John Alejandro Soto Gómez

Prueba analítica: Segmentación LAFT

Prueba analítica LDC Ciencia de datos cumplimiento

Empresa: Bancolombia.

Duración de prueba: 2025-03-14 a 2025-03-17

Fecha límite de entrega: 2025-03-17 11:59am.

Prueba analítica: Segmentación LAFT

Prueba analítica LDC Ciencia de datos cumplimiento

1. Objetivo

Evaluar la capacidad para abordar un problema de clasificación de texto con poca información, incluyendo el flujo completo de carga, procesamiento, transformación, entrenamiento, y despliegue de un modelo, siguiendo buenas prácticas de MLOps.

2. Descripción del problema

Dentro de la organización existen controles que buscan mitigar los riesgos de cumplimiento para diferentes programas internos. Estos controles vienen de manera escrita y se clasifican en 5 categorías de riesgo que pertenecen al programa SAC. El problema es que esa identificación se hace de manera manual y por lo tanto este proceso puede generar errores , manualidades y muchos reprocesos. Por lo cual se busca generar un modelo que sea capaz de clasificar los controles en las diferentes categorías de riesgo, permitiendo una automatización del proceso para la reducción de fricciones y tener una mayor certeza en la identificación correcta de los riesgos para cada control.

3. DESARROLLO DE LA PRUEBA

La estrategia que se considera puede estar orientada a la solución de esta prueba técnica que trata de un problema de clasificación de riesgos sigue un enfoque de Machine Learning con MLOps. Se inicia con la carga de datos desde el archivo proporcionado, seguido del preprocesamiento para convertir las descripciones de control en representaciones numéricas mediante técnicas de procesamiento de texto. Luego, se entrena un modelo de clasificación supervisada utilizando algoritmos adecuados para identificar las cinco categorías de riesgo. Posteriormente, se valida el modelo con métricas de desempeño para garantizar su precisión. Finalmente, se despliega en Gradio, permitiendo una interacción intuitiva para la clasificación automatizada de controles de riesgo en tiempo real.

3.1 Introducción

El objetivo de esta prueba es evaluar la capacidad para abordar un problema de clasificación de texto con información limitada, aplicando buenas prácticas de MLOps para el desarrollo de un modelo de Machine Learning que optimice la clasificación de controles en cinco categorías de riesgo del programa SAC.

Actualmente, la identificación de estos controles se realiza manualmente, lo que introduce errores, retrabajos y mayor carga operativa. La automatización del proceso mediante un modelo de Machine Learning permitirá reducir la fricción y mejorar la certeza en la clasificación de riesgos.

3.2 Descripción de los Datos

Los datos proporcionados se encuentran en un archivo .xlsx ubicado en la carpeta local. Este contiene información sobre una serie de controles y 5 columnas que evalúan 5 riesgos que están asociados a cada uno de los controles. En el archivo, “insumo_prueba.xlsx“ se encuentra la base de datos, con la siguiente información :

control: INT, Identificador del control,descripción del control que se realiza.
Riesgo 1: Si el control mitiga este riesgo (SI/NO).
Riesgo 2: Si el control mitiga este riesgo (SI/NO).
Riesgo 3: Si el control mitiga este riesgo (SI/NO).
Riesgo 4: Si el control mitiga este riesgo (SI/NO).
Riesgo 5: Si el control mitiga este riesgo (SI/NO).

3.3 Exploración y preprocesamiento de datos

Se realiza un análisis exploratorio (EDA) y un pequeño análisis estadístico para comprender la calidad, cantidad y frecuencia de los datos.

