

# SOL319 – Introducción al Aprendizaje de Máquinas para Ciencias Sociales

Primer semestre 2026

Pontificia Universidad Católica de Chile – Instituto de Sociología

## Equipo docente

<b>Profesor</b>	Mitchell Bosley <a href="mailto:mitchell.bosley@uc.cl">mitchell.bosley@uc.cl</a> Horario de consulta: por confirmar (probablemente 2 horas semanales los jueves)
<b>Ayudante</b>	Martín Roseblatt <a href="mailto:martinroseblatt@uc.cl">martinroseblatt@uc.cl</a>

## Información del curso

<b>Sigla</b>	SOL319
<b>Semestre</b>	2026-1
<b>Idioma</b>	Español (clases, módulos y evaluaciones en español; lecturas mayormente en inglés)
<b>Cátedra</b>	Lunes 11:00–13:30
<b>Ayudantía</b>	Miércoles 16:10–17:20
<b>Software</b>	R, RStudio, Quarto
<b>Canal oficial</b>	Canvas

### Fechas clave:

- Inicio de semestre: miércoles 4 de marzo.
- Primera cátedra: lunes 9 de marzo.
- Receso (sin clases): 18–23 de mayo.
- Fin de docencia: viernes 26 de junio.
- Evaluaciones finales: 30 de junio–10 de julio.

**Cambios a este programa.** Este syllabus puede actualizarse durante el semestre (por ejemplo, por ajustes de calendario o logística). Cualquier cambio se comunicará oportunamente por Canvas.

## Descripción y resultados de aprendizaje

### ¿De qué se trata este curso?

Las afirmaciones sobre IA ya están influyendo en decisiones de política pública, inversión, trabajo y educación. La pregunta del curso es simple: ¿qué haría falta para evaluar esas afirmaciones con seriedad? Eso exige pasar de claims amplios a preguntas medibles, con datos, protocolo de evaluación y límites explícitos.

Este curso enseña a hacer ese trabajo con aprendizaje de máquinas. El foco no está en perseguir el modelo más sofisticado, sino en construir análisis aplicados bien hechos: pipelines reproducibles, evaluación honesta y comunicación que no exagere lo que los datos muestran.

La primera mitad del semestre trabaja sólo con datos tabulares. Ahí armamos el pipeline completo, desde la pregunta hasta la evaluación y el reporte, partiendo por baselines simples y llegando a ensamblados. Recién después pasamos a texto, embeddings y modelos de lenguaje. La idea es llegar a esos temas con criterio, no sólo con recetas.

El curso tiene dos ejes que corren en paralelo todo el semestre:

1. **Métodos:** definir bien el problema, separar datos de entrenamiento y evaluación, elegir métricas que reflejen los costos reales del error, no contaminar la evaluación, y comunicar resultados con límites explícitos.
2. **IA y sociedad:** los sistemas de IA son objetos de estudio sociológico (¿qué están haciendo en el trabajo, la educación, los medios?), instrumentos de medición (¿qué tan válidos y estables son?) y entornos donde la gente interactúa.

Para que el curso no se vuelva una introducción genérica a ML, volvemos una y otra vez a preguntas sobre trabajo, educación, instituciones, discurso y medición social. La idea no es aprender técnicas en abstracto, sino ver cómo cambian las decisiones metodológicas cuando la pregunta sustantiva importa.

### Rationale: de fundamentos a LLMs

El curso está diseñado como una rampa de entrada a sistemas modernos de IA desde sus fundamentos. Antes de hablar de modelos más complejos, hay que dominar habilidades portables: formular bien el problema, definir una variable objetivo, construir representaciones útiles y evaluar fuera de muestra con protocolos honestos.

Por eso la primera mitad del semestre consolida el criterio metodológico en datos tabulares: métricas coherentes con costos reales, detección de leakage, comparación contra baselines, análisis de errores y decisiones básicas de pipeline como missingness, codificación, escalamiento, umbrales, calibración y tuning. La segunda mitad reutiliza exactamente esa lógica en texto, embeddings y sistemas generativos.

El cierre con LLMs no cambia la exigencia central del curso. Cambia el tipo de evidencia: rúbricas,

estabilidad, comparación con evaluación humana y documentación de límites. La meta es salir pudiendo construir, criticar y auditar sistemas de IA con una base firme.

