Group 5
Team Batak (TB)

apper.ph

Table of Contents

apper.ph

Meet the Team

Edward Vincent “**VINCE**” Duero

Fave Childhood Series: Naruto
Fave Series Now: Silicon Valley

Rosiel Jazmine “**ROSE**” Villareal

Fave Childhood Series: Avatar - The Last Airbender
Fave Series Now: The Good Place

Jericho Carlo “**ECHO**” Agudo

Fave Childhood Series: Detective Conan
Fave Series Now: Bridgerton

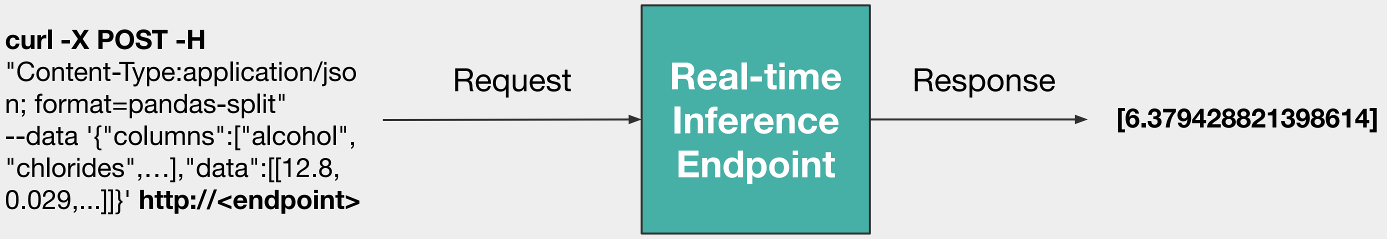
apper.ph

SageMaker Deployment Options

Real-time Inference Endpoint
Batch Transform

apper.ph

Real-Time Inference Endpoint



```
curl -X POST -H
"Content-Type:application/json; format=pandas-split"
--data '{"columns":["alcohol",
"chlorides",...],"data":[[12.8,
0.029,...]]}' http://<endpoint>
```

```
[6.379428821398614]
```

apper.ph

Use SageMaker Real-Time Inference

- For inference workloads when you have **real-time, interactive, low latency** requirements
- For providing a **persistent**, real-time endpoint

- When you want to **call an endpoint** that accepts data input and outputs a model prediction in real-time, i.e., it **gives you a response anytime you want**

apper.ph

SageMaker Real-time Inference Hosting Options

Single model

Multiple models in **one container** behind one endpoint

Multiple models for **serial inference pipeline** behind one endpoint

Multiple models w/c use **different containers** behind one endpoint

SageMaker Real-time Inference Features

Auto **Scale** Models

Host **Storage** Volumes (EBS)

Test Model **Versions** using Production Variants

Note: Specify traffic distribution or variants

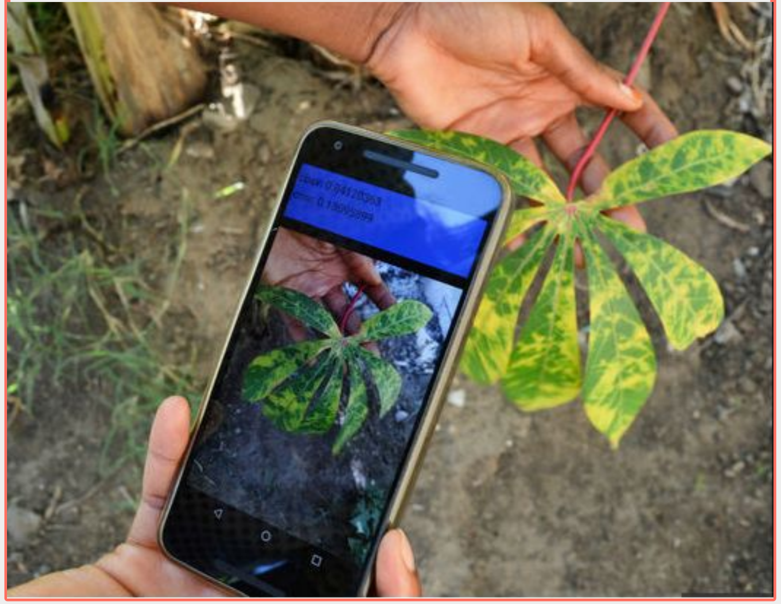
Monitor Drifts in Data, Model, Bias, Feature Attribution

