CA675 Cloud Technologies

Title	:	**Assignment 2**
Group Id	:	**K**
Student Name(s)	:	**Anup Bolli (18210820)** **Gautam Shanbhag (18210455)** **Janice Teresa Rodrigues (18213466)** **Navaneethan Rajasekaran(18210481)**
Email Address	:	**anup.bolli2@mail.dcu.ie** **gautam.shanbhag2@mail.dcu.ie** **janice.rodrigues4@mail.dcu.ie** **navaneethan.rajasekaran2@mail.dcu.ie**
Module Code	:	**CA675**
Submission Date	:	**19th April 2019**

Declaration

We declare that this material, which we now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations set out in the module documentation. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name: Date: 19th April 2019

Anup Bolli (18210820)
Gautam Shanbhag (18210455)
Janice Teresa Rodrigues (18213466)
Navaneethan Rajasekaran (18210481)

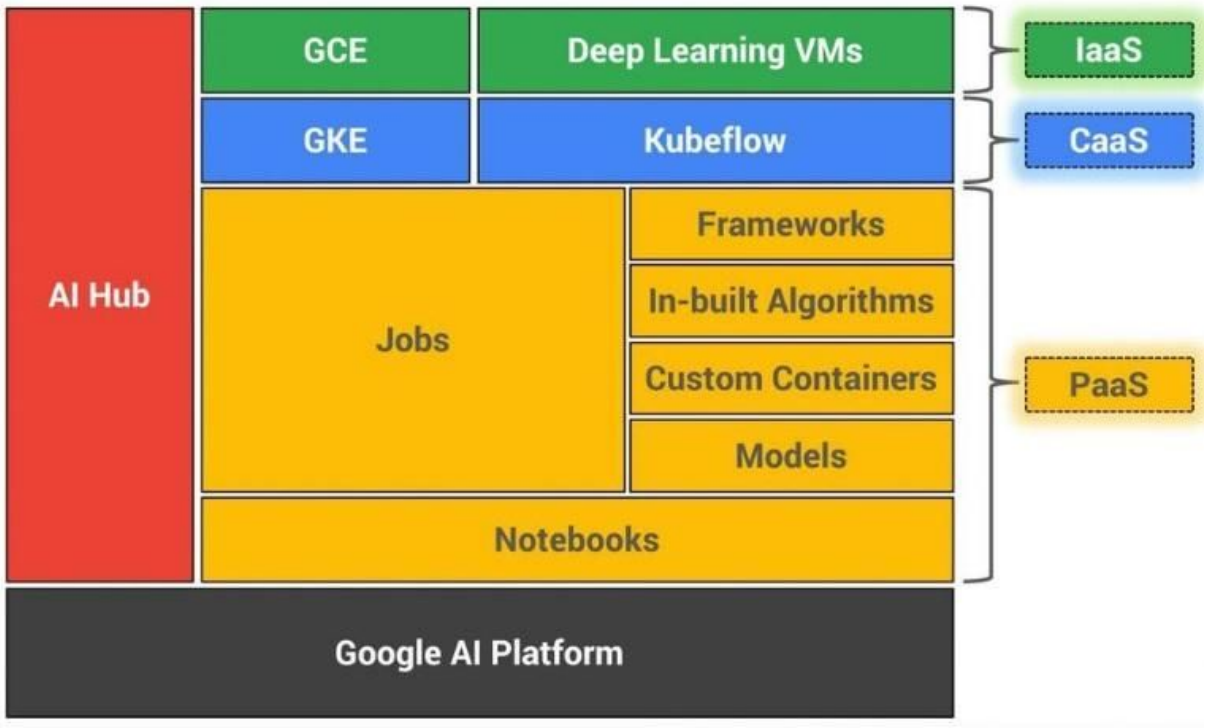
Contents

Introduction

The main objective is to explore and try out various cloud tools and technologies for Big data analytics and AI. We have planned to use GCP because of its various benefits when compared to other cloud platforms. It provides access to the numerous tools like Stackdriver Monitoring, ML engine, Stackdriver Logging, Security scanner service, to help us to focus on building application.

Google used its annual Cloud conference to shine a spotlight on its AI tools. At the core of all new tools and services, the company's plan is to democratize AI and machine learning with pre-built models, and easier to use services, while also giving more advanced developers the tools to build their own custom models.

We have compared Google AutoML with traditional methods normally adopted by Data Scientists in developing Machine Learning models.



Google AI Platform Architecture

Data – source selection, preparation, cleaning

Elo, one of the largest payment brands in Brazil, has built partnerships with merchants in order to offer promotions or discounts to cardholders. The dataset is openly available on Kaggle. Our machine learning model will predict Elo's Customer loyalty score based on the three features developed by feature engineering.

