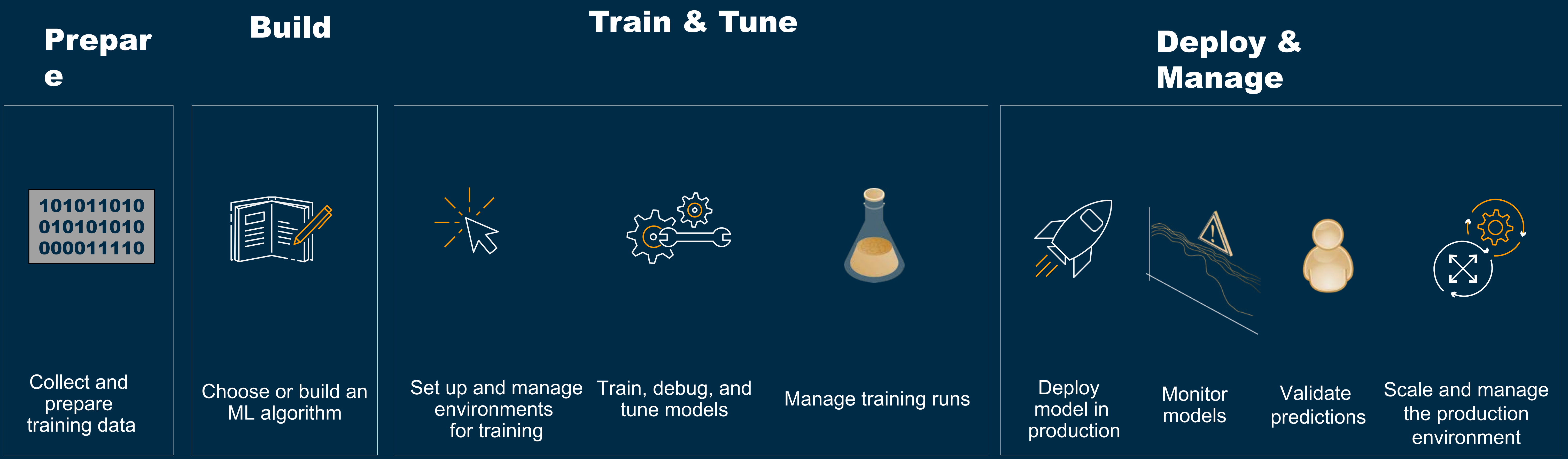
Build, Train, and Deploy Machine Learning Models with Amazon SageMaker

