🤖 Inteligencia Artificial Aplicada para la Economía

👨‍🏫 Profesores

Profesor Magistral

Camilo Vega Barbosa

Asistente de Docencia

Sergio Julian Zona Moreno

Fundamentos de Aprendizaje Supervisado

Una introducción a los algoritmos que aprenden de ejemplos etiquetados

🎯 Tipos de Aprendizaje Automático

🎓 Aprendizaje Supervisado

- Aprende de datos etiquetados
- Predice una salida específica

🔍 Aprendizaje No Supervisado

- Encuentra patrones sin etiquetas
- Agrupa datos similares

🤖 Aprendizaje Autosupervisado

- Genera sus propias etiquetas
- Aprende de contexto

🎮 Aprendizaje por Refuerzo

- Aprende por ensayo y error
- Optimiza recompensas

🎓 Aprendizaje Supervisado: ¿Cómo funciona?

📚 El aprendizaje supervisado es como enseñar con ejemplos. El modelo estudia casos donde ya conocemos la respuesta correcta y aprende a identificar patrones, similar a cómo aprendemos a distinguir objetos viendo muchos ejemplos.

🔑 La clave está en la calidad de los ejemplos. Cuantos más casos representativos procese el modelo, mejor será su capacidad para hacer predicciones precisas sobre nuevos datos.

📊 Ejemplo: Dataset de Animales

Los datos etiquetados son como una tabla donde cada fila es un ejemplo y cada columna una característica:

Peso (kg)	Altura (cm)	Tiene_Pelaje	Tiene_Plumas	Etiqueta
5.2	25	Sí	No	Gato
0.3	15	No	Sí	Pájaro
25.0	60	Sí	No	Perro
0.2	12	No	Sí	Canario
8.5	35	Sí	No	Gato

Tipos de Aprendizaje Supervisado

📊 Clasificación

- Objetivo: Predecir una categoría
- Salida: Valores discretos
- Ejemplos:
  - Riesgo crediticio (Alto/Bajo)
  - Fraude bancario (Sí/No)
  - Tipo de cliente (A/B/C)

📈 Regresión

- Objetivo: Predecir un número
- Salida: Valores continuos
- Ejemplos:
  - Precio de vivienda
  - Demanda de productos
  - PIB esperado

💡 Diferencia Clave:

Clasificación asigna categorías, regresión predice valores numéricos continuos

El Entrenamiento Supervisado 🧠

- 🎓 El aprendizaje supervisado utiliza algoritmos (como KNN, Random Forest o Regresión Logística) que aprenden de ejemplos etiquetados: si les mostramos suficientes casos con sus respuestas correctas, pueden predecir casos nuevos.
- 🤖 Por ejemplo, para predecir el precio de casas, alimentamos un algoritmo con miles de ejemplos:

```
Casa B: 80m², 2 habitaciones, suburbio → $150,000```
El algoritmo aprende los patrones que relacionan características con precios.
```

- 🎯 Este proceso tiene dos fases:

    i. **Entrenamiento**: El algoritmo aprende de muchos ejemplos

    ii. **Prueba**: Verificamos si aprendió bien con casos nuevos

El Proceso de Entrenamiento

🎯 Objetivo del Entrenamiento

- Aprender patrones generales
- Evitar memorizar ejemplos
- Poder predecir casos nuevos
- Encontrar relaciones útiles

⚠️ Retos Comunes

- Sobreajuste (memorizar)
- Subajuste (no aprender)
- Datos desbalanceados
- Ruido en los datos

💡 Analogía:

Es como estudiar con exámenes pasados, pero la prueba final tiene preguntas nuevas

Evaluación del Aprendizaje 📊

Así como un profesor necesita evaluar si sus estudiantes realmente aprendieron, nosotros necesitamos medir qué tan bien aprendió nuestro modelo. No basta con que memorice ejemplos - debe poder generalizar a situaciones nuevas.
Por eso dividimos nuestros datos y evaluamos el desempeño en ambos conjuntos (Entrenamiento y prueba). Esta evaluación nos ayuda a:

