GenZ Working Environment
Modelo de clasificación binaria + API REST (Flask + ngrok)
UDD Bootcamp — Módulo 7

Autor: Elías Aburto

Problema

¿Qué queremos predecir?

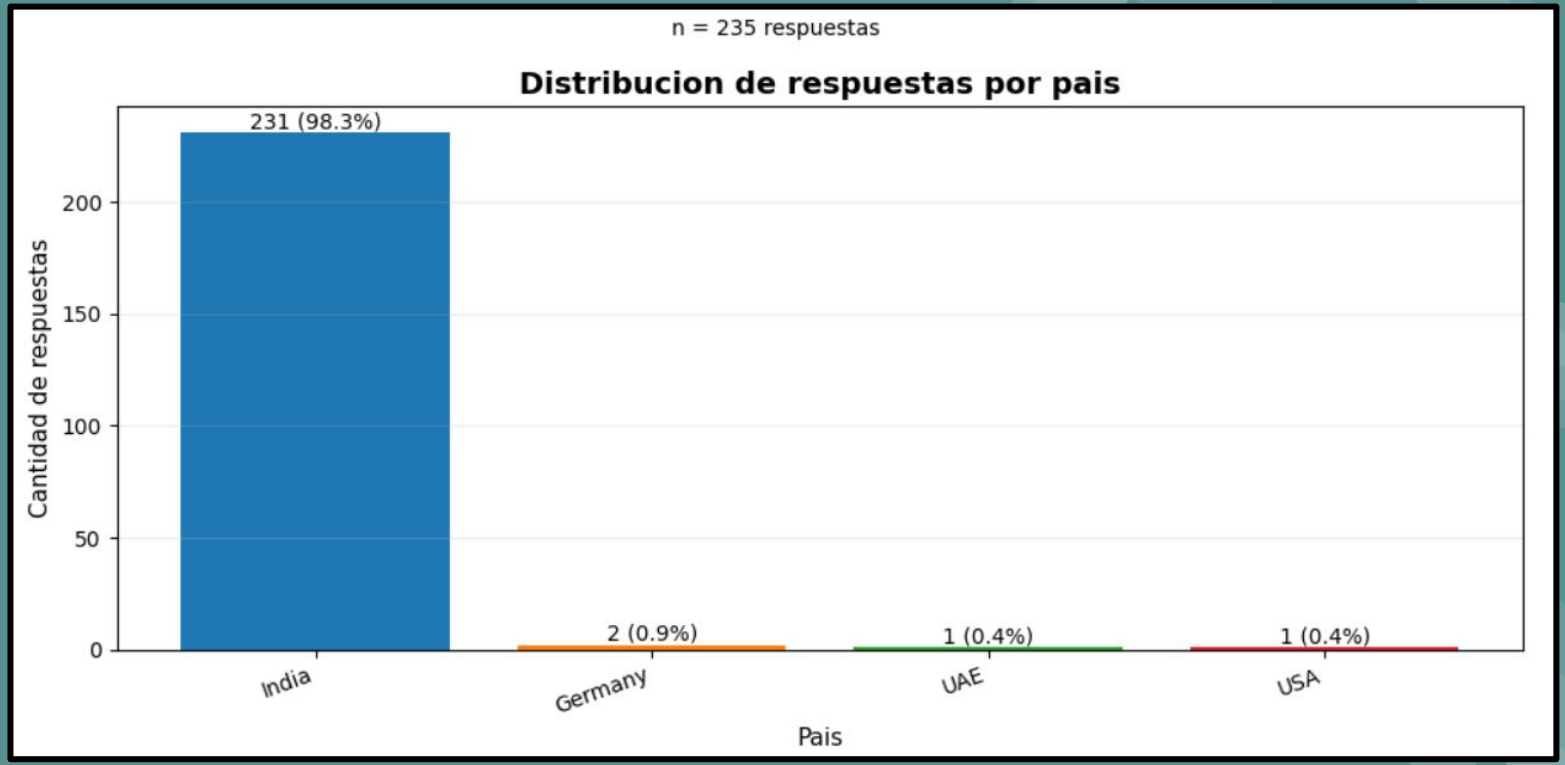
- Preferencia de ambiente laboral de Gen Z
- Dos resultados:
  - **REMOTE** (incluye híbrido y full remote)
  - **FULL_OFFICE** (oficina todos los días)

¿Para qué sirve?

