



Herramientas y Aplicaciones de la IA.

IA en Amazon Web Services

Jaume Jordán

Instalación componentes API AWS

■ Instalación y Configuración de AWS CLI (Command Line Interface):

- Instalación de AWS CLI: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>
- Configuración AWS CLI para lanzar AWS en local por consola y en scripts Python:
 - Obtención de la clave de acceso (token).
 - AWS habitual: Configurar IAM con un usuario (recomendado no root) y crear clave de acceso (token)
 - AWS learner lab: Clicar en *AWS details* y *Cloud Access/AWS CLI*.
 - Configuración de credenciales: `aws configure`
 - AWS habitual: Comando `aws configure` e introducir los tokens, región y formato salida
 - AWS learner lab: Copiar info en `~/.aws/credentials` (Linux) o `C:\Users\<usuario>\.aws\credentials` (Windows). Especificar también `.aws/config` la región adecuada y formato salida.

■ Instalación de Boto3 en Python:

- Comando: `pip install boto3`
- Boto3 es el SDK de Amazon Web Services para Python, que permite a Python hablar con AWS.

Primer script Python utilizando API AWS

- Script Python básico que utiliza la API de AWS para realizar una acción simple
- Pasos a seguir:
 - Importación del módulo Boto3.
 - Creación de un recurso S3 que proporciona una interfaz orientada a objetos para Amazon S3.
 - Iteración y listado de todos los buckets S3 asociados con la cuenta.

```
import boto3
s3 = boto3.resource('s3')
for bucket in s3.buckets.all():
    print(bucket.name)
```

El servicio Amazon Rekognition

- Servicio de análisis de imágenes y vídeos basado en aprendizaje automático.
- Proporciona etiquetado automático, detección de objetos, y reconocimiento facial en medios visuales:
 - Objetos
 - Personas
 - Texto
 - Actividades
- <https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html>

Amazon Rekognition: Trabajando con una imagen en local

- Se puede cargar una imagen local en Amazon Rekognition para su análisis.
- Lectura de la imagen local en bytes.
- Utilizar los métodos de Rekognition pasando esa imagen como parámetro.
- Apertura de una imagen y lectura en formato de bytes.
- Llamada a *detect_labels* para obtener etiquetas de la imagen.

```
import boto3
client = boto3.client('rekognition')
with open("path_to_your_image.jpg", "rb") as image:
    response = client.detect_labels(Image={'Bytes': image.read()})
print(response['Labels'])
```

Amazon Rekognition: Trabajando con una imagen en bucket s3

- Se puede cargar una imagen desde un bucket s3 en Amazon Rekognition para su análisis.
- Pasamos por argumento el bucket de origen y la ruta a la imagen
- Llamada a *detect_labels* para obtener etiquetas de la imagen.

```
import boto3
```

```
bucket_name = "bucketjaume"  
image_path = "img/van.jpg"  
client = boto3.client('rekognition')  
response = client.detect_labels(Image={'S3Object': {'Bucket':  
bucket_name, 'Name': image_path}})  
for label in response['Labels']:  
    print(f"{label['Name']} - Confidence: {label['Confidence']}")
```

Amazon Rekognition: Detección de etiquetas

- Ejemplo de detección de etiquetas:

- `response = client.detect_labels(Image={'Bytes': image.read()})`

- Descripción de cada etiqueta encontrada y su nivel de confianza.

- `for label in response['Labels']:`

- `print(f"{label['Name']} - Confidence: {label['Confidence']}")`

Amazon Rekognition: Comparación de caras

- Comparar dos imágenes que contienen caras:

- `response = client.compare_faces(
 SourceImage={'S3Object':{'Bucket':'bucket_name','Name':'source.jpg'}},
 TargetImage={'S3Object':{'Bucket':'bucket_name','Name':'target.jpg'}})`

- Análisis de las coincidencias de caras y el porcentaje de similitud.

