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

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

1960s

2006

aws

2017

Amazon SageMaker

2019

aws academy

What Amazon SageMaker offer?

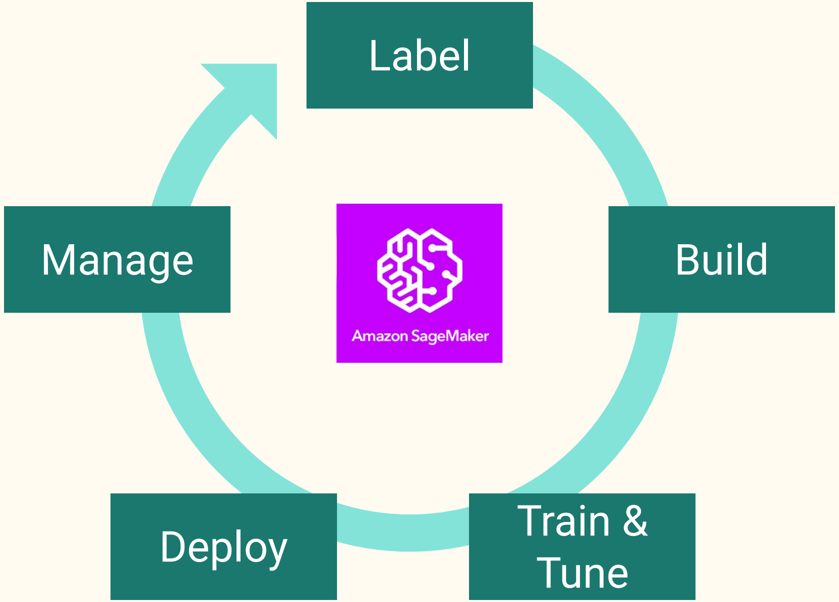
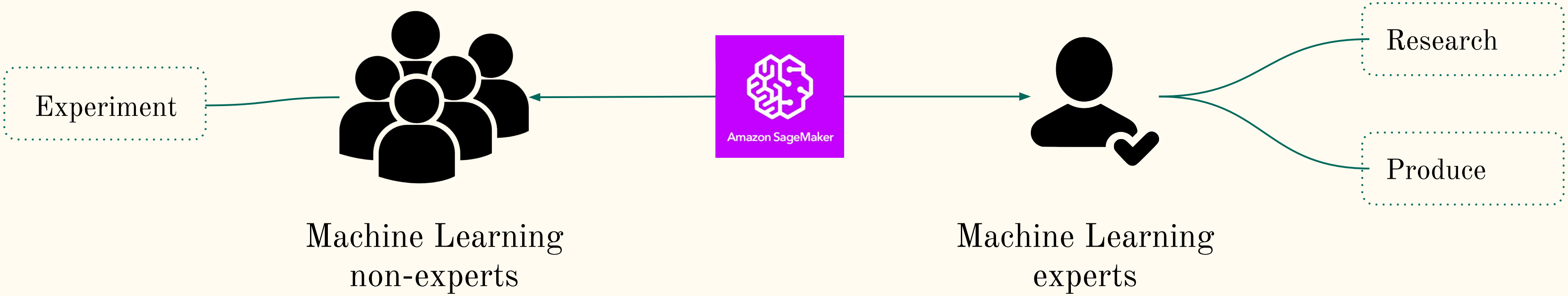


Fig. 1. Amazon SageMaker offerings

Who is Amazon SageMaker for?



How to use Amazon SageMaker ?

- Amazon SageMaker
  - AWS-provided algorithm
    - SageMaker Studio
    - Notebook Instance
    - SageMaker Console
  - AWS Marketplace algorithm
  - Custom algorithm
    - Script mode
    - Docker container
    - Notebook trained locally
    - SageMaker Studio
    - Notebook Instance

Our focus

We explore 3 Approaches

01	**SageMaker Studio**	• fast, fully integrated IDE for machine learning • single, web-based visual interface • notebooks, automatic model creation, etc.
02	**Notebook Instance**	• one-click Jupyter notebooks • fully elastic compute resources • easy sharing with others
03	**SageMaker Console**	• SageMaker main interface • process, train, deploy without any code • Ground Truth, Augmented AI and AWS Marketplace.