- Apoyar decisiones de políticas de trabajo y atracción de talento

Dataset

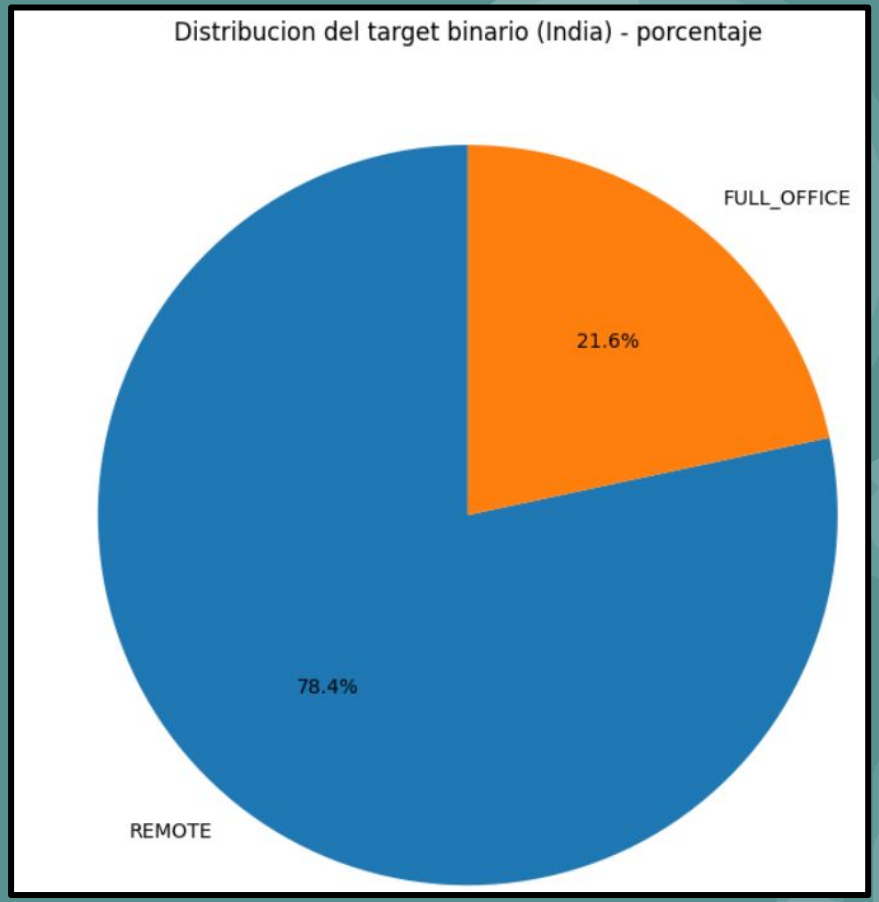
Fuente: Understanding career aspirations of GenZ (KultureHire / Kaggle)
Tamaño: 235 respuestas
Dato clave: ~98% de respuestas provienen de India (limitación de generalización)



Definición del target

Target original: 6 categorías de “working environment”
Nuestra definición binaria:

- FULL_OFFICE = “Every Day Office Environment”
- REMOTE = resto (Hybrid + Fully Remote)



Variables usadas

12 variables de entrada (11 categóricas + 1 numérica)
Ejemplos:

- País, género
- Preferencias de aprendizaje, tipo de manager, setup del equipo
- Escala 0–10 sobre “misión sin impacto social”

