

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



HCMUTE

ĐỒ ÁN MÔN HỌC: ĐIỆN TOÁN Đám Mây
ĐỀ TÀI

TÌM HIỂU VỀ CÁC DỊCH VỤ HỖ TRỢ MACHINE LEARNING
TRÊN AWS VÀ VIẾT ỨNG DỤNG MINH HỌA

GVDH: Thầy Huỳnh Xuân Phụng

SVTH:

Nguyễn Minh Luân 19110395

Đào Quyết Phong 19110427

Thành phố Hồ Chí Minh – Tháng 10/2022

I. TÌM HIỂU VỀ CÁC DỊCH VỤ HỖ TRỢ MACHINE LEARNING

AWS cung cấp một tập hợp sâu rộng nhất gồm các dịch vụ máy học và cơ sở hạ tầng đám mây hỗ trợ, giúp từng nhà phát triển, nhà khoa học dữ liệu và người thao tác có chuyên môn sử dụng công nghệ máy học. Được vinh danh là đơn vị dẫn đầu trong báo cáo Magic Quadrant của Gartner về các dịch vụ dành cho nhà phát triển AI trên đám mây, AWS đang giúp hàng nghìn khách hàng đẩy nhanh hành trình máy học của họ.

Một số dịch vụ hỗ trợ học máy như:

- Amazon Polly
- Amazon Transcribe
- Amazon SageMaker
- Amazon Lex
- Amazon Rekognition
- Amazon Comprehend
- Amazon Translate

1. Amazon Polly

Amazon Polly là dịch vụ chuyển văn bản sang giọng nói chân thực, cho phép bạn tạo các ứng dụng có thể nói chuyện và phát triển những thể loại sản phẩm được trang bị khả năng nói hoàn toàn mới.

Polly là dịch vụ Text-to-Speech (TTS) (chuyển văn bản thành lời nói) sử dụng công nghệ deep learning tiên tiến để tổng hợp thành lời nói tự nhiên của con người. Với hàng chục giọng nói chân thực theo nhiều ngôn ngữ, bạn có thể xây dựng ứng dụng có giọng nói hoạt động ở nhiều nước khác nhau.

Ngoài giọng nói TTS tiêu chuẩn, Amazon Polly còn cung cấp giọng nói Neural Text-to-Speech (NTTS) (chuyển văn bản thành lời nói mạng nơ ron sâu), mang đến những cải tiến nâng cao về chất lượng lời nói thông qua phương pháp máy học mới. Công nghệ TTS Nơ-ron của Polly cũng hỗ trợ kiểu đọc Newscaster được thiết kế cho trường hợp tường thuật tin tức.

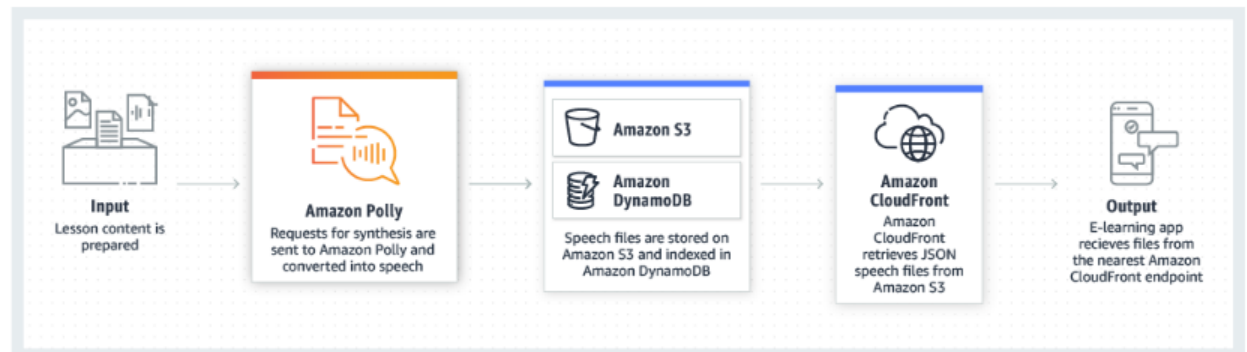
Cuối cùng, dịch vụ Amazon Polly Brand Voice (Giọng nói thương hiệu Amazon Polly) có thể tạo giọng nói tùy chỉnh cho tổ chức của bạn. Đây là một hợp đồng dịch vụ tùy chỉnh mà bạn sẽ làm việc với nhóm Amazon Polly để xây dựng giọng nói NTTS được sử dụng độc quyền cho tổ chức của bạn.

