

Del notebook al pipeline sin dolor con AWS

Sebastián Sandoval García

Objetivo

Generar un flujo automatizado que permita predecir si un pasajero sobrevivió al hundimiento del Titanic.

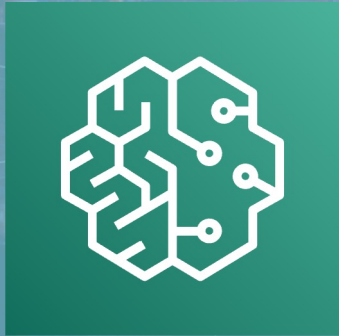
Tomaremos como punto de partida un notebook tradicional de entrenamiento y lo convertiremos en un flujo de Amazon SageMaker Pipelines.

Objetivo



Herramientas

Amazon SageMaker



Amazon S3



Amazon S3 (Simple Storage Service)

- Almacenamiento de objetos accesibles desde la web (a través de API o HTTPS).
- Integración sencilla con otros servicios de AWS.
- Capacidades de seguridad, cumplimiento y auditoría
- Casos de uso: **lagos de datos**, sitios web, aplicaciones móviles, respaldo y restauración, archivo, aplicaciones empresariales, dispositivos IoT y **analítica de datos**.



Amazon SageMaker Studio

SAGEMAKER STUDIO

Prepare data

Store features

Detect bias

Build with notebooks

Train models

Tune parameters

Deploy in production

Explain predictions

Manage and monitor

Amazon SageMaker Studio es un entorno de desarrollo integrado (IDE) basado en la web para preparar datos, crear, entrenar, implementar y monitorear sus modelos de aprendizaje automático.

Amazon SageMaker Studio

Computing Anomaly Scores

Now, let's compute and plot the anomaly scores from the entire taxi dataset.

```
results = rcf_inference.predict(taxi_data_numpy)
scores = [data['score'] for data in results['data']]

# Plot scores to get a feel for the data.
taxi_data['score'] = pd.Series(scores, index=taxi_data.index)
taxi_data.head()

fig, ax1 = plt.subplots()
ax2 = ax1.twinx()

# Try this out - change start and end to zoom in on the
# anomaly found earlier in this notebook
start, end = # (taxi_data
scores = taxi_data['score'].loc[start:end]
taxi_data_subset = taxi_data[start:end]
ax1.plot(taxi_data_subset['score'], color='r', alpha=0.5)
ax2.plot(taxi_data_subset['score'], color='r')
ax1.yrlabel('which score', axis='y')
ax1.set_ylabel('the anomaly', color='r')
ax2.set_ylabel('anomaly score', color='r')
ax1.tick_params(axis='y', color='r')
ax2.tick_params(axis='y', color='r')
ax1.set_ylabel('score')
ax2.set_ylabel('scores', 1.4*ax1.get_yticks())
fig.set_figwidth(10)
```

Note that the anomaly score spikes where our eyeball-norm method suggests there is an anomalous data point as well as in some places where our eyeballs are not as accurate. Below we print and plot any data points with scores greater than 3 standard deviations (approx 99.3% percentile from the mean score).

```
score_mean = taxi_data['score'].mean()
score_std = taxi_data['score'].std()
score_cutoff = score_mean + 3*score_std

anomalous = taxi_data_subset[taxi_data_subset['score'] > score_cutoff]
anomalous
```

The following is a list of known anomalous events which occurred in New York City within this timeframe:

Experiment	Trial	Trial Component	Type
Frutos111	Apple111	DEMO-mineva-2019-11-14-04-26-00-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-14-07-18-55-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-14-17-30-15-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-15-16-25-53-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-15-22-10-02-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-19-22-12-34-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-20-17-59-29-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-2019-11-21-05-21-26-aws-training-job	amazon SageMaker
Frutos111	Apple111	DEMO-mineva-Rap-2019-11-21-18-23-16-aws-training-job	amazon SageMaker

execution-1656442175154

Status: Started time: 6/28/2022, 2:49 PM Elapsed time: 15d5h16m

Graph Parameters Settings

```
graph TD
    cv-week4PreProcess --> BirdClassificationSpotTrain
    cv-week4PreProcess --> BirdClassificationOnDemandTrain
    BirdClassificationSpotTrain --> BirdClassificationSpotEval
    BirdClassificationOnDemandTrain --> BirdClassificationOnDemandEval
    BirdClassificationSpotEval --> BirdClassificationSpotCondition
    BirdClassificationOnDemandEval --> BirdClassificationOnDemandCondition
    BirdClassificationSpotCondition -- true --> RegisterSpotModel
    BirdClassificationOnDemandCondition -- true --> RegisterOnDemandModel
```

Amazon SageMaker Pipelines

The image features a central graphic with the text 'Amazon SageMaker Pipelines' inside a purple circular frame. To the right of this frame, a vertical line of four purple circles is connected by a dashed purple line that curves at the top and bottom. Each circle is connected to a corresponding numbered feature box on the right. The background is a teal-tinted cityscape with a network of white lines and dots overlaid.

1

Automatización de soluciones end to end

2

Trazabilidad

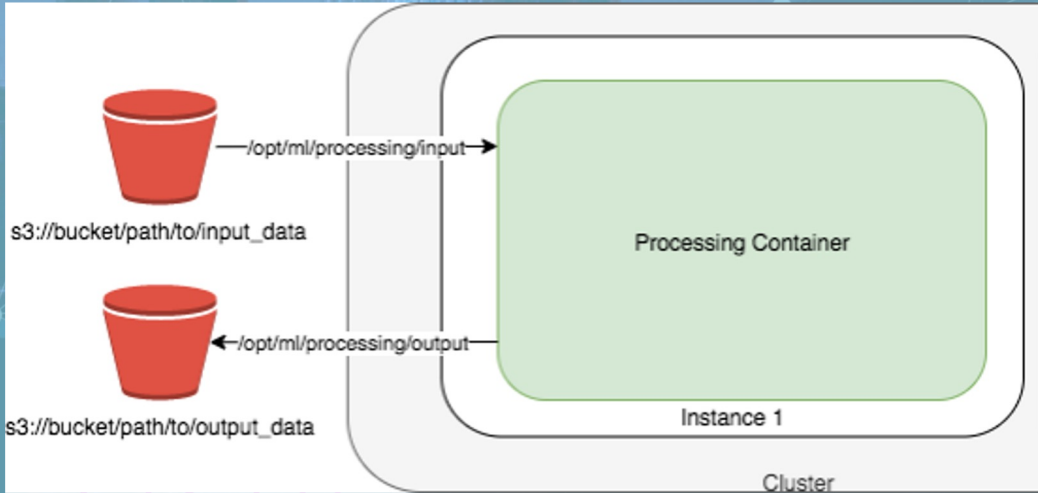
3

CI/CD

4

Reutilizables y escalables

Trabajos de Amazon SageMaker



1. Copia entradas desde s3 al contenedor (datos y código)
2. Ejecuta el código
3. Copia el resultado del contenedor a s3

Taller



<https://bit.ly/3ivvwbe>

Código hash:

5578-1f3eba79f4-f3



¡Gracias!

Sebastián Sandoval García

