#AlwaysTrainingWeights&Biases

Group 5

Team Batak

"You can't spell
MUSCLE
without MLE."
- ssob Marc

apper.ph

Table of Contents

apper.ph

Meet the Team

What I like about Coach Arvs

Edward Vincent “**VINCE**” Duero

I like how he makes the class more interactive and the lessons more hands-on 😁

Rosiel Jazmine “**ROSE**” Villareal

Career advice & life lessons <3

Jericho Carlo “**ECHO**” Agudo

There are no wrong answers in his class
Coach Arvs is a great mentor :)

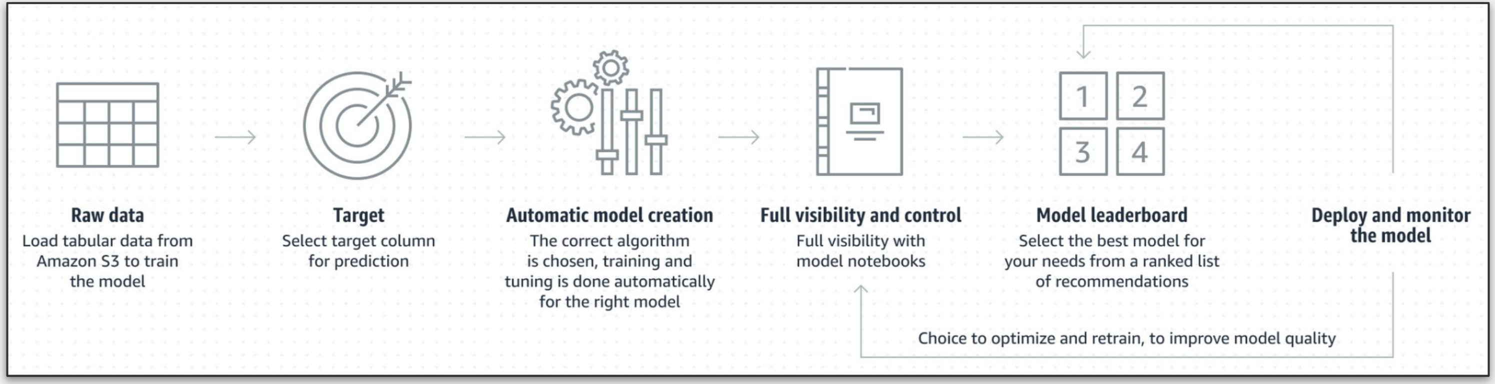
Most Valuable Capability

SageMaker Autopilot

SageMaker Autopilot

SageMaker Autopilot **automatically** builds, trains, and tunes the best ML models based on your data, while **maintaining full control visibility**

SageMaker Autopilot eliminates the **heavy lifting** of building ML models



apper.ph

Rose: What I Like About Autopilot

Even if it automates the different steps of the ML process, we still have **the option to open the notebooks generated by Autopilot** to understand what happened behind the scenes and to **tweak them further** according to our liking :)

Autopilot can free up my time – I usually spend 80% of my time in a data science project on data cleaning and feature engineering – so I can focus on more important matters, such as that of translating data science insights into actionable recommendations to solve business problems.

Echo: What I Like About Autopilot

You may more easily comprehend and explain how models produced make predictions by using the **explainability report** that Amazon SageMaker Autopilot provides. Additionally, you can see the **percentage contribution** of **each attribute** in your training data to the predicted outcome.

