📚 Estadística Aplicada y Modelos de Regresión (2026)

Manual Maestro Unificado y Base de Conocimiento de Cátedra

Institución: Pontificia Universidad Católica Argentina (UCA)
Carreras: Licenciatura en Ciencia de Datos / Ingeniería en Inteligencia Artificial
Profesor Titular / Responsable: Prof. Ivan Gonzalo Vilaboa
Año Lectivo: 2026 — 2º Semestre (Versión Planificada a 13 Semanas)
Destino del Documento: Documento maestro unificado optimizado para indexación, generación de prompts y base de conocimiento en **Google NotebookLM** (`notebooklm.google.com`).

📑 Índice General del Documento Maestro

PARTE I: MARCO INSTITUCIONAL, PROGRAMA Y PLANIFICACIÓN

- 1.1 Programa Oficial de la Asignatura
- 1.2 Cronograma de 13 Semanas y Calendario de Entregas
- 1.3 Articulación Pedagógica: Cadenas de Markov y Modelado Dinámico de Agentes
- 1.4 Guía de Bibliografía y Accesos a Biblioteca Digital O'Reilly

PARTE II: COMPENDIO TEÓRICO Y GUÍAS DE ESTUDIO (UNIDADES 1 A 6)

- 2.1 Unidad 1: Inferencia, Estimación de Parámetros y Tamaño de Muestra
- 2.2 Unidad 1 (Complementario): Cadenas de Markov y Modelado Dinámico
- 2.3 Unidad 2: Contraste de Hipótesis y Análisis de Potencia
- 2.4 Unidad 2: Comparación de Poblaciones y ANOVA de 1 Factor
- 2.5 Unidad 2: Contrastes de Chi-Cuadrado (Bondad de Ajuste e Independencia)
- 2.6 Unidad 3: Regresión Lineal, Diagnóstico y Árbol de Decisiones Rápidas
- 2.7 Unidad 4: Regresión Logística, Odds Ratios y Clasificación Naive Bayes
- 2.8 Unidad 5: Árboles CART, Ensembles (Random Forest), K-Fold CV y Series de Tiempo
- 2.9 Unidad 6: Inferencia Causal, DAGs, Criterio Backdoor y Causal ML (DoWhy / EconML)

PARTE III: GUÍAS DE EJERCITACIÓN PRÁCTICA Y PROBLEMAS DE LIBRO RESUELTOS

- 3.1 Guía de Ejercitación 01: Probabilidad y Procesos Estocásticos
- 3.2 Guía de Ejercitación 02: Inferencia Estadística, Test de Hipótesis y ANOVA
- 3.3 Guía de Ejercitación 03: Regresión Lineal Múltiple, Diagnóstico de Residuos y PCA
- 3.4 Guía de Ejercitación 04: Regresión Logística y Clasificación Naive Bayes
- 3.5 Guía de Ejercitación 05: Árboles de Decisión, Random Forest y Series Temporales
- 3.6 Guía de Ejercitación 06: Inferencia Causal, DAGs y Estimación con DoWhy

PARTE IV: TRABAJOS PRÁCTICOS OFICIALES DE EVALUACIÓN CONTINUA

- 4.1 Trabajo Práctico Nº 1: Exploración de Datos, Probabilidad y Simulación
- 4.2 Trabajo Práctico Nº 2: Inferencia Estadística, Contraste de Hipótesis y ANOVA
- 4.3 Trabajo Práctico Nº 3: Regresión Lineal Múltiple, Residuos e Ingeniería PCA
- 4.4 Trabajo Práctico Nº 4: Regresión Logística y Clasificación Naive Bayes
- 4.5 Trabajo Práctico Nº 5: Árboles de Decisión, Random Forest y Series Temporales
- 4.6 Trabajo Práctico Nº 6: Inferencia Causal, DAGs y Causal ML

PARTE V: TRABAJO PRÁCTICO INTEGRADOR (PROYECTO CAPSTONE)

- 5.1 Enunciado del Proyecto Integrador End-to-End
- 5.2 Rúbrica Detallada de Calificación y Criterios de Evaluación

---

PARTE I: MARCO INSTITUCIONAL, PROGRAMA Y PLANIFICACIÓN

1.1 Programa Oficial de la Asignatura

CARRERA: Ingeniería en Inteligencia Artificial / Lic. en Ciencia de Datos	**CURSO LECTIVO:** 2026
ASIGNATURA: Estadística Aplicada y Modelos de Regresión	**CURSO:** 2° año – 2° semestre
DURACIÓN: Semestral	**Hs. TOTALES:** 64 Hs. Reloj Totales
SEMANAS: 13	**Hs. TEÓRICAS:** 26 Hs. Reloj Totales **Hs. PRÁCTICAS:** 26 Hs. Reloj Totales

PROFESOR PROTITULAR: A DEFINIR
PROFESORES ADJUNTOS: Ivan Gonzalo Vilaboa / A DEFINIR
PROFESORES ASISTENTES: A DEFINIR

1. OBJETIVOS DE LA ASIGNATURA

Que el alumno logre: * Consolidar los conceptos fundamentales de probabilidad, estadística descriptiva e inferencial mediante su aplicación computacional. * Aplicar pruebas de contraste de hipótesis paramétricas y no paramétricas en la resolución de problemas aplicados. * Desarrollar, interpretar y evaluar modelos de regresión lineal (simple y múltiple), regresión no lineal y logística. * Dominar técnicas avanzadas de modelado predictivo como árboles de decisión, bosques aleatorios (Random Forest) y series de tiempo. * Comprender los fundamentos de la inferencia causal y la aplicación de técnicas de Causal Machine Learning para distinguir causalidad de correlación. * Emplear lenguajes de programación (Python) y librerías específicas de ciencia de datos para el análisis estadístico, manipulación de datos, simulación de trayectorias e implementación de modelos. * (Opcional/ Complementario) Entender procesos estocásticos (Cadenas de Markov) aplicados a sistemas multiagente.

2. UNIDADES TEMÁTICAS

(Distribución aproximada: 30% Estadística Aplicada y Procesos Estocásticos, 70% Modelos de Regresión e Inferencia Avanzada)

Unidad 1. Probabilidad Computacional y Análisis Exploratorio Repaso de estadística descriptiva e inferencial. Probabilidad, probabilidad condicional y distribuciones teóricas más conocidas. Implementación de funciones estadísticas usando librerías de Python. Manipulación de datos y visualización con Pandas y Seaborn. (Nota: El estudio de Cadenas de Markov y procesos estocásticos de transición de estados se proporciona como material complementario para su vinculación con la asignatura de Teoría de Juegos y Sistemas Multiagente).

Unidad 2. Inferencia Estadística y Análisis de Varianza Pruebas de normalidad (Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling). Pruebas de bondad de ajuste (Chi-cuadrado). Test de hipótesis paramétricos (test t) y no paramétricos (Wilcoxon, test de la mediana). Comparación de medias univariado y multivariado (ANOVA / Diseño Completamente Aleatorizado). Supuestos del modelo: homogeneidad de varianzas (Levene) y gráficos de normalidad (Q-Q Plot).

Unidad 3. Modelos de Regresión Lineal y Preparación de Datos Covarianza y correlación. Modelo de regresión lineal simple: estimación, evaluación e interpretación. Regresión lineal multivariada. Ingeniería de características (Feature Engineering) y transformación de variables. Detección y tratamiento de valores atípicos (outliers) y datos faltantes (missing data). Reducción de dimensionalidad: Análisis de Componentes Principales (PCA).

Unidad 4. Regresión No Lineal y Modelos de Clasificación Conceptos de modelos no lineales. Regresión logística para problemas de clasificación binaria y multiclase. Interpretación de coeficientes y Odds Ratio. Métricas de evaluación. Clasificador probabilístico Naive Bayes. Aplicaciones prácticas (ej. análisis de riesgo, detección de condiciones clínicas).

Unidad 5. Modelos basados en Árboles y Series de Tiempo Algoritmos basados en árboles de decisión (Clasificación y Regresión). Métodos de ensamble: Bosques Aleatorios (Random Forest). Evaluación de la importancia de características (Feature Importance). Ajuste de hiperparámetros y técnicas de Validación Cruzada (Cross-Validation). Introducción al análisis predictivo de Series de Tiempo (Modelos ARIMA).