Julien Simon

Global Evangelist, AI & Machine Learning
Amazon Web Services
@julsimon

aws

The machine learning workflow is iterative and complex



Prepar e

Collect and prepare training data

Build

Choose or build an ML algorithm

Train & Tune

Set up and manage environments for training

Train, debug, and tune models

Manage training runs

Deploy & Manage

Deploy model in production

Monitor models

Validate predictions

Scale and manage the production environment

Amazon SageMaker helps you build, train, and deploy models

Web-based IDE for machine learning

Prepare

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

Build

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

Train & Tune

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

Visually track and compare experiments

Manage training runs

Deploy & Manage

Automatically build and train n

One-click deployment and autoscaling

Deploy model in production

Automatically spot concept drift

Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

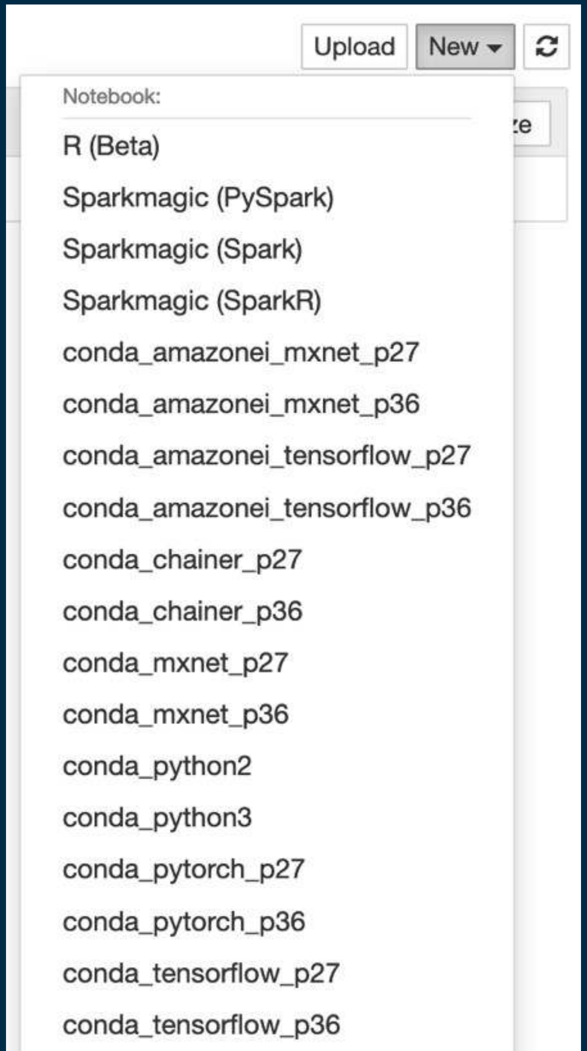
Scale and manage the production environment

Modular service and APIs, from experimentation to production

Build

Amazon SageMaker Notebook Instances

- Fully managed instances, from *ml.t2.medium* to *p3.16xlarge*
- Pre-installed with Jupyter and Conda environments
  - Python 2.7 & 3.6
  - Open-source libraries (TensorFlow, Apache MXNet, etc.)
  - Beta support for R
  - Amazon Elastic Inference for cost-effective GPU acceleration
- Lifecycle configurations
- VPC, encryption, etc.
- Get to work in minutes