We explore 2 Use cases

Experiment

Machine Learning non-experts

Use Case 1

Customer Churn Prediction using Amazon SageMaker Autopilot

Use Case 2

Stock Price Prediction using GluonTS and DeepAR algorithm

Machine Learning experts

Research

Produce

Use Case 1: Customer Churn Prediction using Amazon SageMaker Autopilot

01 SageMaker Studio

02 Notebook Instance

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

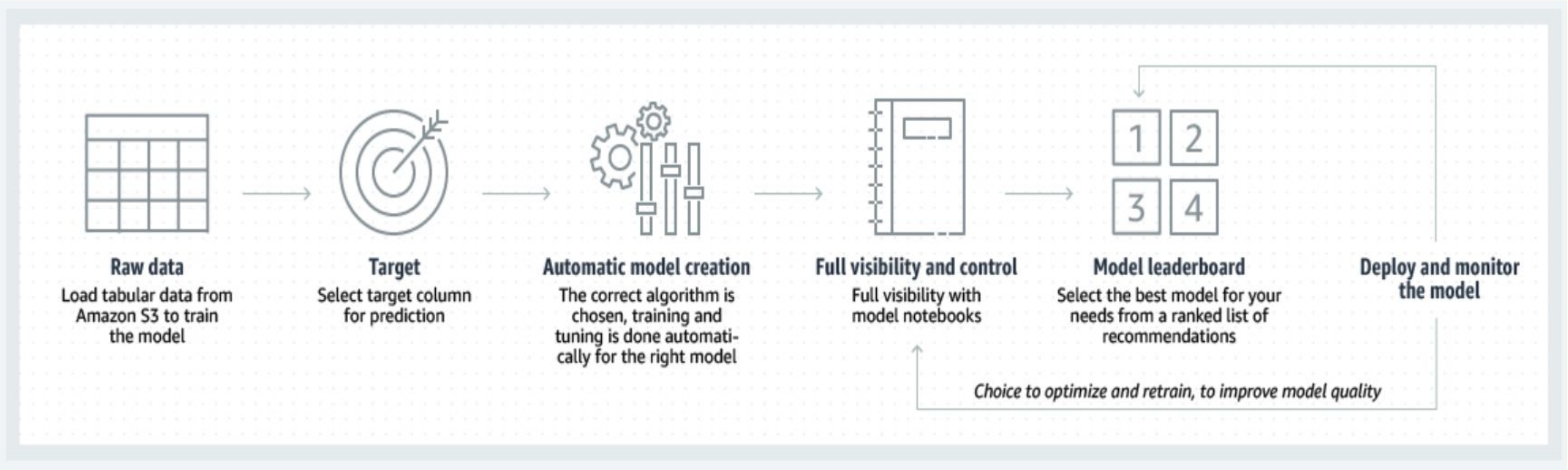
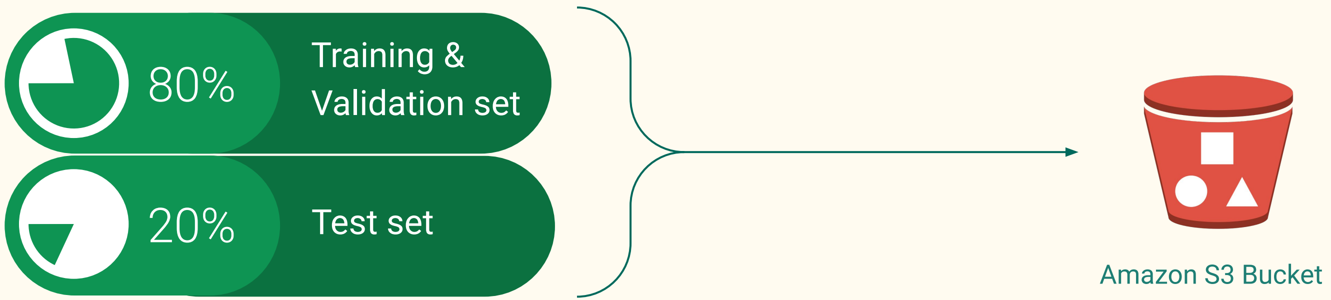


Fig. 2. Amazon SageMaker Autopilot workflow

01 SageMaker Studio

1. **Data Preparation** — Studio Notebook
2. Autopilot Setup — Studio
3. Pre-processing & Training — Studio
4. Deployment — Studio
5. Test & Cleanup — Studio Notebook



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

01 SageMaker Studio

1 — 2 — 3 — 4 — 5

1	2	3	4	5
Data Preparation	**Autopilot Setup**	**Pre-processing & Training**	**Deployment**	**Test & Cleanup**
Studio Notebook	Studio	Studio	Studio	Studio Notebook

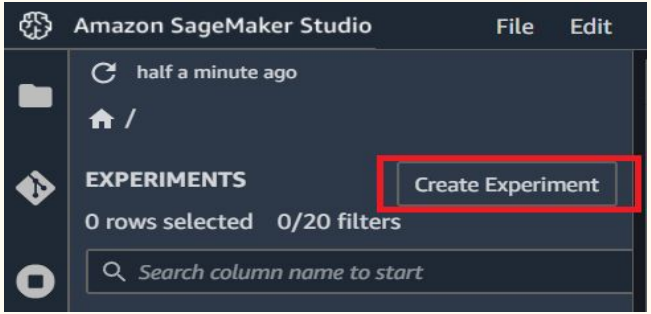


Fig. 3. Create an Autopilot Experiment

01 SageMaker Studio

1. **Data Preparation** — Studio Notebook
2. **Autopilot Setup** — Studio
3. **Pre-processing & Training** — Studio
4. **Deployment** — Studio
5. **Test & Cleanup** — Studio Notebook

