DEV DAY

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

MLS1

Build, Train, and Deploy Your Machine Learning Models

Julien Simon
Global Evangelist, AI & Machine Learning
Amazon Web Services

@julsimon

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The machine learning workflow is iterative and complex

Prepare

101011010
010101010
000011110

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

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker helps you build, train, and deploy models

**Prepar
e**

Build

Train & Tune

Deploy & Manage

Web-based IDE for machine learning

Automatically build and train m

Fully managed data processing jobs and data labeling workflows

101011010
010101010
000011110

Collect and prepare training data

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

Choose or build an ML algorithm

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

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

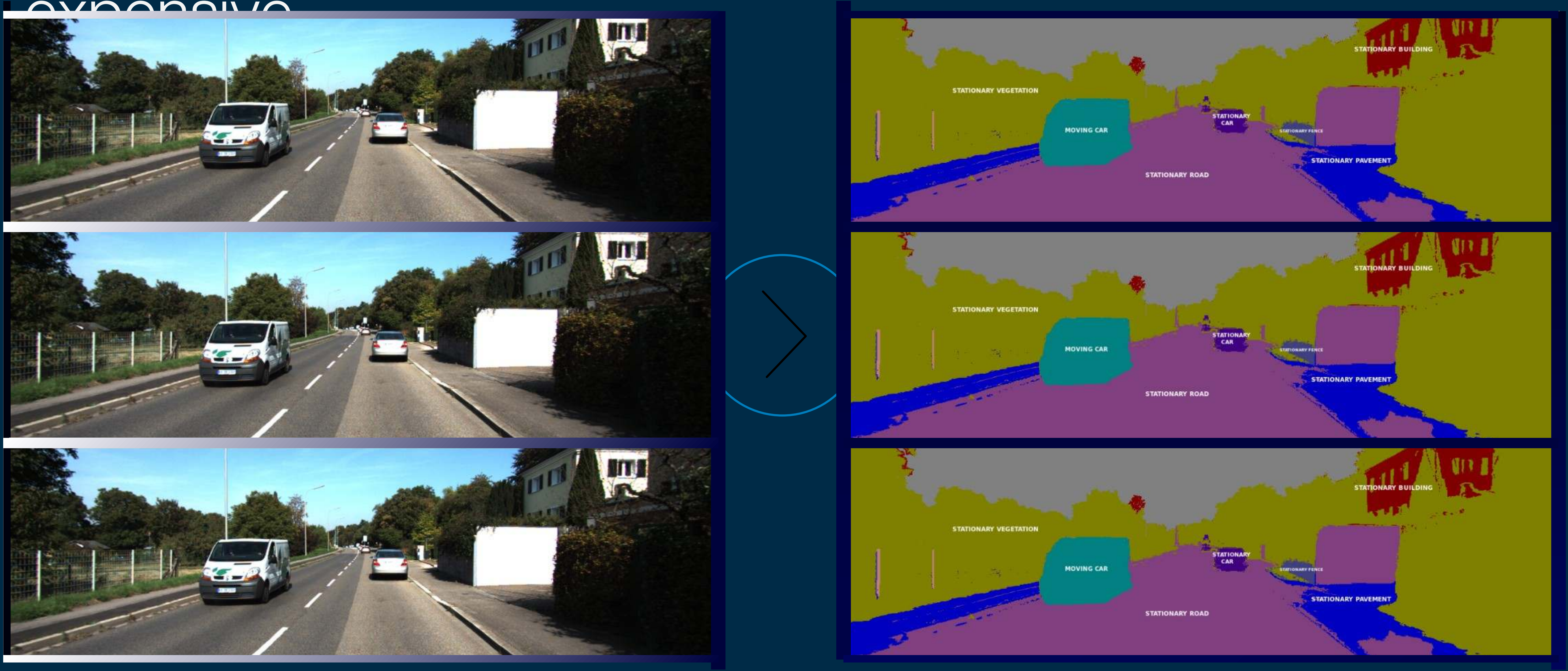
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

Prepare

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Annotating data at scale is time-consuming and expensive