Unidad 6. Inferencia Causal Diferencia entre Correlación y Causalidad. Grafos Acíclicos Dirigidos (DAGs) para la representación estructural. Identificación de efectos causales. El rol de las variables confusoras (confounders). Criterios de ajuste (Backdoor y Frontdoor). Aplicación del enfoque causal a modelos de regresión. Nociones de Causal Machine Learning.

---

3. BIBLIOGRAFÍA

Bibliografía Obligatoria * James, G., Witten, D., Hastie, T., Tibshirani, R. An Introduction to Statistical Learning with Applications in Python. Springer (2023). * Walpole, R., Myers, R., Probabilidad y Estadística para Ingeniería y Ciencias, 9na edición, Pearson (2012). * Pearl, J., Glymour, M., Jewell, N. Causal Inference in Statistics: A Primer. Wiley (2016). * León-García, A. Probability, statistics, and random processes for electrical engineering, 4ta edición, Pearson (2019).

Bibliografía Complementaria * Montgomery, D. C. Design and Analysis of Experiments, 10ma edición, Wiley (2019). * Geron, A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3ra edición, O'Reilly Media (2022). * Facure, M. Causal Inference for The Brave and True (Libro abierto en formato Python / Jupyter Notebook). * Molak, A. Causal Inference and Discovery in Python, Packt Publishing (2023).

4. METODOLOGÍA DE ENSEÑANZA

- **Clases teóricas:** Presentación de los conceptos y fundamentos matemáticos/ estadísticos con enfoque práctico. Se fomenta el debate sobre la interpretación de resultados y la modelización estocástica.
- **Clases prácticas computacionales:** Uso intensivo de Python (NumPy, Pandas, Scikit-Learn, Statsmodels, Scipy, etc.) sobre Jupyter Notebooks/Colab. Resolución de guías y análisis de bases de datos reales.
- **Plataforma EVA / Aula Virtual:** Espacio para foros, entrega de ejercicios, cuestionarios de autoevaluación y material asincrónico.

5. FORMACIÓN PRÁCTICA

El enfoque es "Hands-on", estructurando la práctica en los siguientes ejes: * **TP 1:** Exploración de Datos, Probabilidad y simulaciones en Python. * **TP 2:** Test de Hipótesis y ANOVA aplicados a casos industriales o de ciencias de la salud. * **TP 3:** Regresión Lineal y Feature Engineering. * **TP 4:** Regresión Logística y Naive Bayes. * **TP 5:** Modelado con Árboles y Random Forest. * **TP 6:** Grafos Causales y Ajuste Backdoor.

Trabajo Práctico Integrador: A lo largo del semestre, los alumnos desarrollarán en equipos un flujo de trabajo completo de Machine Learning y Estadística Aplicada: desde el Análisis Exploratorio de Datos (EDA) y tratamiento de anomalías, hasta el modelado, ajuste de hiperparámetros mediante Validación Cruzada e interpretación causal de los resultados.

6. CRITERIO Y MODALIDAD DE EVALUACIÓN

- **Evaluación formativa:** Seguimiento continuo mediante ejercicios en clase, participación y cuestionarios de autoevaluación.
- **Examen Parcial:** Un examen integrador de la primera mitad de la materia (orientado fuertemente a inferencia, hipótesis, procesos markovianos y regresión lineal), con posibilidad de recuperación. Se aprueba resolviendo correctamente el 60%.
- **Trabajo Práctico Integrador:** Entrega y defensa oral grupal obligatoria del proyecto aplicado de modelos de regresión y clasificación.
- **Examen Final:** Evaluación escrita teórica-práctica de carácter integrador.

1.2 Cronograma de 13 Semanas y Calendario de Entregas

Período: 26 de Agosto de 2026 - 18 de Noviembre de 2026
Día y Horario: Miércoles (4 horas cátedra)
Modalidad: Presencial (Teórico-Práctico)

> [!NOTE] El tema de **Cadenas de Markov** se ha estructurado como material de estudio complementario y puente teórico para la asignatura de 4º año (Teoría de Juegos y Sistemas Multiagente), por lo que no se dictará sincrónicamente en clase, pero la bibliografía y notebooks de soporte estarán disponibles. Las últimas semanas del curso incluyen espacios dedicados a consultas y avances del Trabajo Práctico Integrador.

Planificación Semanal

Semana	Fecha	Unidad	Temario Teórico-Práctico (Notebooks)	Lecturas Recomendadas
1	26/08/2026	Unidad 1	**Análisis Exploratorio y Probabilidad** Repaso de estadística descriptiva. Manipulación de datos y visualización con Pandas y Seaborn. Conceptos fundamentales de probabilidad y variables aleatorias.	**[1]** Cap. 1 a 4 **[5]** Cap. 1
2	02/09/2026	Unidad 2	**Inferencia, Estimación y Teorema Central del Límite** Estimación paramétrica (puntual y por intervalos). Tamaños de muestra. Pruebas de normalidad (Shapiro-Wilk, Q-Q Plots).	**[1]** Cap. 8 **[5]** Cap. 2
3	09/09/2026	Unidad 2	**Test de Hipótesis y Análisis de Varianza (ANOVA)** Pruebas paramétricas (t de Student, F) y no paramétricas (Wilcoxon). Análisis de varianza (ANOVA de 1 factor) y comparaciones post-hoc (Tukey).	**[1]** Cap. 8 **[5]** Cap. 2
4	16/09/2026	Unidad 3	**Regresión Lineal Simple y Múltiple** Estimadores MCO, supuestos de Gauss-Markov. Bondad de ajuste (R^2, RMSE). 🛎️ **Entrega de TP 1 y TP 2**	**[5]** Cap. 3 **[6]** Cap. 1 y 2
5	23/09/2026	Unidad 3	**Diagnóstico de Modelos, Outliers y PCA** Análisis de residuos, multicolinealidad (VIF), valores atípicos y apalancamiento (Distancia de Cook). Ingeniería de características y Análisis de Componentes Principales (PCA).	**[4]** Cap. 3 **[6]** Cap. 3 y 4 **[7]** Cap. 3
6	30/09/2026	Unidad 4	**Regresión Logística y Naive Bayes** Clasificación binaria, función sigmoidea, interpretación de	**[4]** Cap. 8 **[7]** Cap. 2

Semana	Fecha	Unidad	Temario Teórico-Práctico (Notebooks)	Lecturas Recomendadas
			Odds Ratios. Clasificador Naive Bayes. Curvas ROC, matrices de confusión y métricas de rendimiento.	
7	07/10/2026	-	**EVALUACIÓN PARCIAL** Evaluación teórico-práctica presencial.	-
8	14/10/2026	Unidad 5	**Modelos Basados en Árboles, Ensembles y Series de Tiempo** Árboles de clasificación y regresión (CART). Random Forest y validación cruzada. Nociones de Series de Tiempo. 🕰 **Entrega de TP 3 y TP 4**	**[4]** Cap. 8 y 9 **[7]** Cap. 2 y 5
9	21/10/2026	Unidad 6	**Inferencia Causal I: Fundamentos y DAGs** Correlación vs Causalidad. Grafos Acíclicos Dirigidos (DAGs) y variables confusoras (confounders).	**[2]** Cap. 1 y 2 **[3]** Cap. 1 a 3
10	28/10/2026	Unidad 6	**Inferencia Causal II: Ajuste y Causal ML / Avances TP Integrador** Criterio Backdoor. Estimación de efectos con DoWhy/EconML. Taller de consultas y avances del TP Integrador.	**[2]** Cap. 3 a 6 **[3]** Cap. 4 a 6
11	04/11/2026	-	**Taller Práctico / Avances TP Integrador** Espacio para aplicar Inferencia Causal a los proyectos finales. Pruebas de refutación de modelos. 🕰 **Entrega de TP 5 y TP 6**	-
12	11/11/2026	-	**Cierre Técnico y Preparación de Defensas** Tutorías por grupo para el cierre de hitos del TP Integrador (Hito 3 y 4). Resolución de dudas y formato de reporte.	-

Semana	Fecha	Unidad	Temario Teórico-Práctico (Notebooks)	Lecturas Recomendadas
13	18/11/2026	-	**RECUPERATORIO / DEFENSA TP INTEGRADOR** Examen recuperatorio y defensa oral obligatoria del Proyecto Integrador. Cierre formal de cursada.	-

Fechas Límite de Entregas (Deadlines):

- **Miércoles 16/09 (Semana 4):** Entrega TP 1 y 2.
- **Miércoles 14/10 (Semana 8):** Entrega TP 3 y 4.
- **Miércoles 04/11 (Semana 11):** Entrega TP 5 y 6.
- **Miércoles 18/11 (Semana 13):** Entrega Final y Defensa Oral del Trabajo Práctico Integrador.