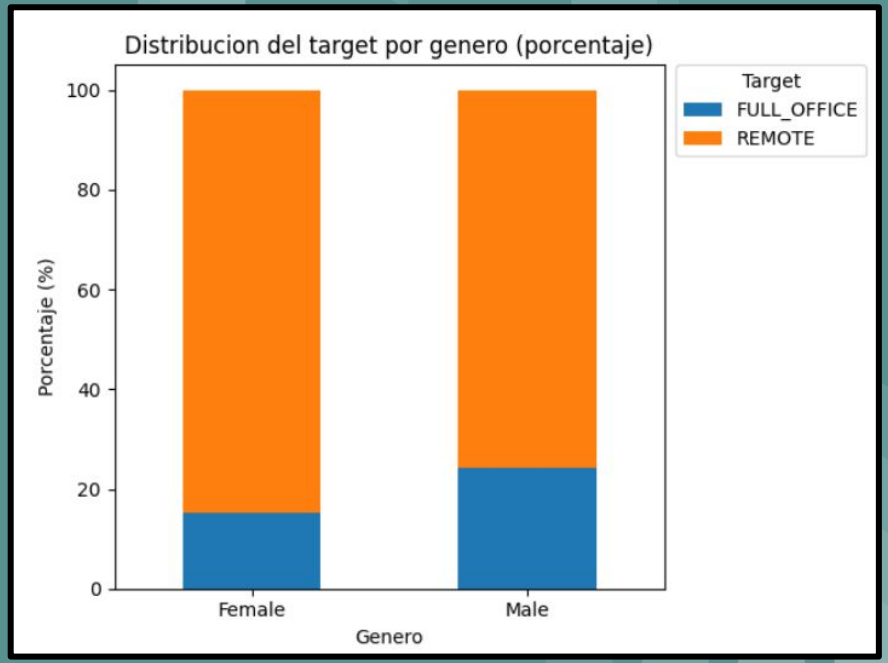
Metodología

Pipeline end-to-end

1. Limpieza y EDA
2. Preprocesamiento:
    - categorías raras → OTHER
    - OneHotEncoder
3. Entrenamiento:
    - Logistic Regression (tuned)
4. Evaluación (test)