Ứng dụng:

- Tạo nội dung



- Nghe bài nói và làm nổi bật chữ được phát âm



2. Amazon Transcribe

Amazon Transcribe sử dụng quy trình học sâu có tên gọi là nhận dạng giọng nói tự động (ASR) để chuyển giọng nói thành văn bản một cách nhanh chóng và chính xác.

Ứng dụng:

- Nhận thông tin chuyên sâu từ các cuộc trò chuyện của khách hàng
- Tìm kiếm và phân tích nội dung đa phương tiện

- Tạo phụ đề và ghi chú cuộc họp
- Cải thiện tài liệu lâm sàng

Một số khách hàng sử dụng Amazon Transcribe



Intuit phân tích 274 triệu phút để cải thiện các cuộc trò chuyện với khách hàng »



Audioburst hàng ngày tạo ra 225.000 phút trò chuyện trực tiếp qua radio có thể tìm kiếm »



NASCAR tạo phụ đề cho nội dung video theo nhu cầu của mình với giá thấp hơn 97% so với ước tính »



Cerner sử dụng Amazon Transcribe Medical để xây dựng một công cụ ghi chép kỹ thuật số »

3. Amazon SageMaker

Amazon SageMaker là dịch vụ được quản lý toàn phần, mang đến cho mọi nhà phát triển và nhà khoa học dữ liệu khả năng xây dựng, đào tạo và triển khai các mô hình máy học (ML) một cách nhanh chóng.

Nhiều tính năng đa dạng và sâu rộng cho vòng đời ML



Hình. Tính năng của Amazon SageMaker

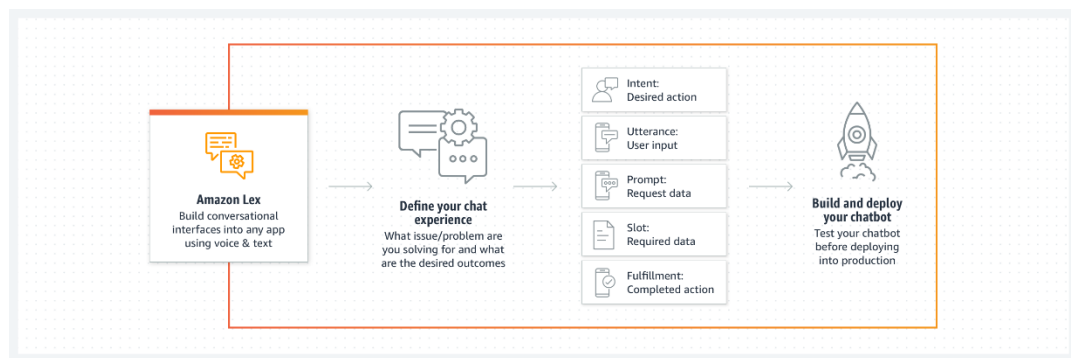
Hỗ trợ các khung ML, bộ công cụ và ngôn ngữ lập trình hàng đầu



4. Amazon Lex

Amazon Lex là dịch vụ xây dựng giao diện hội thoại trên bất cứ ứng dụng nào sử dụng giọng nói và văn bản.

Amazon Lex là một dịch vụ trí tuệ nhân tạo (AI) được quản lý hoàn toàn đi kèm với các mô hình ngôn ngữ tự nhiên nâng cao để thiết kế, xây dựng, kiểm thử và triển khai các giao diện trò chuyện trên các ứng dụng.



