

Michael Smurfit Graduate School of Business
M.Sc. in Business Analytics
Capstone Presentation

Amazon Web Services

Amazon SageMaker - An Exploratory Approach

Nguyen, Hang, BA. and Lalhlimpuii, Rosy, BE.

Supervisor: Dr. Michael MacDonnell

August 2020



University College Dublin
Ireland's Global University



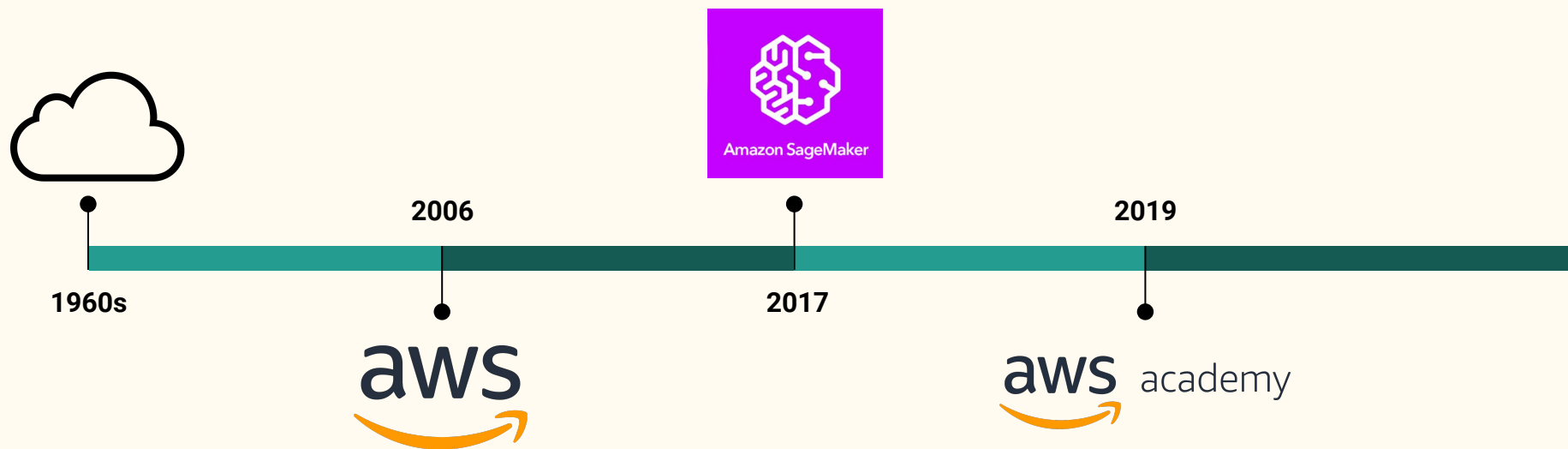
Contents

1. Capstone Project's Objectives
2. Introduction: AWS, AWS Academy, Amazon SageMaker
3. Use Case 1: Customer Churn Prediction using Amazon SageMaker Autopilot
4. Use Case 2: Stock Price Prediction using GluonTS and DeepAR algorithm
5. AWS Products - Module Integration
6. Limitations and Recommendations
7. Learning and Takeaways

Capstone Project's Objectives

- Explore Amazon SageMaker by comparing among 3 approaches (Studio, Notebook Instance, Console) in 2 use cases (Autopilot, Built-in algorithm).
- Recommend AWS Products - Module Integration

Introduction



What Amazon SageMaker offer?

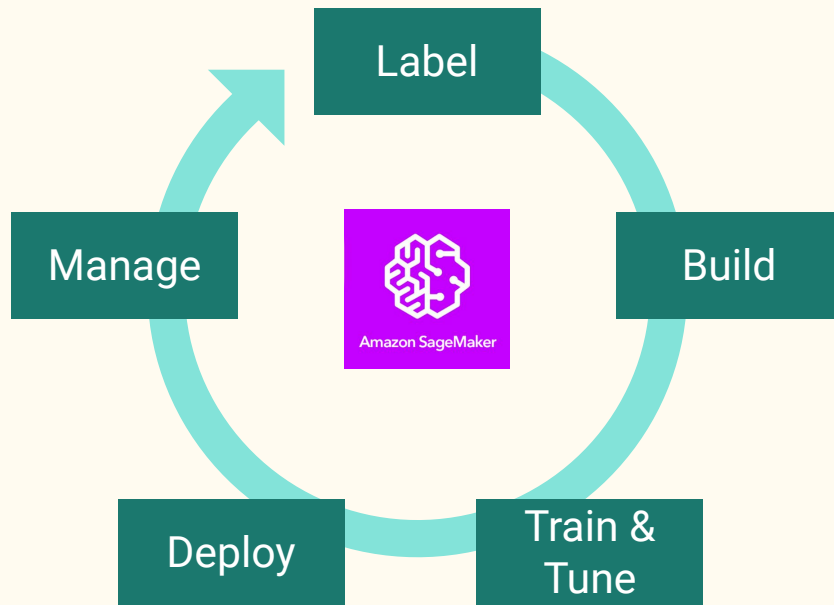
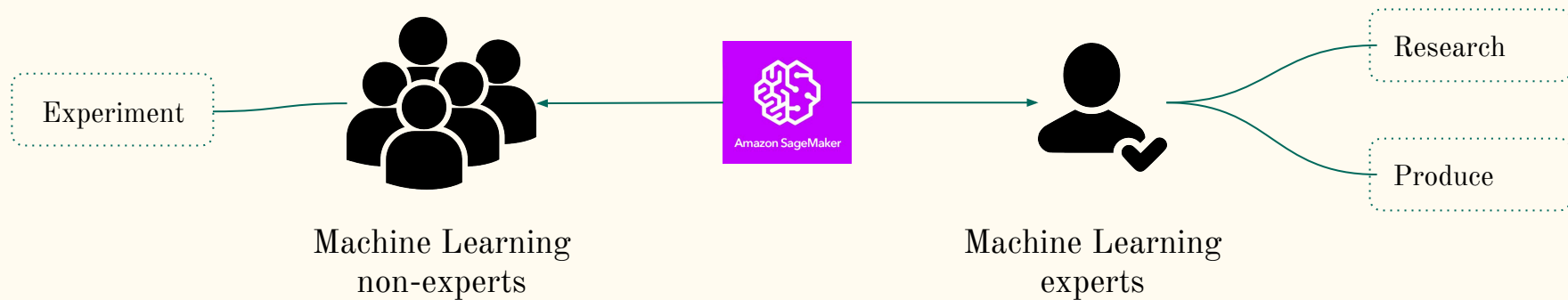
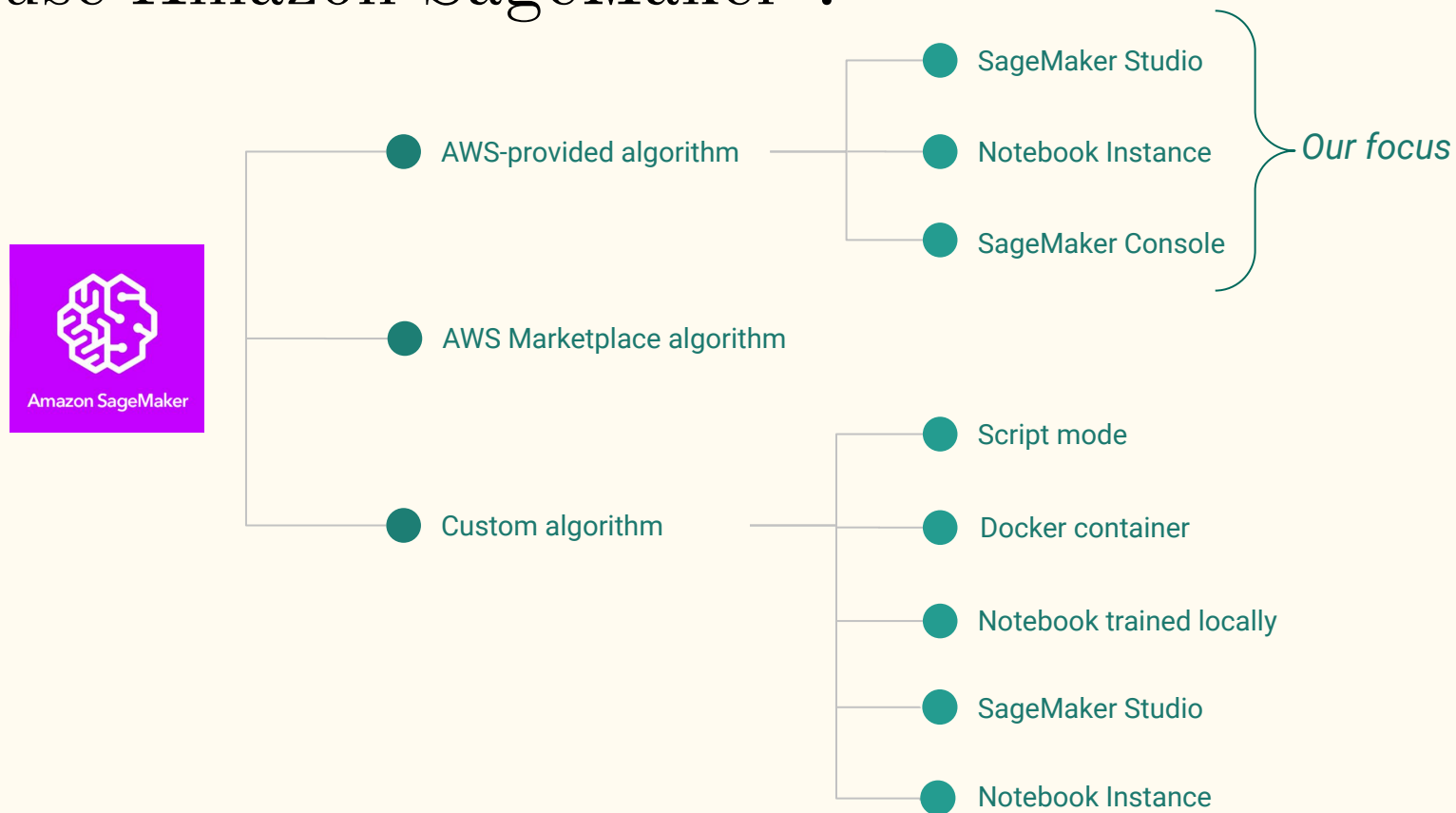