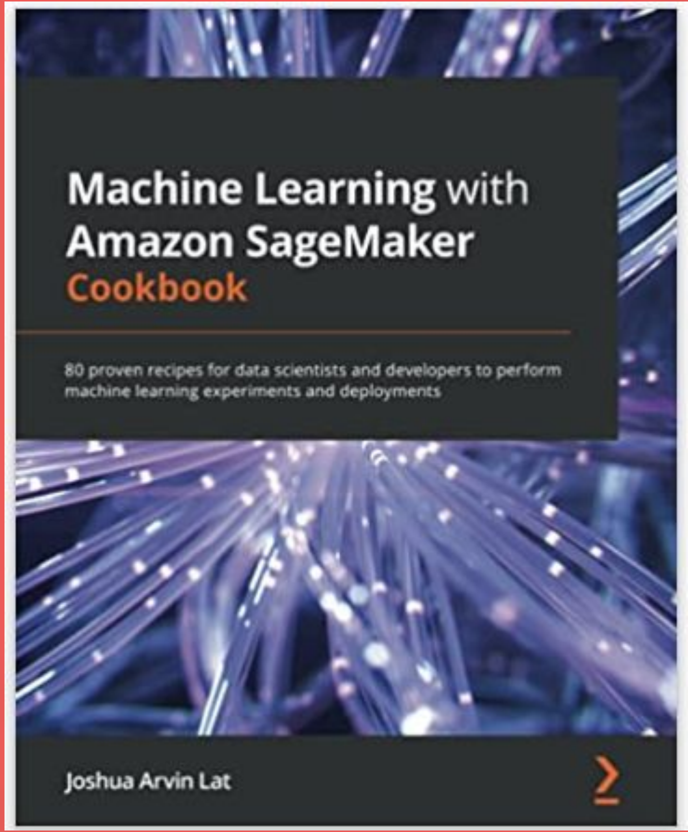
SageMaker Autopilot also allows you to examine and review each ML model that is automatically generated for your data

Vince: What I Like About Autopilot

SageMaker Autopilot automates the machine learning process which is great for developing models for **small** or **common machine learning problems**. This can be helpful for **creating Minimum Viable Products (MVPs)**, **testing** and **comparing machine learning models**, and **quick models for insights** among others

SageMaker Autopilot also **democratizes the machine learning field**. Business analysts and subject matter experts can perform their tasks **without depending to machine learning engineers or data scientists** to generate machine learning models

Amazon SageMaker Cookbook



CHAPTER 7

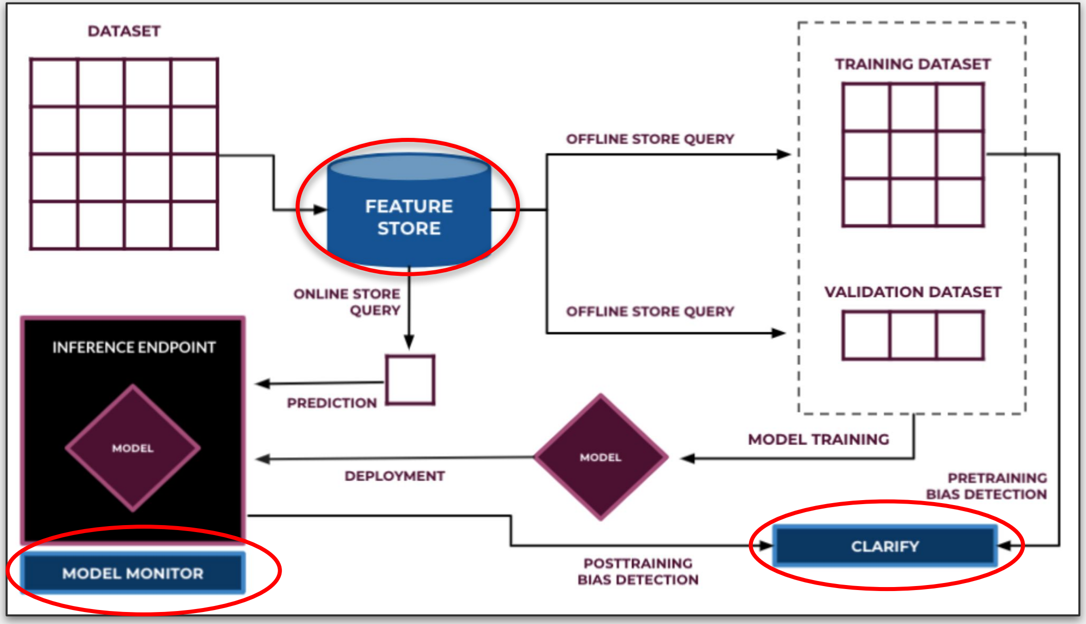
Working with SageMaker Feature Store, SageMaker Clarify, and SageMaker Model Monitor

Chapter 7 SageMaker Feature Store, SageMaker Clarify, and SageMaker Model Monitor

