1.1. Modelo de negocio Big Data: Predicción y prevención de la deserción estudiantil.

Problema u oportunidad

La universidad presenta un nivel relevante de abandono estudiantil, especialmente en los primeros años. Esto afecta al estudiante que interrumpe su formación y reduce sus oportunidades laborales y a la institución, que enfrenta pérdidas económicas, menor eficiencia académica e impacto reputacional.

Este contexto abre una oportunidad estratégica: aplicar Big Data y analítica predictiva para identificar tempranamente a estudiantes en riesgo. La variable objetivo es multiclase: ***abandona, continúa o se gradúa*.**

La predicción temprana permitirá diseñar acciones personalizadas de retención, optimizar la admisión y mejorar la propuesta de valor educativa

Usuario principal

- **Gestión Académica y Bienestar Estudiantil**: seguimiento y acompañamiento.
- **Dirección Académica**: diseño curricular y políticas de permanencia.
- **Marketing y Admisiones**: identificación de perfiles con mayor probabilidad de éxito.

Decisión empresarial que permitirá tomar el modelo

El modelo predictivo permitirá tomar decisiones **basadas en datos** sobre:

- Detectar estudiantes con alto riesgo y priorizar intervenciones.
- Identificar factores académicos, económicos o sociales que explican la deserción.
- Ajustar políticas de admisión y becas.
- Diseñar estrategias de apoyo diferenciadas por tipo de estudiante.

Datos necesarios y justificación

- **Académicos**: rendimiento, aprobaciones, reprobaciones, promedio → reflejan compromiso y desempeño.
- **Socioeconómicos**: nivel financiero, becas, educación familiar → explican factores externos.
- **Administrativos**: modalidad, turno, preferencias al postular → indican motivación y compatibilidad.
- **Contextuales**: empleo, inflación, PIB → capturan presiones económicas externas.

La integración de estas variables permite estimar la probabilidad individual de abandonar y genera *insights* para la toma de decisiones.

Valor que aporta el proyecto:

Para la universidad:

- Reducción del abandono y aumento de la graduación.
- Mejor planificación académica y financiera.
- Mayor reputación y atractivo institucional.
- Decisiones basadas en evidencia.

Para los estudiantes:

- Detección temprana de riesgos.
- Acceso a apoyo académico, vocacional, financiero o psicológico.
- Mayor probabilidad de éxito y continuidad.

Estrategias derivadas del modelo

- Alertas tempranas automatizadas.
- Mentoría personalizada.
- Apoyo psicológico y financiero enfocado en la causa raíz.
- Ajustes académicos o curriculares.
- Optimización de admisiones.
- Dashboards de seguimiento continuo.

Conclusión

El modelo predictivo transforma datos dispersos en conocimiento accionable, permitiendo mejorar la experiencia del estudiante, optimizar recursos institucionales y fortalecer la sostenibilidad académica y financiera.

1.2. Arquitectura Cloud para el Análisis de Datos Académicos

Se diseñó una arquitectura Cloud simple y eficiente, compuesta por:

1. Origen de datos

Dataset externo: *students_dropout_academic_success.csv* con 4424 registros.

2. Ingesta

Carga del dataset al entorno cloud para asegurar disponibilidad y trazabilidad.

3. Almacenamiento

Base de datos **Cloud SQL (Google Cloud)**, garantizando integridad, seguridad y consultas eficientes.

4. Procesamiento y entrenamiento

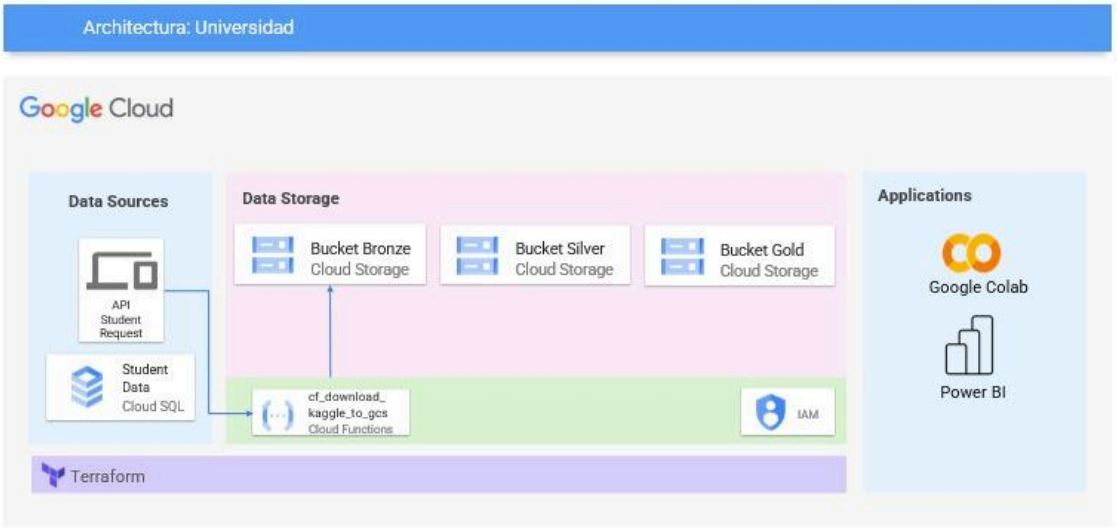
Realizado en notebooks, incluyendo:

- Limpieza, preprocesamiento y transformación.
- Entrenamiento del modelo dentro de un flujo reproducible (pipeline).

5. Consumo

Aplicación **Streamlit**, que funciona como dashboard interactivo para visualizar métricas, interpretar resultados y consultar predicciones.