Fig. 1. Amazon SageMaker offerings

Who is Amazon SageMaker for?



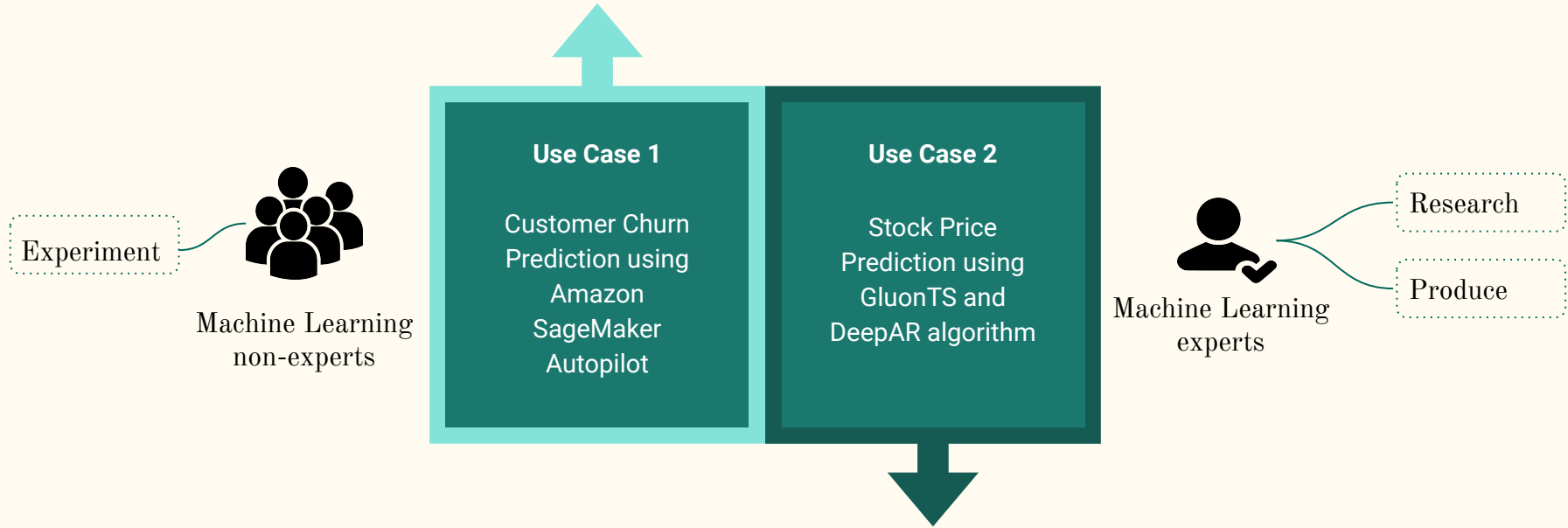
How to use Amazon SageMaker ?



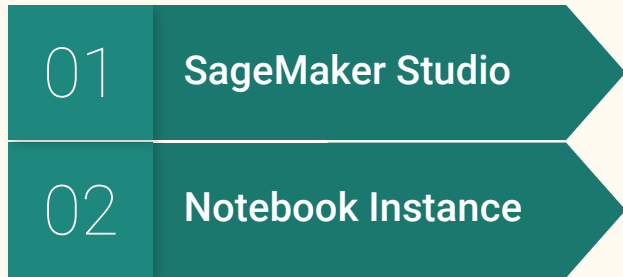
We explore 3 Approaches

01	SageMaker Studio	<ul style="list-style-type: none">• fast, fully integrated IDE for machine learning• single, web-based visual interface• notebooks, automatic model creation, etc.
02	Notebook Instance	<ul style="list-style-type: none">• one-click Jupyter notebooks• fully elastic compute resources• easy sharing with others
03	SageMaker Console	<ul style="list-style-type: none">• SageMaker main interface• process, train, deploy without any code• Ground Truth, Augmented AI and AWS Marketplace.