5. Amazon Rekognition

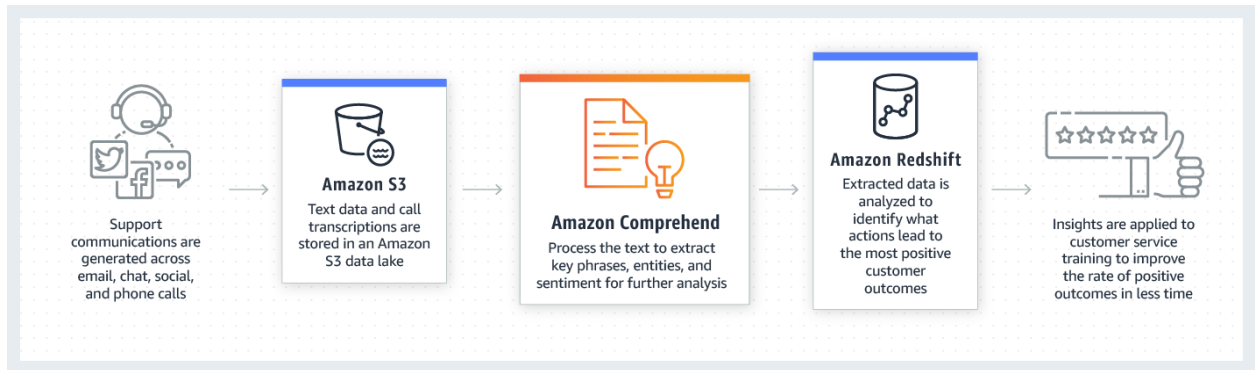
Amazon Rekognition giúp dễ dàng thêm phân tích hình ảnh và video vào ứng dụng của bạn bằng công nghệ học sâu đã được kiểm chứng, có quy mô linh hoạt cao mà không cần có chuyên môn về công nghệ máy học để sử dụng. Với Amazon Rekognition, bạn có thể nhận dạng các đối tượng, con người, văn bản, cảnh và hoạt động trong hình ảnh cũng như video, đồng thời phát hiện bất kỳ nội dung nào không phù hợp.

Một số tính năng:

- Kiểm duyệt nội dung
- Tìm kiếm và so sánh khuôn mặt
- Nhận diện và phân tích khuôn mặt
- Phát hiện phân đoạn, sự kiện trong video

6. Amazon Comprehend

Amazon Comprehend là dịch vụ xử lý ngôn ngữ tự nhiên (NLP) sử dụng công nghệ máy học để phát hiện thông tin chuyên sâu và mối quan hệ trong văn bản. Không yêu cầu kinh nghiệm sử dụng máy học.



