



De mirar el pasado a  
anticipar el futuro



# Machine Learning con Python para la Toma de Decisiones Empresariales

Jorge Israel Frometa Moya

# ¿Por qué falla mi modelo? Diagnóstico y Confianza.

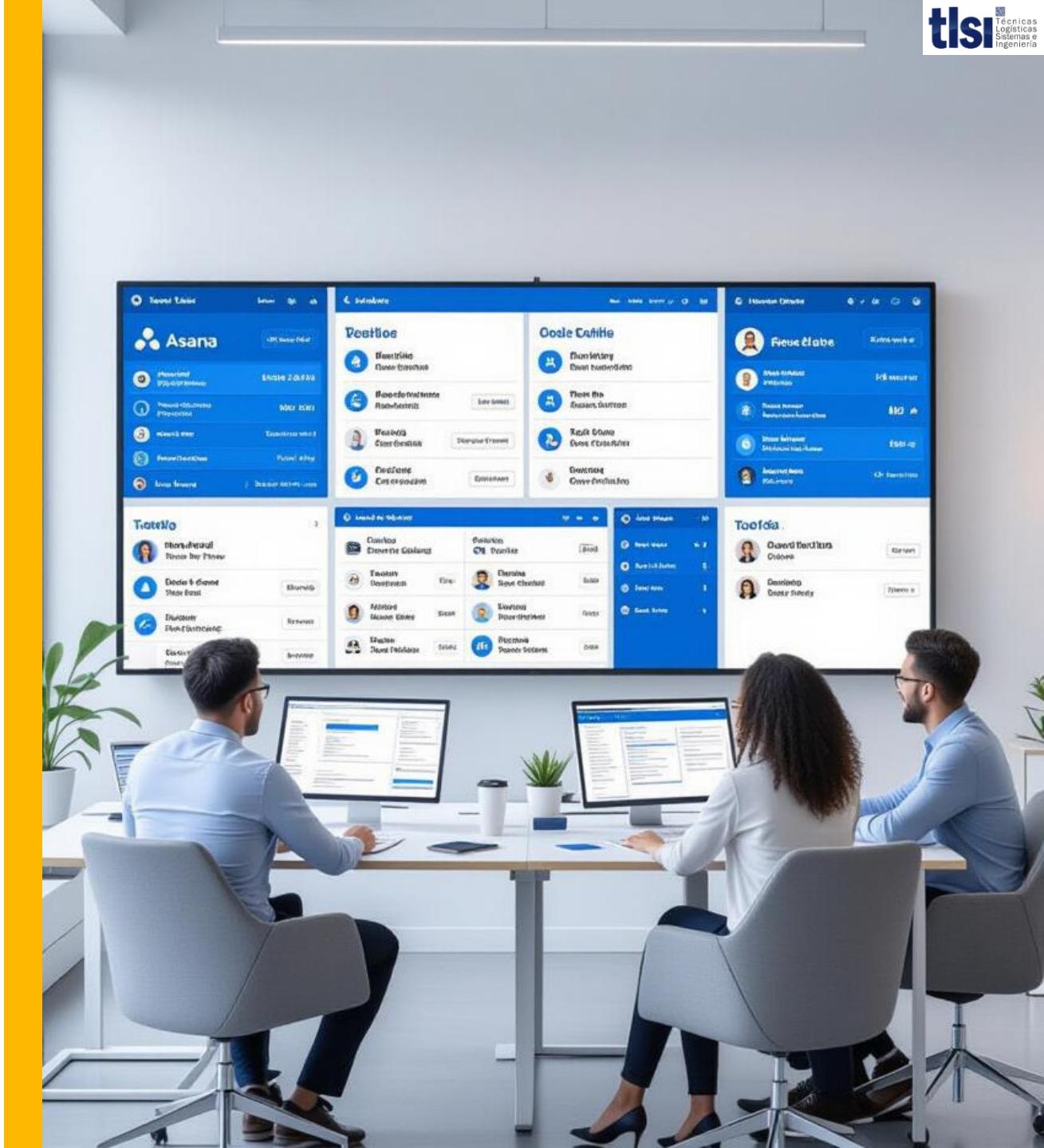


- Equilibrio entre simplicidad (Parsimonia) y complejidad.
- No queremos saber solo qué pasará, sino *por qué* pasará.