We explore 2 Use cases



Use Case 1: Customer Churn Prediction using Amazon SageMaker Autopilot



Use Case 1: Customer Churn Prediction using Amazon SageMaker Autopilot



University of California Irvine
Repository of Machine Learning
Datasets, consisting profile of
customers of an unknown US mobile
operator, collected by Daniel T.
Larose and mentioned in
[Discovering Knowledge in Data](#)



Amazon SageMaker SDK
library



AWS built-in algorithms

- **XGBoost**
- **Linear-Learner**

Experiment with SageMaker Autopilot

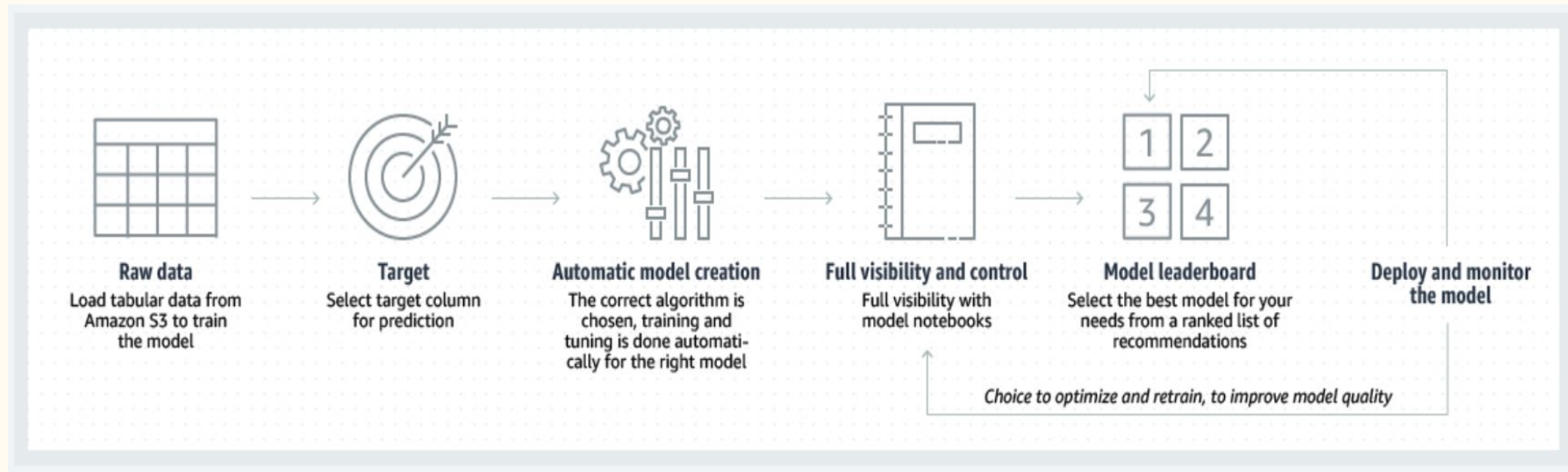
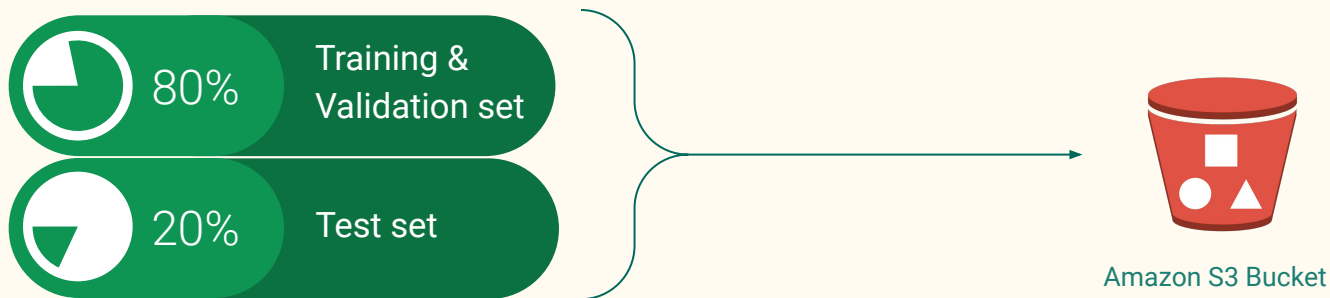
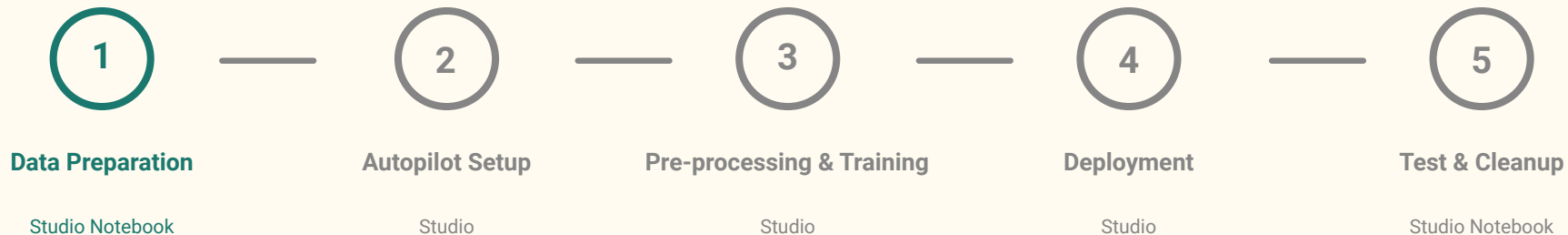


Fig. 2. Amazon SageMaker Autopilot workflow

01

SageMaker Studio



NOTE: Autopilot automatically split Training and Validation set with default ratio = 0.8/0.2

01 SageMaker Studio

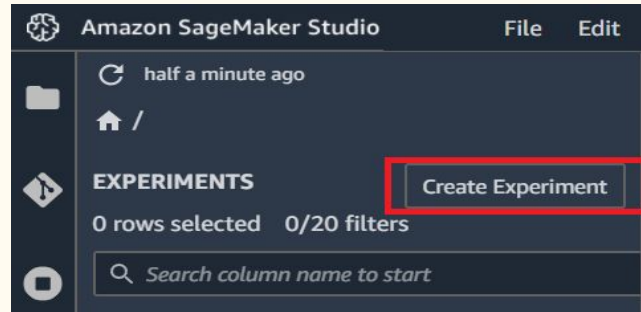
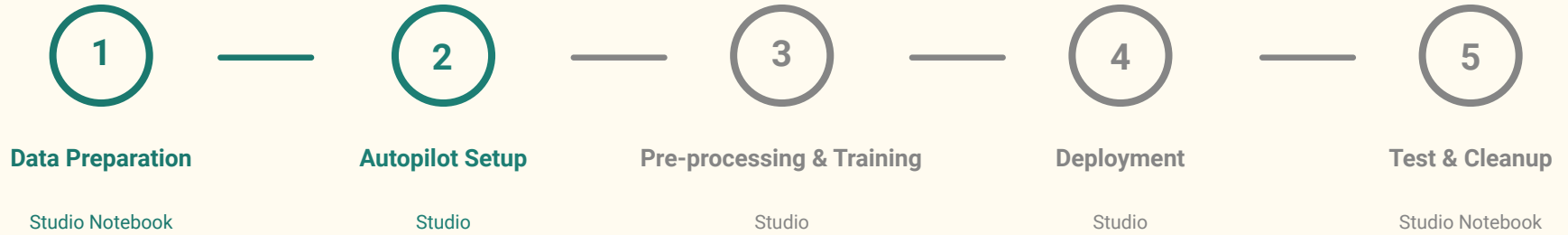