## Resultados de aprendizaje

Al terminar el curso, cada estudiante debería poder:

1. Tomar una pregunta sociológica y traducirla a una tarea de aprendizaje de máquinas (definir unidad de análisis, variable objetivo, características).
2. Armar un pipeline reproducible en R: preparar datos, manejar decisiones básicas de preprocesamiento, ajustar modelos, evaluar y reportar.
3. Evaluar modelos con protocolos apropiados: splits, validación cruzada, métricas relevantes, umbrales/calibración cuando corresponda, y detectar leakage.
4. Comparar baselines con modelos más complejos y justificar cuándo la complejidad adicional vale la pena.
5. Hacer análisis de errores y sensibilidad, e interpretar resultados sin saltar a conclusiones causales.
6. Diseñar una evaluación básica de un sistema generativo (tipo LLM): rúbricas, estabilidad de respuestas, comparación con evaluación humana, y documentación de limitaciones.

## Prerrequisitos

SOL114 y SOL209. En la práctica, lo que necesitan es experiencia con regresión lineal y logística, y manejo básico de R. Si tienen dudas sobre si están preparados/as, escríbanme.

## Materiales

Los textos principales son gratuitos y en línea:

- James et al., *An Introduction to Statistical Learning*, 2a ed. (ISLR): [https://hastie.su.domains/ISLR2/ISLRv2\\_website.pdf](https://hastie.su.domains/ISLR2/ISLRv2_website.pdf)
- Wickham et al., *R for Data Science*, 2a ed.: <https://r4ds.hadley.nz/>
- Kuhn & Silge, *Tidy Modeling with R*: <https://www.tmr.org/>
- Silge & Robinson, *Text Mining with R*: <https://www.tidytextmining.com/>

Además usamos dos ensayos de Dario Amodei (CEO de Anthropic) como textos de encuadre:

- Semana 1: *Machines of Loving Grace* (2024). <https://www.darioamodei.com/essay/machines-of-loving-grace>

- Semana 16: *The Adolescence of Technology* (2026).  
<https://www.darioamodei.com/essay/the-adolescence-of-technology>

Son textos de encuadre. Los usamos para practicar algo que hacemos todo el semestre: tomar una afirmación sobre IA y convertirla en algo que se pueda medir.

Durante las primeras cuatro semanas, la espina principal de lectura será *ISLR + R for Data Science + Tidy Modeling with R*. La idea es construir primero un pipeline completo y portable — pregunta, datos, modelo, evaluación, comunicación — antes de movernos a texto, embeddings y sistemas generativos.

Las lecturas específicas de cada semana están en el módulo semanal correspondiente en Canvas.

## Estructura semanal

Cada semana funciona así:

- 1. Leer antes de la cátedra.** Las lecturas de la semana están en el módulo semanal de Canvas. La cátedra del lunes asume que ya leyeron; es donde discutimos los conceptos y los conectamos con preguntas sociológicas.
- 2. Trabajar el módulo interactivo.** Entre el lunes y el miércoles, cada estudiante completa un módulo guiado con un agente de IA del curso. El módulo los pone a practicar lo que leyeron — preguntas, ejercicios, escenarios — y se adapta según cómo vayan respondiendo. Más detalle abajo.
- 3. Aplicar en el laboratorio (miércoles).** La ayudantía es práctica: implementan lo que aprendieron en R, trabajando con datos reales en notebooks de Quarto. Llegan habiendo leído e interactuado con el módulo; la sesión se dedica a implementar, depurar y resolver dudas sobre el material de la semana.

## Módulos interactivos: el agente de IA del curso

Cada semana incluye un módulo interactivo que trabajan con un agente de IA propio del curso. El agente conoce el contenido de la semana, las lecturas, y lo que necesitan manejar.

### ¿Cómo funciona?

Acceden al agente a través de un enlace en Canvas. El módulo de cada semana tiene una estructura definida — preguntas, ejercicios, escenarios — pero la interacción se adapta al ritmo de cada estudiante: si necesitan más ejemplos, el agente se los da; si ya dominan un concepto, avanzan. El objetivo es practicar los conceptos del curso con feedback inmediato y llegar mejor preparados/as a la ayudantía.