The dataset contains three CSV files with more than 2 million rows. We have used Jupyter notebook on GCP to clean, merge and transform the dataset.

first_active_month	feature_1	feature_2	feature_3
2017-06	5	2	1
2017-01	4	1	0
2016-08	2	2	0
2017-09	4	3	0
2017-11	1	3	0
2016-09	4	2	0
2016-12	3	2	1
2017-09	3	2	1

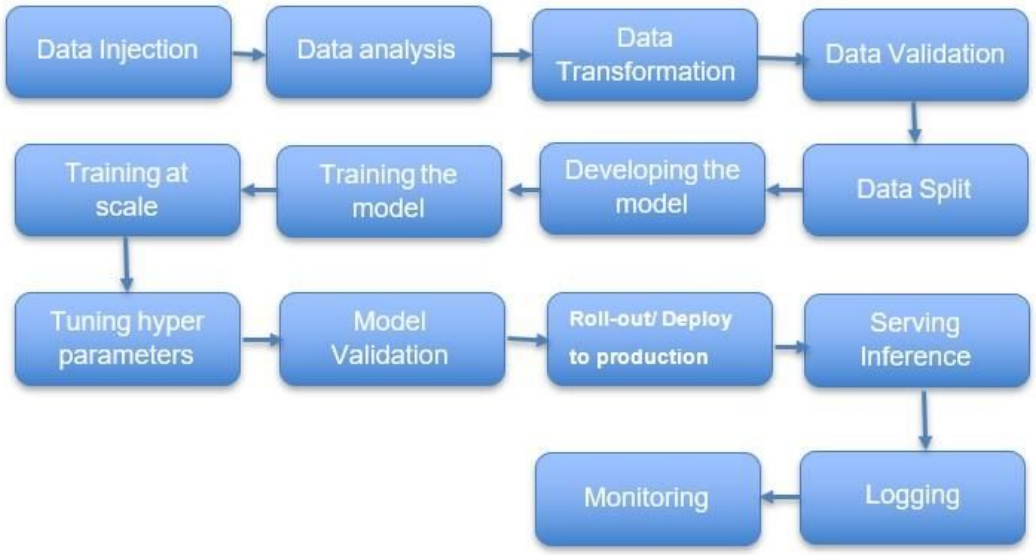
Dataset Snippet

Machine Learning in cloud

Artificial intelligence and machine learning are steadily making their way into enterprise applications in areas such as customer support, fraud detection, and business intelligence. There is every reason to believe that much of it will happen in the cloud.

The top cloud computing platforms are all betting big on democratizing artificial intelligence. Over the past three years, Amazon, Google and Microsoft have made significant investments in artificial intelligence (AI) and machine learning, from rolling out new services to carrying out major reorganizations that place AI strategically in their organizational structures.

Google Cloud Platform offer many machine learning options that don't require deep knowledge of AI, machine learning theory or a team of data scientists. We will try out the traditional approaches and the Google Cloud Technologies for AI below.



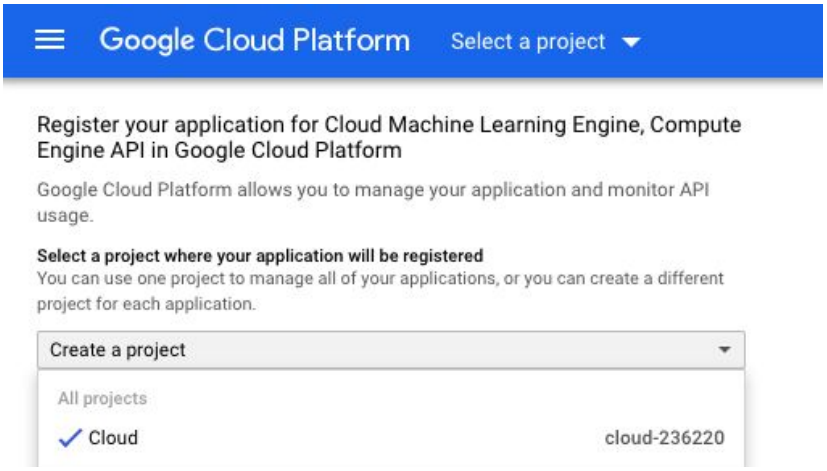
Machine Learning WorkFlow

I. Training Model Locally and getting online prediction

1. The Machine learning model was developed, fined tuned **locally** and exported as pickle file.
2. The exported pickle is stored in Google Bucket.

```
rnavaneeeth4@cloudshell:~ (cloud-236220)$ gsutil cp ./model.pkl gs://buc_06/model.pkl
```

3. Google Cloud project is created and AI Platform API's were enabled



4. Environment variables for the Cloud Storage directory that contains the model framework, and the name of the input file is set.

```
rnavaneeeth4@cloudshell:~ (cloud-236220)$ MODEL_DIR="gs://buc_06/"
rnavaneeeth4@cloudshell:~ (cloud-236220)$ INPUT_FILE="input.json"
rnavaneeeth4@cloudshell:~ (cloud-236220)$ FRAMEWORK="SCIKIT_LEARN"
rnavaneeeth4@cloudshell:~ (cloud-236220)$ VERSION_NAME='elov2'
```

5. GCloud can be used to deploy model for local predictions. This optional step helps us to save time by sanity-checking the model before deploying it to AI Platform.

```
rnavaneeeth4@cloudshell:~ (cloud-236220)$ gcloud ml-engine local predict --model-dir=$MODEL_DIR \
>     --json-instances $INPUT_FILE \
>     --framework $FRAMEWORK
```

6. The model is working fine and we got the expected prediction output.
7. The model resource in AI Platform, version of the model is created.The model version to the model file stored in Cloud Storage was linked