Detalle de Bibliografía y Capítulos Requeridos (Excluyendo Markov de obligatoria)

- **[1] León-García, A. (2019).** Probability, statistics, and random processes for electrical engineering (4th ed.). Pearson.
  - **Cap. 1 a 4:** Probabilidad, variables aleatorias.
  - **Cap. 8:** Inferencia estadística y contraste de hipótesis.
  - Nota: El Cap. 11 (Cadenas de Markov) se asigna como lectura complementaria.
- **[2] Ness, R. O. (2025).** Causal AI. Manning Publications. (Disponible en O'Reilly UCA)
  - **Cap. 1 a 6:** Fundamentos, DAGs, ajuste, y Causal Machine Learning.
- **[3] Molak, A. (2023).** Causal inference and discovery in Python. Packt Publishing. (Disponible en O'Reilly UCA)
  - **Cap. 1 a 6:** Modelos Causales Gráficos, estimación en DoWhy y EconML.
- **[4] Han, J., Kamber, M., & Pei, J. (2012).** Data mining: Concepts and techniques. Morgan Kaufmann.
  - **Cap. 3, 8 y 9:** Limpieza, PCA, Árboles, Naive Bayes y Ensembles.
- **[5] Chan, Y. L., Badano, A., & Rey, M. (2020).** Análisis inteligente de datos con lenguaje R. edUTecNe.
  - **Cap. 1 a 3:** Análisis exploratorio, ANOVA y Regresión múltiple.

- **[6] Szretter Noste, M. E. (2017).** Apunte de regresión lineal. UBA - FCEyN.
  - **Cap. 1 a 4:** Modelo lineal, MCO, diagnóstico de residuos y puntos influyentes.
- **[7] Müller, A. C., & Guido, S. (2016).** Introduction to machine learning with Python. O'Reilly Media.
  - **Cap. 1 a 3 y 5:** Scikit-Learn, regresión, clasificación, PCA y GridSearch.

1.3 Articulación Pedagógica: Cadenas de Markov y Modelado Dinámico de Agentes

Uso previsto: Especificación de contexto y directiva para Agentes de IA en el diseño curricular y generación de contenidos.

---

1. Metadatos y Contexto Curricular

- **Asignatura Base (2º Año – 2º Semestre):** Estadística Aplicada y Modelos de Regresión (64 hs).
- **Asignatura Destino (4º Año – 1º Semestre):** Teoría de Juegos y Sistemas Multiagente (48 hs).
- **Cátedra:** Licenciatura en Ciencia de Datos / Ingeniería en Inteligencia Artificial – UCA.
- **Docente Titular / Responsable:** Prof. Ivan Gonzalo Vilaboa.
- **Objetivo de este Documento:** Definir la trayectoria de aprendizaje vertical donde las **Cadenas de Markov (CM)** actúan como el núcleo conceptual y computacional que conecta la inferencia probabilística con la Teoría de la Decisión, la Teoría de Juegos y la Simulación Basada en Agentes (ABM).

---

2. La Escalera Conceptual: De la Probabilidad a los Sistemas Multiagente

Los agentes de IA que generen material, ejercicios, evaluaciones o notebooks deben respetar y construir sobre los siguientes 4 niveles progresivos de modelado:

```
[Nivel 1: 2° Año - EAyMR]
Cadenas de Markov Estocásticas Puras:  P(S_{t+1} | S_t)
            |
            ▼  (Se agregan Acciones 'A' y Utilidad/Recompensa 'R')
[Nivel 2: 4° Año - Teoría de la Decisión]
Procesos de Decisión de Markov (MDP):  P(S_{t+1} | S_t, A_t)  [Juegos vs. Naturaleza]
            |
            ▼  (Se agregan Múltiples Jugadores con Interacción Estratégica)
[Nivel 3: 4° Año - Teoría de Juegos]
Juegos Estocásticos de Markov:         P(S_{t+1} | S_t, a_1, a_2, ..., a_N) → Nash
            |
            ▼  (Se escala a Poblaciones de Agentes Autónomos y Espacio)
[Nivel 4: 4° Año - Sistemas Multiagente / ABM]
Simulación Basada en Agentes:         Reglas de transición individuales → Comportamiento Emergente
```

---

3. Especificaciones Temáticas por Nivel

Nivel 1: Base en 2º Año (Estadística Aplicada y Modelos de Regresión)

- **Foco:** Entender la pérdida de memoria (propiedad de Markov), el álgebra de matrices estocásticas y la simulación discreta.
- **Conceptos Clave:**
  1. **Propiedad de Markov:** $P(X_{t+1} = s_j \mid X_t = s_i, \dots, X_0 = s_0) = P(X_{t+1} = s_j \mid X_t = s_i) = p_{ij}$.
  2. **Matriz de Transición (P):** Matriz estocástica por filas ($\sum_j p_{ij} = 1$).
  3. **Ecuaciones de Chapman-Kolmogorov:** Evolución temporal $\pi^{(t)} = \pi^{(0)} P^t$.
  4. **Distribución Estacionaria (π):** Vector de equilibrio invariante $\pi = \pi P$ tal que $\sum \pi_i = 1$.
  5. **Implementación en Python:** Simulación de trayectorias individuales con `np.random.choice` y resolución del sistema lineal ($\mathbf{P}^T - \mathbf{I})\boldsymbol{\pi}^T = \mathbf{0}$.

Nivel 2: Puente a la Teoría de la Decisión (Juegos contra la Naturaleza)

- **Foco:** El entorno no es estático: el decisor puede intervenir mediante acciones bajo incertidumbre.
- **Conceptos Clave:**
  1. Criterios de decisión bayesianos y valor esperado de la información.

2. Introducción a los **Procesos de Decisión de Markov (MDP)**: (S, A, P_a, R, γ).
3. Racionalidad acotada y árboles de decisión estocásticos.

Nivel 3: Teoría de Juegos (Juegos entre Jugadores)

- **Foco:** La transición del sistema ya no depende de un solo agente o del azar puro, sino de la acción combinada de agentes estratégicos.
- **Conceptos Clave:**
  1. Juegos de suma cero y suma no nula.
  2. Estrategias mixtas (distribuciones de probabilidad sobre acciones).
  3. Búsqueda de **Equilibrios de Nash** en juegos dinámicos y repetidos.

Nivel 4: Simulación Basada en Agentes (ABM)

- **Foco:** Dinámicas poblacionales complejas y propiedades emergentes.
- **Conceptos Clave:**
  1. Agentes como autómatas o máquinas de estado finito markovianas interactuando en red o grilla espacial.
  2. Transición micro $\rightarrow$ macro: cómo reglas de probabilidad locales generan equilibrios globales estables, ciclos o caos.

---

4. Matriz de Articulación para el Diseño de Actividades Prácticas

Hito / Capacidad Práctica	Requisito en 2º Año (EAyMR)	Continuación en 4º Año (Teoría de Juegos / ABM)
Modelado de Estados	Definir estados disjuntos (S) y armar la matriz P.	Definir el espacio de estados del agente y su matriz de pagos (payoff matrix).
Cálculo de Equilibrio	Calcular la distribución estacionaria analítica π.	Calcular el Equilibrio de Nash y distribuciones estacionarias poblacionales.
Simulación Computacional	Simular la trayectoria de 1 variable a lo largo de T pasos.	Simular N agentes simultáneos interactuando en un simulador ABM (Mesa / Python).
Toma de Decisiones	Inferencia condicional (Naive Bayes y probabilidades a posteriori).	Políticas de decisión óptimas $\pi^*(s)$ y aprendizaje por refuerzo (Reinforcement Learning).

5. Directivas para los Agentes de IA

Cuando los agentes reciban tareas vinculadas a este corpus: 1. **Evitar tratar a las Cadenas de Markov como un tema aislado:** Contextualizarlas siempre como la base matemática de los sistemas dinámicos y la toma de decisiones autónoma. 2. **Priorizar el enfoque constructivo / de simulación:** Además de resolver $\pi = \pi P$ analíticamente, incluir siempre la simulación algorítmica de trayectorias en Python. 3. **Mantener la consistencia notacional:** - $S = \{s_1, \ldots, s_K\}$ para estados. - $P = [p_{ij}]$ para la matriz de transición. - $\pi = [\pi_1, \ldots, \pi_K]$ para distribuciones de probabilidad sobre los estados. - $A = \{a_1, \ldots, a_M\}$ para el conjunto de acciones en etapas posteriores.

