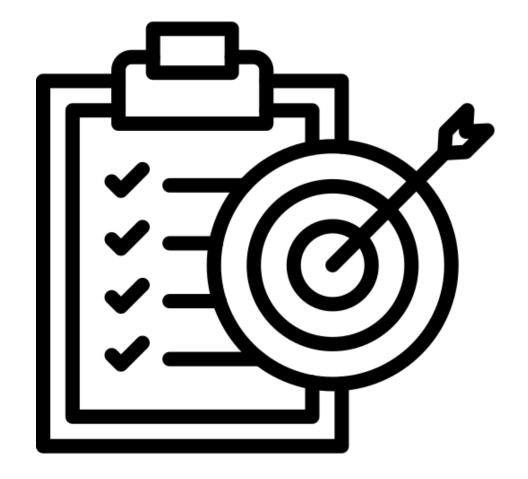
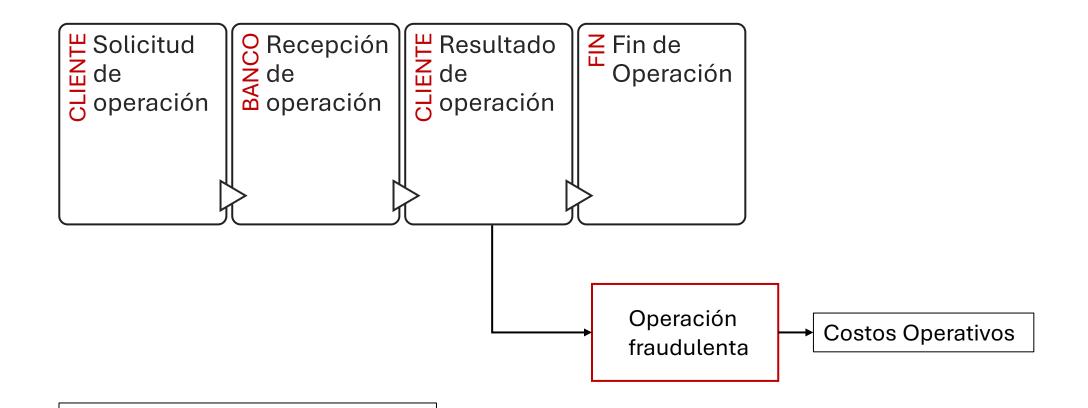
Predicción de transacciones fraudulentas a través de algoritmos de clasificación



Problemática y Objetivos

- Se cometen varias transacciones fraudulentas
 - Costos operativos
 - Fidelización del cliente
 - Mala reputación en el mercado
- Crear un modelo estadístico para predecir si una operación será fraudulenta o no.
 - Ayudar a mejorar la detección de fraude para disminuir los casos asociados a este.





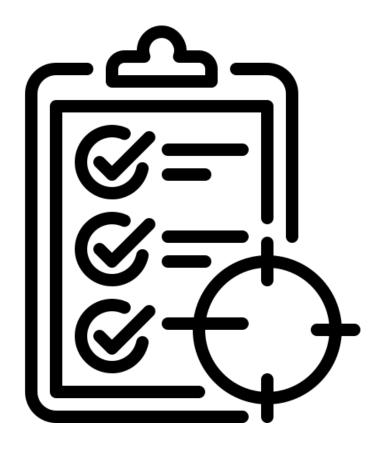
* Robo de tarjetas de crédito

* Transferencias fraudulentas

* Clonación

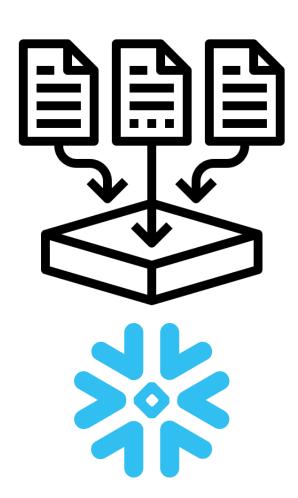
Alcance

- Periodo de prueba inicial:
 - Limitar la base de datos a ciudades específicas.
 - Operaciones transaccionales comunes (retiros, transferencias, pagos).
 - Información de 6 meses histórica
- A excluir:
 - Operaciones transaccionales en persona (directo en la agencia).
 - Periodo de tiempo se limita por cuestiones de procesamiento y recursos.