- `for faceMatch in response['FaceMatches']:
 print(f"Similarity: {faceMatch['Similarity']}%")`

Amazon Rekognition: buscando caras en una colección

■ Creación de una colección de caras:

- `client.create_collection(CollectionId='mycollection')`
- `response = client.index_faces(CollectionId=collection_id,
Image={'S3Object': {'Bucket': bucket, 'Name': filename}},
ExternalImageId=facename,
MaxFaces=1)`

■ Búsqueda en una colección de caras:

- `response = client.search_faces_by_image(CollectionId='mycollection',
Image={'S3Object': {'Bucket': 'bucket_name', 'Name': 'photo.jpg'}},
FaceMatchThreshold=95, MaxFaces=10)`

■ Manejar el listado de caras encontradas y análisis de coincidencias.

Amazon Rekognition: reconociendo famosos

- Cargar una imagen y tratar de reconocer si es de un famoso indexado en AWS.

- ```
client = boto3.client('rekognition')
with open(photo, 'rb') as image:
 response = client.recognize_celebrities(Image={'Bytes':
image.read()})
```

- ```
for celebrity in response['CelebrityFaces']:
```

- ...

Amazon Rekognition: detección de texto en imagen

■ Código para Detectar Texto:

```
□ response =  
  client.detect_text(Image={'S3Object':{'Bucket':'your_bucket','Name':'image_with_text.jpg'}})
```

■ Interpretación del texto detectado y su nivel de confianza.

```
□ for textDetail in response['TextDetections']:  
    print(f"Detected text: {textDetail['DetectedText']} - Confidence:  
    {textDetail['Confidence']}")
```

Amazon Rekognition: Trabajando con vídeos

■ Proceso de Análisis de Vídeo:

```
□ response =  
  client.start_label_detection(Video={'S3Object':{'Bucket':'bucket_name',  
  'Name':'video.mp4'}})  
  jobId = response['JobId']
```

■ Operación asíncrona

```
□ result = client.get_label_detection(JobId=jobId)
```



Amazon Rekognition: Práctica obligatoria

- Se debe realizar solo una de las 3 tareas propuestas a continuación.
- Subir a la Tarea *Práctica AWS Rekognition* de PoliformaT:
 - El código fuente desarrollado que llama a los servicios correspondientes de AWS.
 - Los archivos de prueba utilizados en dicho código fuente (imágenes, diccionarios de palabras...) y cualquier otro archivo para poder ejecutar el código fuente.
 - Una explicación de máximo 2 párrafos especificando qué tarea se ha realizado y cómo se ha implementado la tarea, así como los resultados principales obtenidos.



Amazon Rekognition: Práctica propuesta.

Tarea 1

- Realiza un script que permita detectar imágenes con palabras ofensivas.
- Para ello, sube algunas imágenes a un bucket (algunas con mensajes ofensivos y otras no).
- El script deberá extraer las palabras de cada imagen y compararlas con una pequeña estructura de palabras ofensivas (por ejemplo, una lista), mostrando por pantalla aquellas que tengan mensajes ofensivos.

Amazon Rekognition: Práctica propuesta.

Tarea 2

- Realiza un script que permita al usuario introducir una etiqueta (p.e. ANIMAL) y a partir de esta etiqueta, mostrará el nombre de las imágenes de un bucket que contengan esta etiqueta con una seguridad mayor al 70%.



Amazon Rekognition: Práctica propuesta.

Tarea 3

- Pide a varios de tus compañeros que te pasen una foto suya (o recolecta imágenes de internet).
- Construye un script que permita recibir una foto y que te muestre el nombre de la persona que aparece en la foto.
- Puedes hacerlo comparando por parejas o bien creando una colección.



Líneas de Trabajo de HAIA

- Servicios de IA en la nube
 - Llevar a cabo una revisión de casos de uso que utilicen herramientas en la nube. Descripción y aplicación.
 - Desarrollar un prototipo de aplicación que requiera servicios en la nube.