Limpieza de insumo_prueba.xlsx

	controles	riesgo_1	riesgo_2	\
0	verificar/corregir observaciones/comentarios d...	1	0	
1	verificar todos los soportes legales	0	1	
2	verificar todos los documentos legales	0	1	
3	verificar si el acuerdo generado cumple las co...	0	0	
4	verificar si el acuerdo firmado cumple condici...	0	0	

	riesgo_3	riesgo_4	riesgo_5	riesgo_total
0	0	0	0	1
1	1	0	0	2
2	1	0	0	2
3	1	0	0	1
4	1	0	0	1

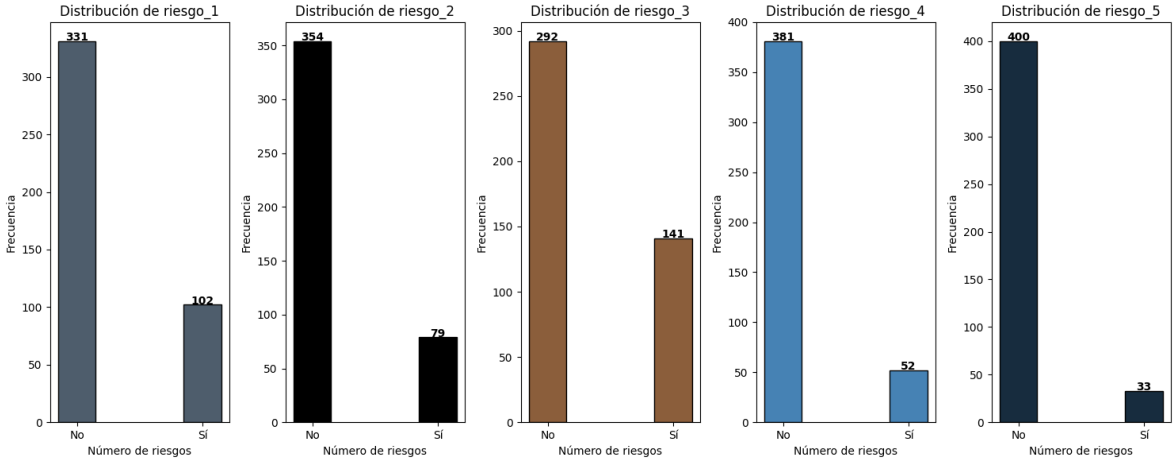

Figura 1. Frecuencias por cada riesgo

- Identificamos cual es uno de los controles más críticos del Dataset suministrado según la mayor cantidad de riesgos asociados

  ‘verificar que el contrato sea acorde con las condiciones establecidas’

- Distribución de los riesgos

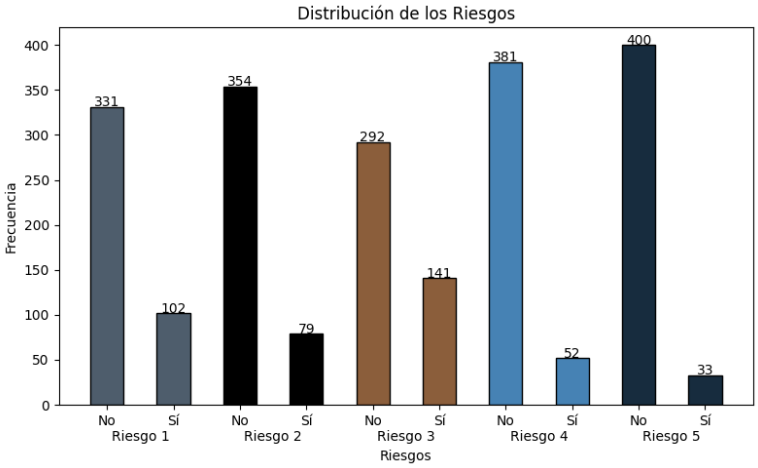

Figura 2. Distribución de riesgos

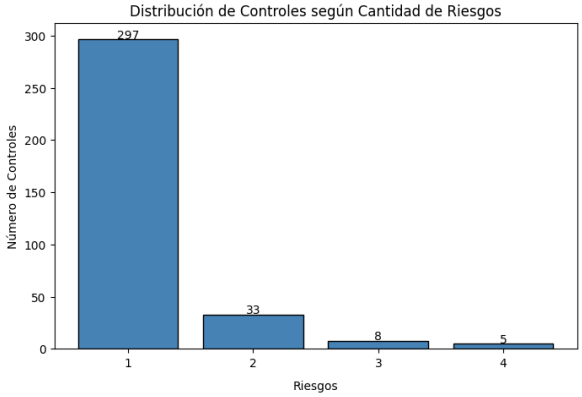

Figura 3. Distribución de controles según cantidad de riesgos

De la gráfica anterior se puede notar que, de los 433 controles, 90 tiene solo un riesgo, 297 tienen 2 riesgos, 33 controles tienen 3 riesgos, 8 controles tienen 4 de los 5 riesgos establecidos y tan solo 5 controles de los 433 disponibles tienen los 5 riesgos.