El tipo de actividades que van a encontrar en los módulos incluye:

- Tomar una afirmación sobre IA y traducirla a una pregunta medible.

- Elegir una métrica para un escenario y justificar por qué.
- Detectar leakage en un pipeline descrito.
- Evaluar si una interpretación de resultados es razonable o sobreinterpreta.
- Diseñar un protocolo de evaluación para un caso nuevo.

Esto es lo mismo que van a ver en los exámenes, así que los módulos son directamente preparación para ellos.

### ¿Qué puede ver el equipo docente?

El sistema registra la completitud de cada módulo (si lo terminaron y cuándo), que es lo que cuenta para la nota de completitud semanal. El equipo docente puede ver patrones agregados (por ejemplo, si un concepto está generando dificultades para varias personas) para ajustar la cátedra o la ayudantía. Por defecto, no se evalúa “la conversación” y no se usa el contenido de mensajes para calificar respuestas una por una.

### ¿Qué pasa si el agente se equivoca?

Es un modelo de lenguaje, y puede cometer errores. Si algo que dice el agente les parece raro o contradice lo que vieron en clase o en las lecturas, confíen en la clase y las lecturas. Traigan la duda a la cátedra o a la ayudantía — de hecho, identificar cuándo un modelo se equivoca es parte del punto del curso.

### Un apunte sobre el uso y fallas técnicas

Este módulo es un espacio de práctica. La evaluación se basa en completitud de la interacción, no en “tener la respuesta correcta a la primera”. Además, al ser una tecnología experimental, los servidores pueden fallar. Si la plataforma se cae o el agente deja de responder un martes en la noche, tomen una captura de pantalla, avísenle al ayudante, y no habrá penalización en su nota de completitud.

## Evaluación

Componente	Ponderación
Completitud semanal (módulos + labs)	10 %
Tareas individuales: A01–A04 (5 % cada una)	20 %
Prueba parcial (en clase)	20 %
Examen final (ventana oficial)	20 %
Proyecto final (póster + repositorio reproducible + componente oral)	30 %

**Complejidad semanal (10 %).** El módulo y el lab de cada semana se evalúan como completo/incompleto. La nota final de este componente es proporcional al porcentaje de ítems completados a tiempo. Pueden faltar a un par de semanas sin consecuencias reales.

**Tareas individuales (20 %).** Cuatro tareas a lo largo del semestre (ver descripciones abajo). Se evalúan con una lógica de complejidad exigente: código reproducible, artefacto completo, interpretación razonable. No se espera que produzcan el mejor modelo posible, sino que demuestren un pipeline limpio y conclusiones honestas.

**Prueba parcial y examen final (40 %).** Ambos son presenciales, individuales, y basados en casos: les presentamos una situación y les pedimos que tomen decisiones metodológicas y las justifiquen. Parte de cada examen incluye fragmentos de código en R que tienen que leer, interpretar y criticar — por ejemplo, identificar si un pipeline tiene leakage, si la métrica elegida tiene sentido para el problema, o si la interpretación de los resultados es razonable. No tienen que escribir código desde cero, pero sí entender qué hace y si está bien hecho. Pueden traer una hoja de apuntes.

**Proyecto final (30 %).** Se recomienda realizarlo en parejas, pero también se puede hacer de forma individual. Consiste en un póster académico y un repositorio reproducible sobre una pregunta sociológica. Se evalúa el artefacto final: reproducibilidad, protocolo de evaluación, análisis de errores y conclusiones proporcionales a la evidencia. Además, incluye un componente oral breve (conversación individual) para verificar comprensión y autoría.

## Escala de notas UC

La evaluación del curso utiliza la escala de notas UC de 1.0 a 7.0, donde 4.0 es la nota mínima de aprobación.

## Tareas individuales

**A01 – Pipeline completo** (entrega: 29 de marzo, 23:59). Primer ejercicio de punta a punta. Toman una pregunta con relevancia sociológica, trabajan con un dataset compartido, hacen un split correcto, y ejecutan una evaluación mínima fuera de muestra con un baseline sin modelo (clase mayoritaria). Lo que importa es hacerlo bien: splits apropiados, métrica razonable, sin leakage. El código debe correr de principio a fin sin errores.