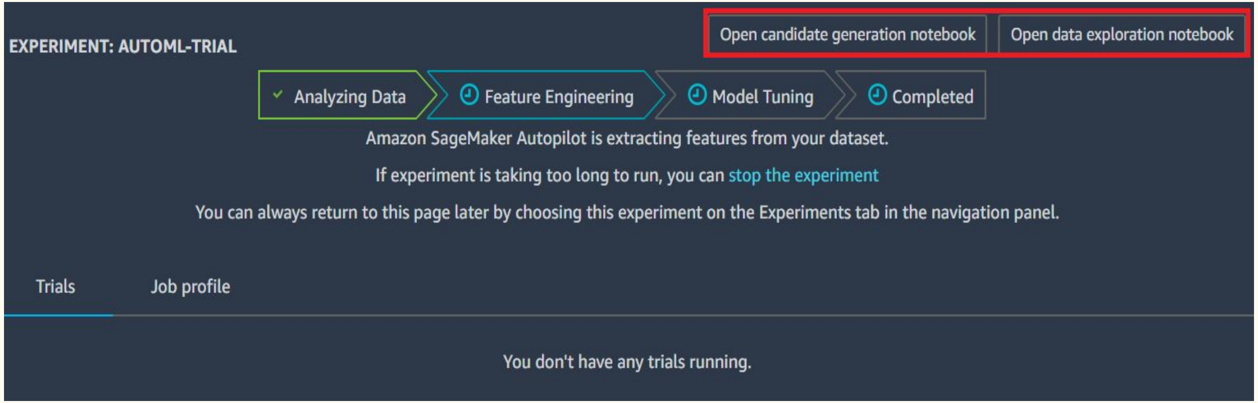


Fig.4. Data Exploration notebook and Candidate Generation notebook in Studio Autopilot Experiment

01 SageMaker Studio

1. **Data Preparation** — Studio Notebook
2. **Autopilot Setup** — Studio
3. **Pre-processing & Training** — Studio
4. **Deployment** — Studio
5. **Test & Cleanup** — Studio Notebook

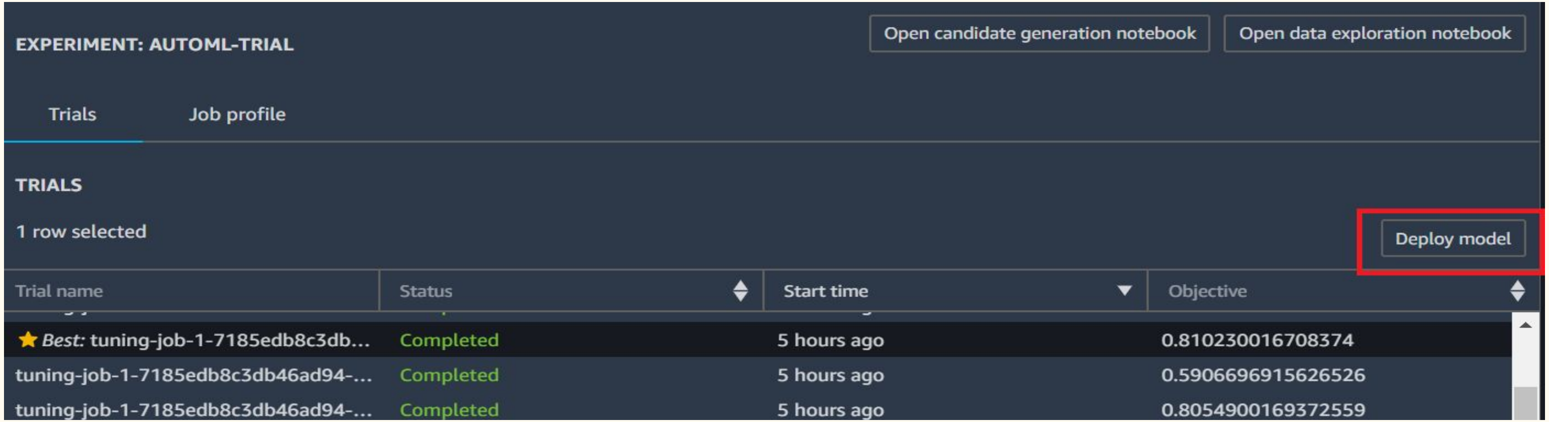
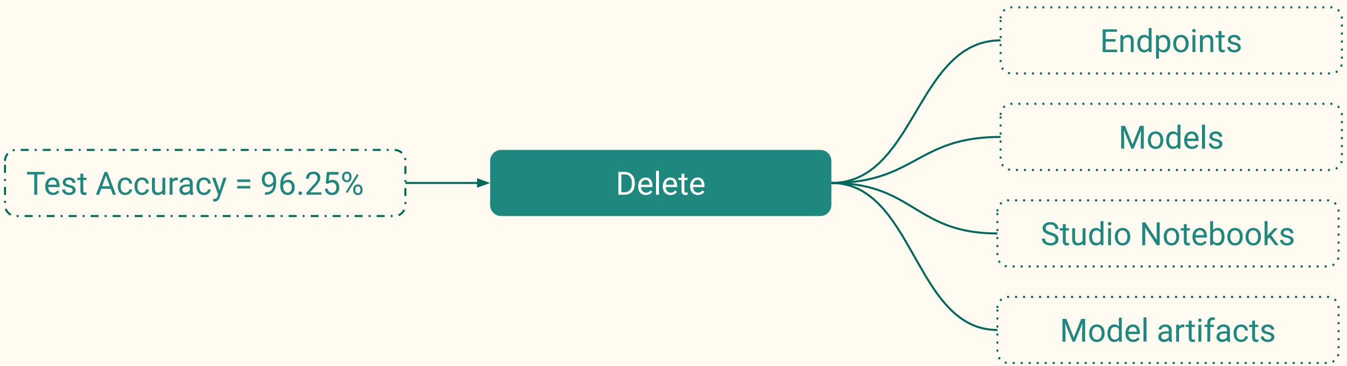