🎯 Saber si el modelo realmente está aprendiendo
🔍 Detectar problemas en el entrenamiento
⚖️ Comparar diferentes modelos objetivamente

La Analogía del Estudiante 👨‍🎓

📝 Evaluación en Entrenamiento:

Es como resolver ejercicios del libro

- El estudiante ya vio las respuestas
- Útil para práctica inicial
- No garantiza aprendizaje real

📋 Evaluación en Prueba:

- Es como el examen final
- Preguntas totalmente nuevas
- Mide verdadera comprensión
- Indica capacidad de generalizar

💡 Clave: Buen desempeño en ambas evaluaciones indica verdadero aprendizaje

Problemas Comunes en el Aprendizaje 🎯

📉 Subajuste (Underfitting):

- Mal desempeño en entrenamiento
- Mal desempeño en prueba
- No captura ni patrones básicos
- Es como estudiar muy poco

📈 Sobreajuste (Overfitting):

- Bien con datos de entrenamiento
- Mal con datos de prueba
- Memoriza en vez de aprender
- Su aprendizaje es límitado

🎯 Objetivo: Encontrar el balance perfecto entre memorización y generalización

La Matriz de Confusión

La matriz de confusión es una herramienta fundamental que nos ayuda a visualizar el desempeño de nuestro modelo. Es como una libreta de calificaciones que muestra no solo cuántas veces acertó nuestro modelo, sino también cómo se equivocó. Compara las predicciones del modelo con lo que realmente ocurrió, permitiéndonos entender sus fortalezas y debilidades.

	Predicción del Modelo	
Realidad	**Positivo**	**Negativo**
Positivo	Verdadero Positivo (VP)	Falso Negativo (FN)
Negativo	Falso Positivo (FP)	Verdadero Negativo (VN)

Métricas de Evaluación 📊

🎯 Medidas de desempeño

- **Accuracy:** Proporción de predicciones correctas sobre el total
- **Precision:** Proporción de verdaderos positivos entre todas las predicciones positivas
- **Sensibilidad (Recall):** Proporción de positivos reales correctamente identificados

📊 Ecuaciones

- **Accuracy** = $\frac{VP+VN}{VP+VN+FP+FN}$
- **Precision** = $\frac{VP}{VP+FP}$
- **Recall** = $\frac{VP}{VP+FN}$

Métricas de Evaluación 📊

🎯 Medidas de desempeño

- **Especificidad:** Proporción de negativos reales correctamente identificados
- **F1-Score:** Media armónica entre precision y recall

📊 Ecuaciones

- **Especificidad** = $\frac{VN}{VN+FP}$
- **F1-Score** = $2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$

💡 Nota:

El F1-Score es especialmente útil cuando las clases están desbalanceadas

Ejemplo Práctico: Matriz de Confusión

📊 Predicción de Pago de Préstamos (100 casos evaluados)

	Predicción del Modelo	
Realidad	**Pagará**	**No Pagará**
Pagó	VP = 30	FN = 10
No Pagó	FP = 10	VN = 50

💡 Interpretación:

- 30 clientes pagaron como se predijo (VP)
- 50 no pagaron como se predijo (VN)
- 10 pagaron cuando se predijo que no (FN)
- 10 no pagaron cuando se predijo que sí (FP)

Ejemplo Práctico: Evaluación del Modelo 📊

Calculemos las métricas principales:

1. **Accuracy (Exactitud)** 🎯

- Fórmula: $\frac{VP+VN}{Total} = \frac{30+50}{100}$
- Resultado: 80%
- Significado: El 80% de todas nuestras predicciones fueron correctas

2. **Precision (Precisión)** 📊

- Fórmula: $\frac{VP}{VP+FP} = \frac{30}{30+10}$
- Resultado: 75%
- Significado: Cuando predecimos que alguien pagará, acertamos el 75% de las veces

Ejemplo Práctico: Evaluación del Modelo 📊

3. **Recall (Sensibilidad)** 🔍

- Fórmula: $\frac{VP}{VP+FN} = \frac{30}{30+10}$
- Resultado: 75%
- Significado: Identificamos correctamente el 75% de todos los clientes que realmente pagaron

Interpretación Final 💡

- El modelo tiene un buen balance entre precision y recall
- Mantiene una exactitud general del 80%
- Hay espacio para mejora en la identificación de pagadores reales

Comparación Train vs Test: ¿Por qué es importante?