SageMaker Feature Store

SageMaker Clarify

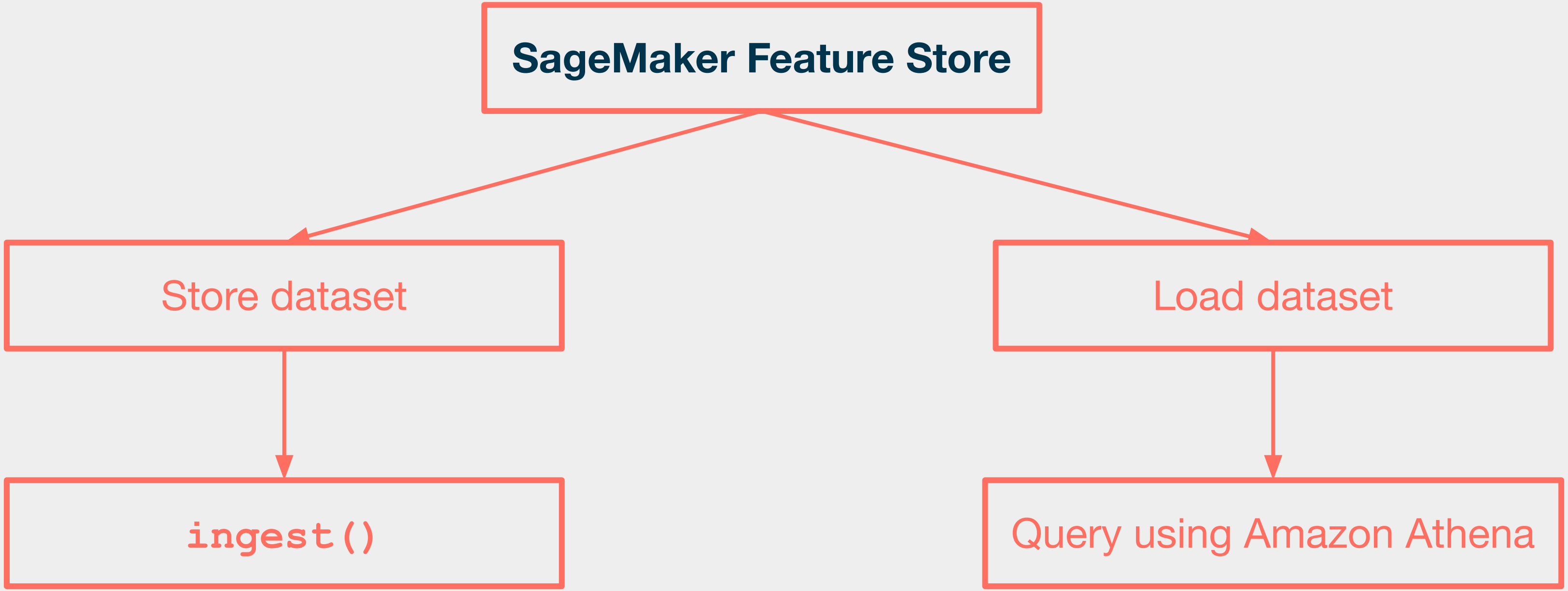
SageMaker Model Monitor



apper.ph

Chapter 7 SageMaker Feature Store

For **storing** and **managing** ML features in our dataset



Chapter 7 SageMaker Clarify

For **detecting pre-training** and **post-training bias**

Pre-Training Bias Metrics

- Difference in Positive Proportions in Labels (DPL)
- Kullback-Liebler divergence (KL)
- Jensen-Shannon divergence (JS)
- LP Norm (LP)
- Total Variation Distance (TVD)
- Komogorov-Smirnov (KS)

Post-Training Bias Metrics

- Disparate Impact (DI)
- Difference in Conditional Acceptance (DCA)
- Difference in Conditional Rejection (DCR)
- Difference in Acceptance Rates (DAR)
- Difference in Rejection Rates (DRR)
- Accuracy Difference (AD)
- Conditional Demographic Disparity of Predicted Labels (CDDPL)
- Treatment Equality (TE)
- Flip Test (FT)

Chapter 7 SageMaker Clarify

Also provides **ML explainability report** for understanding **how a machine learning model performed** a prediction

Creates ML explainability report by computing the **SHAP values**

Shapley values are used to **interpret machine learning models** and provide details regarding the **importance of each feature in the dataset**

Chapter 7 SageMaker Model Monitor

Capturing and **analyzing** the data in our inference endpoint

Monitor and Detect the following Issues

- Drift in data quality
- Drift in model quality metric values
- Bias drift during prediction
- Feature attribution drift