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Ground Truth

Build scalable and cost-effective labeling workflows

Quickly label training data

Easily integrate human labelers

Get accurate results

KEY FEATURES

Automatic labeling via machine learning

Ready-made and custom workflows

Private, 3rd party, and public workforce

aws

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Processing

Analytics jobs for data processing and model evaluation

Fully managed

Achieve distributed processing for clusters

Custom processing

Bring your own script for feature engineering

Container support

Use SageMaker's built-in containers or bring your own

Security and compliance

Leverage SageMaker's security & compliance features

Automatic creation & termination

Your resources are created, configured, & terminated automatically

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

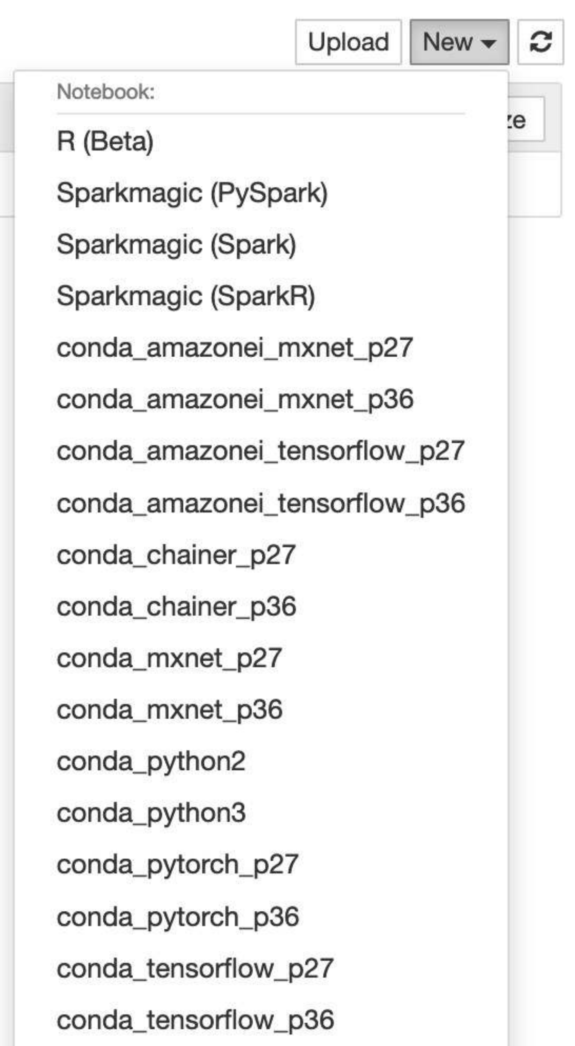
DEV DAY

Build

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

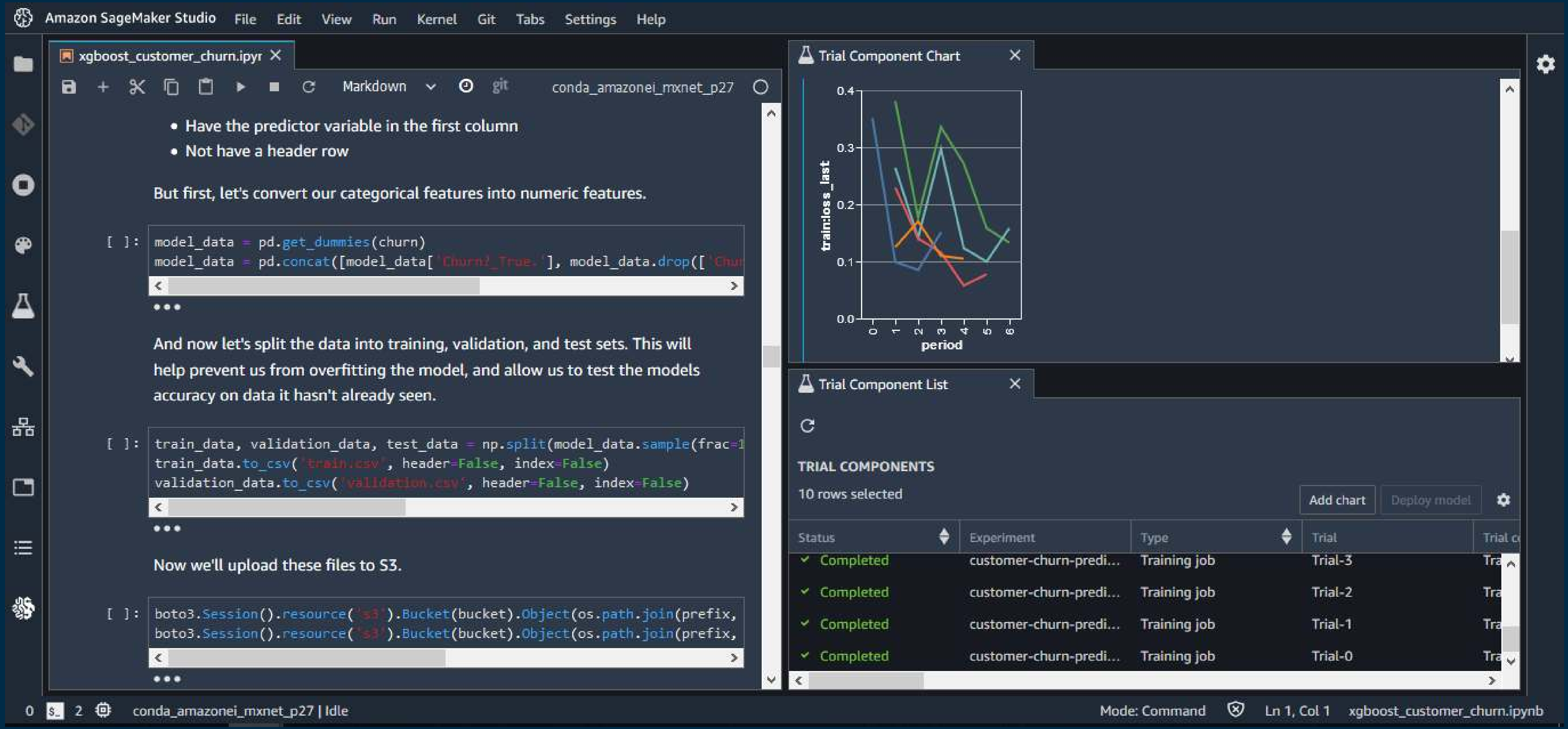
Automatically debug errors, monitor models, & maintain high quality