**A02 – Toolkit supervisado** (entrega: 26 de abril, 23:59). Comparan un baseline con un modelo de ensamble usando el mismo protocolo de evaluación, agregan una exploración no supervisada (clustering o PCA), y escriben una interpretación breve que no sobreinterprete los resultados.

**A03 – Clasificación de texto** (entrega: 17 de mayo, 23:59). Construyen un clasificador de texto (TF-IDF o embeddings), lo evalúan con la misma disciplina que usaron en datos tabulares, y producen un artefacto de comunicación: una model card o un reporte corto de sensibilidad.

**A04 – Mini-auditoría de LLM** (entrega: 7 de junio, 23:59). Diseñan un protocolo de evaluación para un sistema tipo LLM: definen una rúbrica, corren un set de casos, comparan con una referencia humana mínima cuando haga sentido, y reportan sobre estabilidad y sensibilidad de las respuestas.

Esto los prepara directamente para el proyecto final.

Todas las tareas son individuales. Se evalúan con una lógica de completitud exigente: el código debe ser reproducible (correr de principio a fin sin errores), el artefacto entregado debe estar completo, y las preguntas de interpretación deben estar respondidas con criterio razonable. No se espera optimización ni resultados perfectos. Hacer estas tareas bien es la mejor preparación para los exámenes y el proyecto.

## Integridad académica y colaboración

**Regla general.** En este curso se espera trabajo propio y reproducible. Se permite discutir ideas, depurar y pedir ayuda, pero las entregas individuales deben reflejar el trabajo individual.

**Trabajo individual (A01–A04).** Pueden conversar sobre el *enfoque*, interpretar errores, y comparar resultados a nivel conceptual. No pueden compartir código, notebooks o respuestas escritas de manera que otra persona pueda copiarlas o adaptarlas directamente.

**Proyecto (parejas o individual).** Se permite colaboración plena dentro de la pareja. Cada integrante debe comprender el pipeline y las decisiones tomadas; eso se verifica en el componente oral del proyecto.

**Evaluaciones presenciales.** La prueba parcial y el examen final son individuales. No se permite acceso a herramientas de IA ni colaboración durante estas evaluaciones.

## Ajustes razonables, bienestar y accesibilidad

Si necesitan ajustes razonables por razones de accesibilidad, salud, o circunstancias significativas, avisen lo antes posible para coordinar un plan. La coordinación puede involucrar los canales institucionales de apoyo estudiantil de la universidad, cuando corresponda.

## Recalificación (regrade)

Si creen que una evaluación fue calificada incorrectamente, pueden solicitar recalificación por escrito vía Canvas dentro de 7 días corridos desde la publicación de la nota. La recalificación puede resultar en una nota mayor, igual o menor.

## Calendario de evaluaciones

Hito	Fecha
A01	29 de marzo
Prueba parcial (en clase)	20 de abril
A02	26 de abril
Propuesta de proyecto	3 de mayo
A03	17 de mayo
<i>Receso</i>	<i>18–23 de mayo</i>
A04	7 de junio
Borrador de póster	24 de junio
Examen final	Por confirmar (30 jun–10 jul)
Entrega final del proyecto	5 de julio

Todas las entregas son a las 23:59, vía Canvas.

## Proyecto final: póster + artefacto reproducible

El proyecto puede realizarse en parejas (recomendado) o de forma individual. Eligen una pregunta sociológica, la traducen a una tarea de aprendizaje de máquinas, construyen un pipeline reproducible, y comunican los resultados en un póster académico. El repositorio que entregan debe correr de principio a fin sin intervención manual.

### Hitos

#### Propuesta (3 de mayo)

Una página: pregunta, datos que van a usar, tarea de ML que van a implementar, y un bosquejo del plan de evaluación. No necesitan tener resultados — necesitan mostrar que el problema está bien definido y que los datos existen. El equipo docente da retroalimentación escrita.

#### Workshop 1: baseline y evaluación (semana 13)

Traen un pipeline funcionando con un baseline y un protocolo de evaluación. No importa si el modelo es malo; importa que la evaluación sea correcta. Sesión de trabajo en clase con retroalimentación directa.

#### Workshop 2: resultados y análisis de errores (semana 14)

Traen resultados con al menos un modelo más allá del baseline, análisis de errores, y un borrador de las conclusiones. Discutimos qué pueden y qué no pueden afirmar con los resultados que tienen.