1.4 Guía de Bibliografía y Accesos a Biblioteca Digital O’Reilly

Este repositorio contiene los enlaces de acceso, repositorios de código oficiales descargados localmente y accesos a través del portal de **Biblioteca UCA / O’Reilly Learning**.

1. Libros Disponibles en el Portal O'Reilly de UCA

> **Portal UCA O'Reilly:** https://learning-oreilly-com.bibliotecauca.idm.oclc.org/home/
> (Requiere iniciar sesión con tus credenciales UCA)

Referencia	Título	Editorial	Acceso O'Reilly / Alternativa
[7] Müller & Guido (2016)	Introduction to Machine Learning with Python: A Guide for Data Scientists	O'Reilly Media	Buscar en O'Reilly UCA 💻 **Código/Notebooks locales:** `docs/Bibliografia/Muller_Guido_Intro_ML_Python`
[3] Molak, Aleksander (2023)	Causal Inference and Discovery in Python	Packt Publishing	Buscar en O'Reilly UCA 💻 **Código/Notebooks locales:** `docs/Bibliografia/Molak_Causal_Inference_Discovery`
[2] Ness, Robert O. (2025)	Causal AI	Manning Publications	Buscar en O'Reilly UCA 💻 **Materiales/Notebooks locales:** `docs/Bibliografia/Ness_CausalAI`
[4] Han, Kamber & Pei (2012)	Data Mining: Concepts and Techniques (3rd/4th ed.)	Morgan Kaufmann	Buscar en O'Reilly UCA 🌐 Slides oficiales UIUC (Jiawei Han)

2. Materiales en Acceso Abierto y Descargas Locales

Referencia	Título / Autor	Tipo / Formato	Ubicación / Enlace
[6] Szretter Noste, M. E. (2017)	Apunte de Regresión Lineal (UBA - FCEyN)	**PDF Completo Descargado**	📄 `docs/Bibliografia/Szretter_Noste_Apunte_Regresion_Lineal.pdf` (30.8 MB) 🌐 Enlace oficial UBA
[5] Chan, Badano & Rey (2020)	Análisis inteligente de datos con lenguaje R: con aplicaciones a imágenes	Repositorio Abierto UTN (RIA)	🌐 Repositorio RIA-UTN edUTecNe (ISBN: 978-987-4998-22-4)
[1] León-García, A. (2019)	Probability, statistics, and random processes for electrical engineering (3rd/4th ed.)	Libro de Texto (Pearson)	🌐 Disponible en biblioteca física UCA / Pearson.

3. Estructura de Carpetas Descargadas en este Repositorio

```
docs/Bibliografia/
├── GUIA_BIBLIOGRAFIA_Y_DESCARGAS.md
├── Szretter_Noste_Apunte_Regresion_Lineal.pdf       (PDF oficial completo UBA)
├── Muller_Guido_Intro_ML_Python/                    (Notebooks oficiales de O'Reilly)
│   ├── 01-introduction.ipynb
│   ├── 02-supervised-learning.ipynb
│   ├── 03-unsupervised-learning-preprocessing.ipynb
│   └── ...
├── Molak_Causal_Inference_Discovery/                (Notebooks oficiales de Packt)
│   ├── Chapter_01/ ... Chapter_14/
│   └── ...
└── Ness_CausalAI/                                   (Tutoriales y notebooks de Causal AI)
    ├── tutorials/
    └── ...
```

PARTE II: COMPENDIO TEÓRICO Y GUÍAS DE ESTUDIO (UNIDADES 1 A 6)

2.1 Unidad 1: Inferencia, Estimación de Parámetros y Tamaño de Muestra

Esta guía complementa la Unidad 1 y 2 de la materia, condensando los conceptos formales de inferencia univariada, propiedades de estimadores, intervalos de confianza y cálculo de tamaño muestral.

1. Conceptos Fundamentales de la Inferencia Estadística

La **Inferencia Estadística** es el proceso inductivo mediante el cual se extraen conclusiones sobre los parámetros de una **población** (θ) a partir de la información provista por una **muestra aleatoria** ($X_1, X_2, \ldots, X_n$).

- **Población:** Conjunto total de elementos bajo estudio (parámetros: media μ, varianza σ^2, proporción p).
- **Muestra:** Subconjunto representativo de tamaño n.
- **Estadístico (o Estimador):** Variable aleatoria función de la muestra que no depende de parámetros desconocidos ($\bar{X}$, S^2, $\hat{p}$).

1.1. Propiedades Deseables de un Buen Estimador

Sea $\hat{\theta}$ un estimador del parámetro poblacional θ:

1. **Insesgadez (Unbiasedness):** El valor esperado del estimador es igual al parámetro que estima:

   $$\mathbb{E}[\hat{\theta}] = \theta \iff \text{Sesgo}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta = 0$$

   Ejemplo: $\bar{X} = \frac{1}{n}\sum X_i$ es insesgado para μ. La varianza muestral corregida $S^2 = \frac{1}{n-1}\sum (X_i - \bar{X})^2$ es insesgada para σ^2.

2. **Eficiencia (Efficiency):** Dados dos estimadores insesgados $\hat{\theta}_1$ y $\hat{\theta}_2$, $\hat{\theta}_1$ es más eficiente si su varianza es menor:

   $$\mathbb{V}\text{ar}(\hat{\theta}_1) < \mathbb{V}\text{ar}(\hat{\theta}_2)$$

   El estimador de mínima varianza alcanza la cota inferior de Fréchet-Cramér-Rao.

3. **Consistencia (Consistency):** A medida que el tamaño de la muestra tiende a infinito, la probabilidad de que el estimador difiera del parámetro tiende a cero:

   $$\lim_{n \to \infty} P(|\hat{\theta}_n - \theta| > \epsilon) = 0$$

4. **Suficiencia (Sufficiency):** Un estimador es suficiente si resume toda la información que la muestra contiene acerca del parámetro θ (Criterio de factorización de Fisher-Neyman).

---

2. Intervalos de Confianza (Nivel de Confianza $1 - \alpha$)

Un intervalo de confianza al nivel $1 - \alpha$ garantiza que el $(1 - \alpha) \times 100\%$ de los intervalos construidos bajo muestreo repetido contendrán al verdadero parámetro poblacional.



2.1. Intervalo para la Media Poblacional (μ)

Caso A: Desvío poblacional σ CONOCIDO (o $n \geq 30$ por TCL)

Estadístico pivote: $Z = \frac{\bar{X} - \mu}{\sigma / \sqrt{n}} \sim \mathcal{N}(0, 1)$

$$IC_{1-\alpha}(\mu) = \left[\bar{X} - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \; ; \; \bar{X} + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \right]$$

Caso B: Desvío poblacional σ DESCONOCIDO (Población Normal, $n < 30$)

Estadístico pivote: $T = \frac{\bar{X} - \mu}{S / \sqrt{n}} \sim t_{n-1}$ (t de Student con $\nu = n - 1$ grados de libertad)

$$IC_{1-\alpha}(\mu) = \left[\bar{X} - t_{n-1,\, 1-\alpha/2} \frac{S}{\sqrt{n}} \; ; \; \bar{X} + t_{n-1,\, 1-\alpha/2} \frac{S}{\sqrt{n}} \right]$$

2.2. Intervalo para la Varianza Poblacional (σ^2)

Asumiendo población Normal con media μ desconocida. Estadístico pivote: $\chi^2 = \frac{(n-1)S^2}{\sigma^2} \sim \chi^2_{n-1}$

$$IC_{1-\alpha}(\sigma^2) = \left[\frac{(n-1)S^2}{\chi^2_{n-1,\, 1-\alpha/2}} \; ; \; \frac{(n-1)S^2}{\chi^2_{n-1,\, \alpha/2}} \right]$$

Para el desvío σ, se aplica raíz cuadrada a ambos límites.