Fig. 5. Autopilot automatically chooses the best model after Training jobs

01 SageMaker Studio

1. **Data Preparation** — Studio Notebook
2. **Autopilot Setup** — Studio
3. **Pre-processing & Training** — Studio
4. **Deployment** — Studio
5. **Test & Cleanup** — Studio Notebook



02 Notebook Instance

1 Notebook Setup & Data Preparation

- 80% Training & Validation
- 20% Test

2 Autopilot Setup

- input/output S3 location
- target 'Churn?'

3 Pre-processing & Training

- Analysing Data
- Feature Engineering
- Hyperparameter Tuning

4 Deployment

- best model name, arn
- endpoint name, configuration

5 Test & Cleanup

Test Accuracy = 95.65% → Delete

- Endpoints
- Models
- Studio Notebooks
- Model artifacts

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	✓	✗	- 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	✓	✗	- 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**

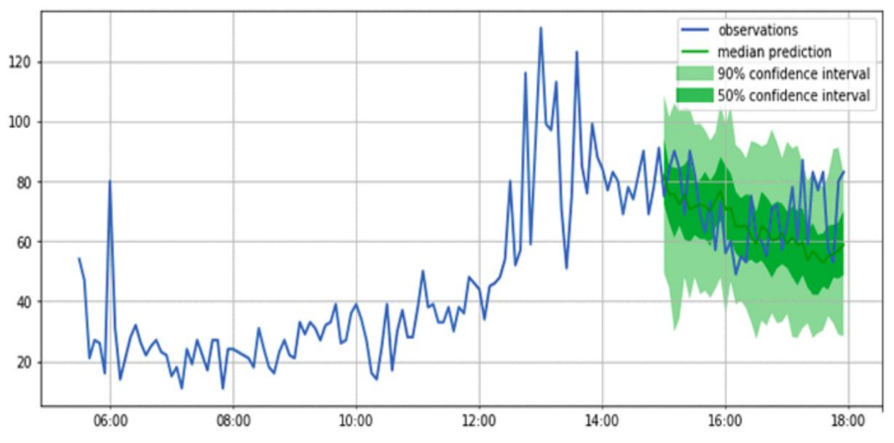


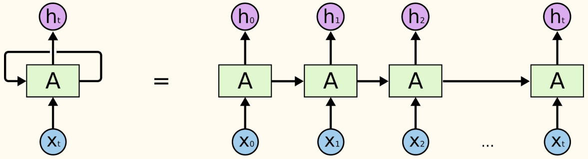
Fig. 6. DeepAR Inference visualisation*

* 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

1 — Notebook Setup

2 — Data Preparation and Preprocessing

3 — Feature Selection

4 — Train & Deploy

5 — Test & Cleanup

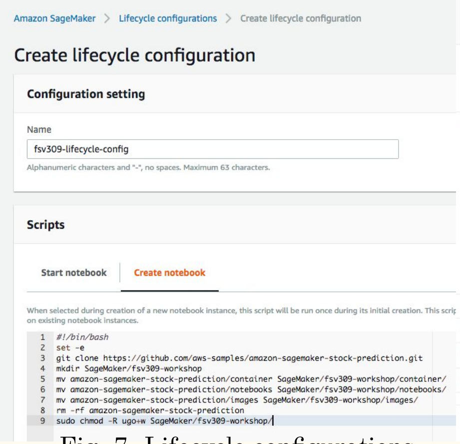


Fig. 7. Lifecycle configurations

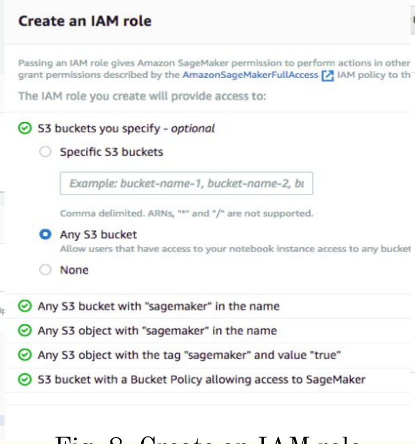


Fig. 8. Create an IAM role

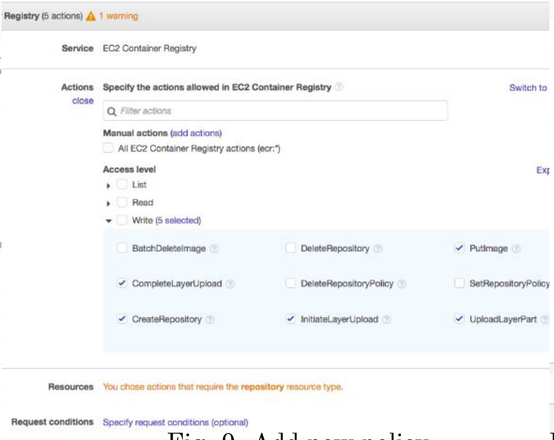


Fig. 9. Add new policy

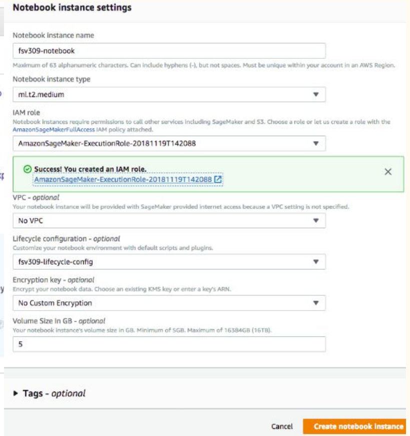


Fig. 10. Create notebook instance

02 Notebook Instance

1 Notebook Setup — 2 Data Preparation and Preprocessing — 3 Feature Selection — 4 Train & Deploy — 5 Test & Cleanup