Increased productivity

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

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Studio



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Model options

AWS Marketplace for Machine Learning

Training code

Amazon SageMaker AutoPilot

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

mxnet
Chainer
TensorFlow
PyTorch
scikit learn
dmlc XGBoost

Built-in Algorithms (17)

No ML coding required

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

aws

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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	

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

Train & Tune

aws

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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.

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Experiments

Organize, track, and compare training experiments

Tracking at scale

Track parameters & metrics across experiments & users

Custom organization

Organize experiments by teams, goals, & hypotheses

Visualization

Easily visualize experiments and compare

Metrics and logging

Log custom metrics using the Python SDK & APIs

Fast Iteration

Quickly go back & forth & maintain high-quality

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Automatic Model Tuning

Automatically tune hyperparameters across algorithms

Tuning at scale

Adjust thousands of different combinations of algorithm parameters

Automated

Uses ML to find the best parameters

Faster

Eliminate days or weeks of tedious manual work

Examples

<u>Decision Trees</u>

Tree depth
Max leaf nodes
Gamma
Eta
Lambda
Alpha

<u>Neural Networks</u>

Number of layers
Hidden layer width
Learning rate
Embedding dimensions
Dropout

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Debugger

Analysis and debugging, explainability, and alert generation

Relevant data capture

Data is automatically captured for analysis

Data analysis & debugging

Analyze & debug data with no code changes

Automatic error detection

Errors are automatically detected based on rules

Improved productivity with alerts

Take corrective action based on alerts

Visual analysis and debugging

Visually analyze & debug from SageMaker Studio

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

Deploy & Manage

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon SageMaker Model Monitor

Continuous monitoring of models in production

Automatic data collection

Data is automatically collected from your endpoints

Continuous Monitoring

Define a monitoring schedule and detect changes in quality against a pre-defined baseline

Flexibility with rules

Use built-in rules to detect data drift or write your own rules for custom analysis

Visual data analysis

See monitoring results, data statistics, and violation reports in SageMaker Studio

CloudWatch Integration

Automate corrective actions based on Amazon CloudWatch alerts

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

Demo

aws

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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