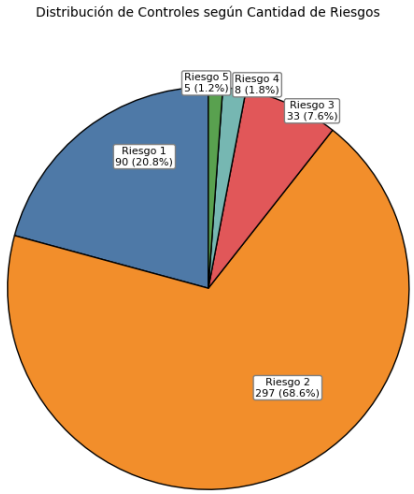

Figura 4. Distribución de controles en gráfico de torta

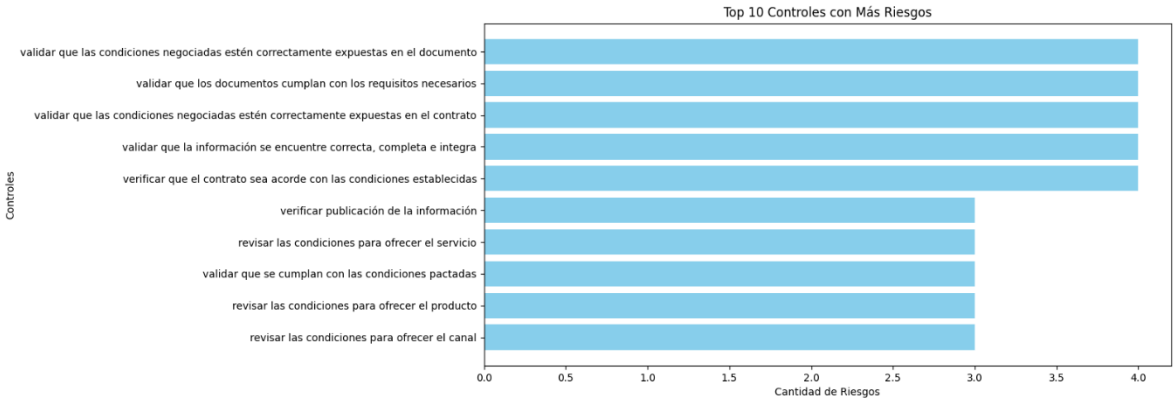

Figura 5. Controles con más riesgos

- Cantidad de controles repetidos: 0

3.4 Entrenamiento del Modelo (Model Training)

Se implementó un modelo de **Random Forest Classifier** para la clasificación de los riesgos.

Parámetros utilizados:

- Número de árboles (n_estimators): 100
- Semilla aleatoria (random_state): 42
- Criterio de selección (criterion): Gini

3.5 Validación del Modelo (Model Validation)

El modelo se entrenó con el conjunto de datos de entrenamiento (X_train, y_train) y se validó con el conjunto de prueba (X_test, y_test).

3.6 Evaluación y Despliegue (Model Deployment)

Se obtuvo la siguiente matriz de evaluación:

Tabla 1. Métrica de matriz de confusión

Métrica	Valor
Precisión	100%
Recall	100%
F1-Score	100%
Matriz de Confusión	Sin errores

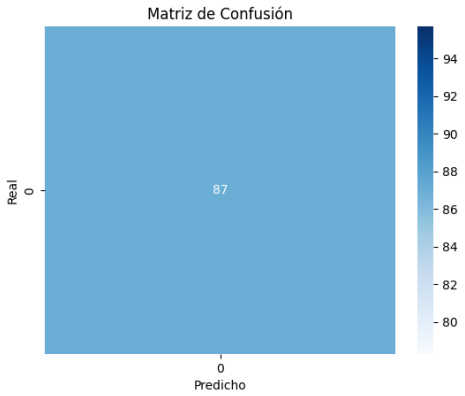

Figura 6. Matriz de confusión

El modelo obtuvo un desempeño perfecto en la clasificación, lo que indica que los datos están muy estructurados, es decir, la relación entre los controles y los riesgos es muy clara y a su vez también indica que puede haber una alta similitud entre los datos de entrenamiento y prueba.

