

Challenge AI: Detección de Fraudes en Transacciones Bancarias

Nombre: Rodrigo Urquizo

1. Descripción del flujo de trabajo

1. Generación de descripciones usando Bedrock

Se empleó el modelo Llama en Bedrock para la generación de descripciones por cada persona, evaluando el riesgo crediticio. El prompt se creó a través de un template donde se toma las variables de edad, trabajo, cuentas, etc, y se solicita al LLM que evalúe el riesgo crediticio indicando si se trata de un “bad risk” o “good risk”.

2. Etiquetado para obtener el target

Como en las descripciones generadas por Llama se encuentra la evaluación del riesgo crediticio, se procedió a extraer y transformar estas etiquetas en valores binarios para el target. Se implementó un script que mapea “good risk” a 0 y “bad risk” a 1, asegurando consistencia en el formato. Estos valores se integraron al conjunto de datos original, en una nueva columna “target”.

3. Preprocesamiento para SageMaker

Se requiere preprocesar la data para poder entrenar al modelo, en este caso, se realizó la codificación de variables categóricas, como por ejemplo Checking account, la cual tiene 3 clases: little, moderate y rich, se asignó una codificación ordinal (little = 0, moderate = 1 y rich = 2) ya que se puede inferir que el tipo de cuenta “rich” en términos financieros es superior a una “moderate”, la cual es superior a una “little”. Por último, se modificó el dataframe de tal forma que la columna target sea la primera, y se quitó el encabezado ya que este es el formato que requiere SageMaker para el entrenamiento.

4. Entrenamiento del modelo en Amazon SageMaker

Se configuró un training job con XGBoost en una instancia ml.m4.xlarge y la versión 1.5-1 del framework, seleccionada tras resolver errores con otras versiones. Los datos preprocesados fueron divididos en un 80% para entrenamiento y un 20% para prueba. El dataset de entrenamiento se subió a un bucket S3. Se definieron hiperparámetros para optimizar la clasificación binaria de riesgo crediticio, logrando un job completado exitosamente.

5. Validación del modelo

Con el dataset de entrenamiento que corresponde al 20%, se realizó la prueba del modelo exportado desde S3, para validar su performance antes de pasar a la etapa de deploy. Se emplearon métricas como accuracy, precisión, recall y F1-Score, AUC-ROC y Matriz de Confusión. Se obtuvieron valores aceptables (accuracy: 0.72, precision: 0.77, recall: 0.81), los cuales se pueden mejorar, pero al ser un prototipado se decidió a pasar a deploy.

6. Despliegue del modelo

El despliegue del modelo se realizó mediante la creación de un endpoint en SageMaker configurando un XGBoostModel a partir del modelo almacenado en S3. El endpoint se activó exitosamente, para validar su funcionalidad se invocó al endpoint, obteniendo predicciones en tiempo real que coincidieron con las métricas previamente calculadas, asegurando que el modelo esté listo para integrarse en aplicaciones.

2. Decisiones técnicas

-Elección del LLM: Se empleó el modelo Llama3-2-90b al tener 90 billones de parámetros, lo que le otorga una capacidad de modelado mucho mayor en comparación con modelos como Llama2-70b. Este modelo demostró respuestas más coherentes al evaluar el riesgo crediticio, evitando ambigüedades que se observaron en otros modelos.

-Preprocesamiento de datos: El preprocesamiento incluyó codificación ordinal para variables como Housing (0: free, 1: rent, 2: own) y Saving accounts (0: little, 1: moderate, 2: rich, 3: quite rich), para representar la predominancia de ciertas clases sobre otras. En el caso de la variable Purpose, se empleó OneHotEncoder ya que las clases como car, radio/TV y education no tienen una relación jerárquica y por ello no se pueden codificar de manera ordinal.

Con esto se logra un formato numérico compatible con los modelos en SageMaker. Los nulos en las columnas Saving accounts y Checking account se dejaron como datos NaN, sumiendo que son casos donde las personas no tienen ningún tipo de cuenta corriente o de ahorro, para este caso el modelo los tomará como parte de otra clase.

-Elección del modelo de clasificación: Se eligió el modelo XGBoost ya que su arquitectura, basada en árboles de decisión ensamblados, permite capturar relaciones no lineales entre variables como edad, monto del crédito, duración, etc. Además, en base a mi expertise, este modelo comparado con otros como Random Forest o redes neuronales, suele mostrar un rendimiento superior en velocidad de entrenamiento y precisión, especialmente con un dataset pequeño con pocas columnas como este.