Datos

- Base de datos con operaciones transaccionales en el alcance definido:
 - ID usuario
 - Monto transacción
 - Ciudad
 - Tipo transacción
 - Dispositivo usado
- Ubicación: Base cargada a Snowflake



Limpieza y Transformación

- Eliminación de valores duplicados
- Tratamiento de valores nulos
- Eliminación de columnas innecesarias
- Creación de variables adicionales



Modelo Estadístico

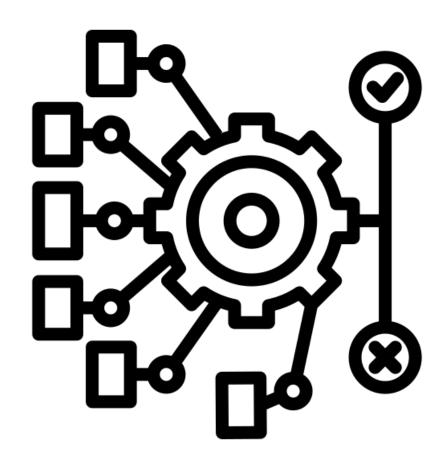
- Algoritmo de clasificación: Operación fraudulenta, Operación NO fraudulenta
- Regresión logística:
 - Eficiente y rápido, fácil interpretación
 - Bajo rendimiento dependiendo el tipo de datos
- Árboles de decisión:
 - Buen manejo de datos mixtos (numéricos y categóricos)
 - Tendencia a overfitting
- Random forest:
 - · Mejor rendimiento, reduce overfitting
 - Requiere más trabajo computacional



