$BUCKET/sampledata/road-not-taken.txt

1. In the Console, on the **Navigation menu** (7a91d354499ac9f1.png) click **Storage** > **Browser**.
2. Click on your bucket.
3. Click on **sampledata**.
4. Some files have already been staged.

## Task 5: Testing Sentiment Analysis with Spark

Examine **02-dataprocML.py** using an editor such as **nano**. Don't make any changes to the file.

1. This program uses Spark RDDs. It reads a small sample file and passes it to the Natural Language Processing service for Sentiment Analysis.
2. Post-processing of the returned data is done in the pipeline using transformations.

### **Run the application**

1. In the Console, on **Navigation menu** (7a91d354499ac9f1.png) click **BIG DATA** > **Dataproc** > **Jobs**. Then click **SUBMIT JOB**.
2. You will need to select the region where your cluster is located, and the cluster, **dataproc-cluster**. The **Job Type** is **PySpark**.
3. In the field for **Main python file**, enter the path to the application file, which is something like this: **gs://<bucket name>/02-dataprocML.py**, where you replace <bucket name> with your bucket name.
4. Click **Submit**. View the output.

## Task 6: Doing Something Useful

Examine **03-dataprocML.py** using an editor such as **nano**. Don't make any changes to the file.

1. This program builds on the previous one. Instead of reading a poem it is going to read an entire book. However, it could just as easily read and process an entire library.
2. Adds filter (in the pipeline) and sort (Python).
3. This gives a list of the lines in the book with the strongest sentiment, both positive and negative.
4. Now, this was just a book. Imagine how you could use this to sort through social media commentary. For example, consider the feedback left by customers on a shopping website. You could use this kind of data analysis to identify the most admired and most despised products.

### **Run the application**

1. In the Console, on **Navigation menu** ( 7a91d354499ac9f1.png) click **Dataproc** > **Jobs**. The click **SUBMIT JOB**.
2. You will need to select the region where your cluster is located, and the cluster, **dataproc-cluster**. The **Job Type** is **PySpark**.
3. In the field for **Main python file**, enter the path to the application file, which is something like this: **gs://<bucket name>/03-dataprocML.py**, where you replace <bucket name> with your bucket name.
4. Click **Submit**. View the output.

## Task 7: Other ML Services

The general pattern for using ML services from a Dataproc application can be adapted to use other ML services.