Note: Currently supports only tabular data, single model endpoint

apper.ph

SageMaker Real-time Inference Use Cases

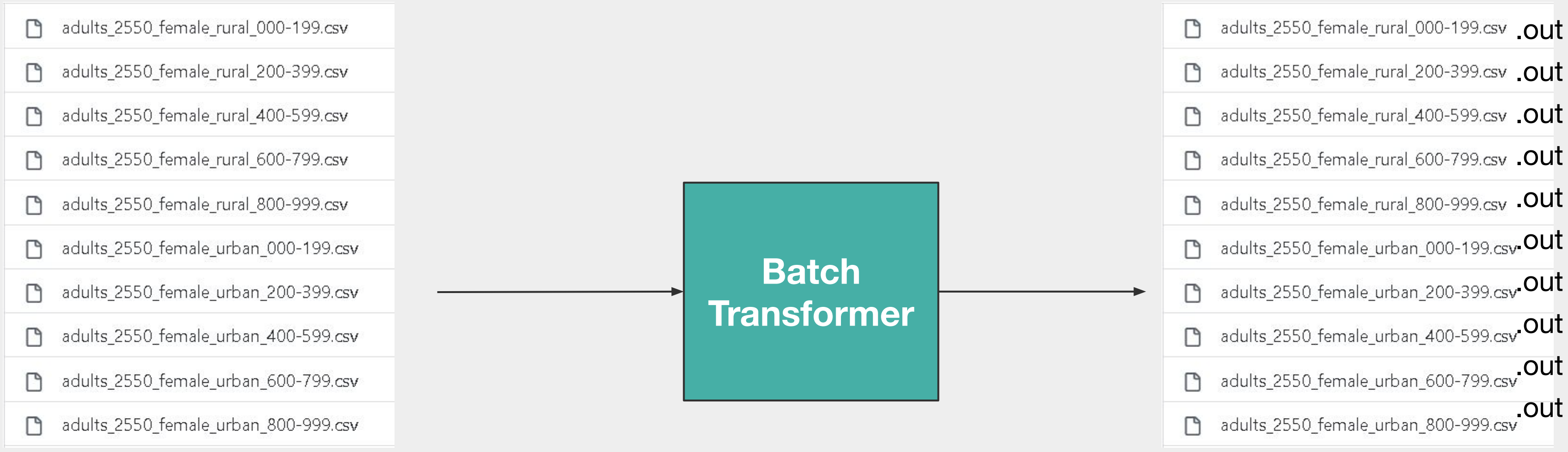
Real-Time Plant Disease Detection



Real-time Fraud Detection

apper.ph

Batch Transform


adults_2550_female_rural_000-199.csv
adults_2550_female_rural_200-399.csv
adults_2550_female_rural_400-599.csv
adults_2550_female_rural_600-799.csv
adults_2550_female_rural_800-999.csv
adults_2550_female_urban_000-199.csv
adults_2550_female_urban_200-399.csv
adults_2550_female_urban_400-599.csv
adults_2550_female_urban_600-799.csv
adults_2550_female_urban_800-999.csv
Batch Transformer
adults_2550_female_rural_000-199.csv.out
adults_2550_female_rural_200-399.csv.out
adults_2550_female_rural_400-599.csv.out
adults_2550_female_rural_600-799.csv.out
adults_2550_female_rural_800-999.csv.out
adults_2550_female_urban_000-199.csv.out
adults_2550_female_urban_200-399.csv.out
adults_2550_female_urban_400-599.csv.out
adults_2550_female_urban_600-799.csv.out
adults_2550_female_urban_800-999.csv.out

Input Data

Output Predictions

apper.ph

Use Batch Transform

- For getting inferences for **large datasets**
- For **preprocessing** datasets to remove noise or bias
- For running inference when a **persistent** endpoint is **not needed**
- For associating input records with inferences to assist **results interpretation**