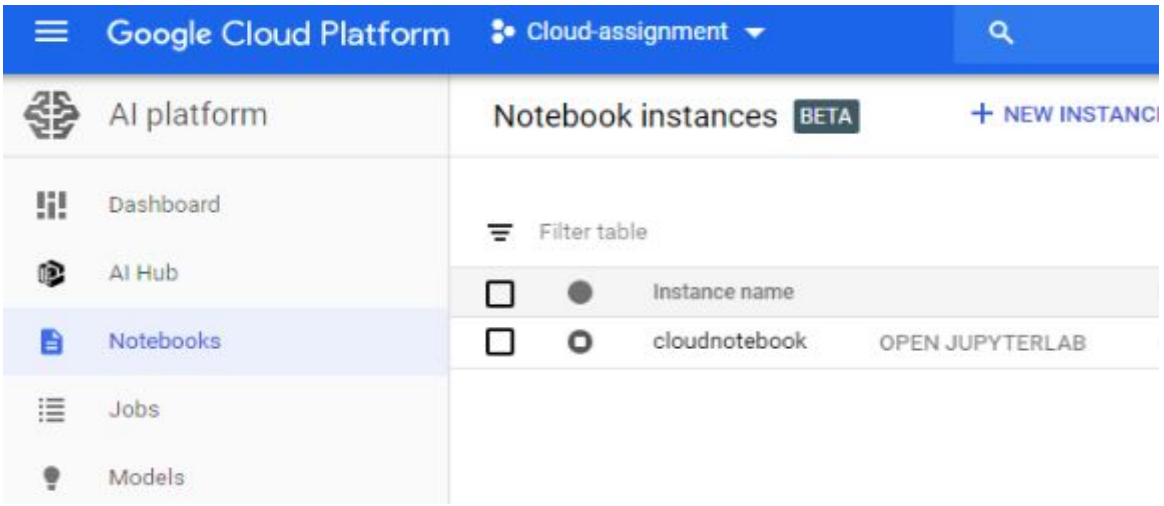
```
rnavaneeeth4@cloudshell:~ (cloud-236220)$ gcloud ml-engine versions create $VERSION_NAME \
>   --model $MODEL_NAME \
>   --origin $MODEL_DIR \
>   --runtime-version=1.13 \
>   --framework $FRAMEWORK \
>   --python-version=3.5
Creating version (this might take a few minutes).....done.
```

8. After successful creation of a model version, AI Platform starts a new server that is ready to serve prediction requests.
9. The Prediction request is sent and we get the following output.

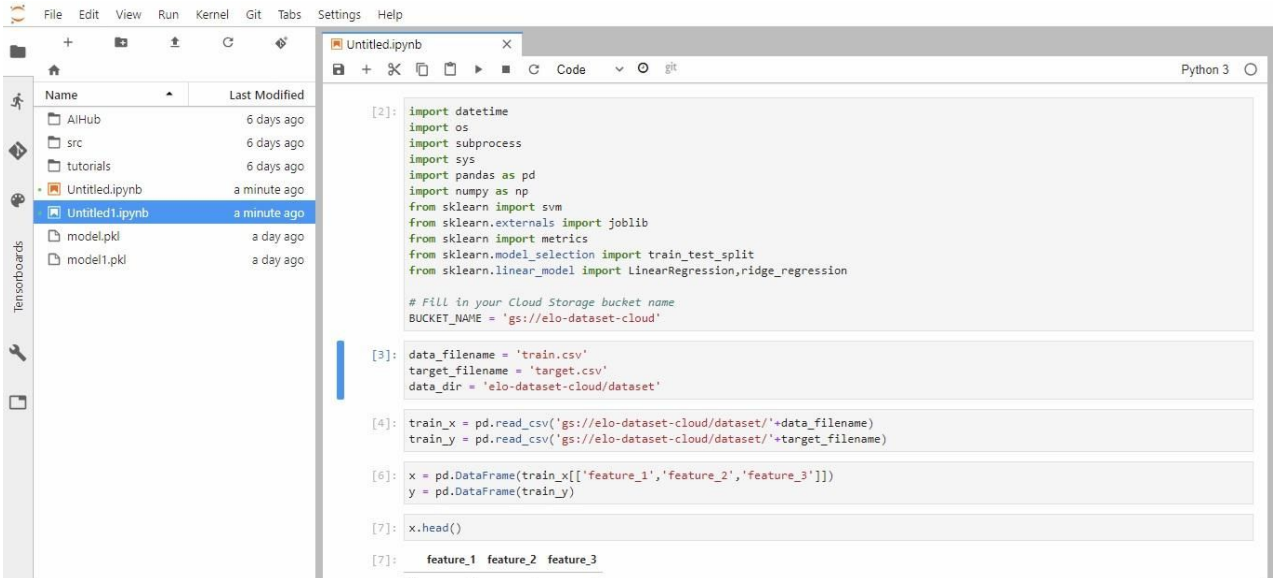
```
rnavaneeeth4@cloudshell:~ (cloud-236220)$ gcloud ml-engine predict --model $MODEL_NAME --version \
>   $VERSION_NAME --json-instances $INPUT_FILE
[[-2.1691783635864557]]
```

II. Training Model on cloud and getting prediction.

1. The Machine learning model is trained on Notebook instance provided by GCP. Notebook instances have JupyterLab pre-installed and are configured with GPU-enabled machine learning frameworks



2. Notebook instances are protected by Google Cloud Platform (GCP) authentication and authorization, and are available using a notebook instance URL. Notebook instances also integrate with GitHub, so that it can easily sync our notebook with a GitHub repository.