https://gitlab.com/juliensimon/dlnotebooks
https://youtube.com/juliensimonfr

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

DEV DAY

Appendix – TensorFlow on SageMaker

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

TensorFlow

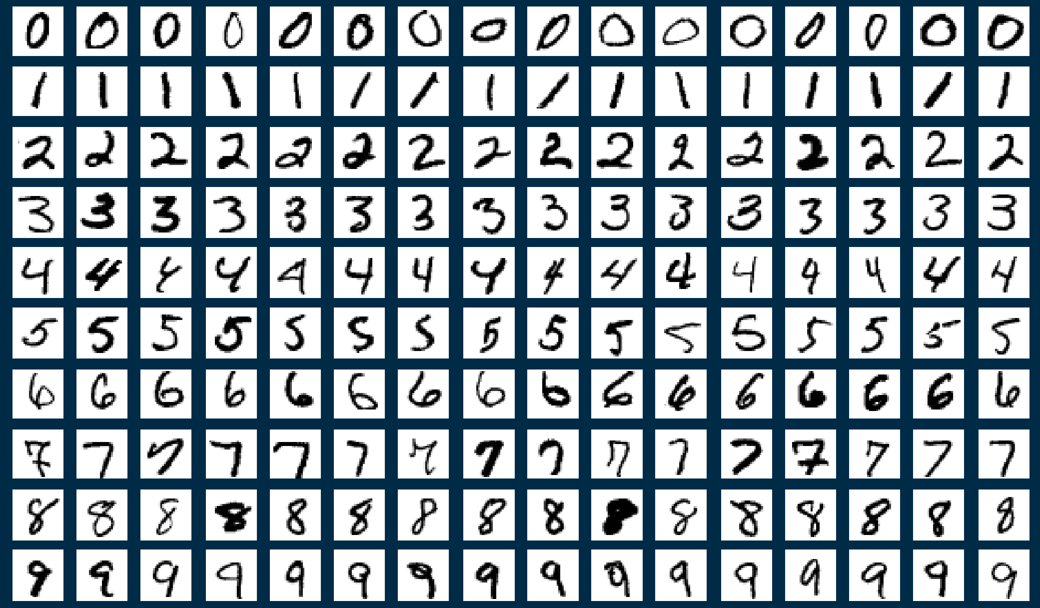
https://www.tensorflow.org

- Main API in Python, with support for Javascript, Java, C++
- TensorFlow 1.x: symbolic execution
  - 'Define then run': build a graph, optimize it, feed data, and compute
  - Low-level API: variables, placeholders, tensor operations
  - High-level API: *tf.estimator.**
  - Keras library: *Sequential* and *Functional* API, predefined layers
- TensorFlow 2.0: imperative execution (aka eager execution)
  - 'Define by run': normal Python code, similar to numpy
  - Run it, inspect it, debug it
  - Keras is the preferred API

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

TF1.x: MNIST with a Fully Connected network

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(512, activation=tf.nn.relu),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS: The platform of choice for TensorFlow

https://aws.amazon.com/tensorflow/

Adobe AUTODESK. edmunds Expedia

SIEMENS grammarly intuit Pinterest

hudl COX AUTOMOTIVE THOMSON REUTERS Zocdoc

GoDaddy News Corp Australia zendesk

zalando babylon

89% of all deep learning workloads in the cloud run on AWS

85% of all TensorFlow workloads in the cloud run on AWS

Source: Nucleus Research, T147, October 2019

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

TensorFlow: a first-class citizen on Amazon SageMaker

- Built-in TensorFlow containers for training and prediction
  - Code available on Github: https://github.com/aws/sagemaker-tensorflow-containers
  - Build it, run it on your own machine, customize it, etc.
  - Versions : 1.4.1 → 1.15, 2.0

- Not just TensorFlow
  - Standard tools: TensorBoard, TensorFlow Serving
  - SageMaker features: Local Mode, Script Mode, Model Tuning, Spot Training, Pipe Mode, Amazon EFS & Amazon FSx for Lustre, Amazon Elastic Inference, etc.
  - Performance optimizations: GPUs and CPUs (AWS, Intel MKL-DNN library)
  - Distributed training: Parameter Server and Horovod

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training a TensorFlow model

- Script mode: simply add your own code.
  - Python 3 required
  - Hyperparameters are passed as command-line arguments
  - Location of training and validation sets are passed as environment variables
  - Location where model must be saved is passed as an environment variable

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(
        entry_point=‘my_script.py’,
        role=role,
        train_instance_count=1, train_instance_type=‘ml.p3.2xlarge’,
        framework_version='1.15', py_version='py3', script_mode=True,
        hyperparameters={'epochs’: 10} )
)

tf_estimator.fit('s3://bucket/path/to/training/data')
```

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training a TensorFlow model in local mode

- You can train on the notebook instance itself, aka local mode.
- This is particularly useful while experimenting:
  you can save time and money by not firing up training instances.

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(
        entry_point='my_script.py',
        role=role,
        train_instance_count=1, train_instance_type='local',    # or 'local_gpu'
        framework_version='1.15', py_version='py3', script_mode=True,
        hyperparameters={'epochs': 10} )
)

tf_estimator.fit('file://path/to/training/data')
```

aws

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training a TensorFlow model on multiple instances

- Aka Distributed Training
- Parameter Server (native mode), or Horovod
- Amazon SageMaker takes care of all infrastructure setup.