Fig. 3. Create an Autopilot Experiment

01 SageMaker Studio

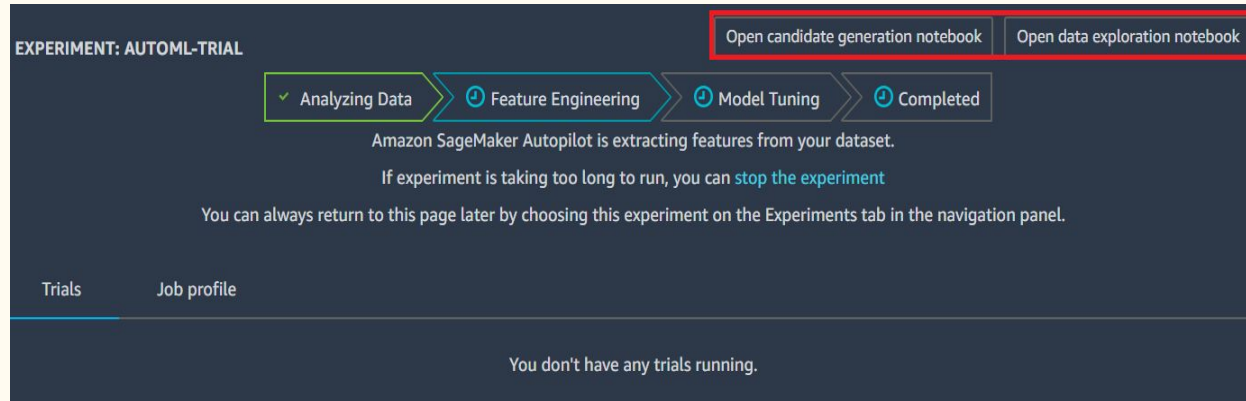
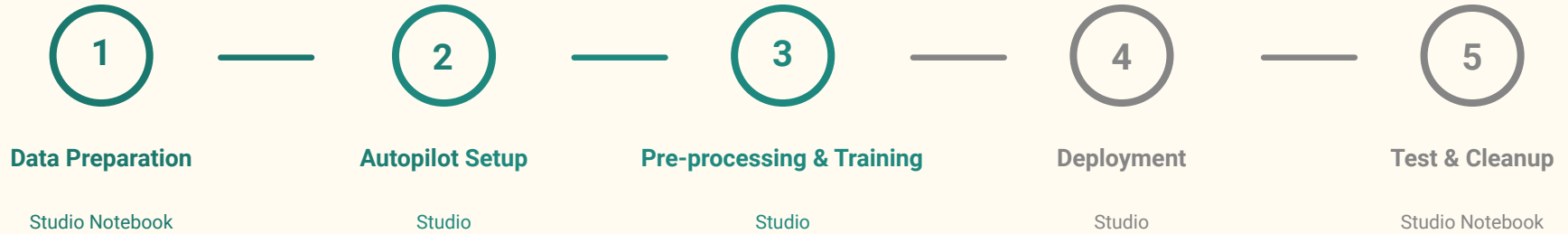
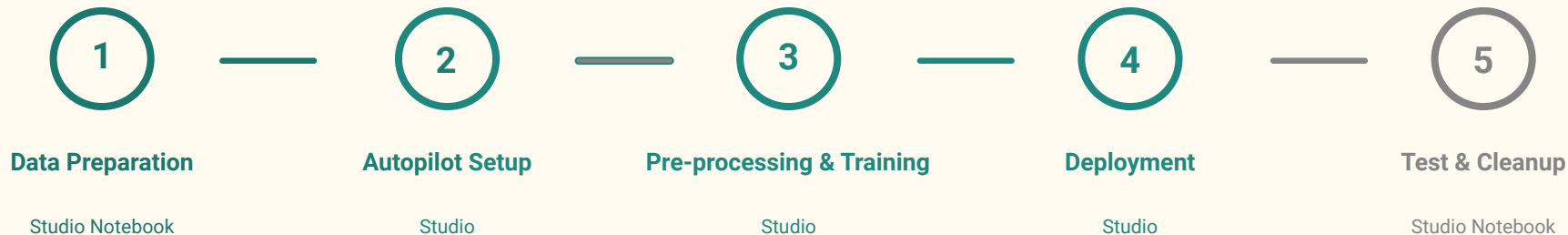


Fig.4. Data Exploration notebook and Candidate Generation notebook in Studio Autopilot Experiment

01

SageMaker Studio



EXPERIMENT: AUTOML-TRIAL

Open candidate generation notebook Open data exploration notebook

Trials Job profile

TRIALS

1 row selected

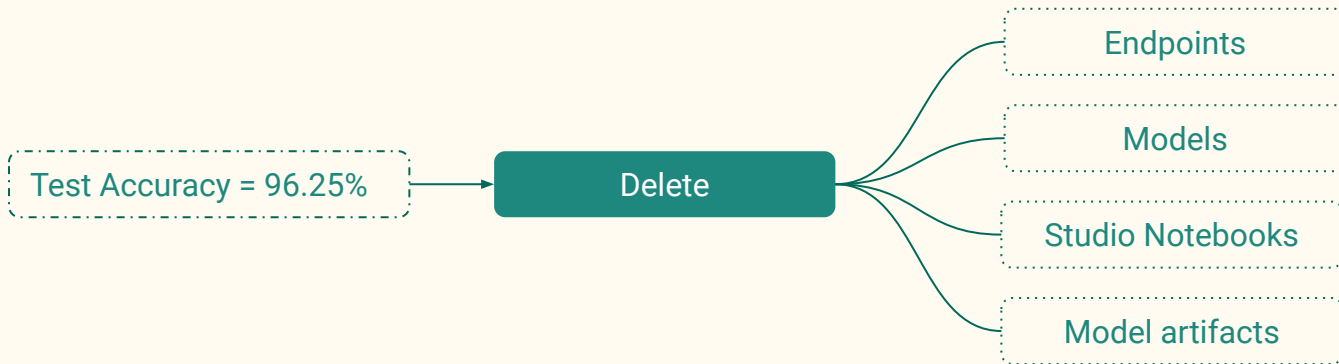
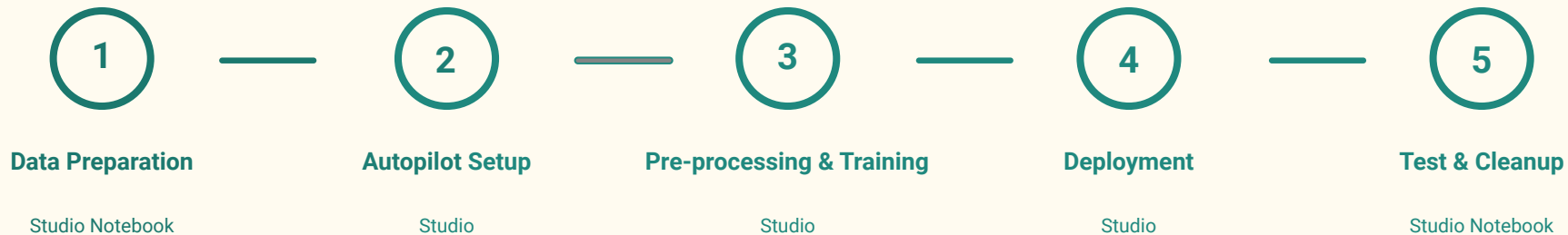
Deploy model

Trial name	Status	Start time	Objective
★ Best: tuning-job-1-7185edb8c3db...	Completed	5 hours ago	0.810230016708374
tuning-job-1-7185edb8c3db46ad94-...	Completed	5 hours ago	0.5906696915626526
tuning-job-1-7185edb8c3db46ad94-...	Completed	5 hours ago	0.8054900169372559

