🎧 ML CONSULTING

Caso SoundFlow: Predicción de Churn

CASO DE NEGOCIO

SoundFlow, una startup de streaming de música con 2 millones de usuarios.

El 8% de usuarios cancela su suscripción premium cada mes (churn). El equipo de marketing quiere predecir qué usuarios van a cancelar para ofrecerles promociones preventivas.

El dataset está muy desbalanceado:

• De 50.000 registros históricos, solo 4.000 son churners (8%)

CONTEXTO DE NEGOCIO

• Cada usuario que se va representa 120€/año de pérdida

• Una promoción preventiva cuesta 15€ por usuario

• El CEO quiere un modelo en producción en 2 semanas

OBJETIVO CLAVE

¿Cómo manejamos el desbalanceo de clases para entrenar un modelo que detecte churners de forma fiable, sin sesgar las predicciones hacia la clase mayoritaria?

⚔️ DINÁMICA DE LA ACTIVIDAD

Cada equipo debe resolver el problema desde una perspectiva técnica diferente:

🔵 EQUIPO A	🔴 EQUIPO B
UNDERSAMPLING	**OVERSAMPLING**

🔵 EQUIPO A: UNDERSAMPLING

OBJETIVO:

Vuestra consultora apuesta por reducir la clase mayoritaria para equilibrar el dataset. Debéis diseñar una estrategia completa de Machine Learning usando esta filosofía.

🎯 Técnicas de Undersampling que DEBÉIS considerar

TÉCNICA	DESCRIPCIÓN
Random Undersampling	Elimina aleatoriamente muestras de la clase mayoritaria hasta igualar la minoritaria. Simple pero puede perder información valiosa.
NearMiss	Selecciona muestras de la clase mayoritaria que están cerca de la frontera de decisión. Más inteligente que aleatorio.
Tomek Links	Elimina pares de muestras muy cercanas de clases opuestas ("links"). Limpia la frontera de decisión.
Cluster Centroids	Agrupa la clase mayoritaria y usa los centroides como representantes. Reduce mucho el tamaño preservando estructura.

💻 Código Python de ejemplo

Instalación necesaria

```
pip install imbalanced-learn
```

```python
# Random Undersampling
from imblearn.under_sampling import RandomUnderSampler

rus = RandomUnderSampler(random_state=42)
X_resampled, y_resampled = rus.fit_resample(X_train, y_train)

# Ver el nuevo balance
print(f'Antes: {Counter(y_train)}')
print(f'Después: {Counter(y_resampled)}')

# ================================================
# NearMiss (versión 1)
from imblearn.under_sampling import NearMiss

nm = NearMiss(version=1)
X_resampled, y_resampled = nm.fit_resample(X_train, y_train)

# ================================================
# Tomek Links
from imblearn.under_sampling import TomekLinks

tl = TomekLinks()
X_resampled, y_resampled = tl.fit_resample(X_train, y_train)
```

Otros ejemplos aquí

EQUIPO B: OVERSAMPLING

OBJETIVO

Vuestra consultora apuesta por generar más ejemplos de la clase minoritaria para equilibrar el dataset. Debéis diseñar una estrategia completa de Machine Learning usando esta filosofía.

Técnicas de Oversampling que DEBÉIS considerar

TÉCNICA	DESCRIPCIÓN
Random Oversampling	Duplica aleatoriamente muestras de la clase minoritaria. Simple pero puede causar overfitting por repetición exacta.
SMOTE	Synthetic Minority Over-sampling Technique. Genera muestras sintéticas interpolando entre vecinos cercanos de la clase minoritaria.
ADASYN	Adaptive Synthetic. Similar a SMOTE pero genera más muestras en regiones donde la clase minoritaria es más difícil de aprender.
BorderlineSMOTE	Genera muestras sintéticas solo para los ejemplos minoritarios que están en la frontera de decisión (los más difíciles).

💻 Código Python de ejemplo

Instalación necesaria

```
pip install imbalanced-learn
```

```python
# SMOTE (el más popular)
from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)

# Ver el nuevo balance
print(f'Antes: {Counter(y_train)}')
print(f'Después: {Counter(y_resampled)}')

# ================================================
# ADASYN (adaptativo)
from imblearn.over_sampling import ADASYN

adasyn = ADASYN(random_state=42)
X_resampled, y_resampled = adasyn.fit_resample(X_train, y_train)

# ================================================
# BorderlineSMOTE
from imblearn.over_sampling import BorderlineSMOTE

bsmote = BorderlineSMOTE(random_state=42, kind='borderline-1')
X_resampled, y_resampled = bsmote.fit_resample(X_train, y_train)
```

Otros ejemplos aquí

PREGUNTAS

OJO

AMBOS equipos debéis responder las preguntas de estrategia.

#	PREGUNTAS ESTRATÉGICAS (NIVEL TÉCNICO)
1	¿Cómo dividís los datos? ¿train/test? ¿en qué proporción? ¿por qué?
2	¿Dónde aplicáis el balanceo? ¿antes o después del split?
3	¿Qué técnica concreta de balanceo usáis? ¿por qué esa y no otra de las disponibles?
4	¿Qué métrica priorizáis? ¿accuracy, precision, recall, f1, auc-roc?
5	¿Qué riesgos tiene vuestro enfoque? ¿qué podría salir mal?

#	PREGUNTAS ESTRATÉGICAS (NIVEL NEGOCIO)
1	¿Qué objetivo de negocio concreto deberíamos optimizar con este proyecto (más allá de “predecir churn”) y qué indicadores usaríais para medir el éxito?
2	Con 120€/año de pérdida por churner y 15€ por promoción, ¿cómo evaluaríais si una campaña preventiva es rentable? Explicad qué números o condiciones miraríais.
3	Si el presupuesto no permite ofrecer promociones a todos, ¿cómo decidiríais a qué usuarios priorizar y por qué?
4	¿Cómo evitaríais “educar” a los usuarios a cancelar para conseguir descuentos? Proponed medidas comerciales o de producto para reducir ese efecto.
5	¿Cómo definiríais “churn” de forma operativa (qué significa exactamente “cancelar premium”) y qué ventana temporal tendría sentido para actuar a tiempo?
6	En este negocio, ¿qué es más dañino: ofrecer promoción a alguien que iba a quedarse (FP) o no ofrecerla a alguien que se iba (FN)?
7	Una vez en producción, ¿qué podría hacer que el sistema se degrade con el tiempo (cambios de producto, estacionalidad, campañas) y qué plan pondríais para monitorizar y actualizarlo?

REGISTRO DE TAREAS DEL EQUIPO

Rellenad esta tabla con las tareas realizadas por cada miembro del equipo:

Enfoque: ☐ Undersampling ☐ Oversampling

NOMBRE	TAREAS REALIZADAS