6. Diagrama de Arquitectura



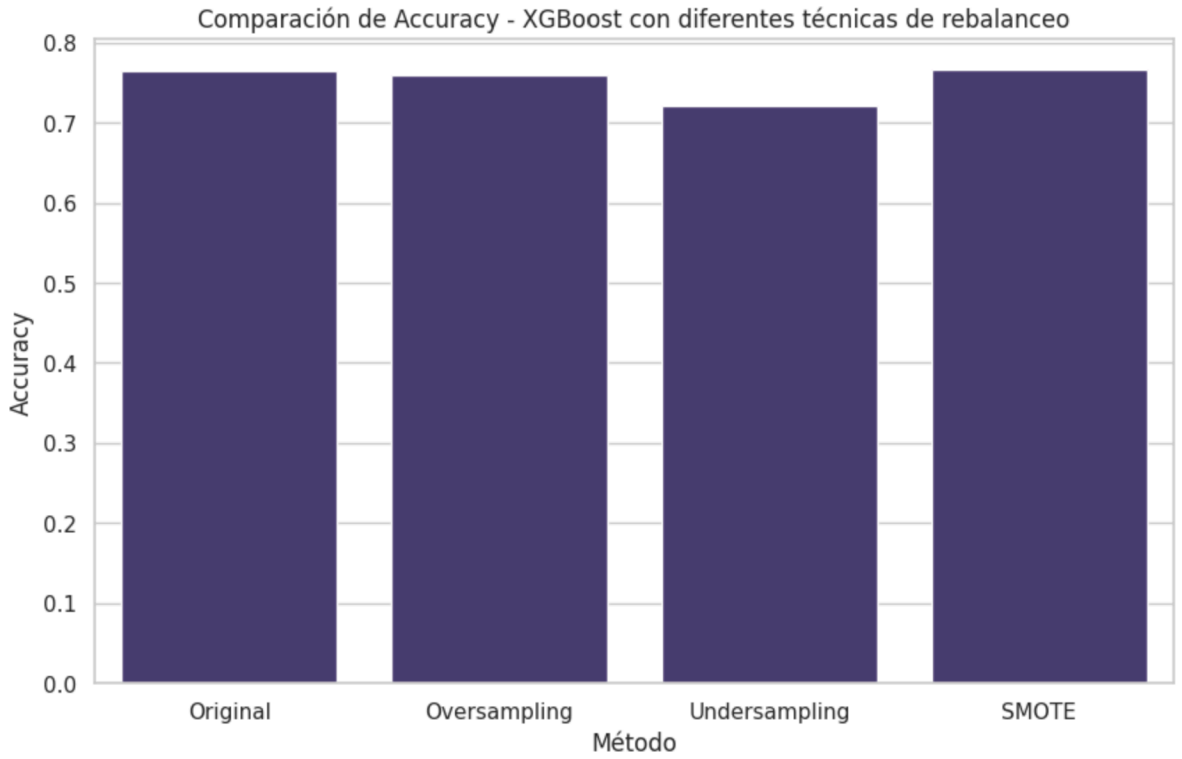
Modelo seleccionado: XGBoost + SMOTE

El uso de **SMOTE** obtuvo el mejor *accuracy* (0.7658).

Motivos para elegir SMOTE

- Aumenta significativamente el *recall* en clases minoritarias.
- Reduce el riesgo de *overfitting* frente al oversampling tradicional.
- Mantiene la información sin eliminar datos como en el undersampling.
- Funciona especialmente bien en problemas de dataset desbalanceado.

Recomendación final: XGBoost o RandomForest combinados con SMOTE.



1.4. Obtención de datos

Se utilizó un dataset público descargado desde **Kaggle**. La ingesta se gestionó mediante el script:

`/src/data_ingestion.py`

Funcionalidades implementadas

- Descarga/carga del dataset desde la fuente pública.
- Lectura y validación del esquema.
- Transformaciones adicionales (tipos de datos, normalización).
- Guardado del dataset limpio en: `data/raw/` o `data/processed/`.

El script asegura un proceso reproducible para análisis y modelado.

1.5. Limpieza, EDA y preprocesamiento

Documentado en un notebook, incluyendo:

Tratamiento de nulos

- Imputación con media/mediana (numéricas) y moda/desconocido (categóricas).
- Eliminación solo cuando fue necesario.

Duplicados

- Identificación por claves.
- Eliminación de duplicados completos.

Outliers

- Detección con IQR o z-score.
- Winsorization o eliminación puntual cuando se detectaron errores evidentes.

Codificación

- One-Hot Encoding para modelos lineales.
- Label Encoding para modelos basados en árboles.

Escalado

- StandardScaler o MinMaxScaler según el modelo.

Feature Engineering

- Nuevas variables a partir de combinaciones relevantes.
- Transformaciones logarítmicas cuando fue necesario.

EDA

Visualizaciones elaboradas:

- Histogramas.
- Heatmap de correlaciones.
- Distribución de la variable objetivo.
- Boxplots.
- Gráficos de barras por categoría.

1.6. Modelo de Machine Learning

Separación Train/Test

- Train: 70–80%
- Test: 20–30%
- `random_state` para reproducibilidad.

Modelos evaluados

- Regresión logística / Random Forest / SVM
- Árbol de decisión / KNN / XGBoost

Métricas

Para clasificación:

- Accuracy, Precision, Recall, F1-score
- Matriz de confusión

El mejor desempeño se obtuvo con **XGBoost + SMOTE**.

1.7. Dashboard analítico

Se utilizó **Power BI** para visualizar datos académicos y proyecciones del estado del estudiante, complementando la interpretación del modelo.