#### Borrador de póster (24 de junio)

Póster completo para retroalimentación de pares y del equipo docente. El formato específico (dimensiones, template) se publica en Canvas.

### Entrega final (5 de julio)

Póster definitivo + repositorio reproducible. El repositorio debe incluir:

- Datos (o instrucciones claras para obtenerlos).
- Código que regenere todos los resultados de principio a fin.
- Un README que explique cómo ejecutar el pipeline.

### Componente oral

Cada estudiante tiene una conversación individual con el profesor, donde les pido que expliquen decisiones del proyecto — por qué eligieron esa métrica, qué encontraron en el análisis de errores, qué limitaciones tiene su trabajo. Este componente forma parte de la evaluación del proyecto (ver sección “Evaluación”).

El estándar es el mismo que en las tareas, pero a mayor escala: código reproducible, evaluación honesta, comunicación que no sobreinterprete. Lo que hace un buen proyecto es que las decisiones estén justificadas, los errores estén analizados, y las conclusiones sean proporcionales a la evidencia.

### Equidad y contribución (parejas)

En proyectos en pareja, se solicitará una breve declaración de contribución (por ejemplo, 5–10 líneas) describiendo qué aportó cada integrante (datos, modelamiento, evaluación, escritura, figuras, etc.). La declaración complementa el componente oral y sirve para transparentar el trabajo realizado y facilitar retroalimentación.

### Programa semanal

Sem.	Tema	Entrega
1	Orientación: pipeline reproducible, predicción vs. explicación	Módulo + lab
2	Wrangling, visualización y formulación de problemas	Módulo + lab
3	Splits, validación cruzada, métricas, leakage y evaluación mínima	A01
4	Baselines: regresión lineal y logística + preprocesamiento básico	Módulo + lab
5	Ensamblados: bagging, boosting y selección de modelo	Módulo + lab
6	Clustering, PCA y representaciones	Módulo + lab
7	<b>Prueba parcial</b> (semana liviana)	Prueba parcial, A02
8	Texto I: TF-IDF como baseline	Propuesta de proyecto
9	Texto II: SVM y embeddings	Módulo + lab
10	Evaluación por subgrupos: fairness, costos y shift <i>Receso (18–23 de mayo)</i>	A03

Sem.	Tema	Entrega
11	¿Cómo funcionan los modelos generativos? (intuición)	Módulo + lab
12	Auditoría de LLM en práctica: rúbricas y estabilidad	A04
13	Workshop de proyecto: baseline y plan de evaluación	Avance
14	Workshop de proyecto: resultados y análisis de errores	Avance
15	Taller de póster y retroalimentación	Borrador de póster
16	Cierre y reflexión final	Entrega final

### Antes de la primera clase

La semana del 4 de marzo van a recibir un correo de bienvenida con instrucciones para instalar R, RStudio y Quarto. Si llegan a la primera cátedra con todo instalado y funcionando, la primera ayudantía va a ser mucho más productiva.

Las primeras cuatro semanas construyen el núcleo metodológico del curso: tomar una pregunta sociológica, traducirla a una tarea medible, definir un protocolo de evaluación honesto, y producir un baseline reproducible que se pueda interpretar sin sobrealcances. Todo lo que viene después — ensambles, texto, embeddings, modelos generativos y auditoría de LLMs — reutiliza exactamente esa lógica.

### Semana 1 — Orientación y workflow reproducible (9–15 mar)

*Módulo (IA):* Diagnóstico de workflow (R/Quarto) y ejercicio afirmación → variable objetivo → evaluación (completitud).

*Ayudantía/Lab:* Setup, Quarto, proyectos, importación de datos y reporte reproducible.

#### Resumen:

- Predicción vs. explicación como mapa inicial del curso.
- Pipeline reproducible: rutas relativas, Quarto y outputs regenerables.
- Primer ejercicio del semestre: afirmación amplia → variable objetivo → evidencia medible.

*Lecturas:* Amodei (2024), *Machines of Loving Grace* (hojear); Breiman (2001), “Statistical Modeling: The Two Cultures”; ISLR caps. 1–2 (hojear); R4DS, capítulos de workflow, importación y scripts.

### Semana 2 — Wrangling, visualización y formulación de problemas (16–22 mar)

*Módulo (IA):* Formulación de problema (unidad de análisis, variable objetivo, características) y práctica de proxies.