2.3. Intervalo para la Proporción Poblacional (p)

Para un proceso de Bernoulli con tamaño muestral grande ($n\hat{p} \geq 5$ y $n(1-\hat{p}) \geq 5$):
Estadístico pivote: $Z = \frac{\hat{p} - p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \sim \mathcal{N}(0, 1)$

$$IC_{1-\alpha}(p) = \left[\hat{p} - z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \; ; \; \hat{p} + z_{1-\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right]$$

---

3. Determinación del Tamaño de Muestra (n)

El margen de error o semiancho del intervalo es $e = |\hat{\theta} - \theta|$.

3.1. Para Estimar la Media μ con Error Máximo e

Fijando un nivel de confianza $1-\alpha$ y un error máximo admisible e:

$$e = z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}} \implies n = \left(\frac{z_{1-\alpha/2}\cdot\sigma}{e}\right)^2$$

> 💡 **Nota práctica:** Si σ es desconocido, puede estimarse mediante una muestra piloto (S) o mediante la regla empírica del rango: $\sigma \approx \frac{\text{Rango}}{4}$ o $\frac{\text{Rango}}{6}$.

3.2. Para Estimar la Proporción p con Error Máximo e

$$e = z_{1-\alpha/2}\sqrt{\frac{p(1-p)}{n}} \implies n = \frac{z_{1-\alpha/2}^2\cdot p(1-p)}{e^2}$$

- **Caso Conservador (Varianza Máxima):** Si no se tiene información previa sobre p, se asume $p = 0.5$ (donde $p(1-p) = 0.25$ es máximo):

$$n_{\text{máx}} = \frac{z_{1-\alpha/2}^2\cdot 0.25}{e^2} = \frac{z_{1-\alpha/2}^2}{4e^2}$$

---

4. Implementación Rápida en Python

```python
import numpy as np
import scipy.stats as stats

# Datos muestrales
data = np.array([23.4, 25.1, 24.8, 26.2, 23.9, 25.5, 24.1, 25.0])
n = len(data)
x_bar = np.mean(data)
s = np.std(data, ddof=1)
alpha = 0.05

# IC para mu con sigma desconocido (t-Student)
t_crit = stats.t.ppf(1 - alpha/2, df=n-1)
ic_mu = (x_bar - t_crit * s / np.sqrt(n), x_bar + t_crit * s /
        np.sqrt(n))

# IC para sigma^2 (Chi-Cuadrado)
chi2_inf = stats.chi2.ppf(alpha/2, df=n-1)
chi2_sup = stats.chi2.ppf(1 - alpha/2, df=n-1)
ic_var = ((n-1)*s**2 / chi2_sup, (n-1)*s**2 / chi2_inf)

print(f"IC Media (95%): [{ic_mu[0]:.3f}, {ic_mu[1]:.3f}]")
print(f"IC Varianza (95%): [{ic_var[0]:.3f}, {ic_var[1]:.3f}]")
```

2.2 Unidad 1 (Complementario): Cadenas de Markov y Modelado Dinámico

Esta guía complementa la Unidad 1 de **Estadística Aplicada y Modelos de Regresión** (2º año) y establece la **articulación vertical** con la asignatura **Teoría de Juegos y Sistemas Multiagente (ABM)** (4º año).

---

1. Fundamentos de las Cadenas de Markov de Tiempo Discreto (DTMC)

Una Cadena de Markov es un proceso estocástico $\{X_t\}_{t \in \mathbb{N}_0}$ que evoluciona sobre un espacio discreto de estados $S = \{s_1, s_2, \ldots, s_K\}$.

1.1. La Propiedad de Markov (Pérdida de Memoria)

La probabilidad del estado futuro depende **únicamente del estado actual**, siendo condicionalmente independiente de toda la trayectoria previa:

$$P(X_{t+1} = s_j \mid X_t = s_i, X_{t-1} = s_{t-1}, \ldots, X_0 = s_0) = P(X_{t+1} = s_j \mid X_t = s_i) = p_{ij}$$

```
  [Estado Pasado X_{t-1}] ---> [Estado Presente X_t] ---> [Estado Futuro X_{t+1}]
                                          |
                                (Toda la información necesaria para
                                 predecir el futuro reside en X_t)
```

---

2. Matriz de Transición de Estados (P) y Diagrama de Grafo

La dinámica del sistema queda determinada por la **matriz estocástica** $P \in \mathbb{R}^{K \times K}$:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1K} \\ p_{21} & p_{22} & \cdots & p_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ p_{K1} & p_{K2} & \cdots & p_{KK} \end{pmatrix}, \quad \text{donde } \sum_{j=1}^{K} p_{ij} = 1 \quad \forall i$$

- Cada fila representa la distribución de probabilidad condicional de salida desde el estado i.
- Cada celda $p_{ij} \geq 0$ representa el peso del arco dirigido en el **Grafo de Transición de Estados**.

```
        p_11                    p_22
       /---\                   /---\
       |   |                   |   |
       v   |       p_12        v   |
  ( Estado 1 ) ------> ( Estado 2 )
       ^                       |
       |       p_21            |
       \-----------------------/
```

---

3. Evolución Temporal y Distribución Estacionaria

Dado un vector fila de probabilidad inicial $\pi^{(0)} = [P(X_0 = s_1), \ldots, P(X_0 = s_K)]$:

3.1. Ecuaciones de Chapman-Kolmogorov

La distribución de probabilidad al paso t se obtiene multiplicando por la potencia t-ésima de la matriz:

$$\pi^{(t)} = \pi^{(0)} \cdot P^t$$

3.2. Distribución Estacionaria o de Equilibrio (π)

Para cadenas irreducibles y aperiódicas (ergódicas), existe una única distribución π tal que el sistema alcanza un **equilibrio dinámico**:

$$\pi = \pi \cdot P \quad \text{sujeto a} \quad \sum_{i=1}^{K} \pi_i = 1,\ \pi_i \geq 0$$

$$\iff \pi(P - I) = 0 \quad (\text{Autovector a izquierda asociado al autovalor } \lambda = 1)$$

---

4. El Puente Pedagógico: De Cadenas de Markov a Teoría de Juegos y ABM

En la carrera de Ciencia de Datos e Inteligencia Artificial, las Cadenas de Markov vistas en 2º año constituyen el **andamio matemático** sobre el cual se construyen los modelos de 4º año:

Nivel / Asignatura	Modelo / Enfoque	Rol de las Cadenas de Markov y Probabilidad
2° Año: Estadística Aplicada y Modelos de Regresión	**Procesos Estocásticos y Dinámica de Estados**	Modelado probabilístico del entorno, matrices P, estados de equilibrio π y simulación computacional de trayectorias individuales.
4° Año: Teoría de la Decisión	**Juegos contra la Naturaleza y Procesos Markovianos**	Se incorporan acciones (A) y recompensas (R) sobre los estados de Markov $\implies$ **Procesos de Decisión de Markov (MDP)**.
4° Año: Teoría de Juegos	**Interacción Estratégica (Múltiples Jugadores)**	Las matrices de transición dependen conjuntamente de las estrategias mixtas de los jugadores $\implies$ **Juegos Estocásticos de Markov**.
4° Año: Simulación Basada en Agentes (ABM)	**Sistemas Multiagente Autónomos**	Cada agente autónomo sigue una máquina de estados finitos markoviana o adaptativa que interactúa en un entorno espacial simulado.

```
[2° Año: Cadena de Markov]  P(S_{t+1} | S_t)
            |
            ▼ (+ Acciones y Recompensas)
[4° Año: Proceso de Decisión de Markov - MDP]  P(S_{t+1} | S_t, A_t)
            |
            ▼ (+ Múltiples Agentes en Interacción Estratégica)
[4° Año: Teoría de Juegos y Simulación Multiagente ABM]  P(S_{t+1} | S_t,
A_t^{(1)}, \dots, A_t^{(N)})
```