Para evaluar realmente si nuestro modelo está aprendiendo bien, debemos comparar su desempeño en ambos conjuntos: **Entrenamiento y Prueba**

Analogía del Estudiante 📚

- Entrenamiento es como practicar con ejercicios del libro
- Prueba es como resolver el examen final con problemas nuevos

Lo que buscamos 🎯

- Rendimiento similar en ambos conjuntos
- Diferencias pequeñas entre métricas
- Buena capacidad de generalización

Ejemplo: Evaluación en Ambos Conjuntos 📊

Entrenamiento (70 casos)

	Predicción	
Real	Pagará	No Pagará
Pagó	22	6
No Pagó	7	35

Accuracy: 81.4%

Prueba (30 casos)

	Predicción	
Real	Pagará	No Pagará
Pagó	8	4
No Pagó	3	15

Accuracy: 76.7%

Análisis de Resultados 💡

📊 Nuestro Modelo

- Accuracy entrenamiento: 81.4%
- Accuracy prueba: 76.7%
- Diferencia: ~5%
- Conclusión: Buen balance, sin sobreajuste

🎯 Balance Ideal

- Diferencia < 10% entre conjuntos
- Buen desempeño en ambos sets
- Métricas estables y consistentes
- Capacidad de generalización

Señales de problemas 💡

🚨 Señales de Sobreajuste

- Accuracy entrenamiento: 95%
- Accuracy prueba: 70%
- Diferencia: > 20%
- Problema: Memorización sin generalización

⚠️ Señales de Subajuste

- Accuracy entrenamiento: 35%
- Accuracy prueba: 33%
- Bajo desempeño en prueba y entrenamiento
- Problema: Aprendizaje insuficiente

Más Allá de la Matriz de Confusión 🎯

La matriz de confusión nos ha permitido entender el desempeño básico de nuestros modelos, pero en el mundo real necesitamos herramientas más sofisticadas. Imagina un médico que no solo quiere saber si su diagnóstico fue correcto o no, sino también qué tan seguro está de cada predicción.

🤔 ¿Por qué necesitamos más?

Nuestros modelos no solo dicen "sí" o "no" - asignan probabilidades a sus predicciones. Necesitamos métricas que capturen estos matices y nos ayuden a tomar mejores decisiones.

La Curva ROC: Conceptos Fundamentales 📈

La curva ROC (Receiver Operating Characteristic) visualiza el balance entre detectar correctamente los positivos (sensibilidad) y evitar falsos positivos (especificidad) a diferentes umbrales de decisión.

Componentes Clave

- **TPR (True Positive Rate)**: Sensibilidad = TP/(TP + FN)
- **FPR (False Positive Rate)**: 1 - Especificidad = FP/(FP + TN)
- **Umbral**: Punto de corte para clasificación binaria

Construcción de la Curva ROC 📊

Proceso Paso a Paso

1. Obtenemos probabilidades del modelo para cada instancia
2. Ordenamos las probabilidades de mayor a menor
3. Para cada umbral posible:
    - Calculamos TPR y FPR
    - Graficamos el punto (FPR, TPR)