Fig. 5. Autopilot automatically chooses the best model after Training jobs

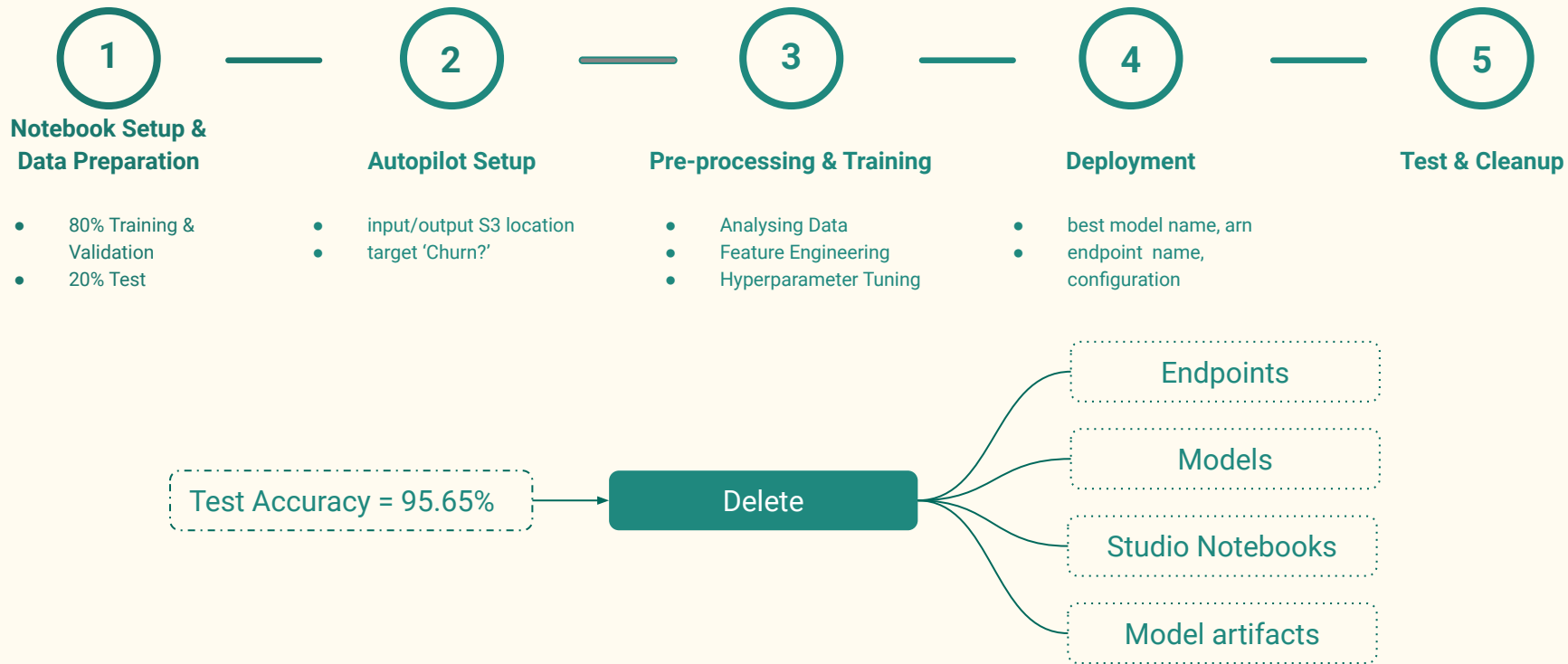
01

SageMaker Studio



02

Notebook Instance



NOTE: Autopilot automatically split Training and Validation set with default ratio = 0.8/0.2

01

SageMaker Studio

OR

02

Notebook Instance

for Autopilot Job

		SageMaker Studio	Notebook Instance	Details
1	Accuracy	96.25%	95.65%	
2	Time	✓	✗	Starting a Studio notebook is typically faster than launching an notebook instance
3	Interpretability	✓	✗	<ul style="list-style-type: none">- Data Exploration notebook from Studio helps users explore the descriptive statistics with no code needed- Trial Component list allows users to compare the metrics among trial models via CloudWatch
4	Transparency	✓	✗	<ul style="list-style-type: none">- Candidate Generation notebook from Studio helps users examine candidates' hyperparameters- Trial Component list: all of model's parameters, configurations and results.

NOTE:

SageMaker Studio is good for Autopilot job, but it is still not matured enough for complicated tasks (research and production)

Use Case 2: Stock Price Prediction using GluonTS and DeepAR algorithm

02

Notebook Instance

03

SageMaker Console

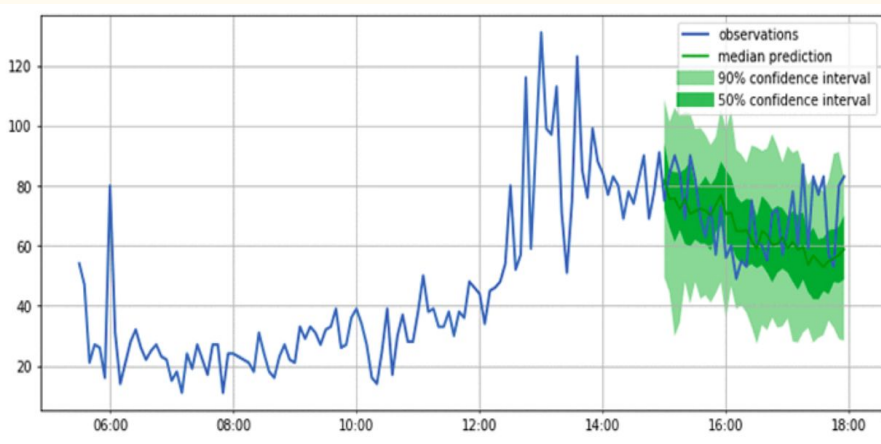


Fig. 6. DeepAR Inference visualisation*

* Gasthaus, J. et al. (2019), Creating neural time series models with Gluon Time Series, *AWS Machine Learning Blog*. [URL](#)

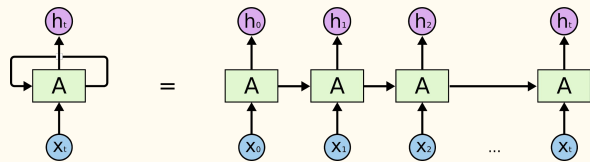
Use Case 2: Stock Price Prediction using GluonTS and DeepAR algorithm



Deutsche Börse Public
Dataset - Xetra, consisting
of daily trading data
(Jul 2017 - Oct 2018)



GluonTS library



AWS built-in algorithm

- Autoregressive Recurrent Network
- High accuracy, global model from all-time series

02

Notebook Instance



Amazon SageMaker > Lifecycle configurations > Create lifecycle configuration

Create lifecycle configuration

Configuration setting

Name
fsv309-lifecycle-config
Alphanumeric characters and "-"; no spaces. Maximum 63 characters.