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(
        entry_point='my_script.py',
        role=role,
        train_instance_count=8, train_instance_type='ml.p3.2xlarge',
        framework_version='1.15', py_version='py3', script_mode=True,
        hyperparameters={'epochs': 10} )
)

tf_estimator.fit('s3://bucket/path/to/training/data')
```

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Training on infinitely large data sets with Pipe Mode

- By default, Amazon SageMaker copies the data set to all training instances.
  - This is the best option when the data set fits in memory.
- For larger data sets, Pipe Mode lets you stream data from Amazon S3.
  - Training starts faster.
  - You can train on infinitely large data sets.

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(
        entry_point='my_script.py',
        role=role,
        train_instance_count=1, train_instance_type='ml.p3.2xlarge',
        framework_version='1.15', py_version='py3', script_mode=True,
        input_mode='Pipe'
)

tf_estimator.fit('s3://bucket/path/to/training/data')
```

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Streaming *TFRecord* files with Pipe Mode

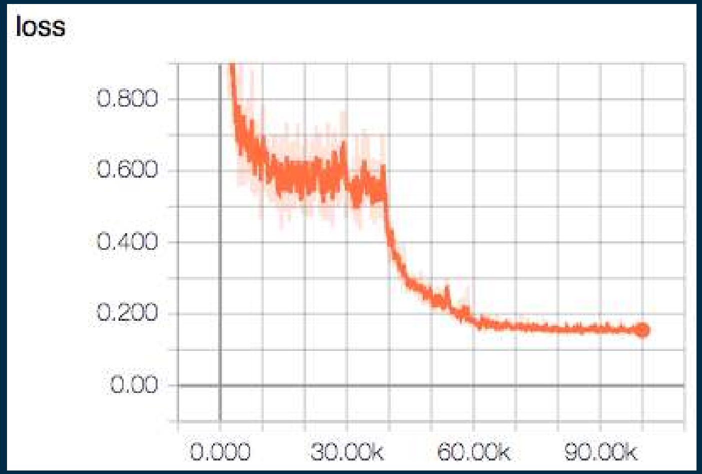
```
from sagemaker_tensorflow import PipeModeDataset

features = { 'data': tf.FixedLenFeature([], tf.string),
             'labels': tf.FixedLenFeature([], tf.int64)}

def parse(record):
        parsed = tf.parse_single_example(record, features)
        return ({ 'data': tf.decode_raw(parsed['data'], tf.float64) }, parsed['labels’])

def train_input_fn(training_dir, hyperparameters):
        ds = PipeModeDataset(channel='training', record_format='TFRecord’)
        ds = ds.repeat(20)
        ds = ds.prefetch(10)
        ds = ds.map(parse, num_parallel_calls=10)
        ds = ds.batch(64)
        return ds
```

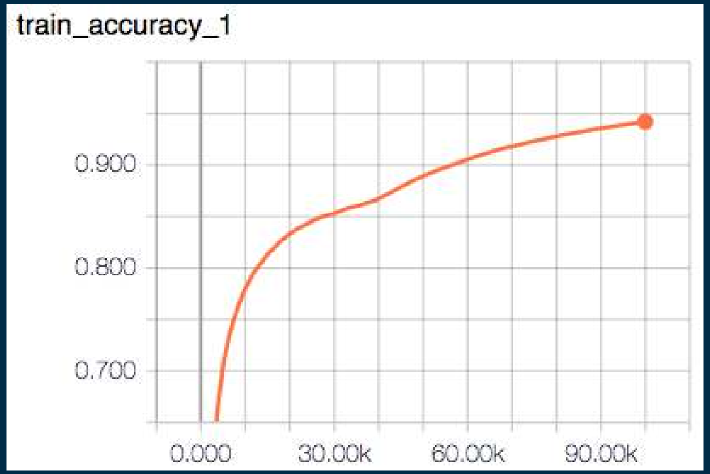
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