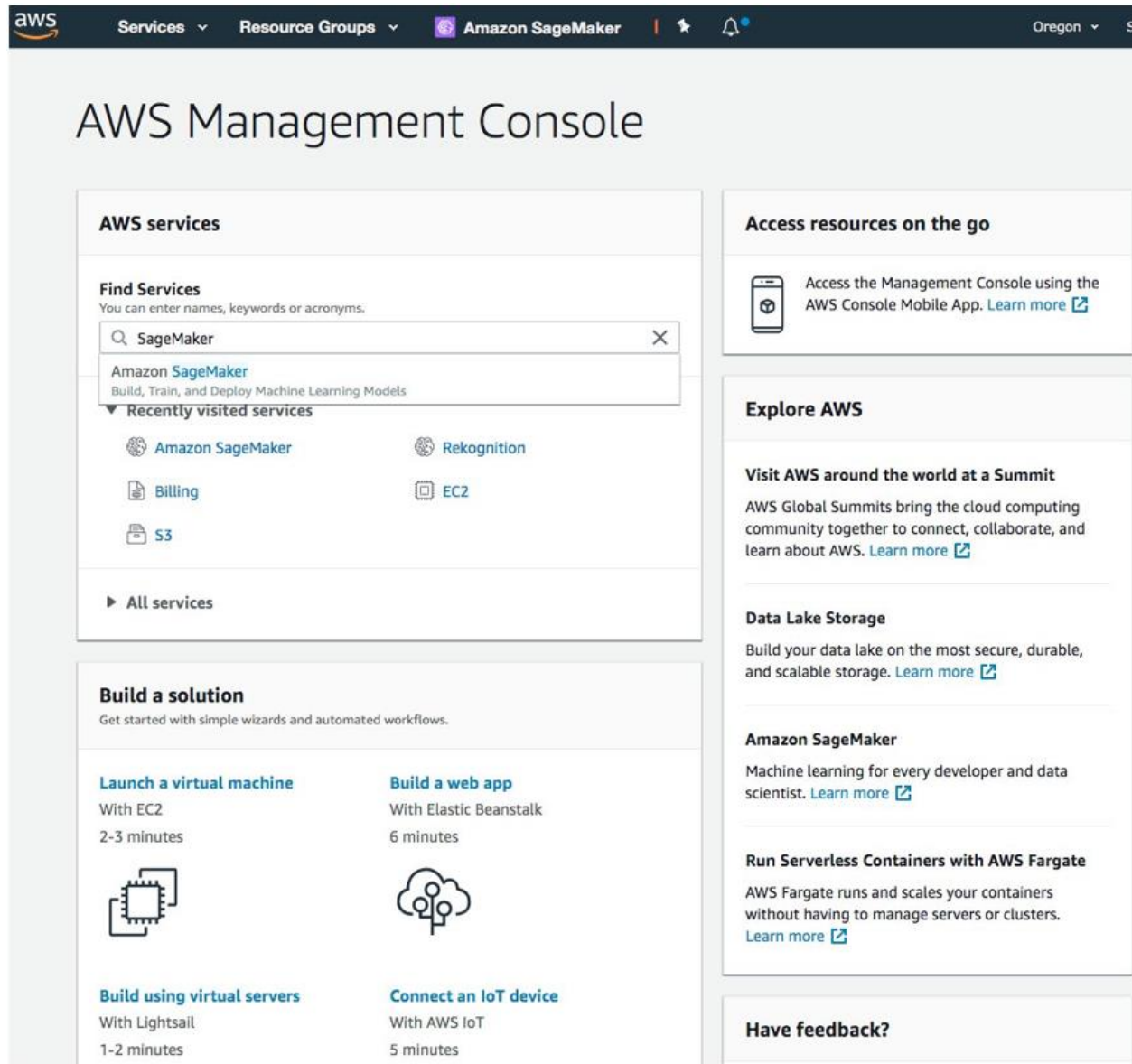
7. Amazon Translate

Amazon Translate là dịch vụ máy dịch ứng dụng mạng nơ-ron để cung cấp bản dịch ngôn ngữ nhanh chóng, chất lượng cao với giá cả phải chăng. Với Amazon Translate, bạn có thể bản địa hóa các nội dung như trang web và ứng dụng cho bộ phận người dùng đa dạng, dễ dàng dịch khối lượng lớn văn bản để phân tích, và cho phép giao tiếp đa ngôn ngữ một cách hiệu quả giữa nhiều người dùng.

II. ỨNG DỤNG MINH HỌA VỚI AMAZON SAGEMAKER

1. Xây dựng, đào tạo và triển khai mô hình Machine Learning bằng Amazon SageMaker

Bước 1. Mở bảng điều khiển Amazon SageMaker



Khi bạn nhấp vào đây, Bảng điều khiển quản lý AWS sẽ mở ra trong cửa sổ mới để bạn có thể giữ hướng dẫn từng bước này luôn mở. Bắt đầu nhập SageMaker vào thanh tìm kiếm và chọn Amazon SageMaker để mở bảng điều khiển dịch vụ.

Bước 2. Tạo phiên bản sổ ghi chép Amazon SageMaker

aws Services Resource Groups Amazon SageMaker Oregon Support

Amazon SageMaker X Amazon SageMaker > Dashboard

Overview [Hide](#)

Ground Truth

Set up and manage labeling jobs for highly accurate training datasets using active learning and human labeling.

[Labeling Jobs](#)

Notebook

Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.

[Notebook Instances](#)

Training

Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.

[Training Jobs](#)

Inference

Create models from training jobs or import external models for hosting to run inferences on new data.

[Models](#)

Recent activity Recent activity within the [Last 7 days](#)

No recent activity.

Learning Content

Amazon SageMaker 10-Minute Tutorial
Step-by-step guide to getting started faster [Learn more](#)

Amazon SageMaker for Telecommunications
Deployment guide for machine learning in telecommunications [Learn more](#)

Feature Spotlight

Amazon SageMaker How-to Blog
AWS machine learning experts showcase how to use Amazon SageMaker [Learn more](#)

Amazon SageMaker Automatic Model Tuning
Skip manual trial-and-error experimentation and

Amazon SageMaker > Notebook Instances > Create notebook instance

Create notebook instance

Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. [Learn more](#)

Notebook instance settings

Notebook instance name

MySageMakerInstance

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Notebook instance type

ml.t2.medium

Elastic Inference [Learn more](#)

none

► Additional configuration

Permissions and encryption

IAM role

Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

Choose an IAM role

Create a new role

Enter a custom IAM role ARN

Use existing role

Encryption key - optional

Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

S3 buckets you specify - optional

Specific S3 buckets

Example: bucket-name-1, bucket-name-2, bi

Comma delimited. ARNs, "*" and "/" are not supported.

Any S3 bucket

Allow users that have access to your notebook instance access to any bucket and its contents in your account.

None

Any S3 bucket with "sagemaker" in the name

Any S3 object with "sagemaker" in the name

Any S3 object with the tag "sagemaker" and value "true"

[See Object tagging](#)

S3 bucket with a Bucket Policy allowing access to SageMaker

[See S3 bucket policies](#)

Cancel

Create role

Permissions and encryption

IAM role
Notebook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the [AmazonSageMakerFullAccess](#) IAM policy attached.