*Ayudantía/Lab:* Limpieza, transformaciones y visualización exploratoria con el dataset compartido.

**Resumen:**

- Pregunta sociológica → unidad de análisis → variable objetivo → features.
- Medir bien importa más que simplemente “tener datos”.
- Wrangling y visualización funcionan como diagnóstico antes de modelar.

*Lecturas:* ISLR cap. 2 (statistical learning, hojear); R4DS, capítulos de visualización y transformación de datos; Molina & Garip (2019), “Machine Learning for Sociology”.

**Semana 3 — Splits, validación cruzada, métricas y leakage (23–29 mar)**

*Módulo (IA):* Selección de métricas bajo costos y detección de leakage en mini-casos (completitud).

*Ayudantía/Lab:* Split train/test + CV + métricas en tidymodels, con un ejemplo guiado de leakage.

**Resumen:**

- Evaluar fuera de muestra para estimar generalización.
- Elegir métricas que reflejen costos reales del error.
- Identificar leakage y dejar planteado que un promedio global puede ocultar fallas por subgrupos.

*Lecturas:* ISLR cap. 5 (resampling methods); TMwR, capítulos de resampling, métricas y train/test workflow.

*Entrega:* A01 (29 de marzo).

**Semana 4 — Baselines: regresión lineal y logística (30 mar–5 abr)**

*Módulo (IA):* Interpretación de baselines y escritura de conclusiones proporcionales a la evidencia (completitud).

*Ayudantía/Lab:* Regresión lineal y logística como baselines; comparación y evaluación fuera de muestra.

**Resumen:**

- Modelos simples como referencia mínima para comparar complejidad.
- Mismo split, misma métrica y misma preparación de datos para comparar de forma justa.
- Missingness, recodificación y escalamiento también forman parte del método.

*Lecturas:* ISLR caps. 3–4; TMwR, capítulos de modelado supervisado y workflows reproducibles.

**Semana 5 — Ensamblados: bagging y boosting (6–12 abr)**

*Módulo (IA):* Decisiones de modelamiento (RF/boosting) y cómo comparar contra un baseline con el mismo protocolo.

*Ayudantía/Lab:* Random forest y gradient boosting con resampling; comparación directa con baseline.

**Resumen:**

- Bagging y boosting como extensión natural del baseline.
- Si cambias el modelo, no cambies el protocolo.
- Tuning básico, umbral y calibración también requieren justificación.

*Lecturas:* ISLR cap. 8 (tree-based methods).

**Semana 6 — Clustering, PCA y representaciones (13–19 abr)**

*Módulo (IA):* Interpretación de clustering/PCA y checklist de decisiones que deben reportarse.

*Ayudantía/Lab:* Clustering y PCA: visualización de resultados y discusión de estabilidad.

**Resumen:**

- Clustering como segmentación exploratoria, no como “tipos reales”.
- Variables, distancia,  $k$  y seeds deben reportarse como decisiones del análisis.
- PCA como intuición geométrica para pensar representaciones y luego embeddings.

*Lecturas:* ISLR cap. 10 (PCA y clustering).

**Semana 7 — Prueba parcial (20–26 abr)**

*Módulo (IA):* Banco de casos de repaso para prueba parcial (optativo).

*Ayudantía/Lab:* Consultas y troubleshooting para A02; revisión de protocolos de evaluación.

**Resumen:**

- Prueba parcial el lunes; el resto de la semana se mantiene liviano.
- Repaso del hilo central: protocolo, métrica, baseline, comparación y análisis de errores.
- Ayudantía enfocada en cerrar A02 con reproducibilidad y evaluación coherentes.

*Lecturas:* Repaso dirigido: apuntes de Semanas 1–6 + blueprint del examen (Canvas).

*Entrega:* A02 (26 de abril).

**Semana 8 — Texto I: TF-IDF como baseline (27 abr–3 may)**

*Módulo (IA):* Representaciones de texto (tokenización, TF-IDF) y diseño de evaluación para clasificación de texto.

*Ayudantía/Lab:* Pipeline de texto con TF-IDF en R; entrenamiento y evaluación fuera de muestra.

**Resumen:**

- Texto → tokens → TF-IDF como baseline fuerte.
- Split, métricas y leakage también aplican en texto.
- La propuesta de proyecto se centra en pregunta, datos reales y plan de evaluación.