CHAPTER 8

Solving NLP, Image Classification, & Time-Series Forecasting Problems with Built-in Algorithms

Chapter 8 Introduction to Built-in Algorithms for Text, Image, & Time-Series Data

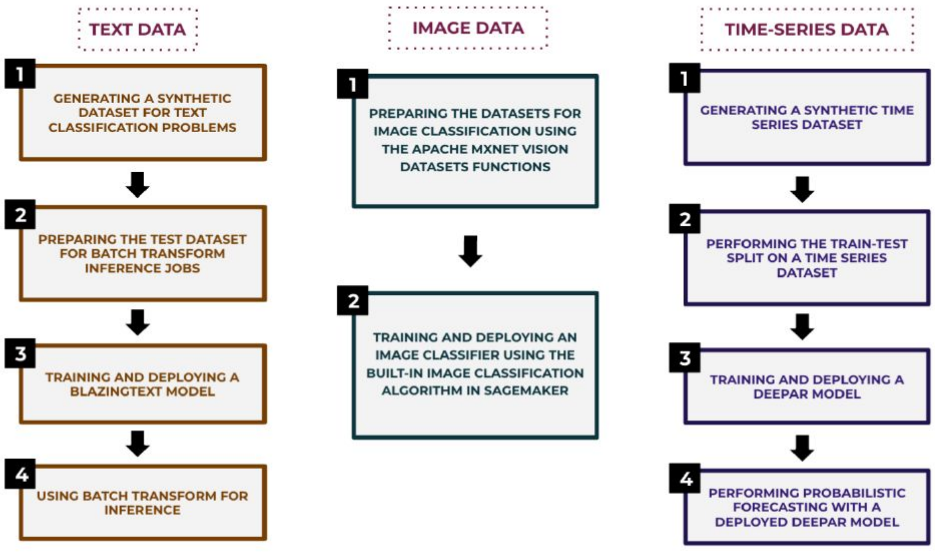


Figure 8.1 – Working with text classification, image classification, and time-series forecasting problems with built-in algorithms

- **BlazingText**
  text classification on reviews
- **Image Classification Algorithm**
  on MNIST handwritten digits
- **DeepAR Forecasting Algorithm**
  on time-series data

Chapter 8 BlazingText for Text Classification of Food Reviews

Generated synthetic **positive** and **negative reviews** using `faker.sentence()`

```
{"source": "There Are Better Restaurants Out There tastes bad spaghetti chicken soup food in the restaurant"}
```

Trained a **BlazingText** model in **supervised mode for text classification**

```
mode="supervised",
min_count=2
```

```
[
  {
    "label": [
      "__label__negative"
    ],
    "prob": [
      0.8144211173057556
    ]
  },
```

(a) Deployed model to a **real-time inference endpoint**
(b) Used **Batch Transform** to predict on entire dataset

Chapter 8 **Image** Classification of MNIST Handwritten Digits

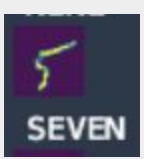
Used **Apache MXNet Vision Datasets MNIST** class to load the dataset & split it into train/valid/test dir

Trained an **Image Classification Algorithm** based on **convolutional neural networks (CNNs)** & set **hyperparameters**

```
num_layers=18,
image_shape = "1,28,28",
num_classes=10,
num_training_samples=600,
mini_batch_size=20,
epochs=5,
learning_rate=0.01,
top_k=2,
```

Deployed model to an **endpoint** and used **argmax**, or class with **highest probability score** as prediction

```
best_index = results_prob_list.index(
    max(results_prob_list)
```



Chapter 8 DeepAR Forecasting on Time-Series Data

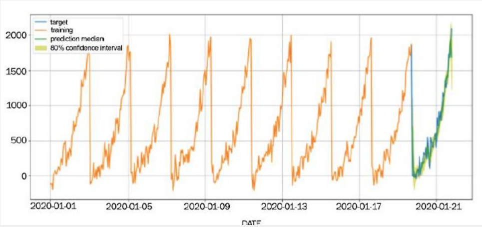
Generated synthetic, hourly time-series data with seasonality & noise using **`numpy`**, **`pandas`**, & **`random`** libraries

```
{'freq': 'H',
 't0': '2020-01-01 00:00:00',
 'length': 500,
 'data': 2020-01-01 00:00:00    -118.
 2020-01-01 01:00:00    -121.417558
 2020-01-01 02:00:00    -113.624594
 2020-01-01 03:00:00    -197.074225
 2020-01-01 04:00:00      11.191463
```