- 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

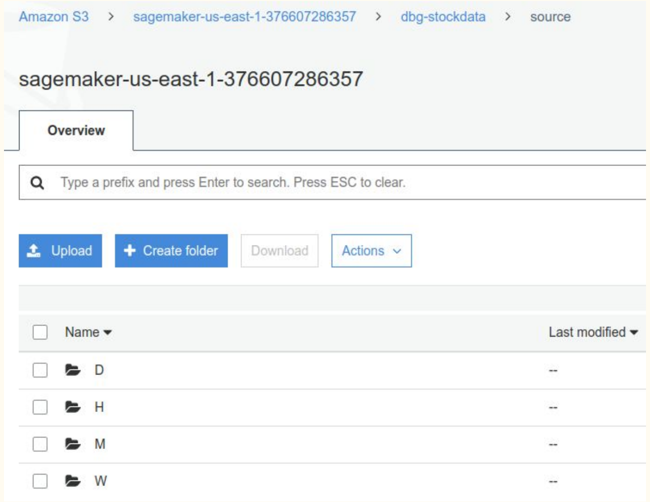


Fig.11. Processed data on S3 bucket

02 Notebook Instance

1. Notebook Setup
2. Data Preparation and Preprocessing
3. Feature Selection
4. Train & Deploy
5. Test & Cleanup

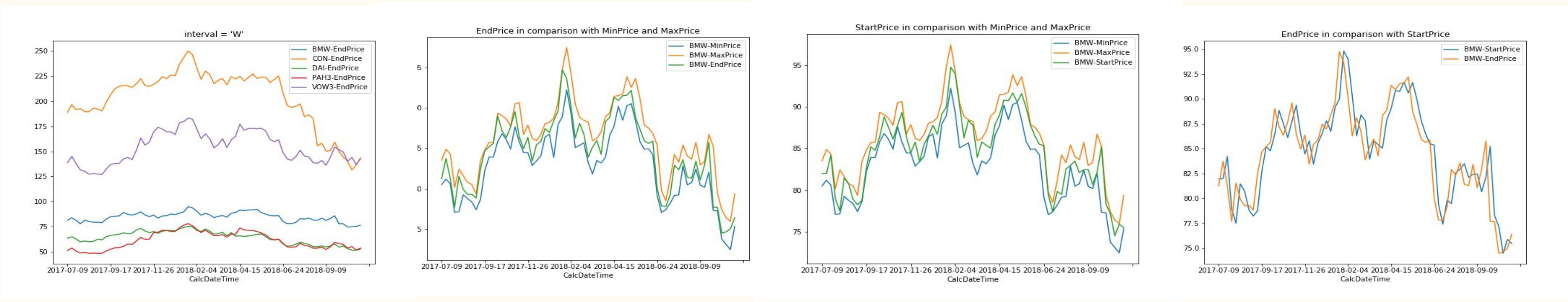


Fig.12. Data exploration

02

Notebook Instance

1 Notebook Setup

2 Data Preparation and Preprocessing

3 Feature Selection

4 Train & Deploy

5 Test & Cleanup

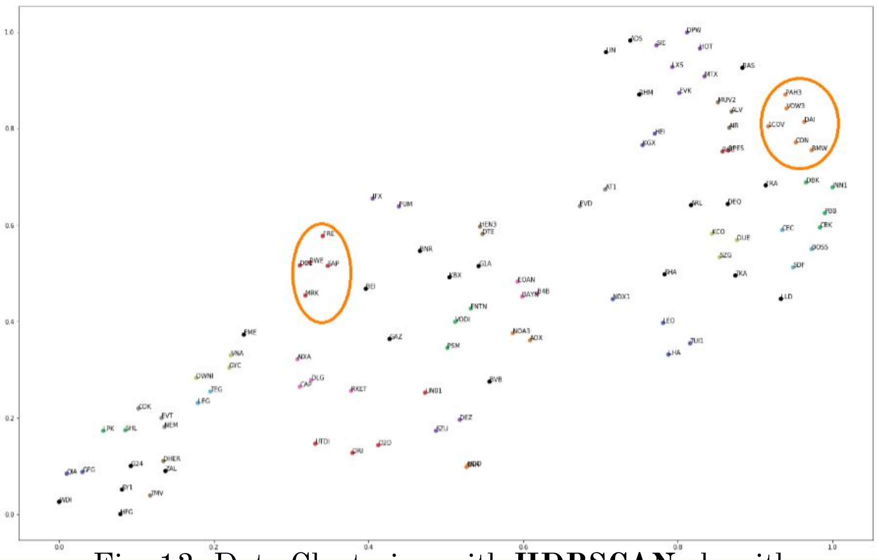


Fig. 13. Data Clustering with **HDBSCAN** algorithm

02 Notebook Instance

1 Notebook Setup — 2 Data Preparation and Preprocessing — 3 Feature Selection — 4 Train & Deploy — 5 Test & Cleanup

80% Training set

20% Test set