Amazon SageMaker Studio

Fully integrated development environment (IDE) for machine learning

Collaboration at scale

Share notebooks without tracking code dependencies

Easy experiment management

Organize, track, and compare thousands of experiments

Automatic model generation

Get accurate models with full visibility & control without writing code

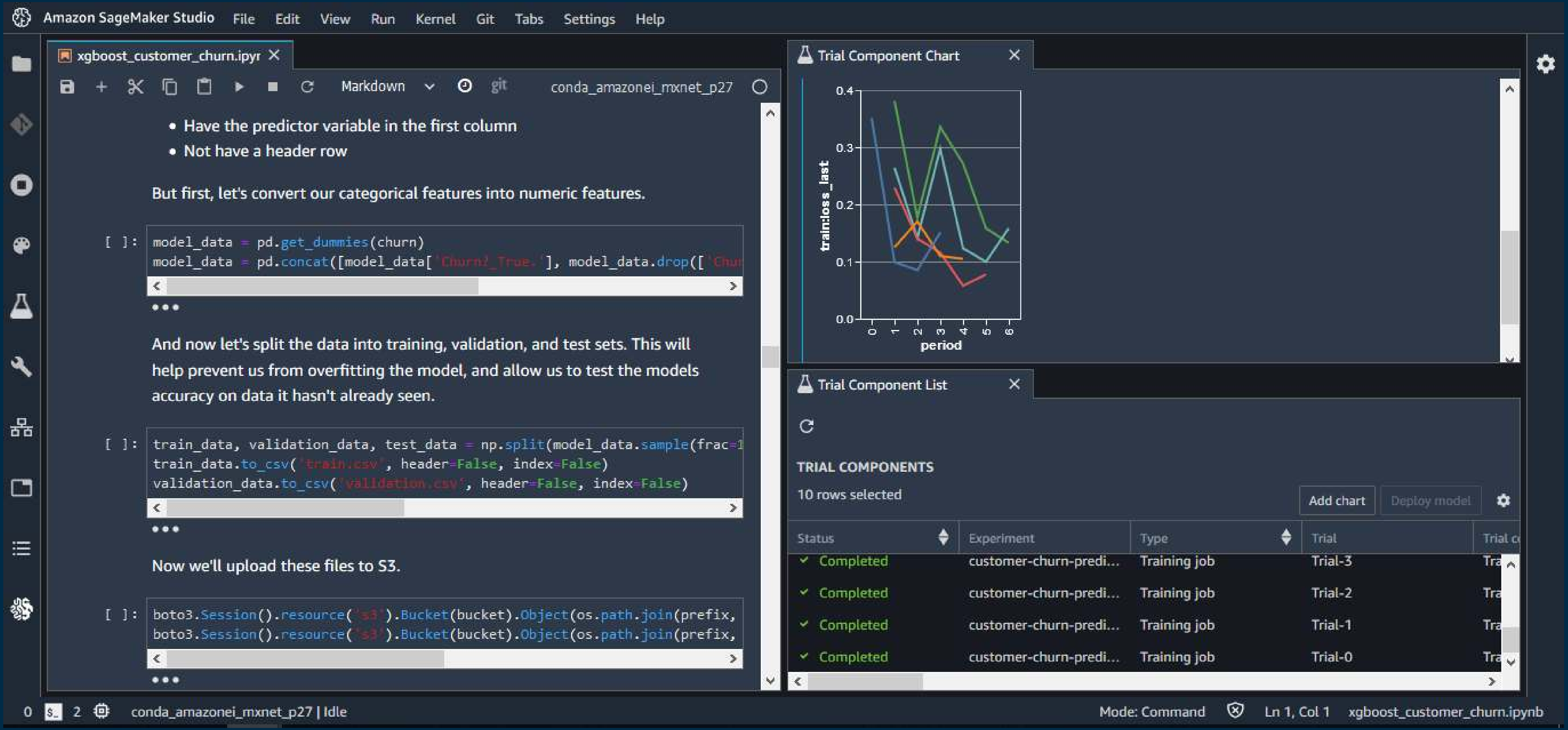
Higher quality ML models

Automatically debug errors, monitor models, & maintain high quality

Increased productivity

Code, build, train, deploy, & monitor in a unified visual interface

Amazon SageMaker Studio



- Have the predictor variable in the first column
- Not have a header row

But first, let's convert our categorical features into numeric features.

```
[ ]: model_data = pd.get_dummies(churn)
     model_data = pd.concat([model_data['Churn?_True.'], model_data.drop(['Chur
```

And now let's split the data into training, validation, and test sets. This will help prevent us from overfitting the model, and allow us to test the models accuracy on data it hasn't already seen.

```
[ ]: train_data, validation_data, test_data = np.split(model_data.sample(frac=1
     train_data.to_csv('train.csv', header=False, index=False)
     validation_data.to_csv('validation.csv', header=False, index=False)
```

Now we'll upload these files to S3.

```
[ ]: boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
     boto3.Session().resource('s3').Bucket(bucket).Object(os.path.join(prefix,
```

TRIAL COMPONENTS

10 rows selected

Status	Experiment	Type	Trial	Trial c
✓ Completed	customer-churn-predi...	Training job	Trial-3	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-2	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-1	Tra
✓ Completed	customer-churn-predi...	Training job	Trial-0	Tra

Model options

AWS Marketplace for Machine Learning

Training code

Amazon SageMaker AutoPilot

Factorization Machines
Linear Learner
Principal Component Analysis
K-Means Clustering
Etc.

Built-in Algorithms (17)
No ML coding required

mxnet
Chainer
TensorFlow
PyTorch
scikit learn
dmlc XGBoost

Built-in Frameworks
Bring your own code
Open source containers

Bring Your Own
Full control, run your container
R, C++, etc.