Scripts

Start notebook **Create notebook**

When selected during creation of a new notebook instance, this script will be run once during its initial creation. This script on existing notebook instances.

```
1 #!/bin/bash
2 set -e
3 git clone https://github.com/aws-samples/amazon-sagemaker-stock-prediction.git
4 mkdir SageMaker/fsv309-workshop
5 mv amazon-sagemaker-stock-prediction/container SageMaker/fsv309-workshop/container/
6 mv amazon-sagemaker-stock-prediction/notebooks SageMaker/fsv309-workshop/notebooks/
7 mv amazon-sagemaker-stock-prediction/images SageMaker/fsv309-workshop/images/
8 rm -rf amazon-sagemaker-stock-prediction
9 sudo chmod -R u+w SageMaker/fsv309-workshop
```

Fig. 7. Lifecycle configurations

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the IAM role you create will provide access to:

☒ **S3 buckets you specify - optional**

☐ Specific S3 buckets

Example: bucket-name-1, bucket-name-2, bucket-name-3

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

☒ **Any S3 bucket**

Allow users that have access to your notebook instance access to any bucket

☐ 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"

☒ S3 bucket with a Bucket Policy allowing access to SageMaker

Fig. 8. Create an IAM role

Registry (5 actions) ⚠ 1 warning

Service: EC2 Container Registry

Actions: Specify the actions allowed in EC2 Container Registry

Filter actions:

Manual actions (add actions)

☐ All EC2 Container Registry actions (ecr:*)

Access level

☐ List

☐ Read

☒ Write (5 selected)

☐ BatchDeleteImage ☐ DeleteRepository ☒ PutImage ☐ SetRepositoryPolicy

☒ CompleteLayerUpload ☐ DeleteRepositoryPolicy ☐ SetRepositoryPolicy

☒ CreateRepository ☒ InitiateLayerUpload ☒ UploadLayerPart

Resources: You chose actions that require the repository resource type.

Request conditions: Specify request conditions (optional)

Fig. 9. Add new policy

Notebook instance settings

Notebook instance name: fsv309-notebook

Notebook instance type: ml.t2.medium

IAM role: AmazonSageMaker-ExecutionRole-20181119T142008

Success! You created an IAM role. AmazonSageMaker-ExecutionRole-20181119T142008

VPC - optional: No VPC

Lifecycle configuration - optional: fsv309-lifecycle-config

Encryption key - optional: No Custom Encryption

Volume Size in GB - optional: 5

Tags - optional

Cancel **Create notebook instance**

Fig. 10. Create notebook instance

02

Notebook Instance



- Download data to local files in Notebook Instance
- Pre-process data:
 - Filter top 100 stocks, by trading volume
 - Delete instance with trading volumes = 0
 - Delete instance outside trading hours
- Resample data according to Month, Week, Day, Hour intervals
- Save and upload data to S3 Bucket

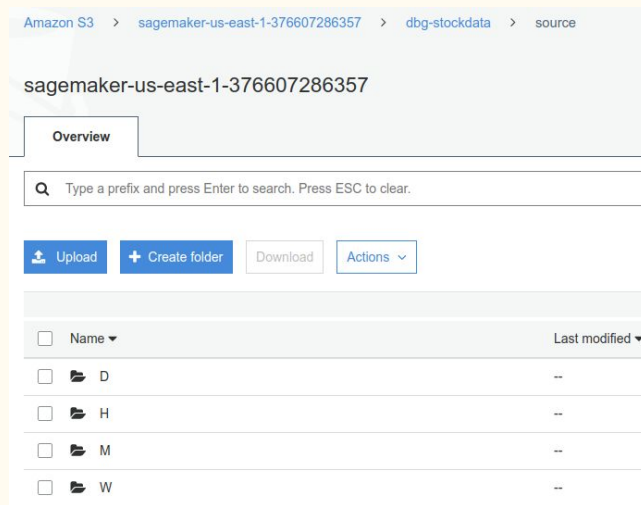


Fig.11. Processed data on S3 bucket

02

Notebook Instance

1

Notebook Setup

2

Data Preparation and
Preprocessing

3

Feature Selection

4

Train & Deploy

5

Test & Cleanup

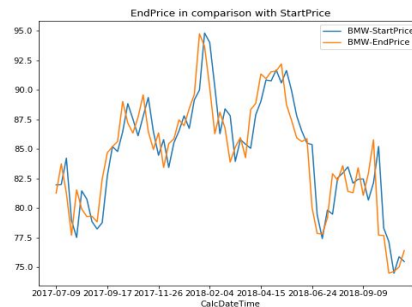
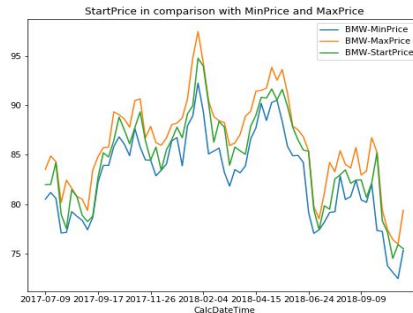
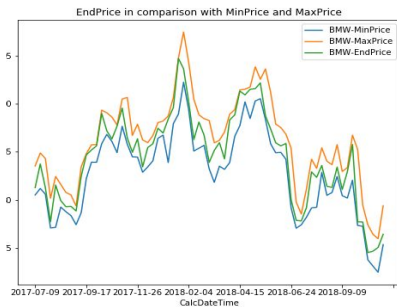
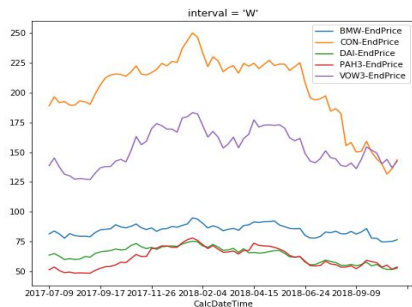