Un modelo que acierta todo en el pasado puede ser un desastre para predecir el futuro.

# Los tres escenarios del ML

- **Underfitting (Subajuste):** El estudiante no ha estudiado lo suficiente. No entiende los patrones ni en los apuntes ni en el examen. (Modelo demasiado simple).
- **Overfitting (Sobreajuste):** El estudiante se ha memorizado los apuntes de memoria. En el examen real, como las preguntas cambian un poco, fracasa. (Modelo demasiado complejo).
- **Modelo Robusto:** El estudiante entiende los conceptos y sabe aplicarlos a situaciones nuevas.



# ¿CONFIAR O NO CONFIAR EN EL MODELO?



## EL PROBLEMA DEL DIRECTOR DE MARKETING:

"El modelo predijo 95% de precisión en pruebas...  
...pero en la realidad, estamos perdiendo dinero."

## PREGUNTA CLAVE:

¿Cómo anticipar si un modelo funcionará EN LA REALIDAD  
antes de invertir miles de dólares?

# LA REGLA DE ORO: NO HAGAS TRAMPA

## ANALOGÍA:

- Entrenamiento = Estudiar con apuntes conocidos
- Test = Examen con preguntas nuevas

## ERROR COMÚN EN EMPRESAS:

Usar MISMO dato para "aprender" y "examinar"

→ Resultado: Modelos que solo funcionan en PowerPoint

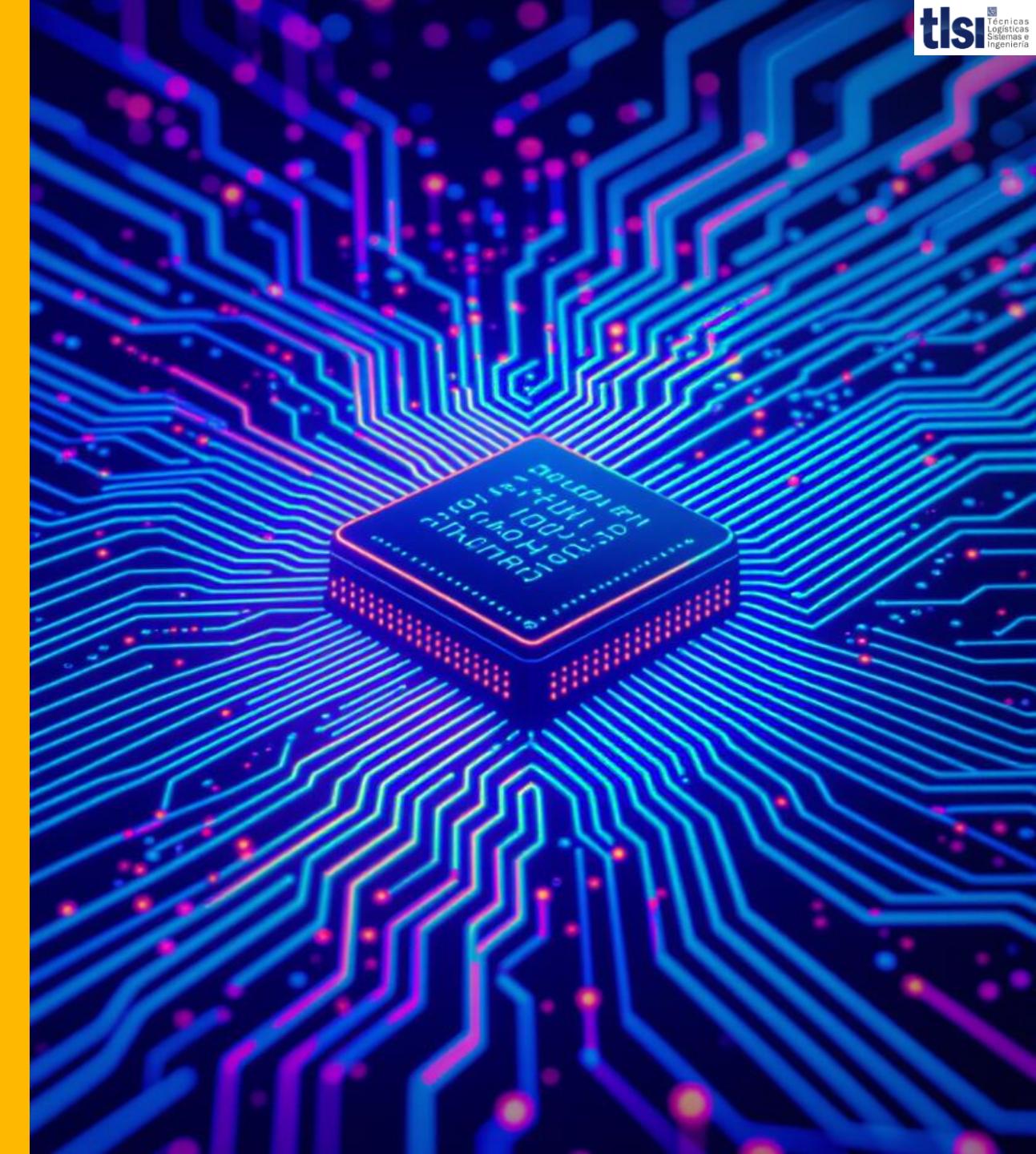
## PRÁCTICA ÉTICA:

80% para aprender patrones

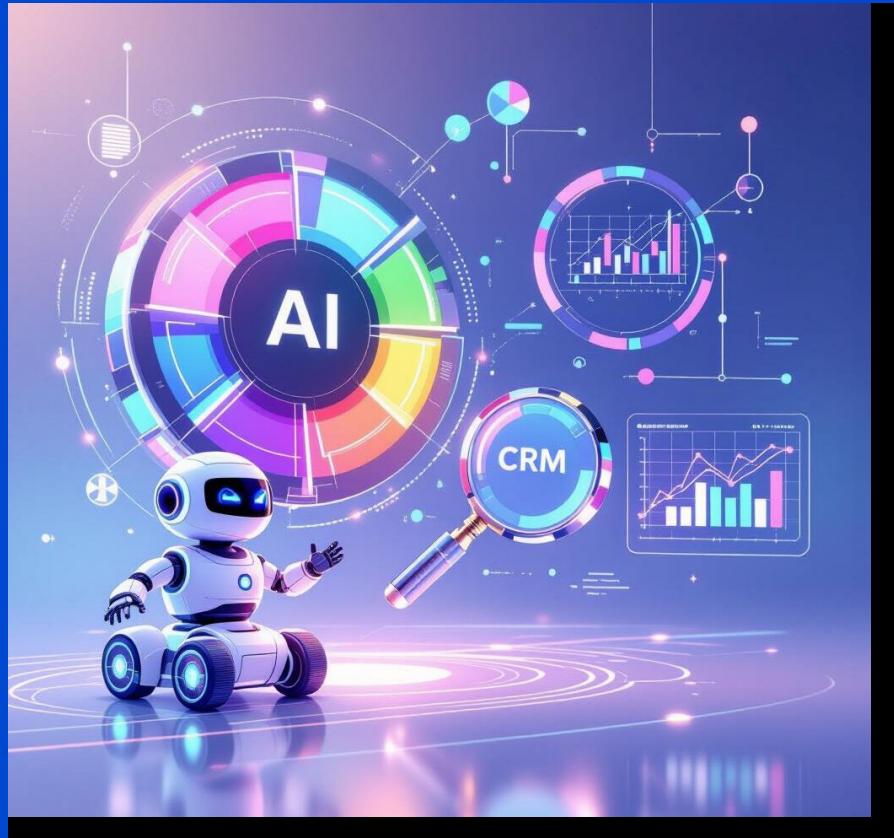
20% para probar en "mundo real simulado"

## PREGUNTA PARA DIRECTORES:

¿Preferirías saber que tu modelo falla AHORA...  
...o cuando ya invertiste \$500K?



# Advertencia temprana



## CÓMO LEER LOS RESULTADOS (SIN SER DATA SCIENTIST)

### SEÑAL DE UNDERFITTING:

"El modelo no entiende ni los datos de entrenamiento"

→ Acción: Añadir variables relevantes, modelo más complejo

### SEÑAL DE OVERFITTING:

"El modelo es genial en nuestros datos, péssimo en nuevos"

→ Acción: Reducir complejidad, más datos

### REGLA SENCILLA:

Si Train R<sup>2</sup> > Test R<sup>2</sup> por más de 0.1 → SOSPECHA

Si ambos son bajos (<=0.6) → PROBLEMA