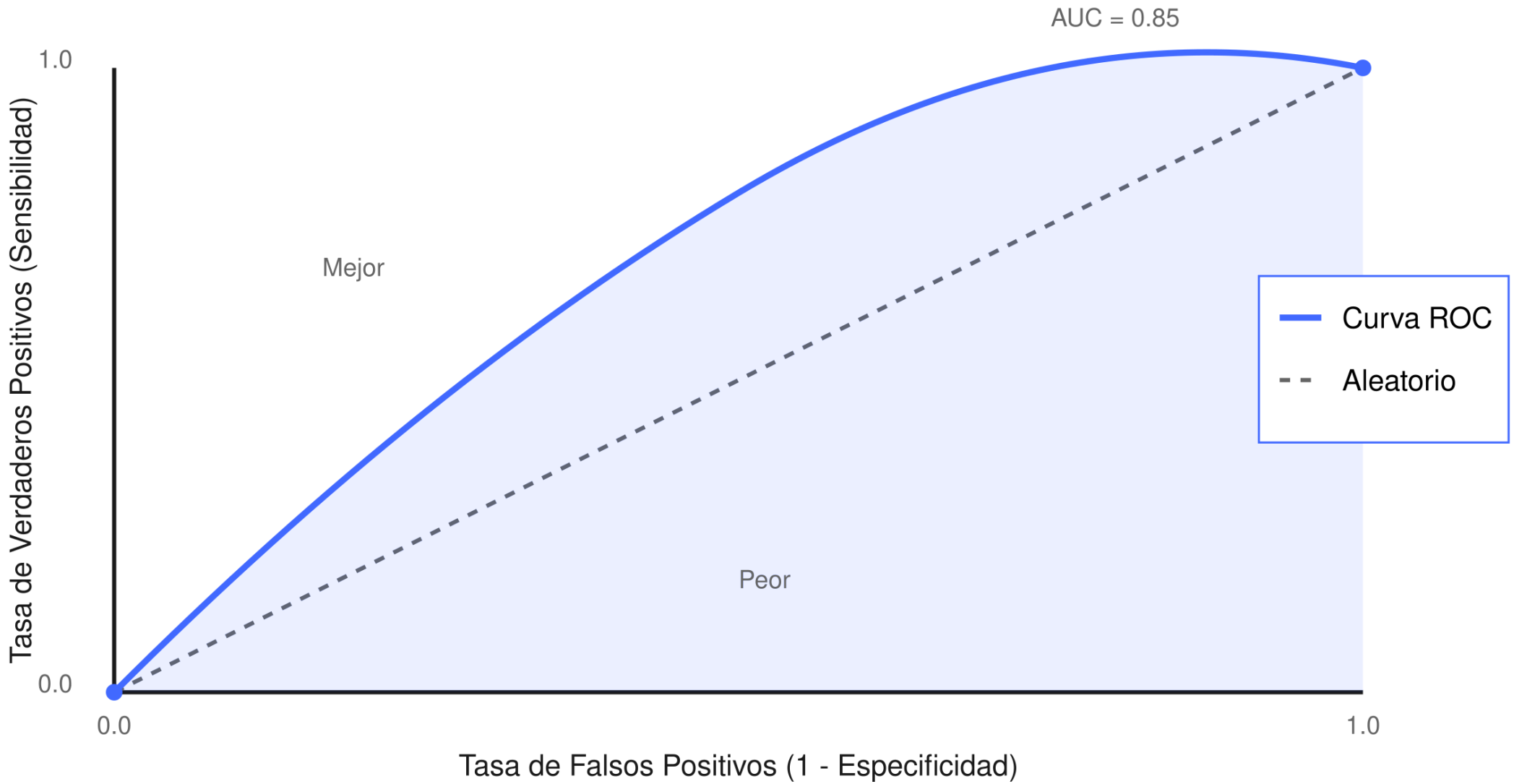
Cálculo del AUC

El AUC (Area Under the Curve) se calcula como:

$$AUC = \sum_{i=1}^{n-1} (x_{i+1} - x_i) \cdot \frac{y_i + y_{i+1}}{2}$$

Donde (x_i, y_i) son los puntos de la curva ROC.