Fig.12. Data exploration

02

Notebook Instance

1

Notebook Setup

2

Data Preparation and
Preprocessing

3

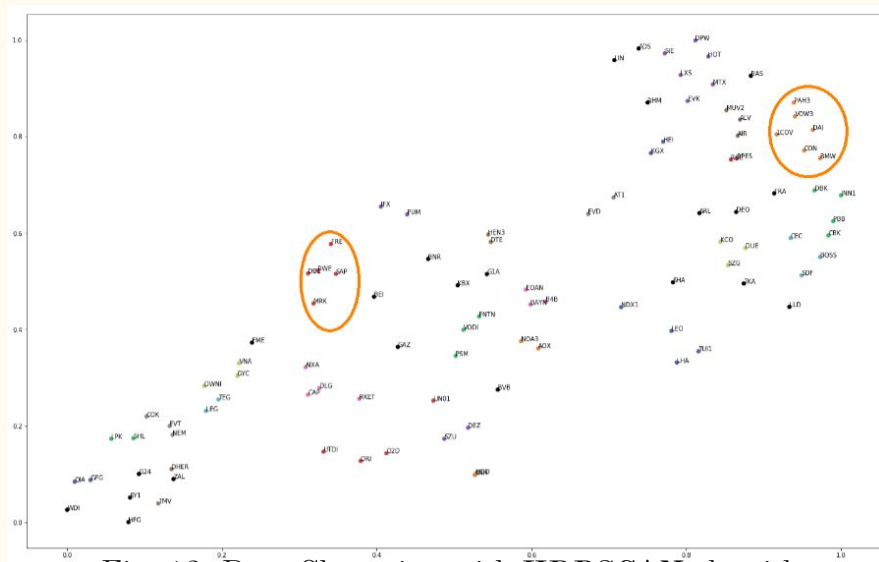
Feature Selection

4

Train & Deploy

5

Test & Cleanup

Fig. 13. Data Clustering with **HDBSCAN** algorithm

02

Notebook Instance

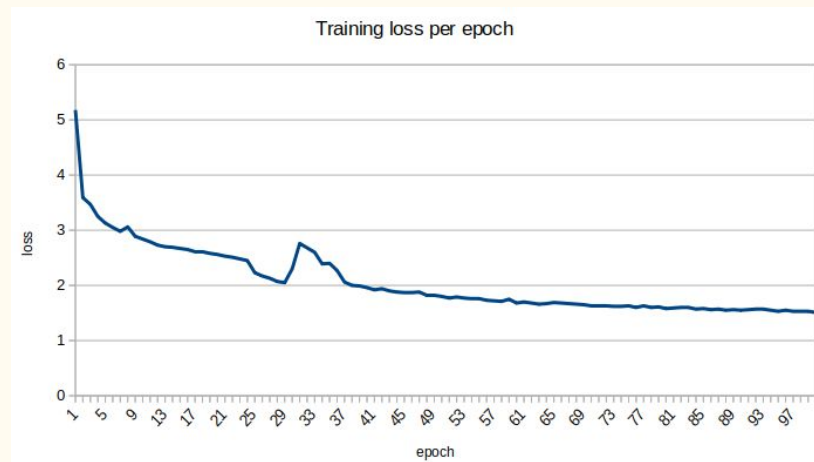
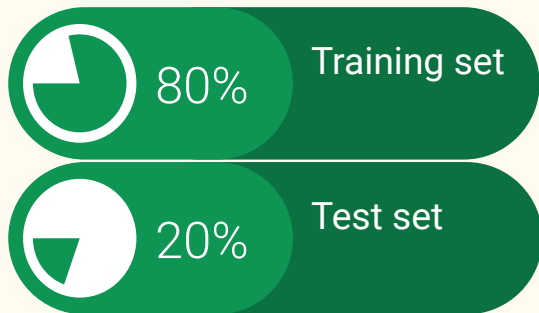
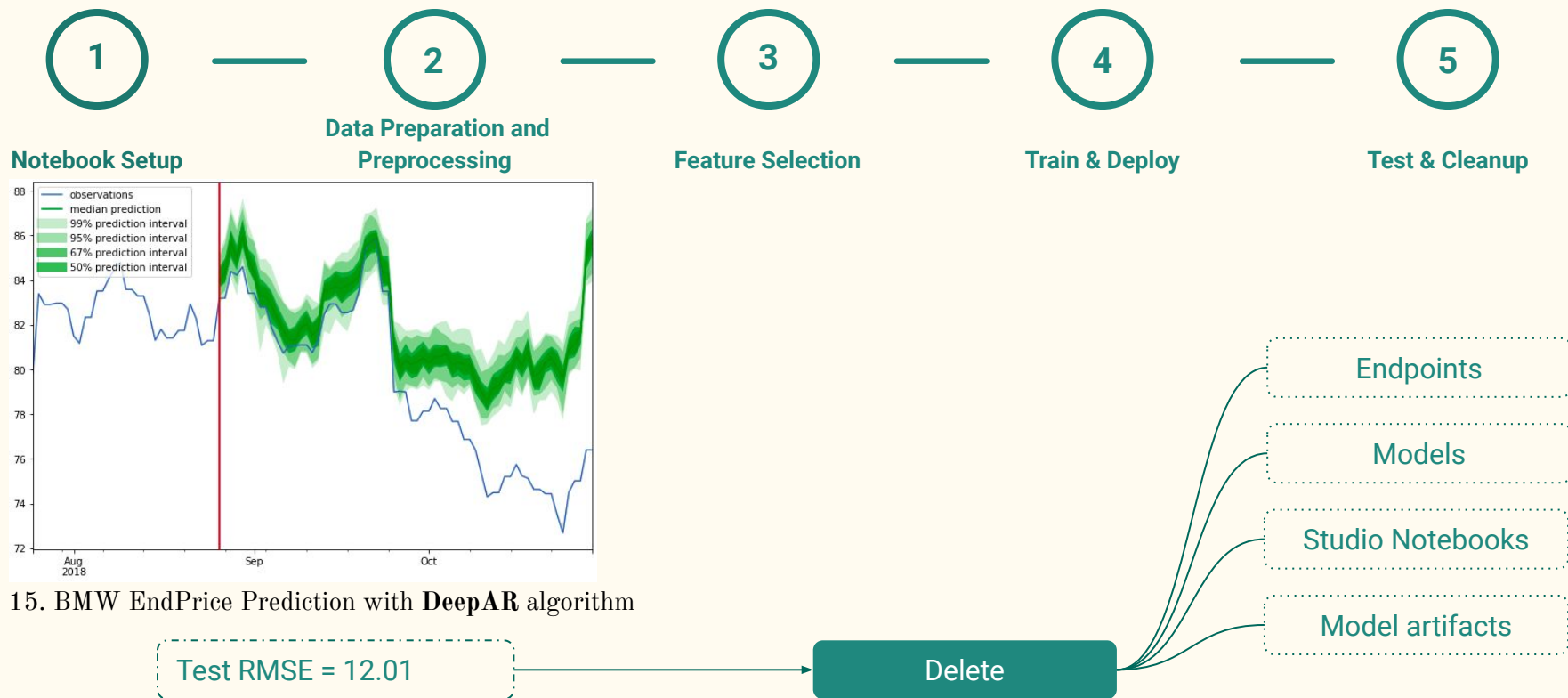


Fig. 14. Training loss per epoch

NOTE: We don't use hyperparameter tuning for DeepAR algorithm as the default hyperparameters are good enough for fast solutions

02

Notebook Instance

Fig. 15. BMW EndPrice Prediction with **DeepAR** algorithm

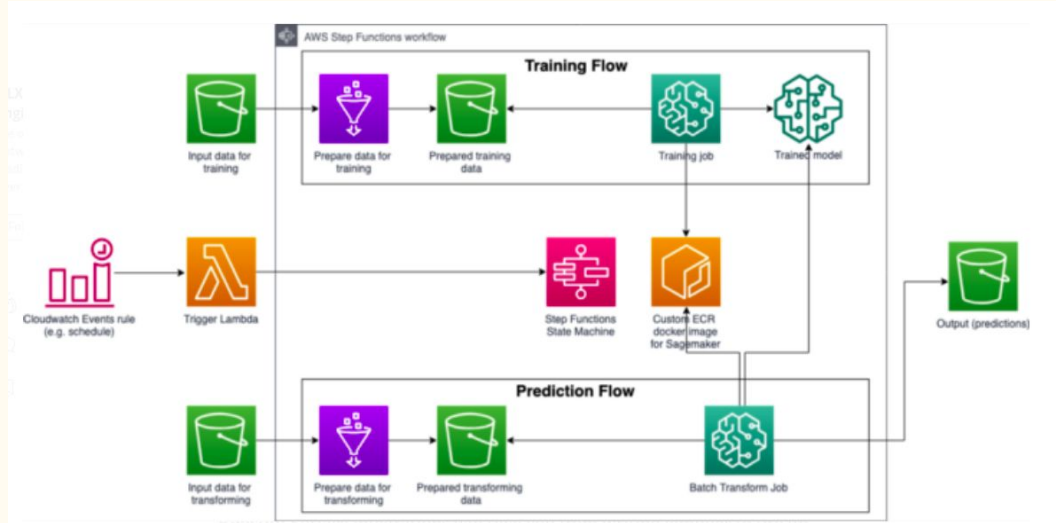


Fig. 16. Suggested Machine learning workflow for production*

- Amazon S3: storage service
- Amazon ECR: Docker container registry
- AWS Step Functions: serverless workflow
- Amazon CloudWatch: metrics recording
- AWS Glue: data extract, transform, and load
- AWS Lambda: automatic computing

* Correa, R. F., (2019), Building an AWS Serverless ML Pipeline with Step Functions, *OLX Group Engineering*. [URL](#)

03

SageMaker Console

The screenshot displays the AWS SageMaker Console interface. On the left is a navigation pane with a search bar and a list of categories: Ground Truth, Notebook, Processing, Training, Inference, Augmented AI, and AWS Marketplace. The 'Processing' and 'Training' categories are highlighted with red boxes. The main content area features five tabs: Ground Truth, Notebook, Training, Inference, and Processing Run. Each tab has a description and a button to view related resources. Below the tabs is a 'Recent activity' section with a dropdown for 'Recent activity within the' set to 'Last 7 days'. This section displays a grid of activity cards for each category, showing counts and status (e.g., '1 In Service', '1 Completed', '1 Created').