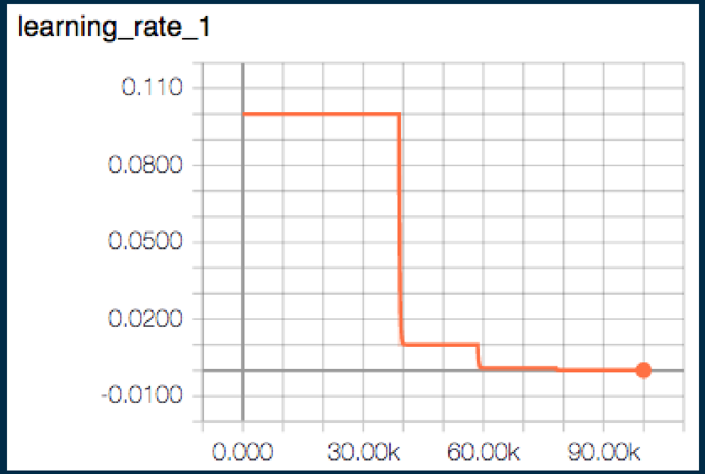
Visualizing training with TensorBoard

- TensorBoard is a suite of visualization tools: graph, metrics, etc.
- When enabled, it will run on the notebook instance.
- You can access it at https://NOTEBOOK_INSTANCE/proxy/6006/

```
tf_estimator.fit(inputs, run_tensorboard_locally=True)
```







© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deploying a TensorFlow model to an HTTPS endpoint

Model trained on-demand

```
from sagemaker.tensorflow import TensorFlow

tf_estimator = TensorFlow(entry_point='tf-train.py', ...)
tf_estimator.fit(inputs)

predictor = tf_estimator.deploy(initial_instance_count=1, instance_type='ml.c4.xlarge')
```

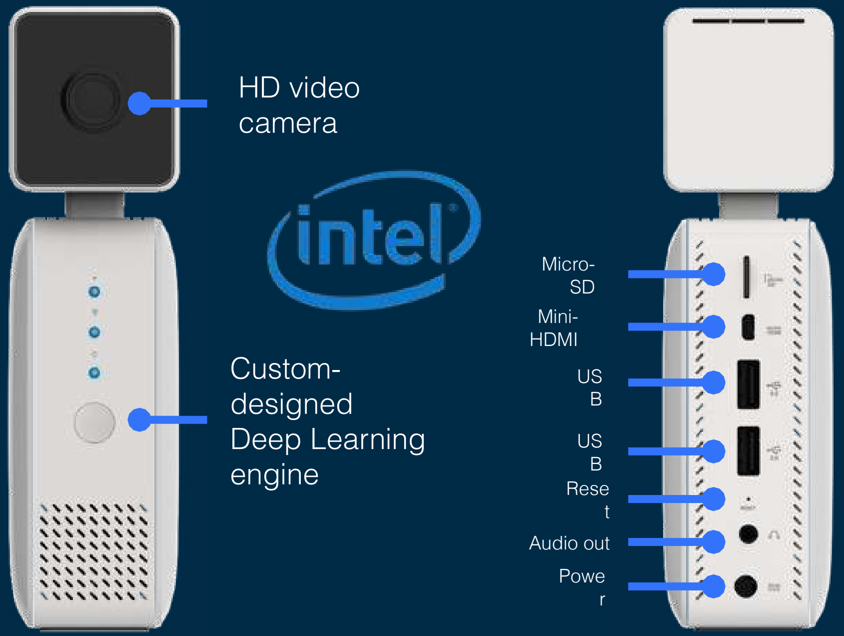
Pretrained Model

```
from sagemaker.tensorflow import TensorFlowModel

tf_model = TensorFlowModel(model_data='s3://mybucket/model.tar.gz’, …,
                                 entry_point='entry.py', name='model_name’)

predictor = tf_model.deploy(initial_instance_count=1, instance_type='ml.c4.xlarge')
```

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Using TensorFlow with AWS DeepLens

- AWS DeepLens can run TensorFlow models.
  - Inception
  - MobileNet
  - NasNet
  - ResNet
  - VGG
- Train or fine-tune your model on Amazon SageMaker.
- Deploy to DeepLens through AWS Greengrass.



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Getting started

http://aws.amazon.com/free

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

https://github.com/aws/sagemaker-python-sdk
https://sagemaker.readthedocs.io/en/stable/using_tf.html
https://github.com/awslabs/amazon-sagemaker-examples

https://gitlab.com/juliensimon/aim410 End to end demo with Keras & SageMaker

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.