Curva ROC y Área Bajo la Curva (AUC)

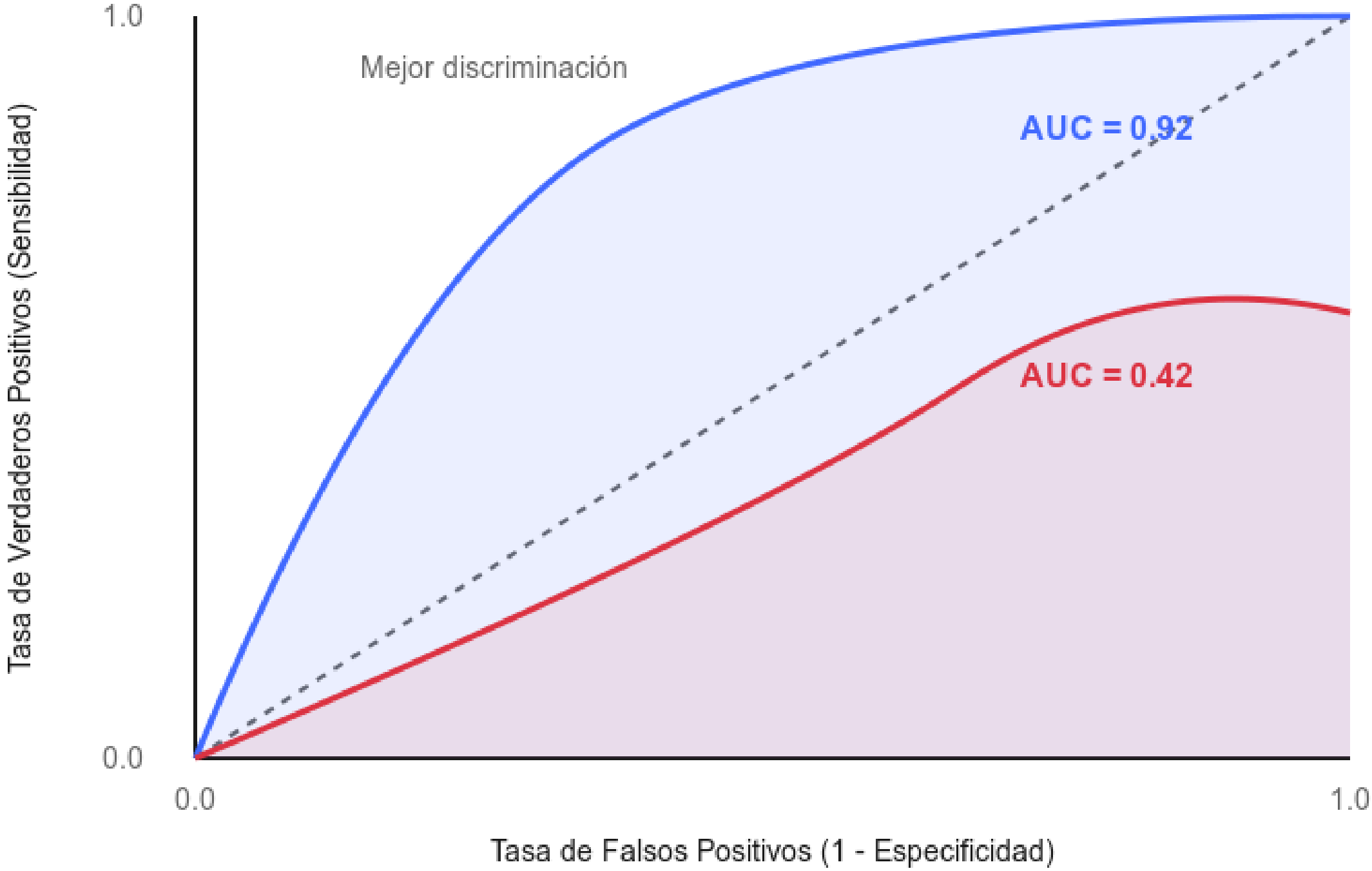


📈 Interpretación de ROC y AUC

La curva ROC visualiza el balance entre **detectar correctamente los positivos** (TPR) y **evitar falsos positivos** (1-FPR) a diferentes umbrales de decisión 🎯.