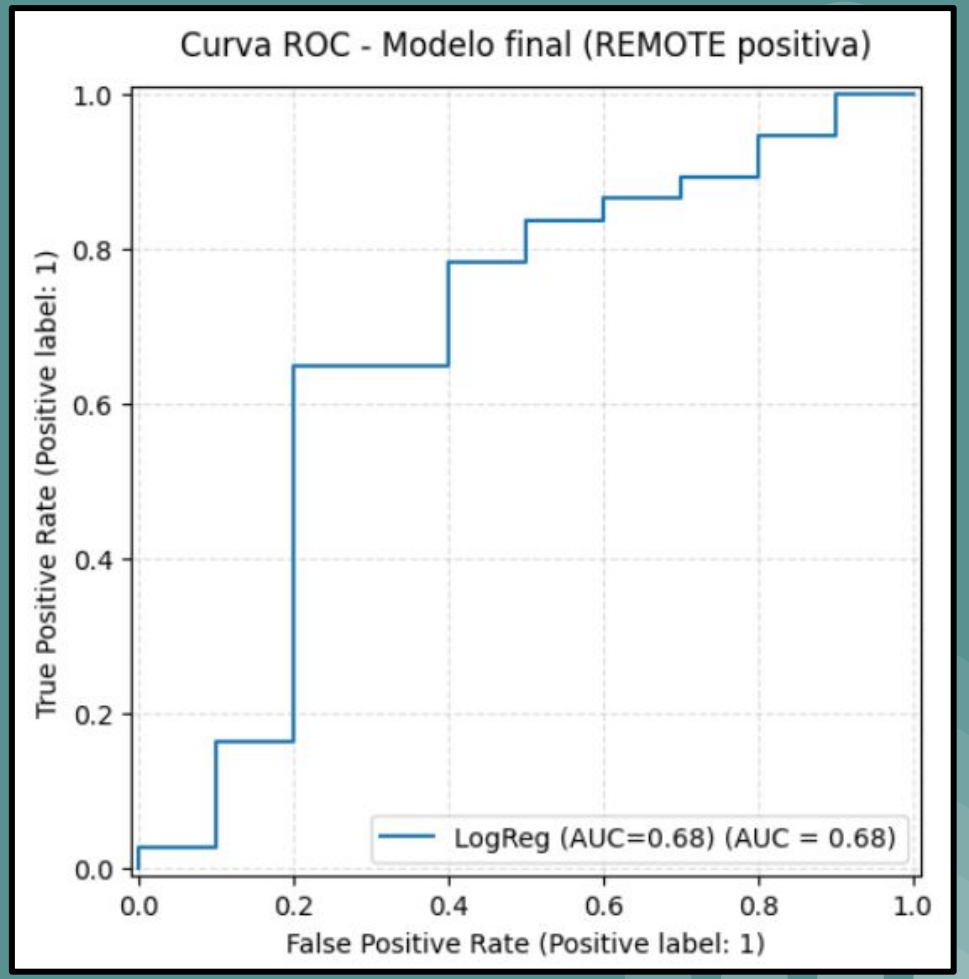
Distribución por género y preferencia

- Comparación del porcentaje de `REMOTE` vs `FULL_OFFICE` por género.
- En este dataset, **ambos géneros muestran alta preferencia por REMOTE**, con diferencias moderadas.
- Sirve para entender si la preferencia por remoto cambia según el perfil.



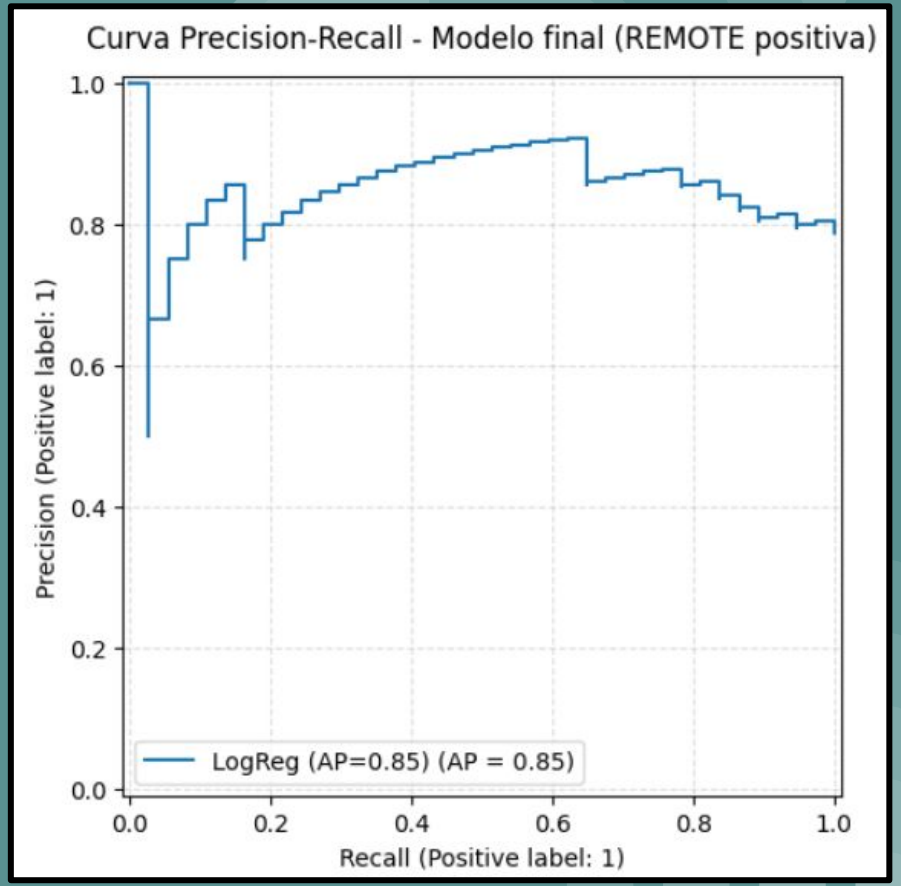
Curva ROC

- **ROC**: muestra el intercambio entre “falsos positivos” y “verdaderos positivos” al variar el umbral.
- **AUC (ROC)** resume el rendimiento general (más alto = mejor separación).



Precision-Recall

- **ROC**: muestra el intercambio entre “falsos positivos” y “verdaderos positivos” al variar el umbral.
- **AUC (ROC)** resume el rendimiento general (más alto = mejor separación).
- **Precision–Recall** es especialmente útil cuando hay **clase minoritaria** (aquí: `FULL_OFFICE`).
- **AP (Average Precision)** resume la curva PR (más alto = mejor balance precisión/recall).