*Lecturas:* Silge & Robinson, *Text Mining with R* (capítulos de TF-IDF); Grimmer et al. (2022), *Text as Data* (Capítulo 1: Introduction).

*Entrega:* Propuesta de proyecto (3 de mayo).

**Semana 9 — Texto II: SVM y embeddings (4–10 may)**

*Módulo (IA):* Comparación de representaciones (TF-IDF vs. embeddings) y criterios de selección de modelo (SVM) bajo el mismo protocolo.

*Ayudantía/Lab:* SVM para texto y primera transición a embeddings como features; comparación de desempeño y errores.

**Resumen:**

- SVM como contraste breve y competitivo para texto.
- Embeddings como features bajo el mismo protocolo de evaluación.
- Puente conceptual entre PCA, geometría y representaciones densas.

*Lecturas:* ISLR cap. 9 (SVM, hojear); *Text Mining with R* (representaciones y modelos).

**Semana 10 — Evaluación por subgrupos: fairness, costos y shift (11–17 may)**

*Módulo (IA):* Diagnóstico por subgrupos; costos del error; sensibilidad a shift y checklist de reporte.

*Ayudantía/Lab:* Evaluación por subgrupos y análisis de errores; sensibilidad y comunicación (Quarto).

**Resumen:**

- Evaluar heterogeneidad y no quedarse sólo con el promedio.
- Métricas y umbrales dependen de costos del error.
- Reportar resultados, riesgos y límites por subgrupo.

*Lecturas:* Barocas, Hardt & Narayanan, *Fairness and Machine Learning* (secciones intro, hojear); Molnar, *Interpretable Machine Learning* (análisis de errores y fairness, hojear).

*Entrega:* A03 (17 de mayo).

## Receso (18–23 de mayo)

Sin clases.

## Semana 11 — ¿Cómo funcionan los modelos generativos? (intuición) (25–31 may)

*Módulo (IA):* Intuición de next-token prediction, muestreo (temperature) y por qué el pre-entrenamiento produce “capacidades”.

*Ayudantía/Lab:* Modelo de lenguaje juguete (bigram) y muestreo; paso a representaciones (embeddings) en texto.

### Resumen:

- Next-token prediction como idea organizadora.
- Embeddings y atención como intuición de contexto, sin implementación pesada.
- Muestreo y temperatura como puente hacia problemas de evaluación en sistemas generativos.

*Lecturas:* Jurafsky & Martin, *Speech and Language Processing* (secciones de modelos de lenguaje y transformers, hojear);

Vaswani et al. (2017), “Attention Is All You Need” (opcional).

## Semana 12 — Auditoría de LLM en práctica: rúbricas, estabilidad, evaluación humana y documentación (1–7 jun)

*Módulo (IA):* Diseñar rúbrica + set de casos; sensibilidad a prompts; checklist de documentación, límites y uso responsable.

*Ayudantía/Lab:* Mini-auditoría de LLM: correr casos, puntuar con rúbrica, analizar errores y reportar.

### Resumen:

- Construir rúbrica y set de casos con criterios explícitos.
- Revisar estabilidad, sensibilidad a prompts y trazabilidad.
- Comparar con juicio humano básico y documentar límites, riesgos y condiciones de uso.

*Lecturas:* Bender et al. (2021), “On the Dangers of Stochastic Parrots”; Liang et al. (2023), HELM.

*Entrega:* A04 (7 de junio).

**Semana 13 — Workshop de proyecto I (8–14 jun)**

*Módulo (IA):* Checklist de milestone 1: pregunta, datos, baseline, protocolo de evaluación y riesgos de leakage.

*Ayudantía/Lab:* Workshop: pipeline corriendo, baseline y plan de evaluación con feedback en vivo.

**Resumen:**

- Pipeline corriendo end-to-end con un baseline real.
- Leakage, proxies y decisiones de medición explicitadas.
- Feedback directo sobre repo y protocolo de evaluación.

*Lecturas:* Milestone 1 + checklist de reproducibilidad (Canvas).

**Semana 14 — Workshop de proyecto II (15–21 jun)**

*Módulo (IA):* Guía de análisis de errores, sensibilidad y redacción de conclusiones para el póster.