---

5. Simulación de Trayectorias de Agentes en Python

El siguiente script muestra cómo simular la trayectoria de un agente y calcular analíticamente su distribución estacionaria:

```python
import numpy as np

# Espacio de Estados: 0: Exploración, 1: Explotación, 2: Descanso
estados = ["Exploración", "Explotación", "Descanso"]
P = np.array([
    [0.60, 0.30, 0.10],  # Desde Exploración
    [0.20, 0.70, 0.10],  # Desde Explotación
    [0.50, 0.00, 0.50]   # Desde Descanso
])

# 1. Cálculo Analítico de la Distribución Estacionaria (pi = pi * P)
# Resolvemos (P.T - I) * pi.T = 0 con sum(pi) = 1
A = np.vstack([P.T - np.eye(3), [1, 1, 1]])
b = np.array([0, 0, 0, 1])
pi_estacionaria = np.linalg.lstsq(A, b, rcond=None)[0]

print("--- Distribución Estacionaria Analítica ---")
for estado, prob in zip(estados, pi_estacionaria):
    print(f"P({estado}) = {prob:.4f}")

# 2. Simulación de Trayectoria de un Agente (Monte Carlo / ABM)
def simular_agente(matriz_P, estado_inicial=0, pasos=1000):
    trayectoria = [estado_inicial]
    estado_actual = estado_inicial
    for _ in range(pasos):
        estado_siguiente = np.random.choice(len(matriz_P),
        p=matriz_P[estado_actual])
        trayectoria.append(estado_siguiente)
        estado_actual = estado_siguiente
    return np.array(trayectoria)

# Simulación de 10,000 pasos temporales
trayectoria = simular_agente(P, estado_inicial=0, pasos=10000)
frecuencias_muestrales = np.bincount(trayectoria) / len(trayectoria)

print("\n--- Distribución Empírica Simulada (10.000 pasos) ---")
for estado, prob in zip(estados, frecuencias_muestrales):
    print(f"Frecuencia Relativa ({estado}) = {prob:.4f}")
```

2.3 Unidad 2: Contraste de Hipótesis y Análisis de Potencia

Esta guía complementa la Unidad 2 de la materia, formalizando la metodología de los ensayos de hipótesis, el cálculo analítico del error Tipo II (β), la función de potencia ($1-\beta$) y el diseño de experimentos para tamaño de muestra óptimo.

---

1. Estructura Formal de un Ensayo de Hipótesis

Un ensayo de hipótesis es una regla de decisión estadística que permite contrastar una **Hipótesis Nula (H_0)** frente a una **Hipótesis Alternativa (H_1)**.

- H_0 **(Hipótesis Nula):** Estado de no efecto, igualdad o status quo (ej. $\mu = \mu_0$, $\sigma^2 = \sigma_0^2$, $p = p_0$). Contiene siempre el signo de igualdad ($=, \leq, \geq$).
- H_1 **(Hipótesis Alternativa):** La hipótesis de investigación o cambio que se busca probar con evidencia suficiente ($\neq, >, <$).

1.1. Tipos de Ensayos

Tipo de Ensayo	Hipótesis Nula H_0	Hipótesis Alternativa H_1	Región de Rechazo (RC)
Bilateral (Dos Colas)	$H_0 : \theta = \theta_0$	$H_1 : \theta \neq \theta_0$	Rechazar si $\lvert Estadístico \rvert > Crítico_{1-\alpha/2}$
Unilateral Derecho	$H_0 : \theta \leq \theta_0$	$H_1 : \theta > \theta_0$	Rechazar si $Estadístico > Crítico_{1-\alpha}$
Unilateral Izquierdo	$H_0 : \theta \geq \theta_0$	$H_1 : \theta < \theta_0$	Rechazar si $Estadístico < -Crítico_{1-\alpha}$

---

2. Tipos de Error y Matriz de Decisión

Al tomar una decisión basada en una muestra, se pueden cometer dos tipos de error:

Decisión Muestral	Estado Real de la Población	
	H_0 es Verdadera	H_0 es Falsa (H_1 es Verdadera)
No Rechazar H_0	**Decisión Correcta** Probabilidad = $1-\alpha$ (Nivel de Confianza)	**Error Tipo II (β)** Falso Negativo (Aceptar H_0 siendo falsa)
Rechazar H_0	**Error Tipo I (α)** Falso Positivo (Nivel de Significación)	**Decisión Correcta** Probabilidad = $1-\beta$ (**Potencia de la Prueba**)

- α **(Nivel de Significación):** Probabilidad de cometer Error Tipo I. Es fijada a priori por el investigador (habitualmente 0.05 o 0.01).
- β **(Probabilidad de Error Tipo II):** Probabilidad de no rechazar H_0 cuando en realidad es falsa y el verdadero parámetro es θ_1.
- **Potencia ($1-\beta$):** Capacidad del test para detectar un efecto real (H_0 falsa).

---

3. El p-valor (Significación Observada)

El p**-valor** es la probabilidad de haber obtenido un estadístico muestral tan o más extremo que el observado, asumiendo que la hipótesis nula H_0 es estrictamente verdadera.

$$\text{Regla de Oro: } \begin{cases} \text{Si } p\text{-valor} \leq \alpha \implies \textbf{Rechazo } H_0 \text{ (Evidencia estadísticamente significativa)} \\ \text{Si } p\text{-valor} > \alpha \implies \textbf{No rechazo } H_0 \text{ (Evidencia insuficiente)} \end{cases}$$

---

4. Cálculo Analítico del Error Tipo II (β) y la Potencia ($1-\beta$)

Consideremos el ensayo unilateral derecho para la media con σ conocido:

$$H_0 : \mu \leq \mu_0 \quad \text{vs} \quad H_1 : \mu > \mu_0$$

1. **Paso 1: Determinar el punto crítico en escala muestral ($\bar{X}_{\text{crítico}}$)** Fijado α:

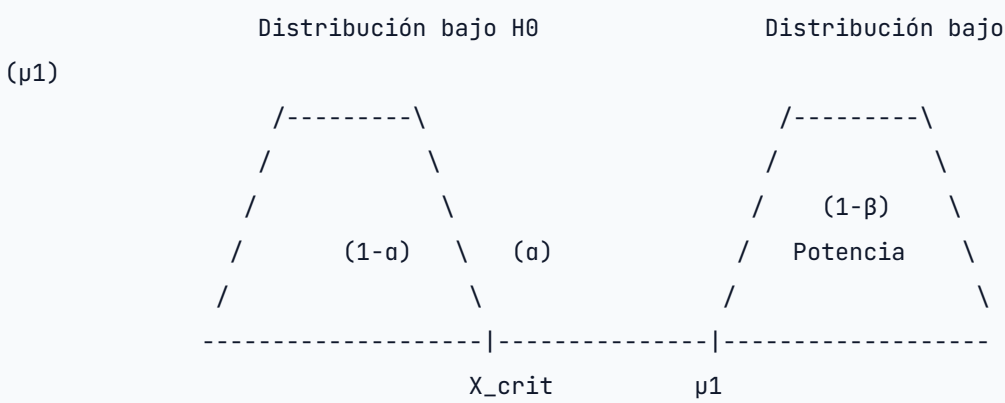
$$\bar{X}_{\text{crítico}} = \mu_0 + z_{1-\alpha}\frac{\sigma}{\sqrt{n}}$$

2. **Paso 2: Evaluar β bajo un valor alternativo específico $\mu_1 > \mu_0$**

$$\beta(\mu_1) = P\left(\bar{X} \leq \bar{X}_{\text{crítico}} \,\middle|\, \mu = \mu_1\right) = \Phi\left(\frac{\bar{X}_{\text{crítico}} - \mu_1}{\sigma/\sqrt{n}}\right)$$

3. **Paso 3: Calcular la Potencia de la Prueba**

$$\text{Potencia}(\mu_1) = 1 - \beta(\mu_1) = 1 - \Phi\left(\frac{\bar{X}_{\text{crítico}} - \mu_1}{\sigma/\sqrt{n}}\right)$$



5. Tamaño de Muestra Mínimo para Fijar α y β Simultáneamente

Para detectar una diferencia mínima relevante $\Delta = |\mu_1 - \mu_0|$ controlando el nivel de significación α y la potencia deseada $1 - \beta$:

$$n = \left(\frac{(z_{1-\alpha} + z_{1-\beta}) \cdot \sigma}{\Delta} \right)^2 \quad \text{(Para test unilateral)}$$

$$n = \left(\frac{(z_{1-\alpha/2} + z_{1-\beta}) \cdot \sigma}{\Delta} \right)^2 \quad \text{(Para test bilateral)}$$