AmazonSageMaker-ExecutionRole-20190816T163711

Success! You created an IAM role.

[AmazonSageMaker-ExecutionRole-20190816T163711](#)

Root access - optional

Enable - Give users root access to the notebook

Disable - Don't give users root access to the notebook

Lifecycle configurations always have root access

Encryption key - optional
Encrypt your notebook data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

► Network - optional

► Git repositories - optional

► Tags - optional

Cancel

Create notebook instance

Bước 3. Chuẩn bị dữ liệu

Amazon SageMaker

Dashboard
Search
Ground Truth
Labeling jobs
Labeling datasets
Labeling workflows

Amazon SageMaker > Notebook Instances

Notebook instances

Search notebook instances

Actions Create notebook instance

Name	Instance	Creation time	Status	Actions
MySageMakerInstance	ml.t2.medium	Aug 13, 2019 20:51 UTC	InService	Open Jupyter Open JupyterLab

Jupyter

Open JupyterLab Quit

Files Running Clusters SageMaker Examples Conda

Select items to perform actions on them.

Upload New

Notebook:

- R (Beta)
- Sparkmagic (PySpark)
- Sparkmagic (Spark)
- Sparkmagic (SparkR)
- conda_amazonai_mxnet_p27
- conda_amazonai_mxnet_p36
- conda_amazonai_tensorflow_p27
- conda_amazonai_tensorflow_p36
- conda_chainer_p27
- conda_chainer_p36
- conda_mxnet_p27
- conda_mxnet_p36
- conda_python2
- conda_python3
- conda_pytorch_p27
- conda_pytorch_p36
- conda_tensorflow_p27
- conda_tensorflow_p36

Other:

- Text File
- Folder
- Terminal

File Edit View Insert Cell Kernel Widgets Help

Trusted conda_python3

Code

```
In [1]: import boto3, re, sys, math, json, os, sagemaker
from sagemaker import get_execution_role
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import Image
from IPython.display import display
from time import gmtime, strftime
from sagemaker.predictor import csv_serializer

# Define IAM role
role = get_execution_role()
prefix = 'sagemaker/DEMO-xgboost-dm'
containers = {'us-west-2': '433757028032.dkr.ecr.us-west-2.amazonaws.com/xgboost:latest',
              'us-east-1': '811284229777.dkr.ecr.us-east-1.amazonaws.com/xgboost:latest',
              'us-east-2': '825641698319.dkr.ecr.us-east-2.amazonaws.com/xgboost:latest',
              'eu-west-1': '685385470294.dkr.ecr.eu-west-1.amazonaws.com/xgboost:latest'} # each region has its XGBoost
my_region = boto3.session.Session().region_name # set the region of the instance
print("Success - the MySageMakerInstance is in the " + my_region + " region. You will use the " + containers[my_region])

Success - the MySageMakerInstance is in the us-east-1 region. You will use the dkr.ecr.us-east-1.amazonaws.com/xgboost:latest container for your SageMaker endpoint.
```

In []:

```
In [2]: bucket_name = 'testyourname' # <--- change this variable to a unique name for your bucket
s3 = boto3.resource('s3')
try:
    if my_region == 'us-east-1':
        s3.create_bucket(Bucket=bucket_name)
    else:
        s3.create_bucket(Bucket=bucket_name, CreateBucketConfiguration={ 'LocationConstraint': my_region })
    print('S3 bucket created successfully')
except Exception as e:
    print('S3 error: ',e)
```

S3 bucket created successfully

```
In [3]: try:
        urllib.request.urlretrieve ("https://dl.awsstatic.com/tmt/build-train-deploy-machine-learning-model-sagemaker/bank_clean.csv",
        print('Success: downloaded bank_clean.csv.')
    except Exception as e:
        print('Data load error: ',e)

    try:
        model_data = pd.read_csv('./bank_clean.csv',index_col=0)
        print('Success: Data loaded into dataframe.')
    except Exception as e:
        print('Data load error: ',e)
```

Success: downloaded bank_clean.csv.
Success: Data loaded into dataframe.

```
In [4]: train_data, test_data = np.split(model_data.sample(frac=1, random_state=1729), [int(0.7 * len(model_data))])
print(train_data.shape, test_data.shape)
```

(28831, 61) (12357, 61)