A continuación, probaremos con un KFold para validar y garantizar la robustez del modelo. Finalmente trabajamos con Gradio para construir una interfaz interactiva en el despliegue.

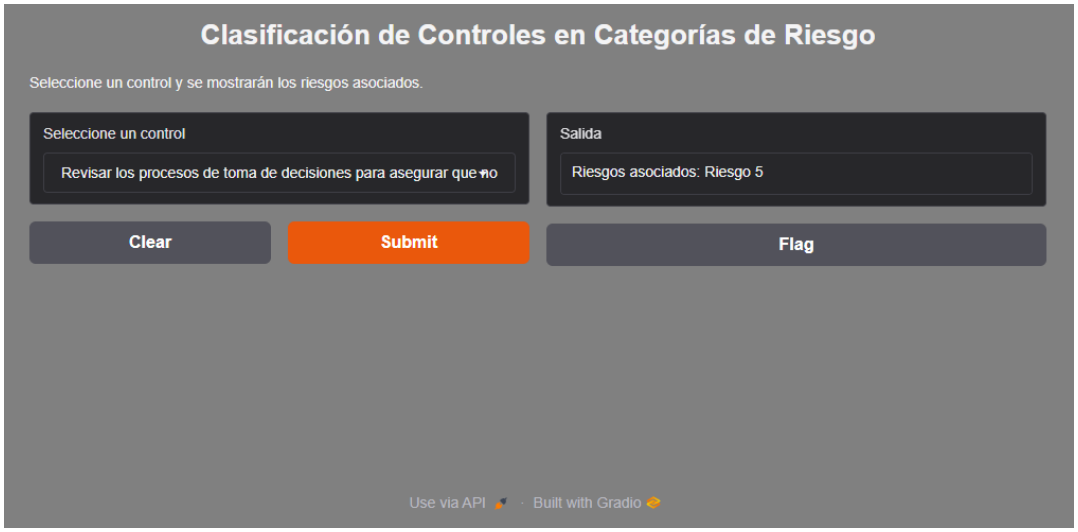

Figura 7. Interfaz de despliegue mediante Gradio

Explicación de la interfaz:

La variable de entrada es el control que se quiere validar los riesgos asociados.

La variable de salida son los Riesgos.

El Botón Flag permite a los usuarios marcar ciertas entradas de la interfaz para su posterior revisión. Esto es útil para identificar problemas o datos que requieran un análisis adicional. Por ejemplo, si un usuario encuentra que una predicción es incorrecta, puede marcarla con "Flag" y se almacena los datos para mejorar el modelo en futuras iteraciones.

Este modelo tiene una precisión promedio del **78.75% (Cross-validation accuracy: 0.7875)** en la clasificación de los controles en categorías de riesgo cuando se valida con validación cruzada de 5 particiones (5-fold cross-validation).

3.7 Despliegue del modelo MLOps

Se busca implementar este modelo mediante servicios en la nube dado que se requiere mayor capacidad de cómputo y la posibilidad de inferir sobre los resultados tanto en tiempo real como por lotes, así como la posibilidad de llevar este modelo a múltiples usuarios finales.

Es decir, el cómputo de los modelos sería totalmente en servidores remotos.