Resumen servicios IA AWS

- Amazon SageMaker: Servicio integral que facilita a los científicos y desarrolladores de datos crear, entrenar y desplegar modelos de machine learning a escala.
- Amazon Rekognition: Servicio que permite añadir análisis de imagen y video a tus aplicaciones para identificar objetos, personas, texto, escenas y actividades.
- Amazon Comprehend: Servicio de procesamiento de lenguaje natural que utiliza machine learning para descubrir información y relaciones en texto.
- Amazon Forecast: Servicio de predicción basado en machine learning que genera predicciones de tiempo series con exactitud para planificación de recursos y más.
- Amazon Translate: Servicio de traducción automática que utiliza técnicas avanzadas de machine learning para traducir texto entre idiomas.
- Amazon Lex: Servicio para construir interfaces de conversación en cualquier aplicación usando voz y texto, impulsado por la misma tecnología que Alexa.
- Amazon Polly: Servicio que convierte el texto en habla realista, permitiendo añadir voces habladas a aplicaciones.
- AWS DeepLens: Cámara de video habilitada para deep learning que puede correr modelos de aprendizaje profundo localmente y en tiempo real.
- Amazon Personalize: Servicio de machine learning que facilita la creación de recomendaciones personalizadas para tus clientes.
- Amazon Kendra: Motor de búsqueda empresarial impulsado por machine learning que proporciona a los usuarios respuestas más precisas a preguntas en lenguaje natural.
- Amazon Textract: Servicio que extrae automáticamente texto y datos de documentos escaneados, incluyendo tablas y formularios.
- AWS DeepRacer: Plataforma autónoma de coche de carreras modelo a escala, diseñada para enseñar los fundamentos de la conducción autónoma y la competencia en la liga de carreras.
- Amazon Fraud Detector: Servicio que identifica actividades fraudulentas en línea en tiempo real, como pagos fraudulentos y cuentas falsas.

Resumen servicios IA AWS (cont.)

- Amazon Lookout for Vision: Detecta defectos y anomalías en imágenes físicas utilizando machine learning para mejorar la calidad del producto.
- Amazon Lookout for Equipment: Analiza los datos de los sensores de equipos para detectar anomalías inusuales en el funcionamiento de las máquinas, lo que ayuda en el mantenimiento predictivo.
- Amazon Lookout for Metrics: Identifica y diagnostica anomalías en tus métricas empresariales automáticamente, facilitando la identificación rápida de problemas inesperados.
- Amazon HealthLake: Permite almacenar, transformar y analizar datos de salud en grandes escalas, facilitando la gestión y análisis de información médica con formatos compatibles como FHIR (Fast Healthcare Interoperability Resources).
- AWS Panorama: Permite a las empresas añadir capacidades de aprendizaje automático a sus cámaras de CCTV existentes para analizar imágenes de video y tomar decisiones automatizadas localmente.
- Amazon Augmented AI (A2I): Facilita la implementación de revisiones humanas para las predicciones de machine learning, permitiendo a los desarrolladores construir flujos de trabajo para la validación humana de las decisiones de IA.
- Amazon CodeGuru: Herramienta que revisa el código automáticamente y ofrece recomendaciones de optimización para mejorar la calidad del software y reducir costes.
- AWS DeepComposer: Servicio que proporciona una manera creativa y divertida de aprender sobre machine learning, permitiendo a los usuarios crear música con modelos generativos.
- Amazon Elastic Inference: Permite adjuntar capacidades de computación de inferencia acelerada a cualquier instancia de Amazon EC2 o Amazon SageMaker, reduciendo así los costos de inferencia de machine learning.
- AWS Glue Databrew: Herramienta de preparación de datos visuales que permite a los científicos de datos limpiar y normalizar datos sin escribir código.
- Amazon Monitron: Sistema de monitorización de condiciones de la máquina que usa sensores y machine learning para detectar anomalías en equipos industriales, ayudando a prevenir fallas no planificadas.
- Amazon Neptune ML: Servicio que facilita la creación y ejecución de predicciones de machine learning utilizando gráficos de bases de datos, ideal para conexiones complejas entre datos.

Extensiones para TFM de IA en AWS

- Predicción y configuración de portfolio de inversión en AWS
- Sistemas de recomendación en AWS
- Chatbots en AWS
- Otras propuestas...
- Contactar con Juanmi Alberola (jalberola@dsic.upv.es) y Jaume Jordán (jjordan@dsic.upv.es)