*Ayudantía/Lab:* Workshop: resultados, análisis de errores y revisión de límites del resultado.

**Resumen:**

- Al menos un modelo más allá del baseline, con comparación justa.
- Análisis de errores y una sensibilidad relevante del pipeline.
- Conclusiones proporcionales a la evidencia y límites explícitos.

*Lecturas:* Milestone 2 + guía de escritura de evaluación/métodos (Canvas).

**Semana 15 — Taller de póster y retroalimentación (22–28 jun)**

*Módulo (IA):* Prompts de crítica y checklist de póster (claridad, evidencia, límites, reproducibilidad).

*Ayudantía/Lab:* Taller de póster: feedback de pares y revisión de reproducibilidad del repositorio.

**Resumen:**

- Póster completo con pregunta, datos, método, evaluación y mensaje central.
- Revisión por pares enfocada en evidencia, claridad y límites.
- Checklist final de reproducibilidad antes de entregar.

*Lecturas:* Rúbrica del póster + guía de diseño (Canvas).

*Entrega:* Borrador de póster (24 de junio).

## Semana 16 — Cierre y reflexión final (29 jun–5 jul)

*Módulo (IA):* Reflexión final: afirmación → medición → evaluación y checklist de entrega final (completitud).

*Ayudantía/Lab:* Sesión de cierre: dudas finales, revisión de entregas y pasos de reproducibilidad.

### Resumen:

- Volver al hilo inicial con un lente más estricto de medición y evaluación.
- Cerrar sobre qué se puede medir, qué falta y qué límites permanecen.
- Entrega final: póster definitivo, repo reproducible y coordinación del componente oral.

*Lecturas:* Amodei (2026), *The Adolescence of Technology*.

*Entrega:* Póster definitivo + repositorio reproducible (5 de julio).

## Política de entregas atrasadas

### Entregas semanales (módulos, labs).

No se aceptan atrasadas — son de completitud y el plazo es parte del diseño. La nota final de completitud tolera algunas ausencias sin penalización significativa.

### Tareas individuales (A01–A04).

Ventana de gracia de 48 horas con un descuento de 20 %. Después de eso, no se aceptan salvo circunstancias excepcionales documentadas y coordinadas *antes* del plazo.

### Entregas del proyecto (propuesta, borrador, entrega final).

Los plazos son firmes porque dependen de retroalimentación del equipo docente. Si tienen un problema, avisen con anticipación.

## Uso de IA

Pueden usar herramientas de IA (ChatGPT, Claude, Copilot, etc.) para el trabajo semanal y el proyecto. De hecho, los módulos semanales están diseñados para que interactúen con un asistente de IA como parte del aprendizaje. No necesitan esconder que usan estas herramientas.

Hay dos restricciones:

1. **Exámenes:** el midterm y el final son presenciales, sin acceso a herramientas de IA. Estos son los momentos donde verificamos comprensión individual.
2. **Componente oral del proyecto:** la conversación sobre el proyecto es sin herramientas. Tienen que poder explicar qué hicieron y por qué lo hicieron.

En las tareas y el proyecto, lo que evaluamos es el *criterio*: ¿eligieron bien la métrica?, ¿el split es correcto?, ¿la interpretación es honesta? Si una herramienta de IA les generó el código pero ustedes no pueden explicar por qué el pipeline es correcto, eso se va a notar en el examen — donde van a tener que leer código R y argumentar si lo que hace es apropiado — y en el componente oral del proyecto.

Si una herramienta de IA contribuyó de forma importante a una entrega (por ejemplo, generó una función clave o sugirió una estrategia de modelamiento), pongan una nota breve diciendo qué usaron y para qué. Basta una línea al final del documento.

## Comunicación

Canvas es el canal oficial para instrucciones, ajustes de calendario y avisos. Si tienen dudas sobre una tarea o el contenido del curso, escriban por Canvas o vengan a horario de consulta. Para temas administrativos urgentes, mándenme un correo. En general, respondemos mensajes dentro de 2 días hábiles.

## Asistencia y participación

Se espera asistencia regular a la cátedra y, especialmente, a la ayudantía/lab (donde se implementan los métodos). Si no pueden asistir a una sesión, revisen el material en Canvas y coordinen con anticipación cuando necesiten apoyo para ponerse al día.