Fully managed training, spot instances included

Built-in algorithms

Orange: supervised, yellow: unsupervised

Linear Learner: Regression, classification	**Image Classification**: Deep learning (ResNet)
Factorization Machines: Regression, classification, recommendation	**Object Detection (SSD)**: Deep learning (VGG or ResNet)
K-Nearest Neighbors: Non-parametric regression and classification	**Neural Topic Model**: Topic modeling
XGBoost: Regression, classification, ranking https://github.com/dmlc/xgboost	**Latent Dirichlet Allocation**: Topic modeling (mostly)
K-Means: Clustering	**BlazingText**: GPU-based Word2Vec, and text classification
Principal Component Analysis: Dimensionality reduction	**Sequence to Sequence**: Machine translation, speech to text and more
Random Cut Forest: Anomaly detection	**DeepAR**: Time-series forecasting (RNN)
Object2Vec: General-purpose embedding	**IP Insights**: Usage patterns for IP addresses
Semantic Segmentation: Deep learning	

Built-in frameworks: Just add your code

TensorFlow mxnet PYTORCH

Keras GLUON Chainer dmlc XGBoost

- Built-in containers for training and prediction
  - Open-source, e.g., https://github.com/aws/sagemaker-tensorflow-containers
  - Build them, run them on your own machine, customize them, etc.

- Local mode: Train and predict on your notebook instance, or on your local machine

- Script mode: Reuse existing code with minimal changes

Amazon SageMaker Autopilot

Automatic model creation with full visibility & control

Quick to start

Provide your data in a tabular form & specify target prediction

Automatic model creation

Get ML models with feature engineering & model tuning automatically done

Visibility & control

Get notebooks for your models with source code

Recommendations & Optimization

Get a leaderboard & continue to improve your model

Train

The Amazon SageMaker API

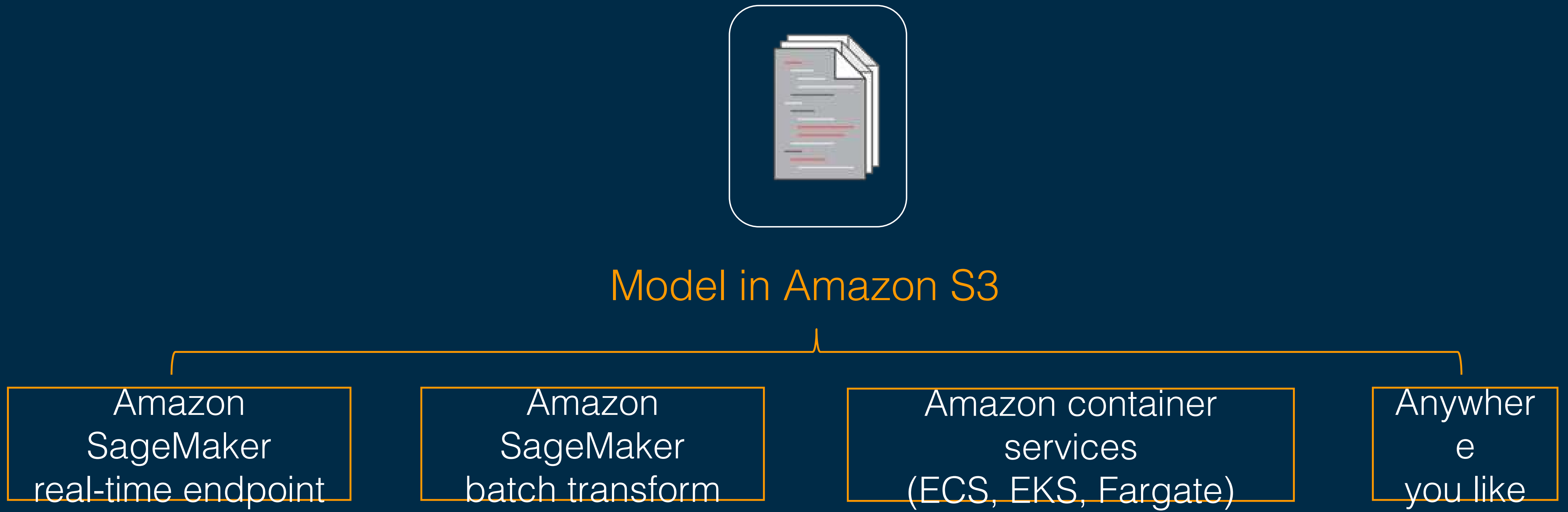
- Python SDK orchestrating all Amazon SageMaker activity
  - High-level objects for algorithm selection, training, deploying, etc. https://github.com/aws/sagemaker-python-sdk