3. The trained model is exported as pickle file to the cloud bucket.

```
!gsutil cp ./model1.pkl gs://elo-dataset-cloud/jupyter_model/model1.pkl
```

4. Environment variables for the Cloud Storage directory that contains the model framework and the name of the input file has been set.
5. The model resource in AI Platform, version of the model is created.The model version to the model file stored in Cloud Storage was linked.

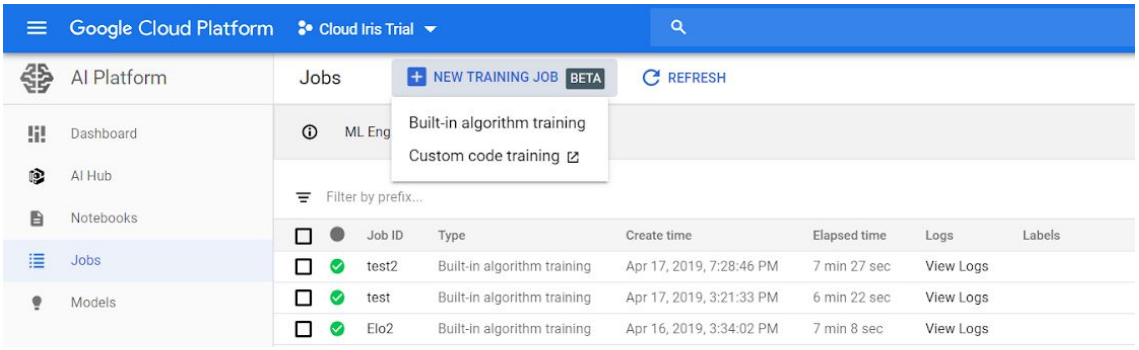
```
gautam_shanbhag2@cloudshell:~ (brave-sunspot-235818)$ gcloud ml-engine versions create $VERSION_NAME \
>   --model $MODEL_NAME \
>   --origin $MODEL_DIR \
>   --runtime-version=1.13 \
>   --framework $FRAMEWORK \
>   --python-version=3.5
Creating version (this might take a few minutes).....
```

6. After successful creation of a model version, AI Platform starts a new server that is ready to serve prediction requests.
7. The Prediction request is sent and we get the following output.

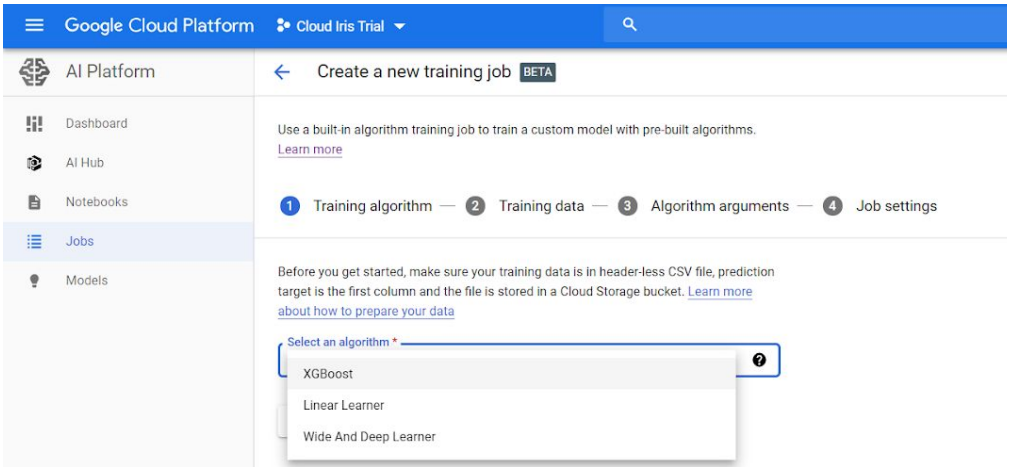
```
gautam_shanbhag2@cloudshell:~ (brave-sunspot-235818)$ gcloud ml-engine predict --model $MODEL_NAME --version \
>   $VERSION_NAME --json-instances $INPUT_FILE
[[-0.43703330338816737]]
```

III. Google AUTOML – Built-in Learner algorithm

1. To use built-in algorithm, header row must be removed from the CSV file and target field to be predicted, should be the first column.
2. Navigate to the AI platform jobs page in Google Cloud Platform Console. Click on the New training job button, Select built-in algorithm training.