- For getting **predictions** for datasets in **bulk or batches** instead of streams
- For **preprocessing entire** datasets

apper.ph

How SageMaker Batch Transform Works

- **Distributes workload** between compute instances

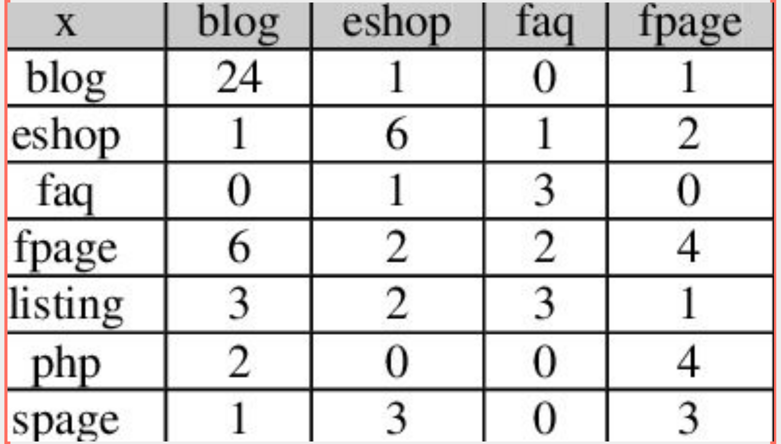
- Processes **one file per instance** or splits a **file into mini-batches**

- Stores **output files** in a specified location, e.g. S3 bucket, with same names as input files + `.out` file extension

apper.ph

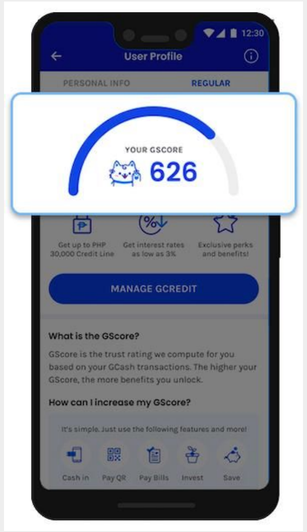
SageMaker Batch Transform Use Cases

NLP Preprocessing



x	blog	eshop	faq	fpage
blog	24	1	0	1
eshop	1	6	1	2
faq	0	1	3	0
fpage	6	2	2	4
listing	3	2	3	1
php	2	0	0	4
spage	1	3	0	3

Credit Scoring



What is the GScore?

GScore is the trust rating we compute for you based on your GCash transactions. The higher your GScore, the more benefits you unlock.

How can I increase my GScore?

It's simple. Just use the following features and more!

Real-time Inference

- Persistent - instances stay running until shut down
- Predicts for individual records in real-time

Batch Transform

- Instances torn down when job completes
- Predicts for groups of records

Amazon SageMaker Cookbook

apper.ph

Chapter 1 Summary

- gentle introduction in using Amazon SageMaker

What we learned:

- How to perform a basic machine learning project in Amazon SageMaker
    - How to perform a machine learning project the AWS way
- How to deploy an inference endpoint for the trained model and how to invoke the endpoint to retrieve predictions