📊 Valores de AUC y su Significado

- **AUC = 1.0**: Clasificación perfecta ✨
- **AUC = 0.5**: Clasificación aleatoria (línea diagonal) 🎲
- **AUC < 0.5**: ¡No significa modelo inútil! 🔄


1.0
Mejor discriminación
AUC = 0,92
AUC = 0.42
Tasa de Verdaderos Positivos (Sensibilidad)
0.0
0.0
1.0
Tasa de Falsos Positivos (1 - Especificidad)

🔑 El Truco de la Inversión

La **forma de la curva** revela si el modelo necesita inversión 📐:

📈 **Curva cóncava hacia arriba (AUC > 0.5)**:

- Modelo clasifica **correctamente**
- Usar predicciones **tal como están** ✅

📉 **Curva cóncava hacia abajo (AUC < 0.5)**:

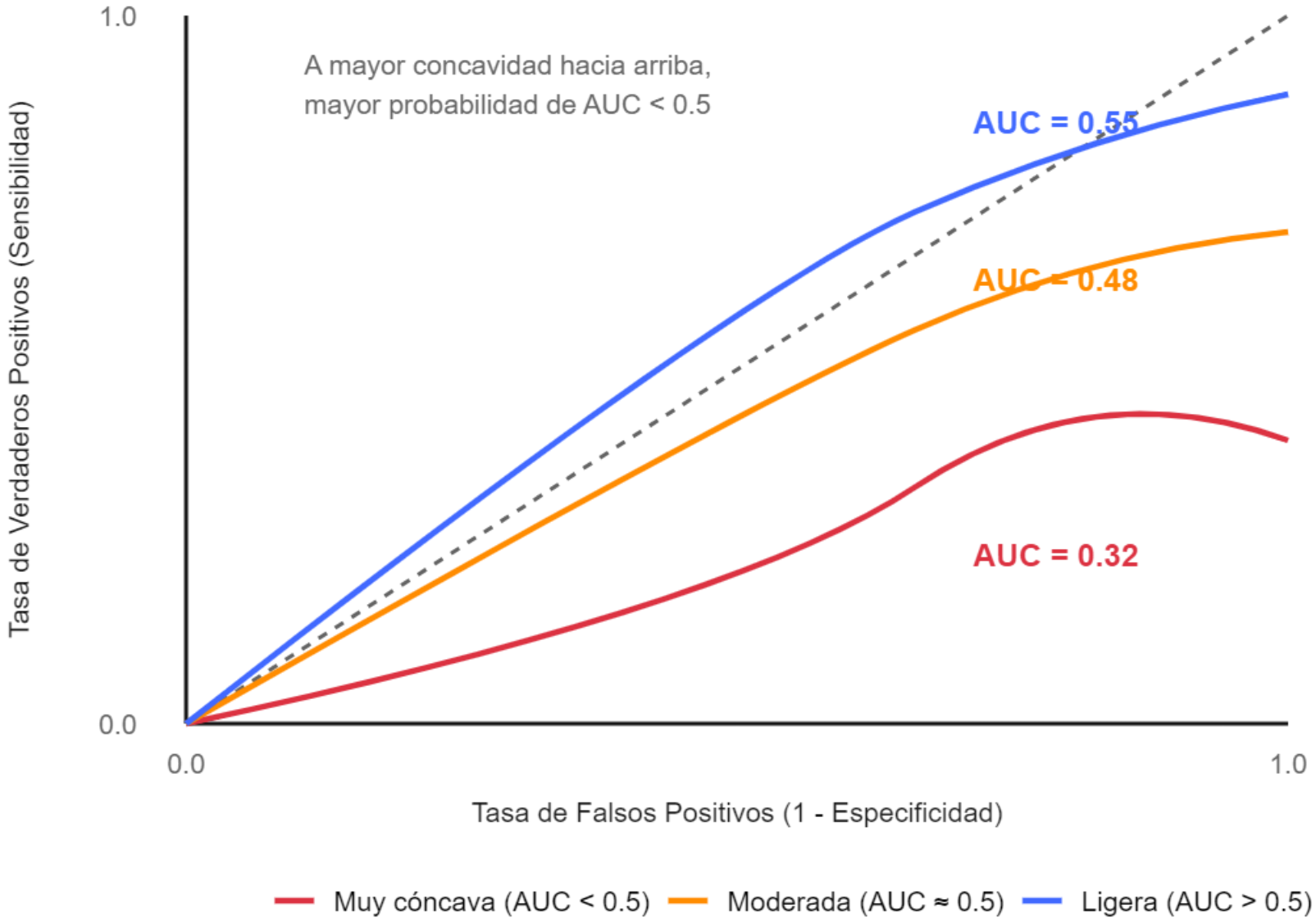
- Modelo tiene la lógica **invertida**
- **Invertir predicciones** para obtener AUC = 1 - AUC_original 🔄