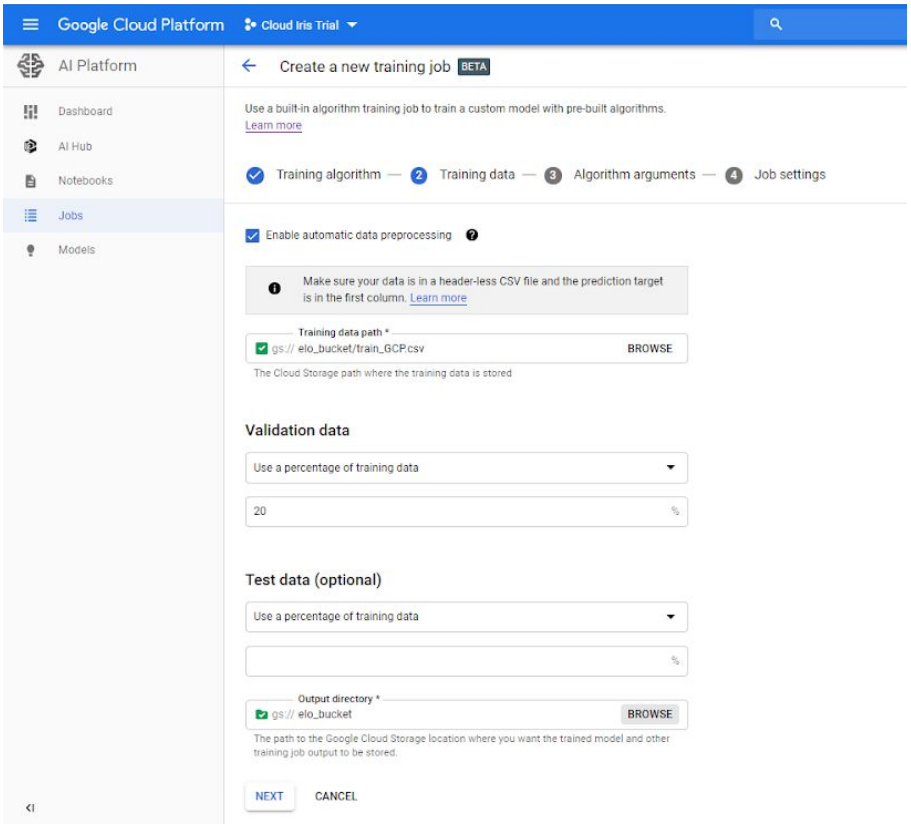
3. The Create a new training job page is displayed, where we have selected Linear learner algorithm for predicting loyalty score.



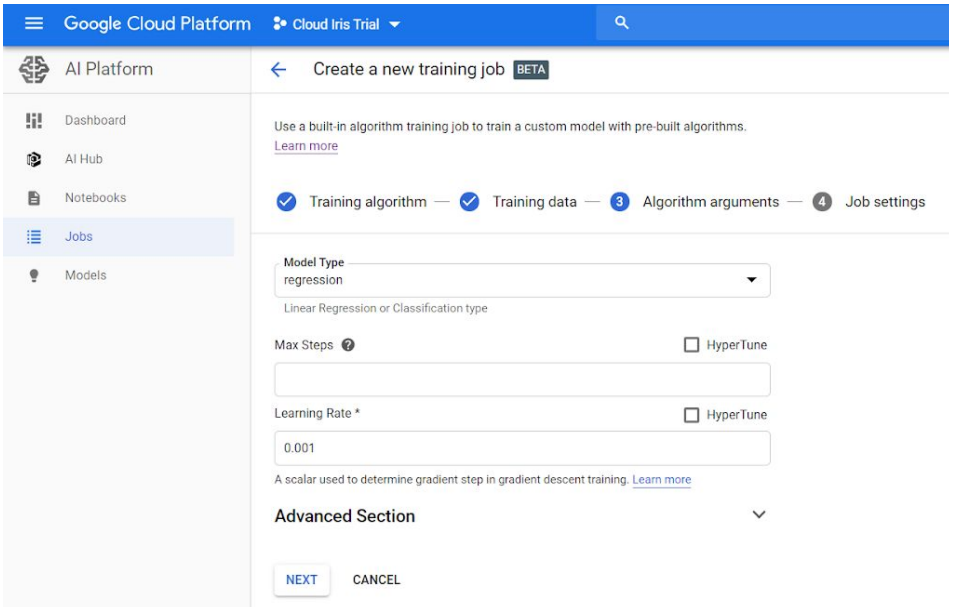
4. After selecting the training algorithm, the next is providing training data to the model.

   For Training data path, select the name of the cloud storage bucket where the training data is stored.

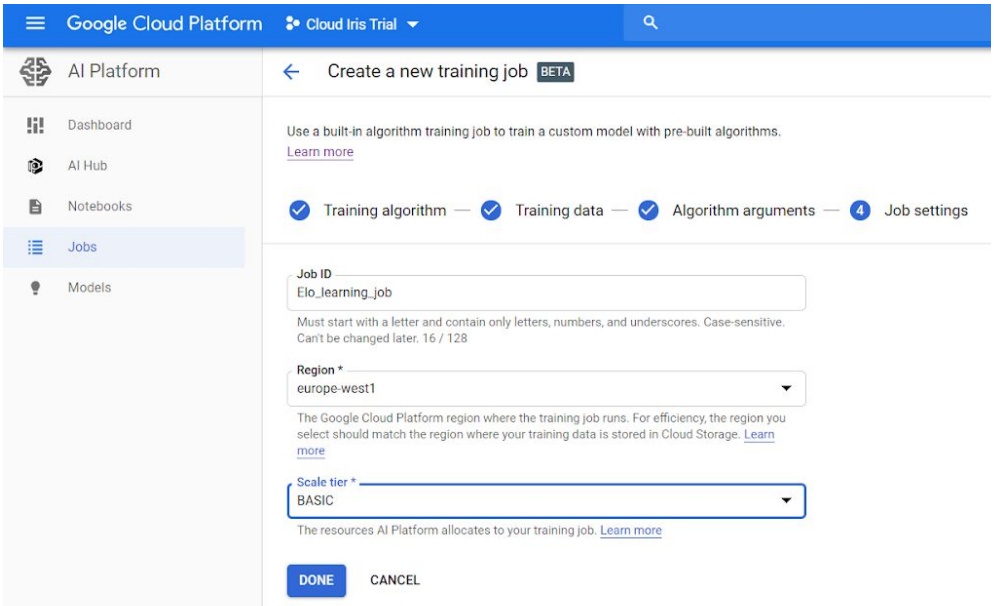
   In output directory, enter the path to your cloud storage bucket where the output of the algorithm is to be stored and click Next.



