



Amazon SageMaker

Machine learning en AWS

Maria Gaska, Mayo 2020

El Stack de servicios de AI/ML en AWS

AI SERVICES

VISION



Amazon
Rekognition

SPEECH



Amazon
Polly
+Medical

TEXT



Amazon
Comprehend
+Medical

SEARCH



Amazon
Translate

SEARCH



Amazon
Kendra

CHATBOTS



Amazon
Lex

PERSONALIZATION



Amazon
Personalize

FORECASTING



Amazon
Forecast

FRAUD



Amazon
Fraud Detector

DEVELOPMENT



Amazon
CodeGuru

CONTACT CENTERS



Contact Lens
For Amazon Connect

ML SERVICES



Amazon SageMaker

Ground
Truth

AWS
Marketplace
for ML



Neo

Augmented
AI

ML FRAMEWORKS & INFRASTRUCTURE



PYTORCH



Deep Learning
AMIs & Containers

GPUs &
CPUs

Elastic
Inference

Inferentia

FPGA

Decenas de miles de empresas usan Amazon SageMaker



Amazon SageMaker is the best place to run TensorFlow



- Containers de Entrenamiento y Deploy
- Escala casi linealmente en 100s de GPUs
- 3x mayor throughput de red con las instancias P3

65% TensorFlow Distribución Libre

90% Tensorflow optimizado por AWS

Scaling efficiency with 256 GPUs

Alguna funcionalidad de SageMaker

- SageMaker operator para Kubernetes
- Autopilot
- Experiment tracker
- SageMaker Groundtruth
- Amazon SageMaker Studio (IDE + Debugger)
- SageMaker Neo
- Amazon Augmented AI

Amazon SageMaker Studio

The screenshot shows the Amazon SageMaker Studio interface. On the left, there's a sidebar with various icons for file operations like Open, Save, and Run. The main area has a Jupyter notebook titled "xgboost_customer_churn.ipynb". The notebook contains the following code and text:

```
[ ]: model_data = pd.get_dummies(churn)
model_data = pd.concat([model_data['Churn?_True.'], model_data.drop(['Churn?_True.'])])
< ... >
...
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), 3)
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,
< ... >
```

On the right side, there are two charts and a table:

- Trial Component Chart**: A line chart showing "trainloss_last" over "period". It displays four colored lines (blue, red, green, orange) fluctuating between 0.0 and 0.4.
- Trial Component List**: A table listing "TRIAL COMPONENTS" with 10 rows selected. The columns are Status, Experiment, Type, Trial, and Trial Order. All entries are "Completed".

Status	Experiment	Type	Trial	Trial Order
Completed	customer-churn-predic...	Training job	Trial-3	Training job
Completed	customer-churn-predic...	Training job	Trial-2	Training job
Completed	customer-churn-predic...	Training job	Trial-1	Training job
Completed	customer-churn-predic...	Training job	Trial-0	Training job
Completed	customer-churn-predic...	Training job	Trial-3	Training job
Completed	customer-churn-predic...	Training job	Trial-2	Training job
Completed	customer-churn-predic...	Training job	Trial-1	Training job
Completed	customer-churn-predic...	Training job	Trial-0	Training job
Completed	customer-churn-predic...	Training job	Trial-3	Training job

Amazon SageMaker Autopilot (solo para datos tabulares)

The screenshot shows the AWS SageMaker Autopilot interface within a Jupyter Notebook environment. The top navigation bar includes File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. The left sidebar has icons for AWS Lambda, S3, Glue, Redshift, and SageMaker. The main content area displays an experiment named "EXPERIMENT: MY-SAGEMAKER-AUTOPILOT". It shows a list of "TRIALS" with the following details:

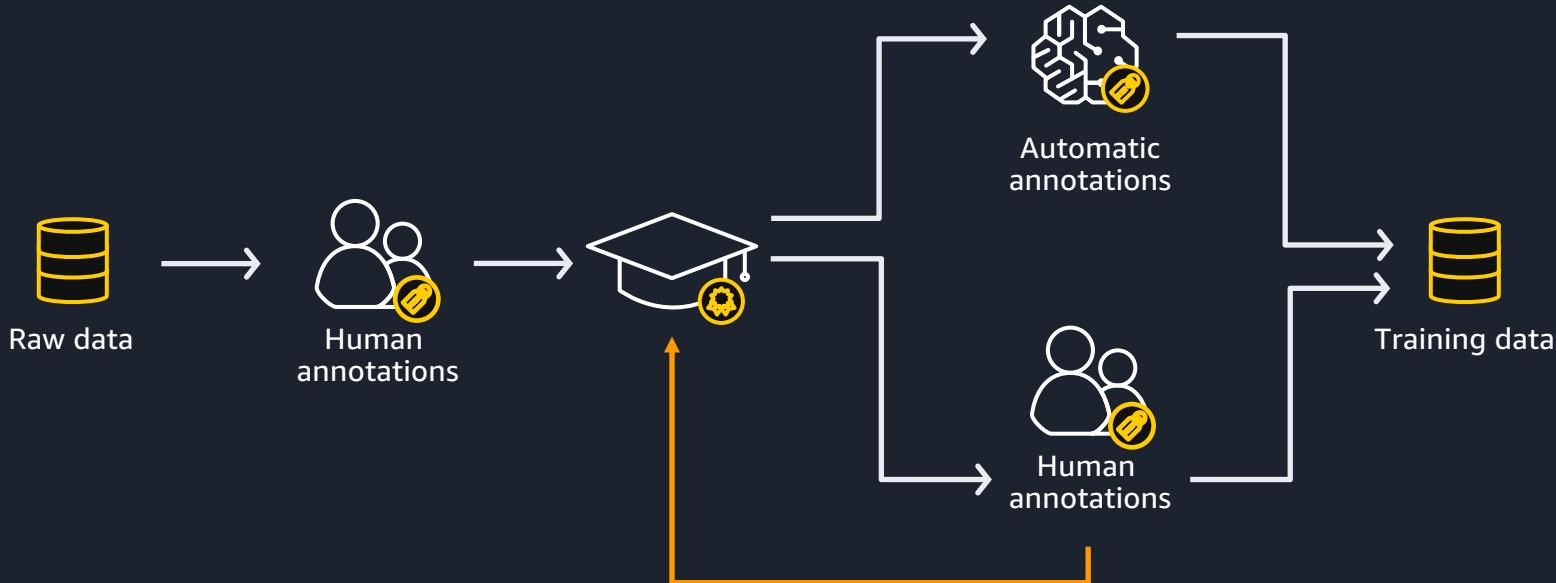
Trial name	Status	Start time	End time	Objective
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9206119775772095
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9202479720115662
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9200050234794617
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9195190072059631
my-sagemaker-tuning-job-...	Completed	9 hours ago		0.9191550016403198
my-sagemaker-tuning-job-...	Completed	7 hours ago		0.9190340042114258
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9189119935035706
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825
my-sagemaker-tuning-job-...	Completed	8 hours ago		0.9186699986457825

Buttons for "Open candidate generation notebook" and "Open data exploration notebook" are also present.