Bước 4. Đào tạo mô hình từ dữ liệu

```
In [7]: xgb.fit({'train': s3_input_train})
```

```
[17:36:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 10 extra nodes, 14 pruned nodes, max_depth=5
[93]#011train-error:0.095314
[17:36:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 24 extra nodes, 30 pruned nodes, max_depth=5
[94]#011train-error:0.095314
[17:36:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 6 extra nodes, 24 pruned nodes, max_depth=3
[95]#011train-error:0.095314
[17:36:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 12 extra nodes, 30 pruned nodes, max_depth=5
[96]#011train-error:0.095279
[17:36:25] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 18 extra nodes, 12 pruned nodes, max_depth=5
[97]#011train-error:0.094828
[17:36:26] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 4 extra nodes, 22 pruned nodes, max_depth=2
[98]#011train-error:0.094863
[17:36:26] src/tree/updater_prune.cc:74: tree pruning end, 1 roots, 30 extra nodes, 12 pruned nodes, max_depth=5
[99]#011train-error:0.094759
```

2019-08-15 17:36:34 Uploading - Uploading generated training model
2019-08-15 17:36:34 Completed - Training job completed
Billable seconds: 56

In []:

Bước 5. Triển khai mô hình

```
In [20]: test_data_array = test_data.drop(['y_no', 'y_yes'], axis=1).values #load the data into an array
xgb_predictor.content_type = 'text/csv' # set the data type for an inference
xgb_predictor.serializer = csv_serializer # set the serializer type
predictions = xgb_predictor.predict(test_data_array).decode('utf-8') # predict!
predictions_array = np.fromstring(predictions[1:], sep=',') # and turn the prediction into an array
print(predictions_array.shape)
```

(12357,)

Bước 6. Đánh giá hiệu năng của mô hình

```
In [12]: cm = pd.crosstab(index=test_data['y_yes'], columns=np.round(predictions_array), rownames=['Observed'], colnames=['Predi
tn = cm.iloc[0,0]; fn = cm.iloc[1,0]; tp = cm.iloc[1,1]; fp = cm.iloc[0,1]; p = (tp+tn)/(tp+tn+fp+fn)*100
print("\n{0:<20}{1:<4.1f}%\n".format("Overall Classification Rate: ", p))
print("{0:<15}{1:<15}{2:>8}".format("Predicted", "No Purchase", "Purchase"))
print("Observed")
print("{0:<15}{1:<2.0f}% ({2:<}){3:>6.0f}% ({4:<})".format("No Purchase", tn/(tn+fn)*100,tn, fp/(tp+fp)*100, fp))
print("{0:<16}{1:<1.0f}% ({2:<}){3:>7.0f}% ({4:<}) \n".format("Purchase", fn/(tn+fn)*100,fn, tp/(tp+fp)*100, tp))
```

Overall Classification Rate: 89.5%

Predicted	No Purchase	Purchase
Observed		
No Purchase	90% (10785)	35% (151)
Purchase	10% (1143)	65% (278)

Bước 7. Chấm dứt tài nguyên

```
In [22]: sagemaker.Session().delete_endpoint(xgb_predictor.endpoint)
bucket_to_delete = boto3.resource('s3').Bucket(bucket_name)
bucket_to_delete.objects.all().delete()

Out[22]: [{"ResponseMetadata": {"RequestId": "28E42783694CB6C0",
'HostId': 'j7ajjzyseZHs5ruIrbmaKuXeJh/I7EjYGs9VADB9wY5HTbuG24AFetOjta603cWo39XPXBkf7XA=',
'HTTPStatusCode': 200,
'HTTPHeaders': {'x-amz-id-2': 'j7ajjzyseZHs5ruIrbmaKuXeJh/I7EjYGs9VADB9wY5HTbuG24AFetOjta603cWo39XPXBkf7XA=',
'x-amz-request-id': '28E42783694CB6C0',
'date': 'Thu, 15 Aug 2019 21:21:43 GMT',
'connection': 'close',
'content-type': 'application/xml',
'transfer-encoding': 'chunked',
'server': 'AmazonS3'},
'RetryAttempts': 0},
'Deleted': [{"Key": 'sagemaker/DEMO-xgboost-dm/output/xgboost-2019-08-15-18-39-29-757/output/model.tar.gz'},
{'Key': 'sagemaker/DEMO-xgboost-dm/train/train.csv'}]}
```

Việc chấm dứt các tài nguyên không được sử dụng chủ động sẽ làm giảm chi phí và là biện pháp tốt nhất. Nếu bạn không chấm dứt tài nguyên thì sẽ phát sinh thêm chi phí.