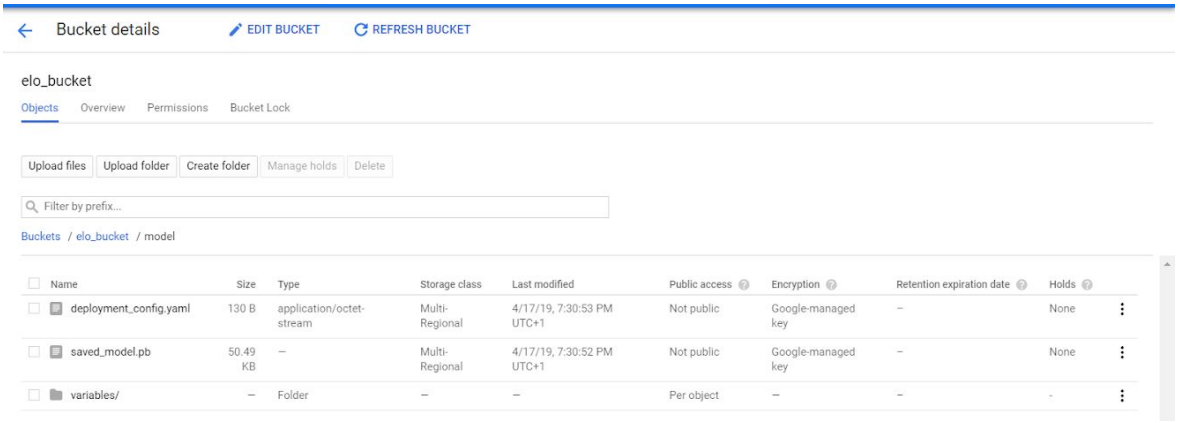
5. Next, we have selected algorithm arguments, such as Model type as Regression and left the other fields as default. Google also provides option to Hyper Tune the algorithm by allowing us to select Max Steps and Learning rate.



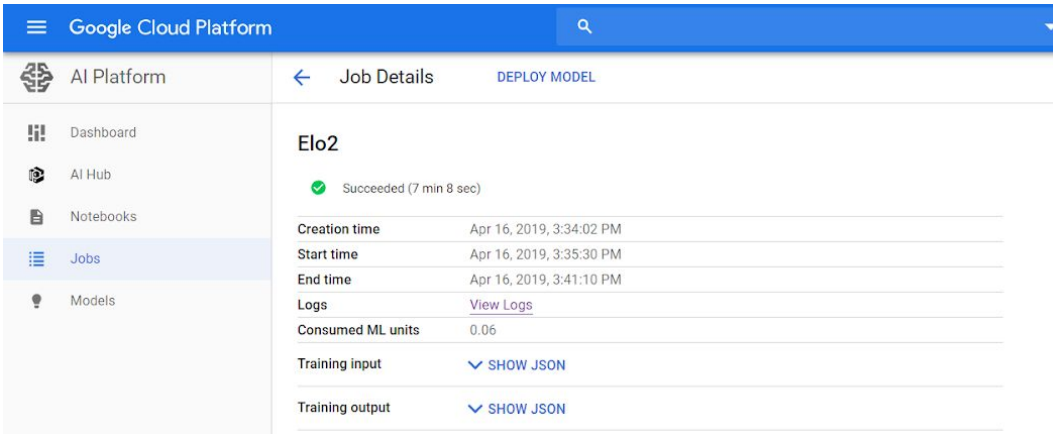
6. On the job setting tab, we have provided unique Job id, Region and Scale tier as Basic. Click done to submit the training job.