Hoy lo más valioso es la generación de datos

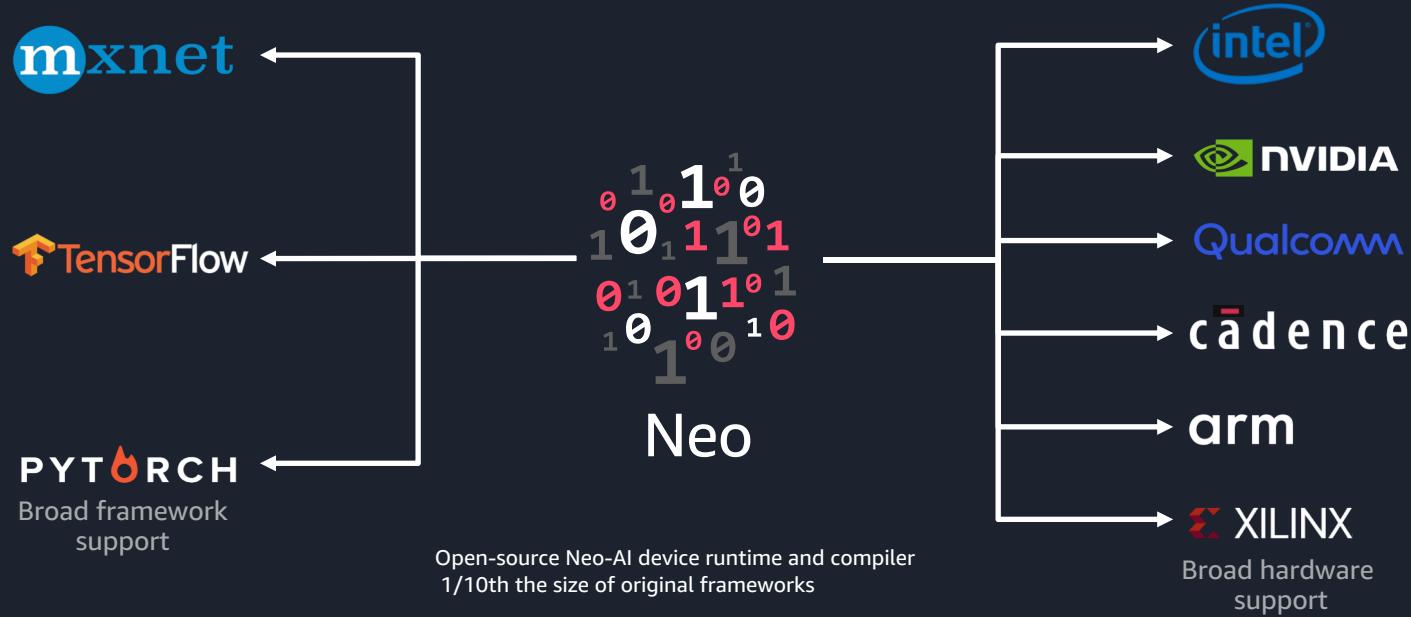


Cómo funciona SageMaker Ground Truth

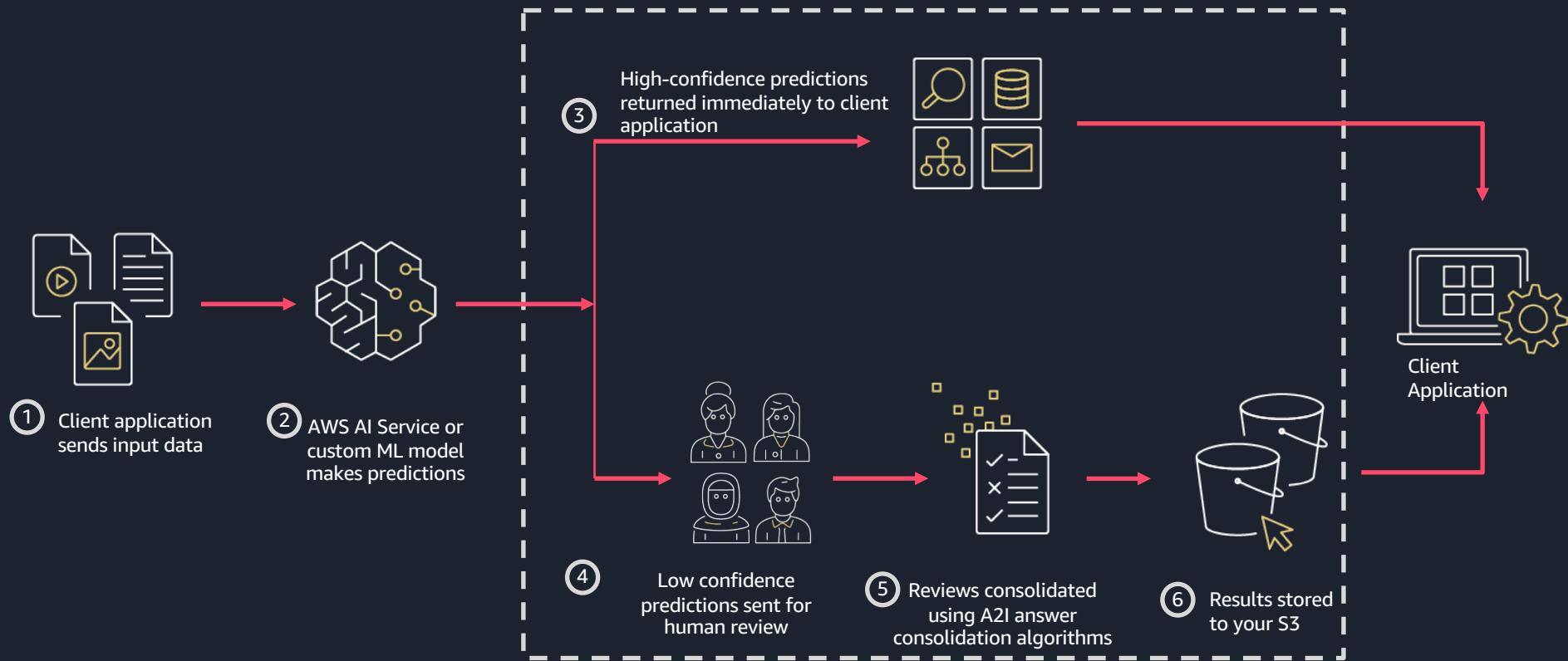


Amazon SageMaker Neo

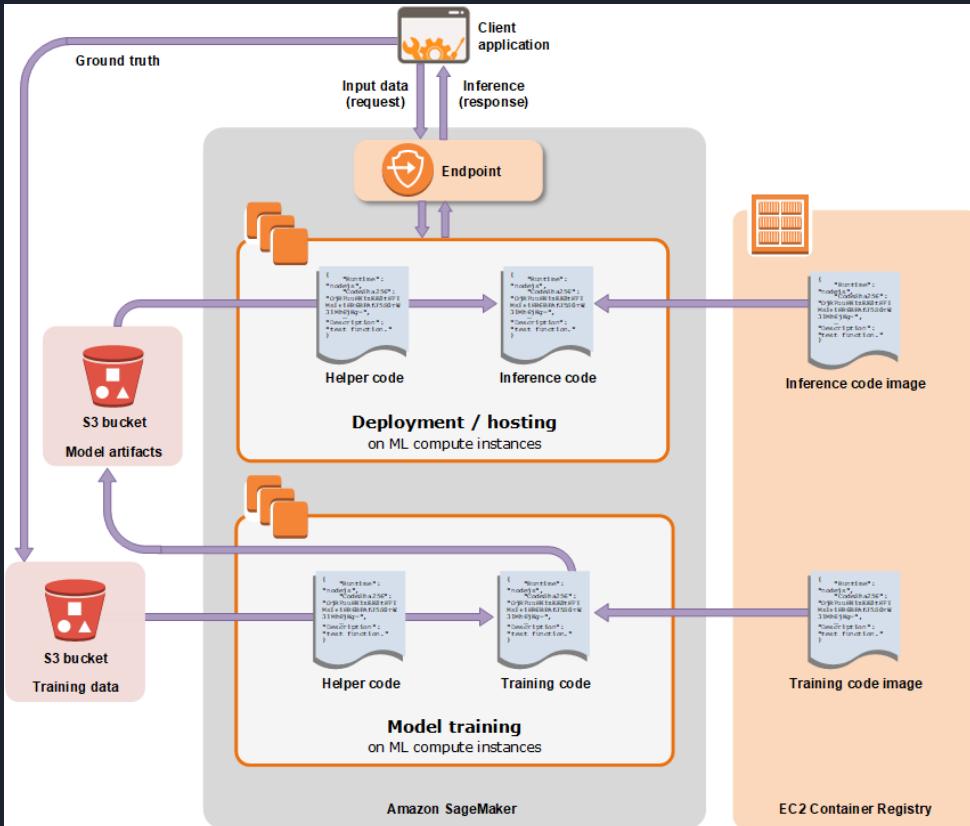
Train once and run anywhere with 2x performance



Cómo funciona Amazon Augmented AI



Arquitectura de SageMaker



Algoritmos Built-in en SageMaker

Classification

- Linear Learner
- XGBoost
- KNN

Working with Text

- BlazingText
- Supervised
- Unsupervised

Sequence Translation

- Seq2Seq

Computer Vision

- Image Classification
- Object Detection
- Semantic Segmentation

Recommendation

- Factorization Machines

Anomaly Detection

- Random Cut Forests
- IP Insights

Regression

- Linear Learner
- XGBoost
- KNN

Topic Modeling

- LDA
- NTM

Forecasting

- DeepAR

Clustering

- KMeans

Feature Reduction

- PCA
- Object2Vec

Tensorflow workshop

<https://github.com/githubmg/sagemaker-tensorflow2.git>