Chapter 2 Summary

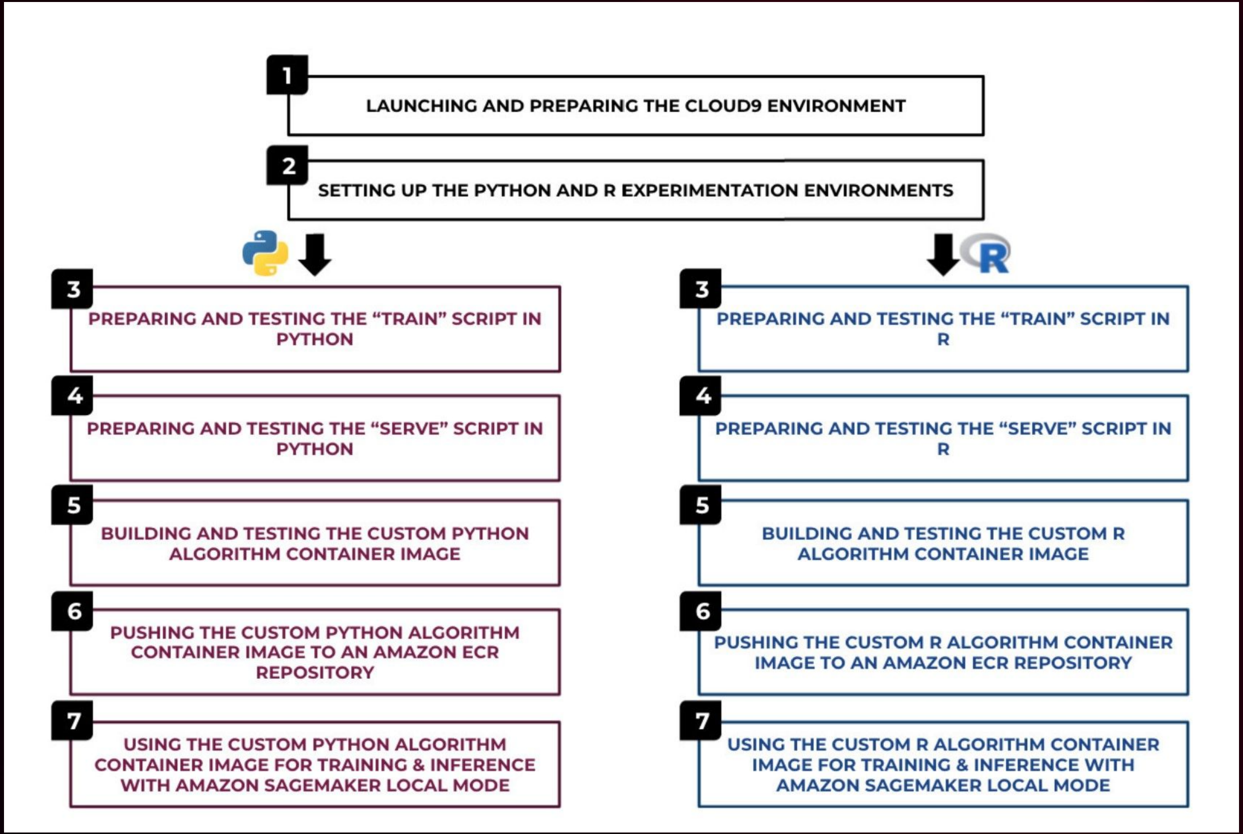
- deep dive into how Amazon SageMaker works internally

What we learned:

- How to implement a custom model in Python and R
  - Create train and serve scripts
  - Build docker image from scripts and upload to a repository
  - Use image to create a machine learning model in a project

apper.ph

Chapter 2 Summary


1
LAUNCHING AND PREPARING THE CLOUD9 ENVIRONMENT
2
SETTING UP THE PYTHON AND R EXPERIMENTATION ENVIRONMENTS
3
PREPARING AND TESTING THE "TRAIN" SCRIPT IN PYTHON
4
PREPARING AND TESTING THE "SERVE" SCRIPT IN PYTHON
5
BUILDING AND TESTING THE CUSTOM PYTHON ALGORITHM CONTAINER IMAGE
6
PUSHING THE CUSTOM PYTHON ALGORITHM CONTAINER IMAGE TO AN AMAZON ECR REPOSITORY
7
USING THE CUSTOM PYTHON ALGORITHM CONTAINER IMAGE FOR TRAINING & INFERENCE WITH AMAZON SAGEMAKER LOCAL MODE
3
PREPARING AND TESTING THE "TRAIN" SCRIPT IN R
4
PREPARING AND TESTING THE "SERVE" SCRIPT IN R
5
BUILDING AND TESTING THE CUSTOM R ALGORITHM CONTAINER IMAGE
6
PUSHING THE CUSTOM R ALGORITHM CONTAINER IMAGE TO AN AMAZON ECR REPOSITORY
7
USING THE CUSTOM R ALGORITHM CONTAINER IMAGE FOR TRAINING & INFERENCE WITH AMAZON SAGEMAKER LOCAL MODE

apper.ph

Chapter 3 Summary

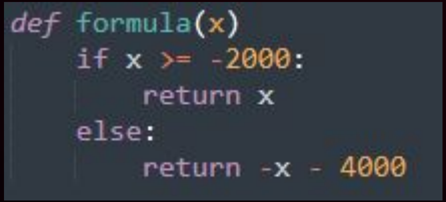
- demonstration on the different machine learning libraries and frameworks in Amazon Sagemaker

What we learned:

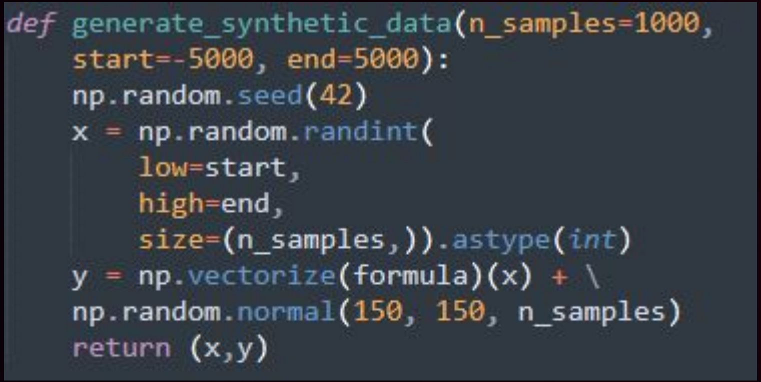
- How to generate a synthetic dataset
- How to train and deploy a model using the following:
    - TensorFlow and Keras
    - PyTorch
    - scikit-learn

apper.ph

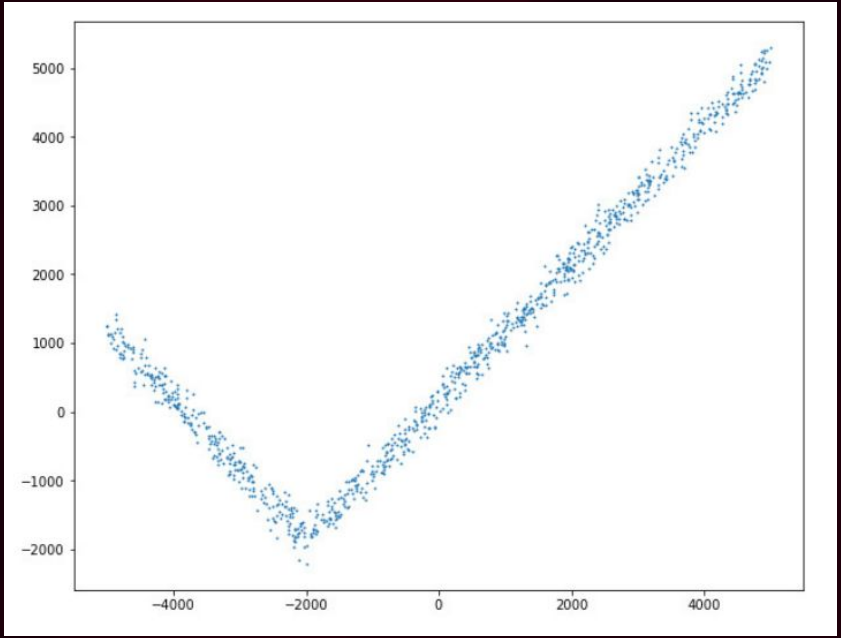
Chapter 3 Summary - Synthetic dataset



```python
def formula(x)
    if x >= -2000:
        return x
    else:
        return -x - 4000
```



```python
def generate_synthetic_data(n_samples=1000,
    start=-5000, end=5000):
    np.random.seed(42)
    x = np.random.randint(
        low=start,
        high=end,
        size=(n_samples,)).astype(int)
    y = np.vectorize(formula)(x) + \
    np.random.normal(150, 150, n_samples)
    return (x,y)
```



apper.ph

Chapter 3 Summary - Libraries and frameworks

TensorFlow and Keras

Entrypoint:
- can use the environment variables set by SageMaker for the entrypoint script.

Deployment:
- can directly use deploy() function.

Notes:
- industry-focused

PyTorch

Entrypoint:
- separate inference script for deployment.

Deployment:
- PyTorchModel needs to be used and initialized.

Notes:
- research-oriented

scikit-learn

Entrypoint:
- model_fn() function needs to be defined.

Deployment:
- can directly use deploy() function.

Notes:
- simple and beginner friendly

apper.ph

Summary

SageMaker Deployment Options

- ✅ Real-Time Inference Endpoint
- ✅ Batch Transform

Amazon SageMaker Cookbook

- ✅ Chapter 1
- ✅ Chapter 2
- ✅ Chapter 3

apper.ph