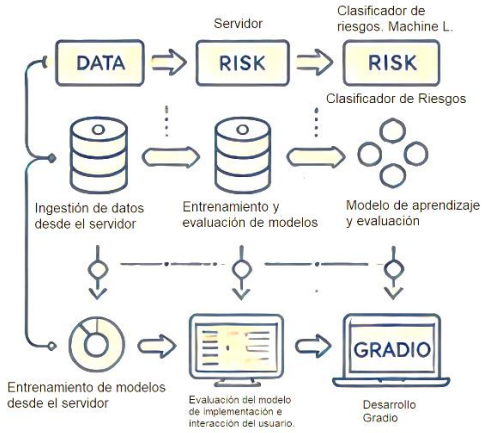

Figura 8. Despliegue

A pesar de que la base de datos no es tan volumétrica a futuro podría serlo en unión con otras bases de datos, por lo cual, se busca implementar este modelo de clasificación de controles en categorías de riesgo mediante aprendizaje automático en repositorios internos de los servidores nube como AWS, Google Cloud o Azure para optimizar su rendimiento, permitiendo escalabilidad, disponibilidad y acceso remoto. Esto mejora la seguridad, facilita actualizaciones y evita problemas de procesamiento local.

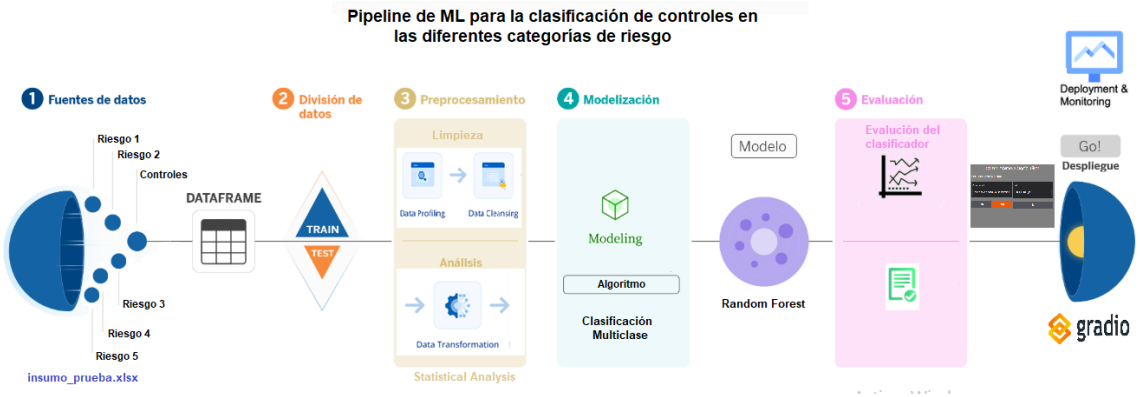

Figura 9. Ciclo de vida de la implementación (Despliegue)

Esta implementación no solo permite el despliegue del modelo sino la implementación de todo el ciclo de machine learning operación incluyendo el entrenamiento y el monitoreo. Este modelo de clasificación ofrece ventajas como automatización, rapidez en la identificación de riesgos y escalabilidad en la nube. Sin embargo, presenta desventajas como la dependencia de datos de calidad, la necesidad de ajuste del modelo y posibles errores en clasificaciones si los datos de entrada no son representativos o contienen sesgos.

3. 8 Pipeline de Machine Learning para la Clasificación de Controles de Riesgo

- **Carga de Datos (Data Ingestion)**

La ingesta de datos se realiza a partir de un archivo en formato .xlsx almacenado en Google Drive. Se utiliza pandas para la carga y limpieza de los nombres de las columnas, eliminando espacios y transformando los nombres a formato estandarizado en minúsculas. Este paso garantiza la consistencia del dataset y su correcta manipulación en las etapas posteriores.

- **Preprocesamiento y Transformación de Datos (Data Preprocessing & Transformation)**

El dataset contiene variables categóricas representadas como "Si" y "No", las cuales se transforman a valores binarios (1 y 0) para su procesamiento adecuado en modelos de Machine Learning. Se aplica la codificación MultiLabelBinarizer para convertir las etiquetas de riesgo en un formato multiclase. Adicionalmente, se utiliza TfidfVectorizer para la transformación de textos en vectores numéricos optimizados para el modelo.