6. Implementación en Python

```python
import numpy as np
import scipy.stats as stats

# Parámetros del diseño experimental
mu_0 = 100.0      # Valor bajo H0
mu_1 = 105.0      # Efecto mínimo a detectar (H1)
sigma = 12.0      # Desvío poblacional
alpha = 0.05      # Nivel de significación
power_target = 0.80 # Potencia deseada (80%)
beta_target = 1 - power_target

# 1. Cálculo de tamaño muestral n necesario
z_alpha = stats.norm.ppf(1 - alpha)
z_beta = stats.norm.ppf(1 - beta_target)
delta = mu_1 - mu_0

n_required = int(np.ceil(((z_alpha + z_beta) * sigma / delta)**2))
print(f"Tamaño de muestra n necesario: {n_required}")

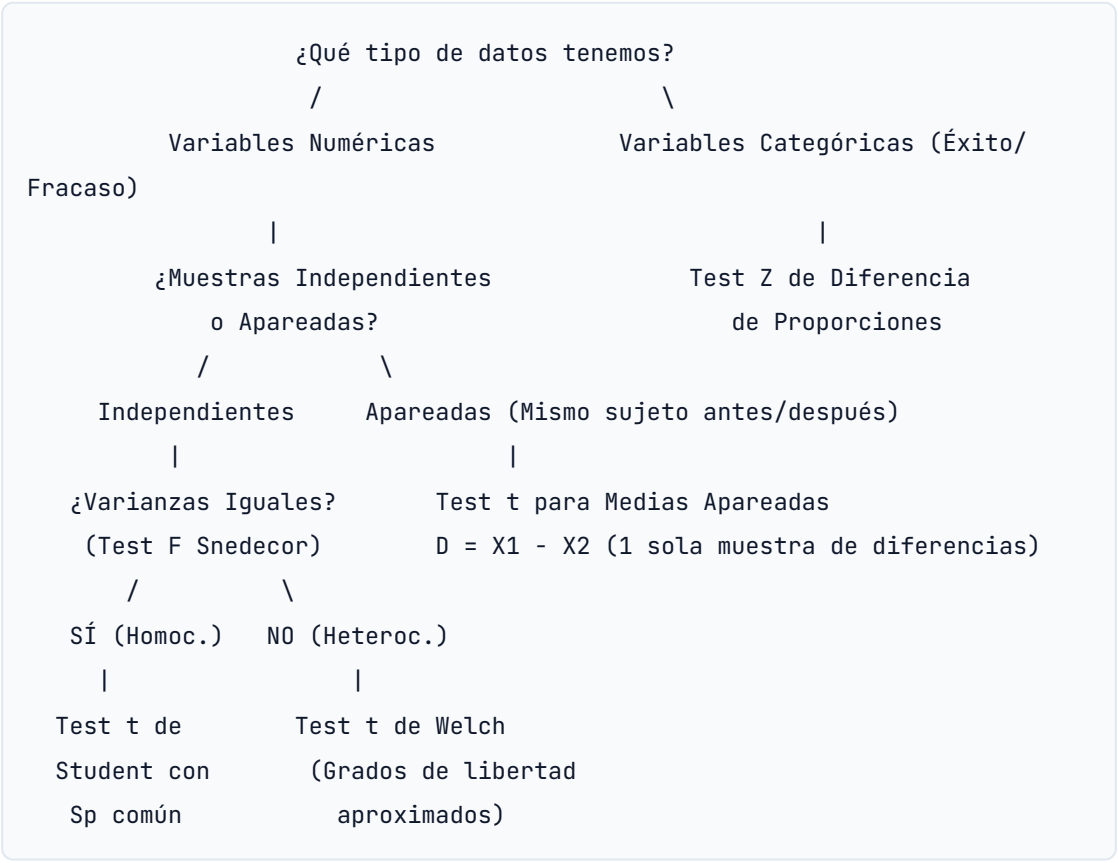
# 2. Verificación de Punto Crítico y Potencia Real
x_crit = mu_0 + z_alpha * (sigma / np.sqrt(n_required))
z_beta_real = (x_crit - mu_1) / (sigma / np.sqrt(n_required))
beta_real = stats.norm.cdf(z_beta_real)
potencia_real = 1 - beta_real

print(f"X crítico: {x_crit:.3f}")
print(f"Error Tipo II (beta): {beta_real:.4f}")
print(f"Potencia Real (1 - beta): {potencia_real:.4f}")
```

2.4 Unidad 2: Comparación de Poblaciones y ANOVA de 1 Factor

Esta guía complementa las Unidades 2 y 4 de la materia, presentando el árbol de decisión completo para la comparación de parámetros entre dos o más poblaciones y el Análisis de Varianza (ANOVA).

1. Árbol de Decisión para Comparar Dos Poblaciones



---

2. Comparación de Varianzas de Dos Poblaciones (Test F)

Para contrastar $H_0 : \sigma_1^2 = \sigma_2^2$ vs $H_1 : \sigma_1^2 \neq \sigma_2^2$ (asumiendo poblaciones normales):

Estadístico de prueba:

$$F_{\text{obs}} = \frac{S_1^2}{S_2^2} \sim F_{\nu_1 = n_1 - 1,\, \nu_2 = n_2 - 1}$$

- **Regla mnemotécnica:** Se suele colocar en el numerador la muestra con mayor varianza muestral ($S_1^2 > S_2^2$) para evaluar contra el cuantil superior $F_{n_1-1,\, n_2-1,\, 1-\alpha/2}$.
- Si el p-valor $\geq \alpha$, **no se rechaza** la igualdad de varianzas $\implies$ se asume homocedasticidad.

---

3. Comparación de Medias de Dos Poblaciones

3.1. Caso 1: Muestras Independientes con Varianzas Iguales ($\sigma_1^2 = \sigma_2^2 = \sigma^2$)

Se calcula la **Varianza Combinada (Pooled Variance)**:

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

Estadístico de prueba:

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{S_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim t_{n_1+n_2-2}$$

$$IC_{1-\alpha}(\mu_1 - \mu_2) = (\bar{X}_1 - \bar{X}_2) \pm t_{n_1+n_2-2,\, 1-\alpha/2} \cdot S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

3.2. Caso 2: Muestras Independientes con Varianzas Distintas ($\sigma_1^2 \neq \sigma_2^2$) — Test de Welch

Estadístico de prueba:

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \sim t_{\nu}$$

Grados de libertad aproximados (Satterthwaite):

$$\nu = \frac{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2}{\frac{(S_1^2/n_1)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}}$$

3.3. Caso 3: Muestras Apareadas (Diseño por Bloques o Antes/Después)

Se define la variable diferencia para cada par: $D_i = X_{1i} - X_{2i}$ para $i = 1, \ldots, n$. Se calculan la media muestral de las diferencias $\bar{D}$ y su desviación estándar S_D.

Estadístico de prueba:

$$T = \frac{\bar{D} - \mu_D}{S_D / \sqrt{n}} \sim t_{n-1}$$

---

4. Análisis de Varianza (ANOVA de 1 Factor / DCA)

El ANOVA permite comparar simultáneamente las medias de $k \geq 3$ poblaciones o tratamientos.

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \quad \text{vs} \quad H_1 : \text{Al menos una media es diferente}$$

4.1. Modelo Matemático y Descomposición de Sumas de Cuadrados

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij}, \quad \epsilon_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$$

$$\text{Suma de Cuadrados Total (SST)} = \text{Suma de Cuadrados de Tratamientos (SSA)} + \text{Suma de Cuadrados del}$$

$$\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\bar{Y}_{\bullet\bullet})^2=\sum_{i=1}^{k}n_i(\bar{Y}_{i\bullet}-\bar{Y}_{\bullet\bullet})^2+\sum_{i=1}^{k}\sum_{j=1}^{n_i}(Y_{ij}-\bar{Y}_{i\bullet})^2$$

4.2. Tabla de Análisis de Varianza (ANOVA)

Fuente de Variación	Suma de Cuadrados (SS)	Grados de Libertad (GL)	Cuadrado Medio (MS)	Estadístico F_{obs}
Entre Tratamientos (Grupos)	SSA	$k-1$	$MSA=\frac{SSA}{k-1}$	$F=\frac{MSA}{MSE}\sim F_{k-1,\,N-k}$
Dentro de los Grupos (Error)	SSE	$N-k$	$MSE=\frac{SSE}{N-k}$	
Total	SST	$N-1$	-	-

Donde $N=\sum_{i=1}^{k}n_i$ es el total de observaciones.

4.3. Validación de Supuestos del ANOVA

1. **Normalidad de los residuos:** Test de Shapiro-Wilk y gráfico Q-Q Plot.
2. **Homogeneidad de varianzas (Homocedasticidad):** Test de Levene o Bartlett.
3. **Independencia de los errores:** Diseño aleatorizado y análisis de residuos vs orden.