Trained a **DeepAR Forecasting Algorithm** based on **recurrent neural networks (RNNs)** & set **hyperparameters**

```
time_freq=freq,
context_length=str(context_length),
prediction_length=str(prediction_length),
num_cells=40,
num_layers=3,
likelihood="gaussian",
epochs=20,
mini_batch_size=32,
```

Deployed model to an **endpoint** and used **quantiles** for **probabilistic** forecasting of future values from historical values



apper.ph

CHAPTER 9

Managing Machine Learning Workflows and Deployments

Chapter 9 Summary

Focuses on more complex set of deployment solutions for **real-time endpoint deployments** and **automated workflows**

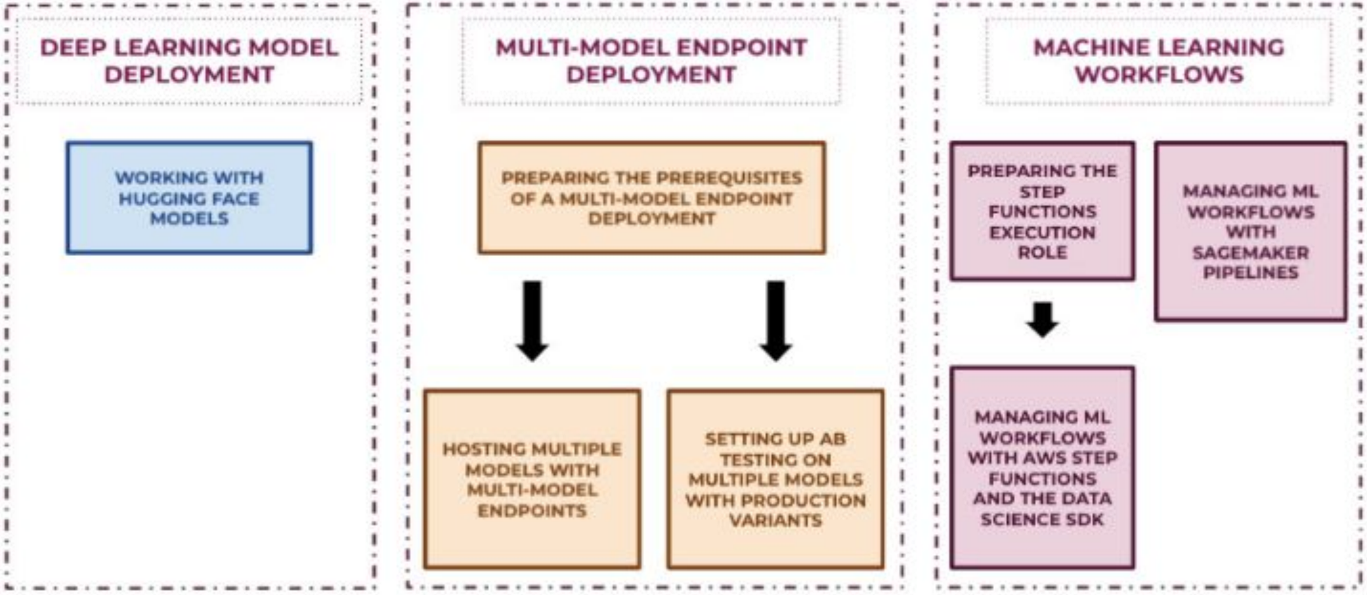


Figure 9.1 – How the recipes in this chapter are divided

Chapter 9 Summary

What we learned:

- Working with Hugging Face Models
- Preparing the prerequisites of multi-model endpoint deployment
- Hosting multiple models with multi-model endpoints
- Setting up A/B testing on multiple models with production variants
- Preparing the Step Functions execution role
- Managing ML workflows with AWS Step Functions and the Data Science SDK
- Managing ML workflows with SageMaker Pipelines

Summary

Meet the Team

Most Valuable SageMaker Capability

✅ SageMaker Autopilot

Amazon SageMaker Cookbook

✅ Chapter 7: Feature Store, Clarify, Model Monitor
✅ Chapter 8: Built-in Algorithms for Text, Image, Time-Series Data
✅ Chapter 9: ML Workflows & Deployments

apper.ph