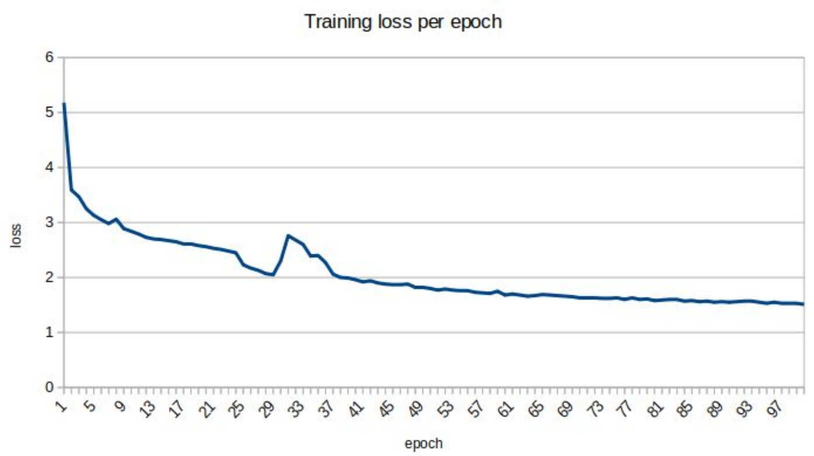


Fig. 14. Training loss per epoch

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

02 Notebook Instance

1 — Notebook Setup

2 — Data Preparation and Preprocessing

3 — Feature Selection

4 — Train & Deploy

5 — Test & Cleanup

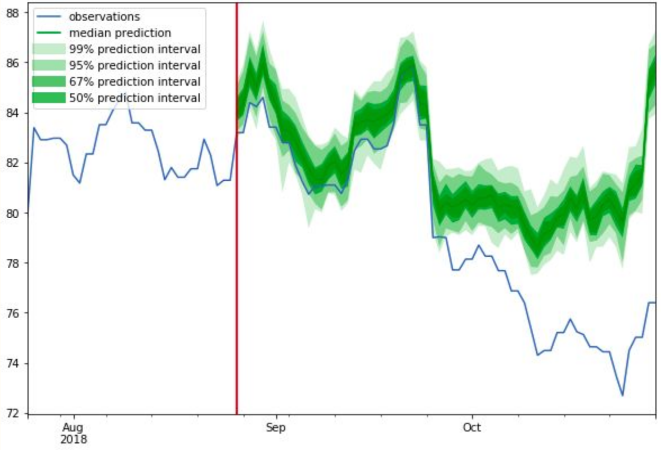


Fig. 15. BMW EndPrice Prediction with **DeepAR** algorithm

Test RMSE = 12.01

Delete

- Endpoints
- Models
- Studio Notebooks
- Model artifacts

03 SageMaker Console

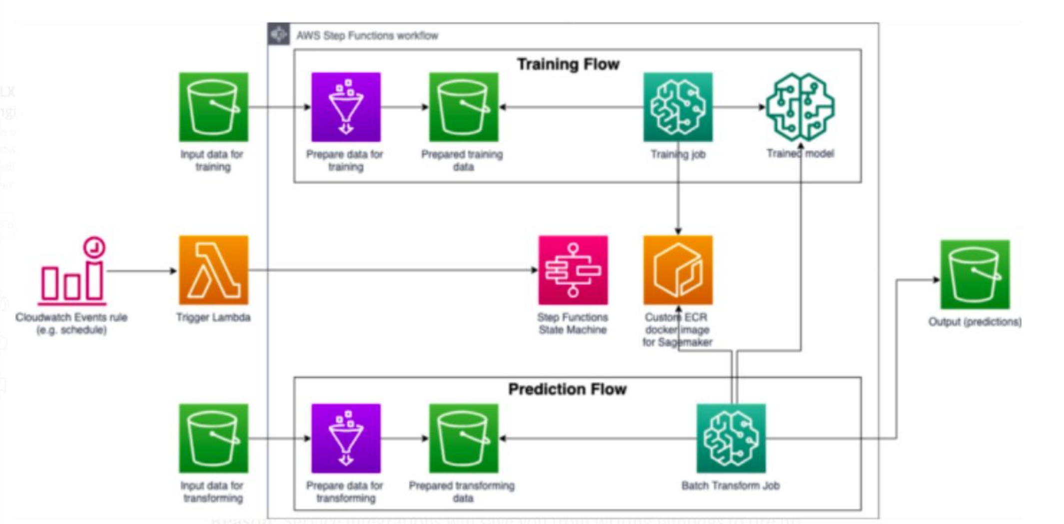


Fig. 16. Suggested Machine learning workflow for production*

- 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

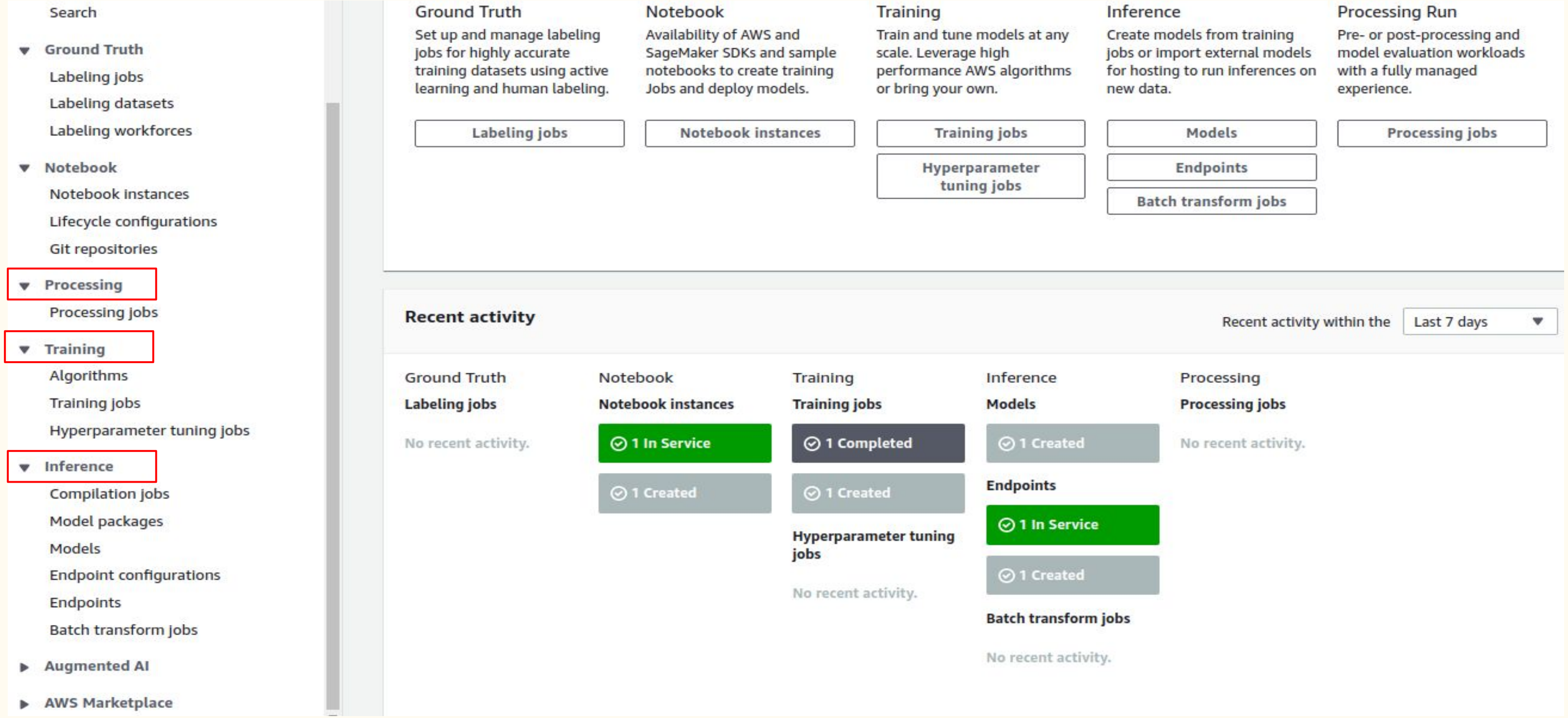


Fig. 17. SageMaker Console

03 SageMaker Console

1 Preprocessing — 2 Training — 3 Inference