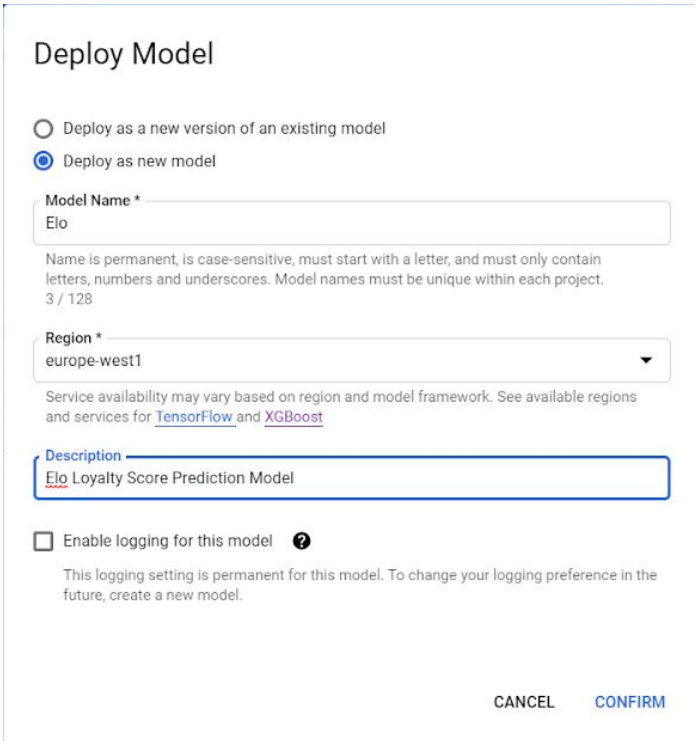


7. After successful completion of training job, Google AI creates a trained model in bucket.

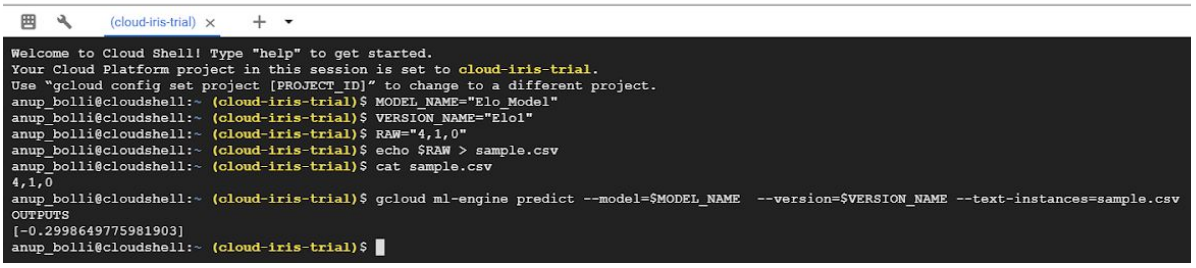


8. Deploying the trained model: Deploy Model Button appears on the job details page after job is executed successfully. Select Deploy Model button then we have provided the details of the model such as Name and version of the Model.





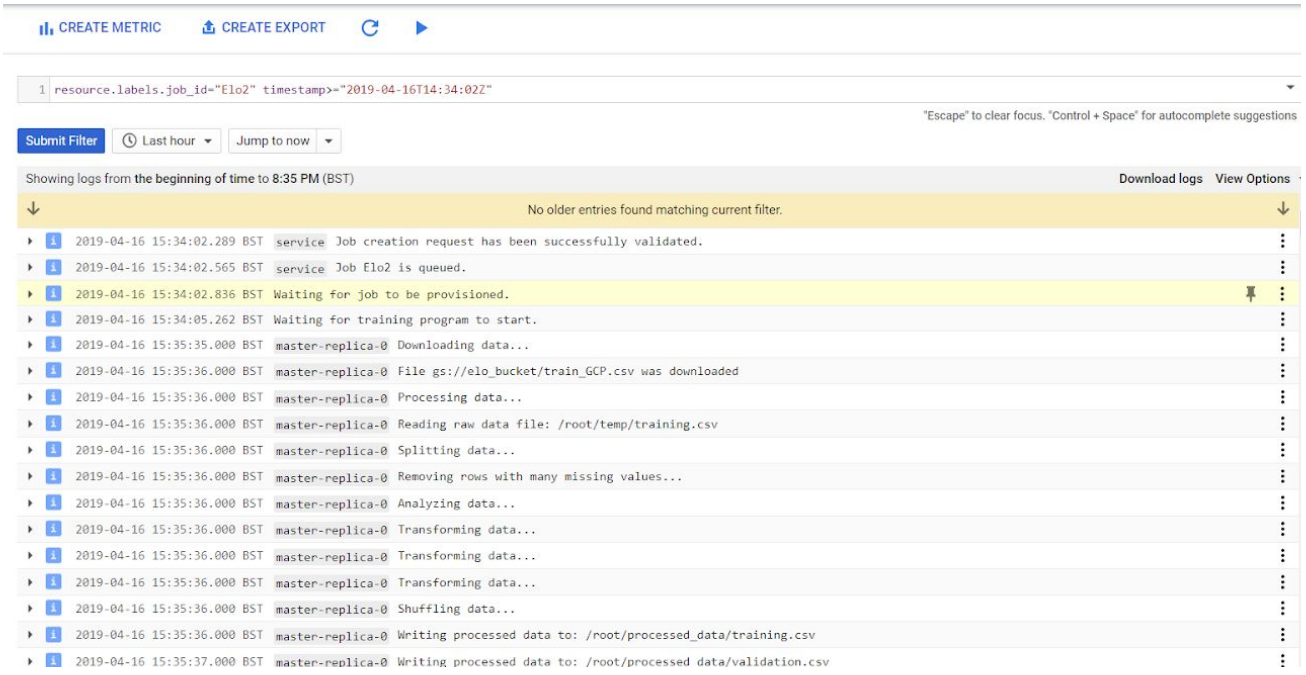
9. After successfully deploying the model, our model is ready for predicting loyalty score for new data.



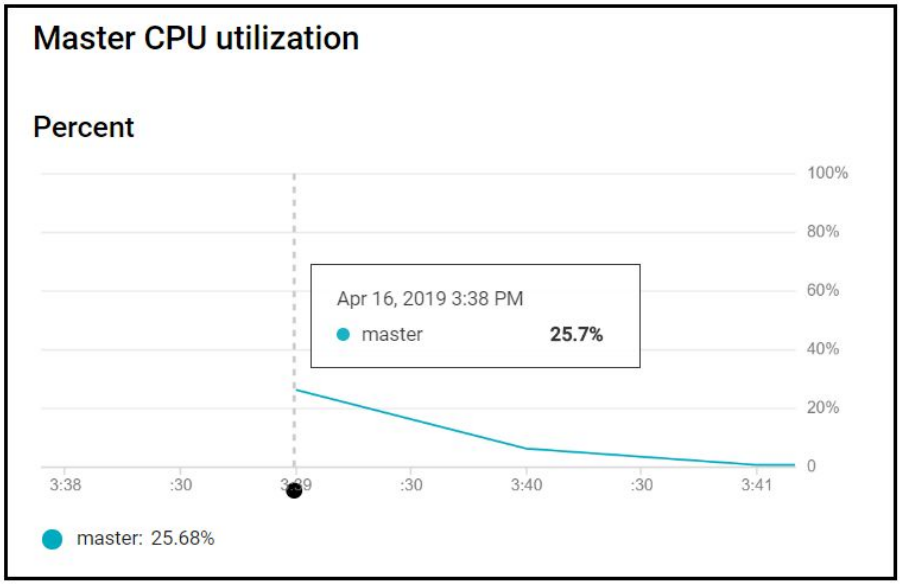
StackDriver Logging:

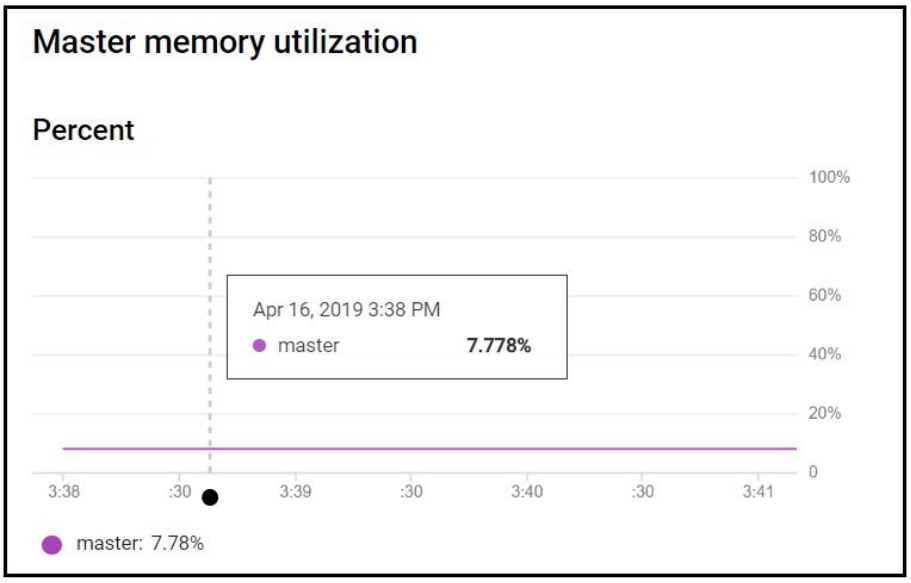
Stackdriver is a monitoring service that provides Developers with performance measurement of applications and virtual machines running on Google Cloud Platform and Amazon Web Services. Stackdriver Logging provides real-time analysis of log management and analysis of cloud applications.

The sample log generated by the StackDriver during the process of training the Elo Prediction model.



The CPU utilization is the percentage of the allocated CPU that is currently used, and Memory utilization is the percentage of the allocated memory that is currently used. These two metrics are be very useful for load balancing, provisioning of new computer resources and to avoid resource bottleneck.





Related Work

As Artificial Intelligence and Machine Learning are steadily making a significant impact on enterprise applications such as Fraud detection, Customer Analytics, Recommender Systems etc. The major cloud companies are trying their best to make these technologies available on cloud, offering general-purpose machine learning algorithms for training and deploying models to the cloud.

Amazon SageMaker: service provided by Amazon for building, training and deploying machine learning models at scale. It was developed for seamless adoption by Machine Learning community and easy deployment of trained models to production. It has 12 common machine learning algorithms, with TensorFlow, Apache MXNet pre-installed and option to bring other frameworks and libraries, training with “Zero-Configuration” workflow, out-of-the-box support for multi-node training and easy deployment to production.

Azure ML Studio: provides an interactive, visual workspace to easily develop, train and test on a predictive analysis model. It is as simple as drag and drop datasets and analysis modules onto interactive canvas connecting them to form an experiment. It supports various machine learning frameworks such as PyTorch, TensorFlow and scikit-Learn.

Challenges and lessons learned

Local Model Deployment	Online Model	Google AUTOML
Training and Tuning Model	Connecting Model with Google Storage	Understanding of ML Algorithm
Manual Configuration & Deployment	Permission & Security Issues which was hard to diagnose	Checking Accuracy & Precision
Big Data Computation Issues due to limited resources		Settings are often spread out in a non-intuitive way

Cloud technologies Used

Jupter Notebook

Google Storage

Google ML Engine

GCP Console

Stackdriver Logging & Monitoring

Responsibility

The work was divided equally amongst us

Response to peer feedback

As per the feedback provided by the peer group, we have focused on various cloud technologies rather than the quality of the Machine learning model.

Conclusion

After trying out different approaches to develop model locally, using cloud notebooks and using AutoML, we discovered that the driving force behind ML & AI development has become easier for technical & non-technical people, as more efforts were given to the problem statement rather than reinventing the wheel of developing ML Algorithms. The efforts invested, time taken for developing, solving the problem statement has drastically reduced by using GCP, which is clearly visible from our comparisons. Google's AutoML seems easy, convenient and user friendly as compared to Azure and AWS at this stage.

References

1. https://cloud.google.com/storage/docs/quickstart-gsutil
2. https://cloud.google.com/ml-engine/docs/
3. https://cloud.google.com/ml-engine/docs/scikit/quickstart
4. https://cloud.google.com/ml-engine/docs/scikit/working-with-cloud-storage