💡 Ejemplo Práctico

Si AUC = 0.3 con curva cóncava hacia abajo → El modelo predice **incorrectamente** pero de forma **consistente**
Solución: **Invertir las predicciones** → AUC efectivo = **0.7** ✅

Curvas ROC con Concavidad Hacia Arriba

Diferentes grados de concavidad y su relación con el rendimiento aleatorio



Entropía y Log Loss 🎲

🚀 En el corazón de la teoría de la información, la entropía mide la incertidumbre. Cuando aplicamos este concepto al aprendizaje automático, obtenemos una forma más sofisticada de evaluar nuestros modelos.

🎯 Es sobre todo útil cuando debemos decidir entre un modelo u otro con indicadores similares de exactitud.

Entropía y Log Loss 🎲

Ejemplo Práctico

Dos modelos, ambos con 80% de exactitud:

```
# Modelo A: Predicciones consistentes
predicciones_A = [0.8, 0.8, 0.8, 0.8]

# Modelo B: Predicciones variables
predicciones_B = [0.99, 0.51, 0.99, 0.51]
```

Log Loss

$$H(y, p) = -\sum_i y_i \log(p_i)$$

- y_i = valor real (0 o 1)
- p_i = probabilidad predicha
- Menor valor = mejor modelo

Validación Cruzada: Más Allá del Train-Test Split 🔄

🏆 Dividir nuestros datos en entrenamiento y prueba es como juzgar a un estudiante por un solo examen. La validación cruzada es como tomar varios exámenes diferentes para tener una evaluación más completa.

Validación Cruzada 🔄

Proceso del K-Fold Cross Validation

1. Dividir datos en K partes
2. Entrenar con K-1 partes
3. Validar en parte restante
4. Repetir K veces
5. Promediar resultados

Tipos Principales

- **K-Fold Básico**: K particiones iguales
- **Estratificado**: Mantiene distribución
- **Leave-One-Out**: K = tamaño datos
- **Time Series**: Para datos temporales

💡 Ventaja Principal

Evaluación más robusta y confiable del modelo

⚖️ El Dilema Sesgo-Varianza

Uno de los conceptos más fundamentales en machine learning es entender el **trade-off** entre sesgo y varianza 🎯. Es como un arquero que debe equilibrar **precisión** con **consistencia** para dar en el blanco.

📉 Sesgo (Bias)

El **sesgo** representa el error sistemático del modelo - qué tan lejos están las predicciones de la realidad **en promedio** 📊. Un modelo con **alto sesgo** es como un arquero que siempre apunta hacia la izquierda del blanco:

- Predicciones **consistentemente** alejadas de la realidad
- Modelo **demasiado simple** para capturar la complejidad
- Resultado: **Subajuste** (underfitting) 📉

📈 Varianza

La **varianza** mide la sensibilidad del modelo a cambios en los datos 🔄. Un modelo con **alta varianza** es como un arquero nervioso con flechas muy dispersas:

- Predicciones cambian **dramáticamente** con nuevos datasets
- Modelo **demasiado complejo**, memoriza ruido
- Resultado: **Sobreajuste** (overfitting) 📈

🎯 Error Total = Sesgo² + Varianza + Ruido

El arte está en encontrar el **equilibrio perfecto**: modelos simples (alto sesgo, baja varianza) vs modelos complejos (bajo sesgo, alta varianza) ⚖️

🛠️ Recursos del Curso

📌 Plataformas y Enlaces Principales

📂 GitHub del curso

🔗 github.com/CamiloVga/IA_Aplicada

🤖 Asistente IA para el curso

🔗 Google Notebook LLM