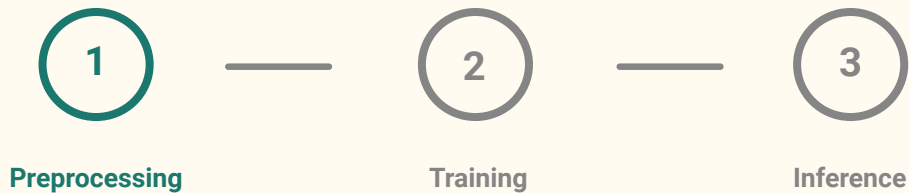
Category	Description	Button
Ground Truth	Set up and manage labeling jobs for highly accurate training datasets using active learning and human labeling.	Labeling jobs
Notebook	Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.	Notebook instances
Training	Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.	Training jobs Hyperparameter tuning jobs
Inference	Create models from training jobs or import external models for hosting to run inferences on new data.	Models Endpoints Batch transform jobs
Processing Run	Pre- or post-processing and model evaluation workloads with a fully managed experience.	Processing jobs

Category	Recent activity
Ground Truth	No recent activity.
Notebook	Notebook instances 1 In Service 1 Created
Training	Training jobs 1 Completed 1 Created Hyperparameter tuning jobs No recent activity.
Inference	Models 1 Created Endpoints 1 In Service 1 Created Batch transform jobs No recent activity.
Processing	No recent activity.

Fig. 17. SageMaker Console

03

SageMaker Console



Create processing job

When you create a processing job, Amazon SageMaker sets up the distributed compute cluster, executes the computation, and deletes the cluster when processing has completed. The results are stored in the location you specified when you created the processing job. [Learn more](#)

Job settings

Job name

stock-data-analysis

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

Container

The registry path where the processing image is stored in Amazon ECR.

806410988728.dkr.ecr.eu-west-1.amazonaws.com/stock-prediction-console

IAM role

Amazon SageMaker requires permissions to call other services on your behalf. Choose a role or let us create a role that has the [AmazonSageMakerFullAccess](#) IAM policy attached.

AmazonSageMaker-ExecutionRole-20200816T143852

Resource configuration

Instance type

mLm4.xlarge

Instance count

1

Additional storage volume per instance (GB)

1

Fig. 18. Create a processing job from Console

03

SageMaker Console

1



2



3

Preprocessing

Training

Inference

DeepAR-2

Exit full screen

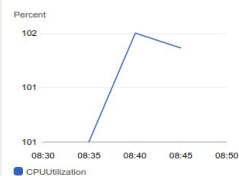
Light

Dark

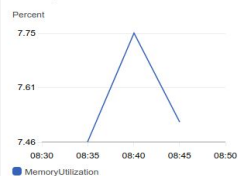
2020-09-02 (08:33:00) - 2020-09-02 (08:52:00)



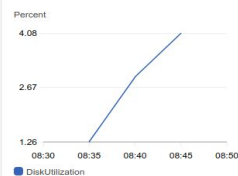
CPUUtilization



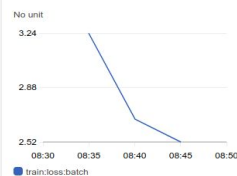
MemoryUtilization



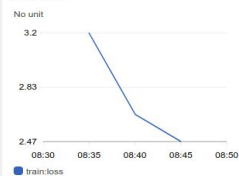
DiskUtilization



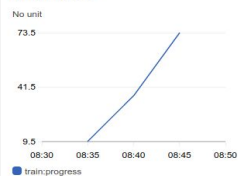
train:loss:batch



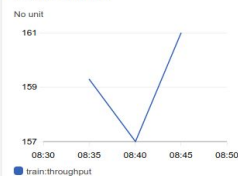
train:loss



train:progress



train:throughput



test:RMSE



train:final_loss

2.17

test:mean_wQuantileLoss

0.04

Fig. 19. Train/Test metrics

03

SageMaker Console

1

Preprocessing

2

Training

3

Inference

Test RMSE = 14.6

**Prediction on an
entire dataset**

Batch Transforms without
create an endpoint

**Real-time prediction on
large scale datasets**

Create an endpoint
configuration and an HTTPS
endpoint

02

Notebook Instance

OR

03

SageMaker Console

		Notebook Instance	SageMaker Console
1	RMSE	12.01	14.6
2	Code	<ul style="list-style-type: none">conda-installedother data science packages	<ul style="list-style-type: none">process, train and deploy with one click
3	Research workflow	<ul style="list-style-type: none">data explorationexecute and share 'technical' papers	<ul style="list-style-type: none">create Lifecycle Configuration, IAM, VPC, and label data
4	Production workflow	<ul style="list-style-type: none">need advanced infrastructures (e.g. Netflix) to use at scale	<ul style="list-style-type: none">suit for script mode, docker container when using at scale

AWS Products - Module Integration

Module	AWS Product
MIS41150: Introduction to Business Analytics	AWS Cloud Security
MIS41110: Programming for Analytics	AWS Developer Tools
MIS41130: Statistics & Simulation Methods	AWS Analytics
MIS41090: Advanced Operations Research	AWS Machine Learning
MIS41270: Data Management & Mining	AWS Machine Learning
MIS41040: Business Intelligence & Visual Analytics	Amazon QuickSight
MIS41050: Consulting Change & Project Management	AWS IoT, AWS Management and Governance
MIS41120: Statistical Learning	AWS Machine Learning

Tab. 1. AWS Products - Module Integration

Limitations and Recommendations

Limitations of SageMaker

- Complicated connection with other AWS services and in-progress pipeline development.
- Local mode is only available in Notebook Instance, not in Studio Notebook.
- Data pre-processing and Cross-validation are not well-supported.
- Detailed permissions system is needed when starting notebook instance and when onboarding to Studio.
- Python SDK is incomplete and some algorithms are not open-sourced.

Recommendations for SageMaker practitioners

- In-depth research on other AWS services (S3, ECR, Step Functions, CloudWatch, Glue, Lambda)
- Keep up-to-date with AWS SageMaker newly-launched features and libraries

Limitations and Recommendations

Capstone Project Limitations

- Do not cover all AWS Machine Learning services (Augmented AI, Forecast, Fraud Detector, Personalize, Rekognition, etc.) and approaches (script mode, docker container, AWS Marketplace and local model training).
- Have not fully explore Inference/Test step in Use Case 2.

Amazon Web Services

Amazon SageMaker - An Exploratory Approach

Learning and Takeaways

- Accelerate Machine Learning with Cloud Computing
 - Big Data infrastructure
 - High-quality models with minimal effort and machine learning expertise
 - Automated deployment and inference



University College Dublin
Ireland's Global University



Nguyen, Hang, BA. and Lalhlimpuii, Rosy, BE.
Supervisor: Dr. Michael MacDonnell
Sponsor contact: Ivan Obarski, AWS

August 2020