  - Spark SDK (Python & Scala) https://github.com/aws/sagemaker-spark/tree/master/sagemaker-spark-sdk

- AWS SDK
  - Service-level APIs for scripting and automation
  - CLI: '*aws sagemaker*'
  - Language SDKs: boto3, etc.

Deploy

Deployment options



Model in Amazon S3

Amazon SageMaker real-time endpoint

1 line of code

Vanilla HTTPS

Post data, get a prediction

Any tool, any language

Auto Scaling available

Amazon SageMaker batch transform

1 line of code

Predict data stored in S3

Read results from S3

Amazon container services (ECS, EKS, Fargate)

Use AWS Deep Learning containers

Use your own container

Anywher e you like

Grab the model in S3 and run!

Fully managed deployment

Getting started

http://aws.amazon.com/free

https://ml.aws
https://aws.amazon.com/sagemaker

https://github.com/aws/sagemaker-python-sdk
https://github.com/awslabs/amazon-sagemaker-examples

https://amazonsagemakerfridays.splashthat.com/
https://reinvent.awsevents.com/

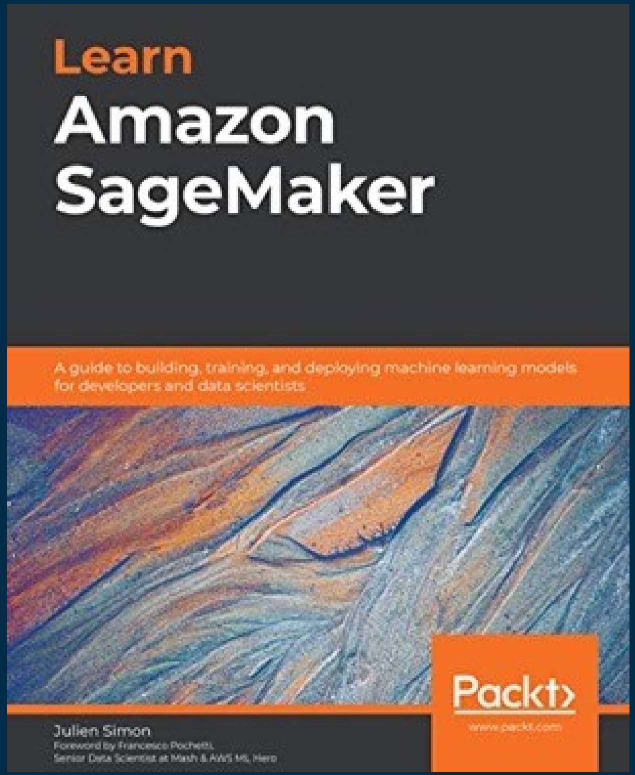
https://youtube.com/juliensimonfr
https://medium.com/@julsimon

Published August 2020

Discount link for the paper edition on Amazon (US only)
https://www.amazon.com/gp/mpc/AOHJSZC7A0AV5

Discount code for the e-book edition on Packt
20SAGEMAKER
https://www.packtpub.com/product/learn-amazon-sagemaker/9781800208919

Valid until November 11th



Learn
Amazon
SageMaker

A guide to building, training, and deploying machine learning models for developers and data scientists

Julien Simon
Foreword by Francesco Pochetti,
Senior Data Scientist at Mash & AWS ML Hero

Packt
www.packt.com