

## **Diseño Arquitectónico para Industrialización**

El pipeline propuesto permite automatizar el ciclo completo del modelo, desde la lectura de datos telemáticos en tiempo real hasta la generación del “Driver Score” y su visualización teniendo la posibilidad de escalabilidad, gobernanza y reproducibilidad.

### **Pipeline de MLOps en AWS:**

Haciendo uso de Step Functions en AWS se gestiona el entrenamiento y reentrenamiento que contiene lo siguiente:

1. Extracción y limpieza de datos desde S3 usando AWS Glue.
2. Generación de características (Feature Engineering).
3. El modelo se entrena con Scikit-learn dentro de SageMaker Training Jobs.
4. Validación automática teniendo en cuenta las métricas F1, ROC-AUC u otros según sea el caso
5. Registro del modelo y metadatos en el SageMaker Model Registry.

Para la automatización es importante tener en cuenta el tiempo, de manera que la tarea sea programada y se ejecute de forma semanal, diaria u otras. En caso de que se detecte un cambio en los datos la tarea debe ser hacer un cambio en el modelo.

### **Artefactos:**

Todos los modelos, datos procesados y resultados se almacenan versionados en **S3** para trazabilidad completa.

### **Despliegue e Inferencia en Tiempo Real**

El modelo es dockerizado utilizando una API REST y se publica en Amazon ECR. Luego se despliega como endpoint gestionado en Amazon SageMaker, con capacidad de autoescalado.

Los datos llegan en streaming a Amazon MSK (Kafka), donde un consumidor en Apache Flink o Kafka Streams procesa la información en tiempo real para generar las características de cada viaje (avg\_speed, speed\_volatility, hard\_event\_count, etc.).

Estas características se envían al endpoint de SageMaker para obtener el Score, que luego se publica en un nuevo tópico Kafka y se almacena en Apache Druid para visualización.

## **Monitoreo y Reentrenamiento**

La supervisión del modelo se realiza de dos maneras:

- AWS CloudWatch para latencia, errores u otros. Y SageMaker Model Monitor para métricas de calidad como drift en features, cambio en distribución de scores y/o degradación del modelo.
- Cuando se detectan cambios significativos, se lanza automáticamente el reentrenamiento que tiene un despliegue seguro y controlado, dejando el modelo actual en producción, utilizando un porcentaje de la data real en el nuevo modelo y en caso de superar las métricas del proceso actual, se lleva el nuevo a producción.

## **Analítica y Visualización (Apache Druid)**

Una vez la información es modelada y se tiene resultados se hará uso de Apache Druid para crear dashboards de consumo de data en tiempo real. Estos dashboards (usando Superset o Grafana) podrían mostrar:

La información por flota, alertas sobre viajes de alto riesgo, la distribución del Driver Score por vehículo, cliente, los resultados históricos del modelo, entre otros,