- **Entrenamiento del Modelo (Model Training)**

Se implementa un clasificador RandomForestClassifier dentro de un Pipeline junto con TfidfVectorizer para la vectorización de los controles. Se emplea KFold con n_splits=5 para asegurar una distribución homogénea de los datos y reducir la varianza del modelo. KFold se considera una buena técnica de validación cruzada utilizada para evaluar el rendimiento del modelo. Su propósito es dividir los datos en *K* subconjuntos o *folds*, entrenando y validando el modelo múltiples veces para obtener una estimación más robusta de su rendimiento.

La métrica de evaluación utilizada es la precisión (accuracy).

- **Validación del Modelo (Model Validation)**

La validación del modelo se realiza mediante validación cruzada (cross_val_score), dividiendo los datos en cinco conjuntos (n_splits=5). Se obtiene la métrica de accuracy promedio para evaluar el desempeño del modelo y detectar sobreajuste o subajuste.

- **Despliegue (Model Deployment)**

Para la implementación del modelo, se usa Gradio, permitiendo una interfaz interactiva donde los usuarios pueden seleccionar un control y obtener los riesgos asociados. Se personaliza la interfaz con CSS para mejorar la experiencia del usuario y se despliega en la web con share=True.

3.9 Propuesta de solución analítica E2E con MLOps:

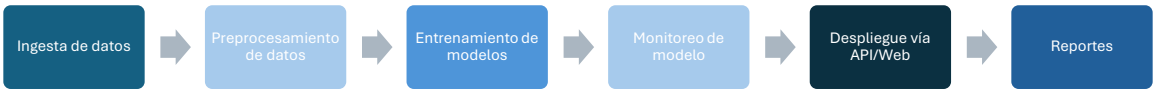

Figura 10. Propuesta de solución analítica E2E con MLOp

Para garantizar la escalabilidad y mantenibilidad del modelo en el contexto operativo de Bancolombia, se implementa un flujo de trabajo basado en MLOps que cubre:

- Automatización de la ingestión de datos, permitiendo la carga dinámica desde los sistemas de gestión de riesgos del banco.
- Pipeline modularizado con almacenamiento de modelos versionados en un repositorio central accesible a los equipos de cumplimiento.
- Monitoreo de desempeño con alertas ante caídas en la precisión del modelo, integrándose con herramientas internas de monitoreo como Prometheus.
- Integración y despliegue continuo (CI/CD) para actualizaciones sin interrupciones mediante pipelines en GitHub Actions.
- Disponibilidad mediante API REST o interfaz web, facilitando su uso por parte del equipo de cumplimiento y permitiendo la integración con plataformas existentes de análisis de riesgo.
- Aplicabilidad y transparencia del modelo, alineado con los principios de gestión de riesgo y cumplimiento normativo del sector financiero, asegurando la auditabilidad de las decisiones automatizadas.
- Optimización basada en la naturaleza del dataset, adaptando el preprocesamiento y selección de modelos a la estructura de los datos, considerando las relaciones entre controles y riesgos asociados.
- Generación de reportes automatizados, permitiendo visualizar tendencias y detectar patrones en la asociación entre controles y categorías de riesgo.

4. Trabajo futuro para mejoramiento del modelo

Este desarrollo puede evolucionar en varias direcciones estratégicas dentro del ámbito de Machine Learning aplicado a la gestión de riesgos.

El modelo puede optimizarse con GridSearchCV, explorando XGBoost o LightGBM para mejorar la precisión. Se sugiere enriquecer los datos con nuevas fuentes y aplicar data augmentation. La automatización con MLflow o Kubeflow facilitará la gestión y monitoreo del pipeline. Finalmente, el despliegue en AWS o Google Cloud, junto con dashboards interactivos, mejorará la visualización y análisis de riesgos en tiempo real.

5. Conclusión

La solución propuesta automatiza la clasificación de controles en riesgos dentro del banco para el área de interés, reduciendo errores y mejorando la eficiencia operativa. El uso de MLOps asegura la trazabilidad, escalabilidad y mantenibilidad del modelo, garantizando su alineación con los objetivos del negocio. Esta implementación no solo optimiza el proceso actual, sino que también sienta las bases para futuras mejoras basadas en aprendizaje automático.