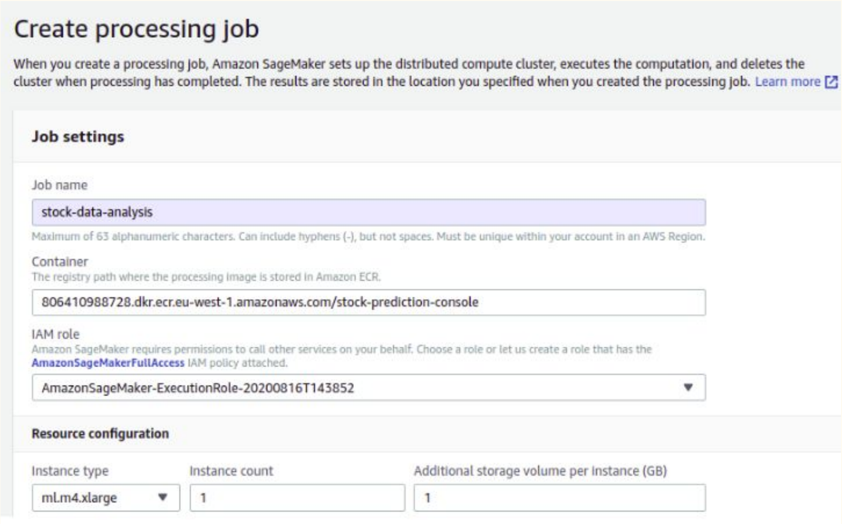


Fig. 18. Create a processing job from Console

03 SageMaker Console

1 — 2 — 3

Preprocessing Training Inference

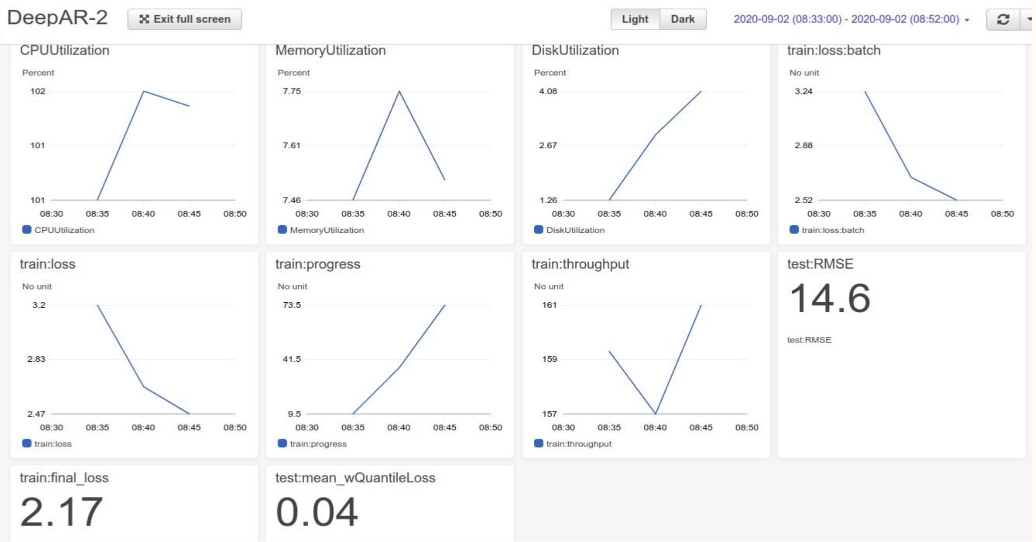


Fig. 19. Train/Test metrics

03

SageMaker Console

1 — 2 — 3

Preprocessing

Training

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	• conda-installed • other data science packages	• process, train and deploy with one click
3	Research workflow	• data exploration • execute and share 'technical' papers	• create Lifecycle Configuration, IAM, VPC, and label data
4	Production workflow	• need advanced infrastructures (e.g. Netflix) to use at scale	• 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.

M.Sc. in Business Analytics

Capstone Presentation

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