-Hiperparámetros:

- **objective='binary:logistic':** Se trata de un problema de predicción entre dos clases: “good risk” y “bad risk”, por lo que la clasificación es binaria. Es logístico porque generará probabilidades continuas entre 0 y 1, ya que luego se definirá un umbral para poder binarizar las predicciones en 0s y 1s.
- **num_round=100:** Establecido para un número moderado de iteraciones, balanceando precisión y tiempo de entrenamiento, evitando sobreajuste tras pruebas con 150 rondas.
- **max_depth=6:** Limitado a 6 para controlar la complejidad del árbol, reduciendo el riesgo de sobreajuste.
- **eta=0.2:** Configurado en 0.2 para regular el paso de aprendizaje y optimizar la convergencia y evitando pasos excesivamente grandes que afecten la generalización del modelo, lo cual fue validado por mejores métricas.
- **subsample=0.8:** Establecido en 0.8 para usar el 80% de los datos por árbol, para agregar variabilidad y reducir el sobreajuste, con resultados mejores en comparación con el valor predeterminado de 1.0.

-Configuración de la instancia y training job: La configuración del training job en SageMaker se diseñó utilizando una instancia ml.m4.xlarge, al tener errores de cuota con instancias como ml.m5.xlarge. La configuración incluyó un tiempo máximo de ejecución de 1h, que es más que suficiente para este dataset, cambiando el valor de 24h por defecto.

3. Métricas de desempeño del modelo

-Accuracy: El 72% de las predicciones fueron correctas en el conjunto de prueba del 20%. En un dataset desbalanceado, puede estar inflado si el modelo predice mayoritariamente la clase dominante ("bad risk"). Aquí, refleja un desempeño moderado.

-Precision: El 77% de las predicciones "bad risk" fueron correctas. Es útil cuando los falsos positivos (predecir "bad risk" cuando es "good risk") es perjudicial, por ejm: rechazar un crédito válido.

-Recall: El 81% de los "bad risk" reales fueron detectados. Es clave cuando los falsos negativos (no detectar un "bad risk") son críticos, por ejm: aprobar un crédito riesgoso.

-F1-Score: El valor de 79% muestra un balance entre precision y recall. Es una métrica robusta para datasets desbalanceados, indicando que el modelo mantiene un balance en capturar "bad risk" y evitar falsos positivos.

-AUC-ROC (0.74): El valor de 74% del Area bajo la curva ROC indica una capacidad aceptable del modelo para distinguir entre las clases bajo diferentes umbrales.

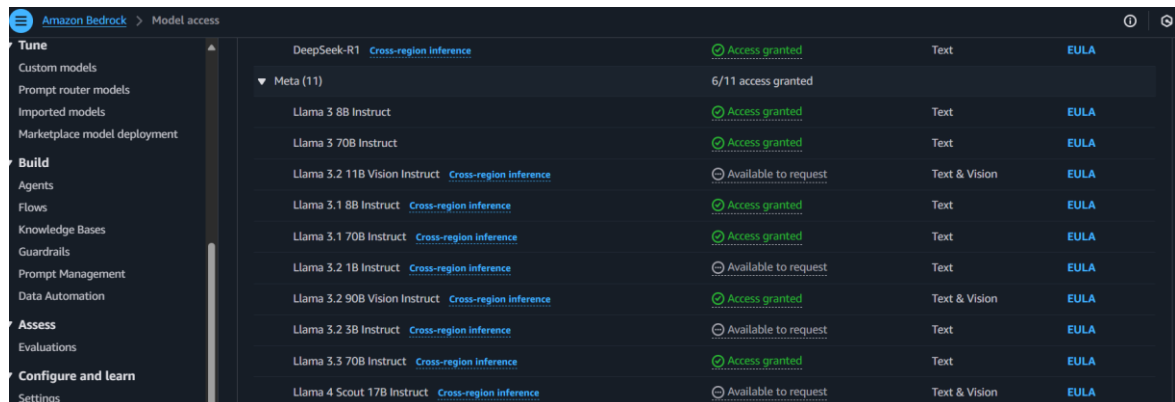
-Matriz de confusión:

- Verdaderos Negativos (TN = 40): 40 instancias de "good risk" correctamente predichas.
- Falsos Positivos (FP = 31): 31 "good risk" erróneamente clasificados como "bad risk".
- Falsos Negativos (FN = 25): 25 "bad risk" no detectados, clasificados como "good risk".
- Verdaderos Positivos (TP = 104): 104 "bad risk" correctamente identificados.

```
Accuracy: 0.72
Precision: 0.77
Recall: 0.81
F1-Score: 0.79
AUC-ROC: 0.74
Confusion Matrix:
[[ 40  31]
 [ 25 104]]
```

4. Evidencias (Screenshots o logs de Bedrock y SageMaker):

Amazon Bedrock – Model access:

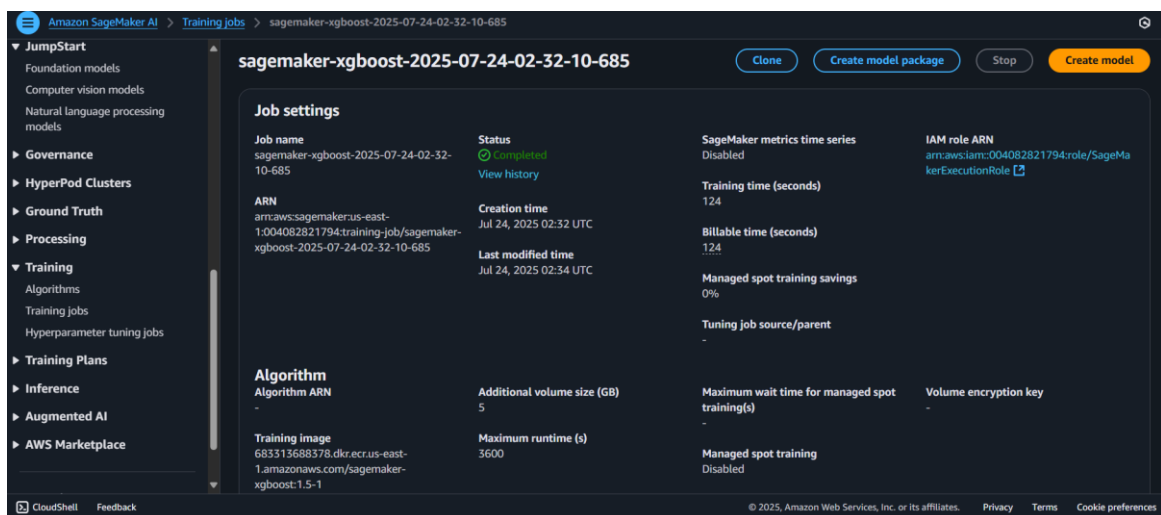


The screenshot shows the Amazon Bedrock console's 'Model access' page. On the left is a navigation menu with sections: Tune (Custom models, Prompt router models, Imported models, Marketplace model deployment), Build (Agents, Flows, Knowledge Bases, Guardrails, Prompt Management, Data Automation), Assess (Evaluations), and Configure and learn (Settings). The main area displays a table of models and their access status.

Model	Access status	Modality	EULA
DeepSeek-R1	Cross-region inference	Text	EULA
▼ Meta (11)			
Llama 3 8B Instruct	6/11 access granted		
Llama 3 70B Instruct	Access granted	Text	EULA
Llama 3.2 11B Vision Instruct	Cross-region inference	Text & Vision	EULA
Llama 3.1 8B Instruct	Cross-region inference	Text	EULA
Llama 3.1 70B Instruct	Cross-region inference	Text	EULA
Llama 3.2 1B Instruct	Cross-region inference	Text	EULA
Llama 3.2 90B Vision Instruct	Cross-region inference	Text & Vision	EULA
Llama 3.2 3B Instruct	Cross-region inference	Text	EULA
Llama 3.3 70B Instruct	Cross-region inference	Text	EULA
Llama 4 Scout 17B Instruct	Cross-region inference	Text & Vision	EULA

Amazon SageMaker:

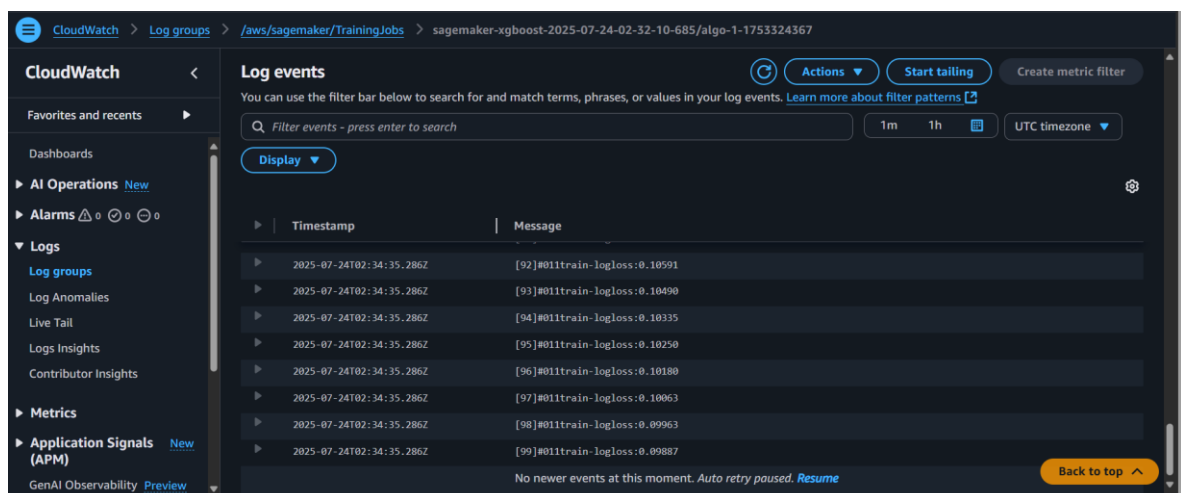
-Training job:



The screenshot shows the details of a SageMaker training job named 'sagemaker-xgboost-2025-07-24-02-32-10-685'. The left sidebar contains a navigation menu with sections: JumpStart (Foundation models, Computer vision models, Natural language processing models), Governance, HyperPod Clusters, Ground Truth, Processing, Training (Algorithms, Training jobs, Hyperparameter tuning jobs), Training Plans, Inference, Augmented AI, and AWS Marketplace. The main area displays job settings and algorithm details.

Job settings			
Job name sagemaker-xgboost-2025-07-24-02-32-10-685	Status Completed View history	SageMaker metrics time series Disabled	IAM role ARN arn:aws:iam::004082821794:role/SageMakerExecutionRole
ARN arn:aws:sagemaker:us-east-1:004082821794:training-job/sagemaker-xgboost-2025-07-24-02-32-10-685	Creation time Jul 24, 2025 02:32 UTC	Training time (seconds) 124	Billable time (seconds) 124
	Last modified time Jul 24, 2025 02:34 UTC	Managed spot training savings 0%	Tuning job source/parent -
Algorithm			
Algorithm ARN -	Additional volume size (GB) 5	Maximum wait time for managed spot training(s) -	Volume encryption key -
Training image 683313688378.dkr.ecr.us-east-1.amazonaws.com/sagemaker-xgboost:1.5-1	Maximum runtime (s) 3600	Managed spot training Disabled	

-Logs del training job en SageMaker:

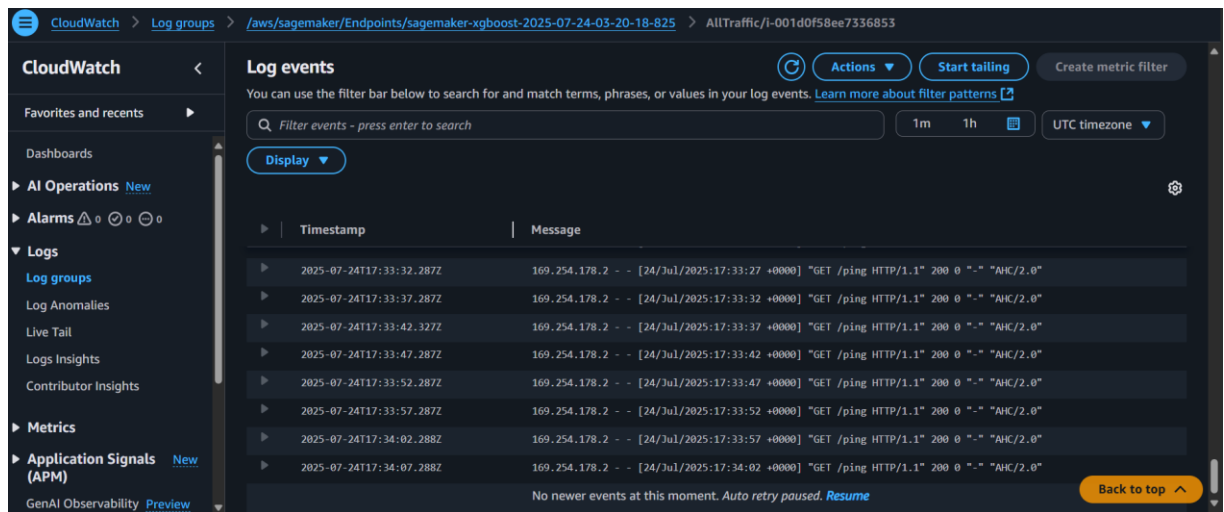


The screenshot shows the Amazon CloudWatch console's 'Log events' page for the training job. The left sidebar contains a navigation menu with sections: CloudWatch, Favorites and recents, Dashboards, AI Operations, Alarms, Logs (Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights), Metrics, Application Signals (APM), and GenAI Observability. The main area displays log events for the group '/aws/sagemaker/TrainingJobs'.

Timestamp	Message
2025-07-24T02:34:35.286Z	[92]#011train-logloss:0.10591
2025-07-24T02:34:35.286Z	[93]#011train-logloss:0.10490
2025-07-24T02:34:35.286Z	[94]#011train-logloss:0.10335
2025-07-24T02:34:35.286Z	[95]#011train-logloss:0.10250
2025-07-24T02:34:35.286Z	[96]#011train-logloss:0.10180
2025-07-24T02:34:35.286Z	[97]#011train-logloss:0.10063
2025-07-24T02:34:35.286Z	[98]#011train-logloss:0.09963
2025-07-24T02:34:35.286Z	[99]#011train-logloss:0.09887

No newer events at this moment. Auto retry paused. [Resume](#)

-Logs del Endpoint en SageMaker:



The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation at the top indicates the path: CloudWatch > Log groups > /aws/sagemaker/Endpoints/sagemaker-xgboost-2025-07-24-03-20-18-825 > AllTraffic/j-001d0f58ee7336853. The left sidebar contains navigation options: Favorites and recents, Dashboards, AI Operations, Alarms, Logs (selected), Log groups, Log Anomalies, Live Tail, Logs Insights, Contributor Insights, Metrics, Application Signals (APM), and GenAI Observability. The main panel is titled 'Log events' and includes a search bar with the placeholder 'Filter events - press enter to search', a 'Display' button, and filters for '1m' (1 minute), '1h' (1 hour), and 'UTC timezone'. Below the search bar, a table displays log events with columns for 'Timestamp' and 'Message'. The messages are HTTP GET requests to the endpoint, all returning a 200 status code and an 'AHC/2.0' response. At the bottom of the log list, a message states 'No newer events at this moment. Auto retry paused. Resume' with a 'Resume' link. A 'Back to top' button is located in the bottom right corner of the log list area.

Timestamp	Message
2025-07-24T17:33:32.287Z	169.254.178.2 - - [24/Jul/2025:17:33:27 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:33:37.287Z	169.254.178.2 - - [24/Jul/2025:17:33:32 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:33:42.327Z	169.254.178.2 - - [24/Jul/2025:17:33:37 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:33:47.287Z	169.254.178.2 - - [24/Jul/2025:17:33:42 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:33:52.287Z	169.254.178.2 - - [24/Jul/2025:17:33:47 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:33:57.287Z	169.254.178.2 - - [24/Jul/2025:17:33:52 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:34:02.288Z	169.254.178.2 - - [24/Jul/2025:17:33:57 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"
2025-07-24T17:34:07.288Z	169.254.178.2 - - [24/Jul/2025:17:34:02 +0000] "GET /ping HTTP/1.1" 200 0 "-" "AHC/2.0"

-Consideraciones:

El nombre del endpoint es:

sagemaker-xgboost-2025-07-24-03-20-18-825

URL del endpoint:

<https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/sagemaker-xgboost-2025-07-24-03-20-18-825/invocations>

Enlace al vídeo:

<https://youtu.be/wYHicxTEm5Y>