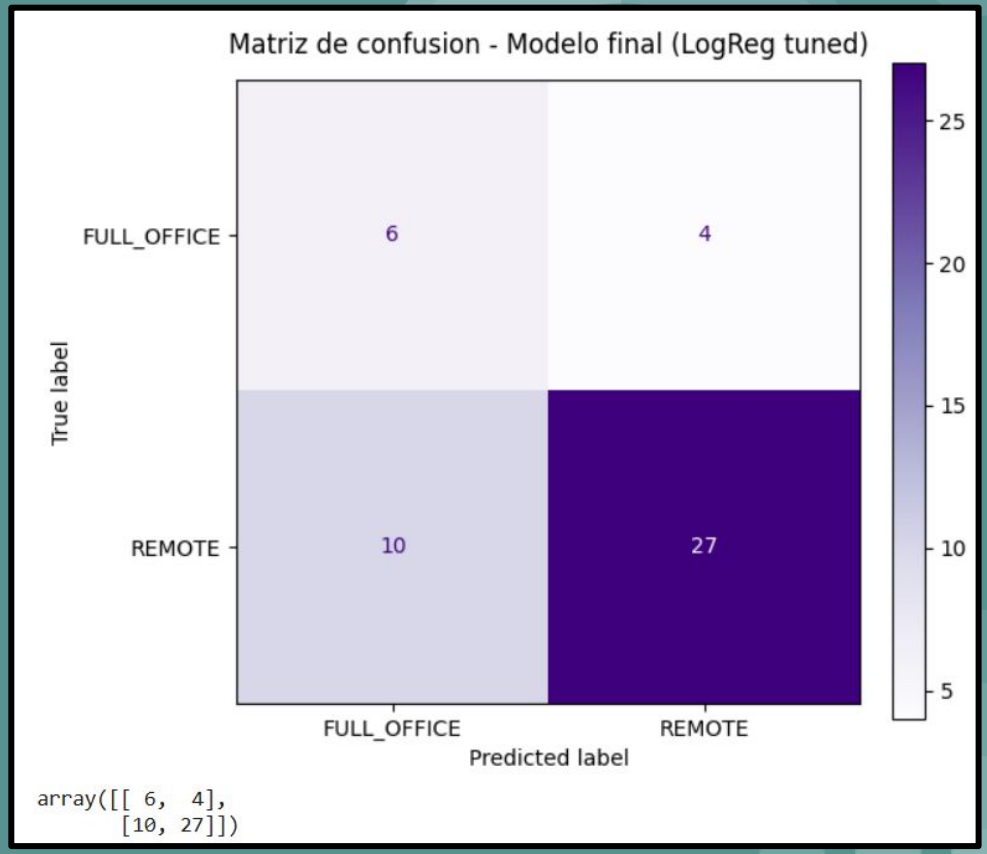


Resultados del modelo (métricas)

Accuracy (test): 0.7021
F1 Macro: 0.6278

Por clase (idea simple):

- REMOTE se predice mejor (clase mayoritaria)
- FULL_OFFICE es más difícil (clase minoritaria)



Tuning y ensambles

Qué se hizo para mejorar el modelo:

- Tuning de hiperparámetros en Logistic Regression
- Comparación con modelos base (ej.: RandomForest / GradientBoosting)
- Selección final por balance entre métricas (F1 macro + estabilidad)

Mensaje: intentamos reducir varianza y mejorar rendimiento antes de producción

API REST pública

Tecnología: Flask + ngrok
Endpoint: `POST /predict`
Respuesta: predicción + confidence

Conclusiones

- **El proyecto cumple un flujo end-to-end reproducible:** EDA → modelo → tuning/ensamble → API pública con confianza.

- **La principal limitación es la generalización:** la muestra está concentrada en India y `FULL_OFFICE` es minoritaria; mejorar esto requerirá más datos y/o estrategias de balanceo.
- Buen rendimiento en REMOTE, menor en FULL_OFFICE por desbalance.