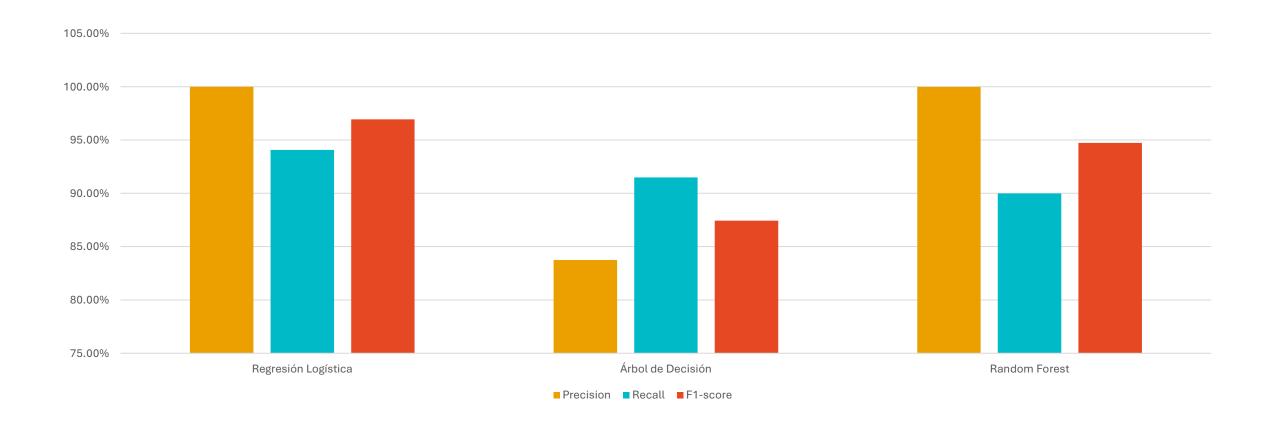
Recall > 85%



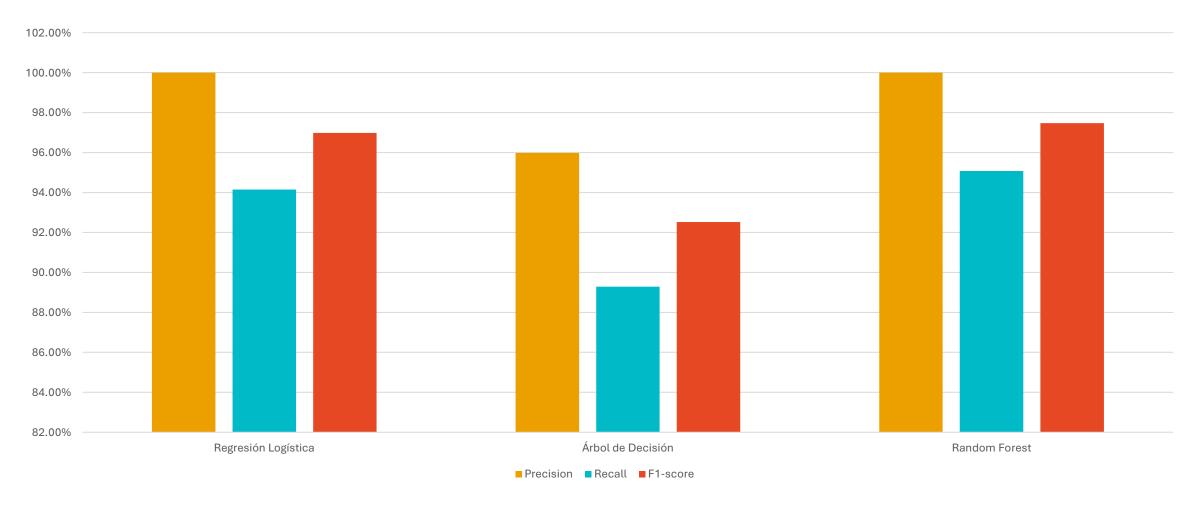
Reducir costos operacionales



Resultados (Entrenamiento)

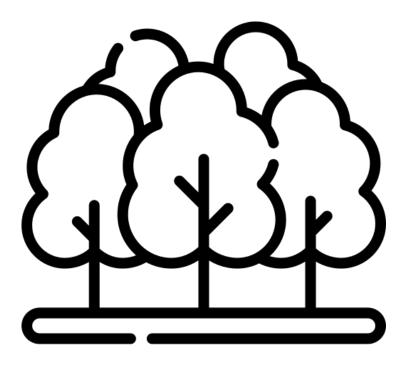


Resultados (Test)



Mejor Modelo

- Random Forest
 - Resultado consistente tanto en entrenamiento como en prueba
 - Recall mejora en prueba lo que significa que captura más fraudes y el f1-score aumenta ya que hay un balance entre la precisión y el recall.
 - Menos probabilidad de overfitting para futuros escenarios.



Plan de Implementación

Desarrollo del Modelo Validación y despliegue a Despliegue del modelo producción • Definir problema • Integración del modelo: • Recoleccion y limpieza de datos • API: Docker • Validación con datos independientes • Feature Engineering Batch: Airflow, Spark • Pruebas de robustez • Selección de modelos • Nube: AWS SageMaker • Guardar el modelo (joblib) • Determinar si es en la nube o on- Evaluación de rendimiento Versionar

Monitoreo

Documentar

- Monitorear métricas en producción
- Detectar cambios en patrones
- Establecer alertas (dashboards)

Reentrenamiento

- Recolección de nuevos datos
- Ciclos de reentrenamiento

• Crear scripts de predicción

• Actualizar modelo en producción

premise

Conclusiones y Recomendaciones

- Se logró encontrar un modelo de clasificación adecuado para la base de datos obtenida (Random Forest)
- Se logró el objetivo de encontrar un recall mayor al 85%
- Se recomienda aumentar el alcance de los datos para probar si de verdad es compatible el modelo con más información.