---

5. Implementación en Python

```
import numpy as np
import scipy.stats as stats

# Datos de 3 tratamientos (ANOVA 1 factor)
grupo_A = np.array([24.5, 23.8, 25.1, 24.9, 25.4])
grupo_B = np.array([28.1, 27.5, 28.9, 27.8, 28.2])
grupo_C = np.array([21.2, 22.0, 21.8, 22.5, 21.9])

# 1. Test de Levene para Homocedasticidad
stat_lev, p_lev = stats.levene(grupo_A, grupo_B, grupo_C)
print(f"Levene Test: p-valor = {p_lev:.4f} (Homocedasticidad {'OK' if
        p_lev > 0.05 else 'RECHAZADA'})")

# 2. ANOVA 1 Factor (F-test)
f_stat, p_val = stats.f_oneway(grupo_A, grupo_B, grupo_C)
print(f"ANOVA F-statistic = {f_stat:.3f}, p-valor = {p_val:.4e}")

if p_val < 0.05:
        print("Resultado: Hay diferencias significativas entre al menos
        dos grupos.")
```

2.5 Unidad 2: Contrastes de Chi-Cuadrado (Bondad de Ajuste e Independencia)

Esta guía complementa la Unidad 2 de la materia, formalizando los contrastes no paramétricos basados en la distribución Chi-Cuadrado: Pruebas de Bondad de Ajuste y Tablas de Contingencia (Independencia y Homogeneidad).

1. El Estadístico de Pearson (χ^2)

Para comparar un conjunto de frecuencias observadas (O_i) frente a frecuencias teóricas esperadas bajo la hipótesis nula (E_i):

$$\chi^2_{\text{obs}} = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i} \sim \chi^2_\nu$$

> ⚠ **Regla de Cochran (Condición de Validez):** 1. El tamaño total de la muestra debe ser $n \geq 30$. 2. Ninguna frecuencia esperada puede ser menor a 1 ($E_i \geq 1$). 3. Al menos el 80% de las clases deben tener $E_i \geq 5$. En la práctica académica habitual, si algún $E_i < 5$, **se deben agrupar clases adyacentes** antes de calcular el estadístico, recalculando los grados de libertad.

---

2. Pruebas de Bondad de Ajuste

Evalúan si una muestra aleatoria proviene de una distribución de probabilidad teórica específica (Uniforme, Binomial, Poisson, Exponencial o Normal).

H_0 : La variable sigue la distribución $F_0(x)$ vs H_1 : La variable NO sigue la distribución $F_0(x)$

2.1. Regla de Grados de Libertad (ν)

$$\nu = k - 1 - m$$

- k: Número de categorías o intervalos finales (tras eventuales agrupamientos).
- m: Número de parámetros poblacionales **estimados a partir de la muestra** mediante máxima verosimilitud o momentos.

Distribución a Contrastar	Parámetros Estimados (m)	Grados de Libertad (ν)
Uniforme / Discreta conocida a priori	$m = 0$	$\nu = k - 1$
Poisson (λ desconocido)	$m = 1$ ($\hat{\lambda} = \bar{X}$)	$\nu = k - 2$
Binomial (p desconocido)	$m = 1$ ($\hat{p} = \bar{X}/n$)	$\nu = k - 2$
Normal (μ y σ desconocidos)	$m = 2$ ($\hat{\mu} = \bar{X},\ \hat{\sigma} = S$)	$\nu = k - 3$

---

3. Tablas de Contingencia (Filas $r \times$ Columnas c)

Permiten analizar la relación entre dos variables cualitativas o categóricas.

3.1. Test de Independencia vs Test de Homogeneidad

- **Test de Independencia:** Se extrae **una sola muestra aleatoria** de tamaño N y se clasifican los individuos según dos atributos A (filas) y B (columnas).

  $$H_0: \text{El atributo A y el atributo B son independientes} \quad (P(A_i \cap B_j) = P(A_i)P(B_j))$$

- **Test de Homogeneidad:** Se toman r **muestras independientes** de poblaciones distintas y se evalúa si la distribución del atributo en estudio es igual en todas ellas.

$$H_0: P(B_j|\text{Pob}_1) = P(B_j|\text{Pob}_2) = \cdots = P(B_j|\text{Pob}_r)$$

3.2. Frecuencias Esperadas y Grados de Libertad

Bajo la hipótesis nula de independencia u homogeneidad:

$$E_{ij} = \frac{(\text{Total Fila } i) \times (\text{Total Columna } j)}{\text{Total General } N} = \frac{R_i \cdot C_j}{N}$$

$$\chi^2_{\text{obs}} = \sum_{i=1}^{r}\sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \sim \chi^2_{\nu}$$

$$\text{Grados de Libertad: } \nu = (r-1) \times (c-1)$$

---

4. Implementación en Python

```python
import numpy as np
import scipy.stats as stats

# Ejemplo: Tabla de contingencia 2x3 (Preferencia de producto según
        rango etario)
# Filas: [Jóvenes, Adultos], Columnas: [Prod A, Prod B, Prod C]
observados = np.array([
    [35, 40, 25],  # Jóvenes
    [20, 30, 50]   # Adultos
])

chi2_stat, p_val, dof, esperados = stats.chi2_contingency(observados)

print(f"Estadístico Chi-Cuadrado: {chi2_stat:.3f}")
print(f"Grados de libertad: {dof}")
print(f"p-valor: {p_val:.4e}")
print("\nFrecuencias Esperadas:\n", np.round(esperados, 2))

if p_val < 0.05:
    print("\nConclusión: Se rechaza H0 → Existe relación de
        dependencia significativa.")
```

2.6 Unidad 3: Regresión Lineal, Diagnóstico y Árbol de Decisiones Rápidas

Este documento funciona como **hoja de fórmulas de referencia rápida y guía de decisión** para resolver problemas prácticos, parciales y trabajos prácticos de la materia.

---

1. Tabla Maestra de Intervalos de Confianza y Contrastes de Hipótesis

Parámetro	Supuestos / Condiciones	Estadístico Pivote / Prueba	Distribución
Media μ (1 Pob.)	σ conocido (o $n \geq 30$)	$Z = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{n}}$	$\mathcal{N}(0,1)$
	σ desconocido, Pob. Normal, $n < 30$	$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}$	t_{n-1}
Varianza σ^2 (1 Pob.)	Pob. Normal, μ desconocida	$\chi^2 = \frac{(n-1)S^2}{\sigma_0^2}$	χ^2_{n-1}
Proporción p (1 Pob.)	$n\hat{p} \geq 5,\ n(1-\hat{p}) \geq 5$	$Z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$	$\mathcal{N}(0,1)$
Cociente de Varianzas $\frac{\sigma_1^2}{\sigma_2^2}$	Poblaciones Normales Independientes	$F = \frac{S_1^2}{S_2^2}$	$F_{n_1-1,\, n_2-1}$
Diferencia de Medias $\mu_1 - \mu_2$	Muestras Indep., $\sigma_1^2 = \sigma_2^2 = \sigma^2$ (Homocedasticidad)	$T = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$	$t_{n_1+n_2-2}$
	Muestras Indep., $\sigma_1^2 \neq \sigma_2^2$ (Welch)	$T = \frac{(\bar{X}_1 - \bar{X}_2) - \Delta_0}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}}$	t_ν (Satterthwaite)
	Muestras Apareadas ($D = X_1 - X_2$)	$T = \frac{\bar{D} - \mu_D}{S_D/\sqrt{n}}$	t_{n-1}
Bondad de Ajuste / Contingencia	$E_i = np_i \geq 5$ (o agrupar clases)	$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$	χ^2_{k-1-m} ó $\chi^2_{(r-1)(c-1)}$
ANOVA (1 Factor)	Normalidad y Homocedasticidad	$F = \frac{MSA}{MSE}$	$F_{k-1,\, N-k}$

2. Fórmulas de Teoría de la Regresión Lineal

2.1. Modelo Lineal Simple: $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$

- **Pendiente MCO:**

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2} = \frac{n\sum X_i Y_i - \sum X_i \sum Y_i}{n\sum X_i^2 - (\sum X_i)^2}$$

- **Ordenada al Origen MCO:**

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

- **Varianza Residual del Modelo (S_e^2 ó MSE):**

$$S_e^2 = \frac{\sum (Y_i - \hat{Y}_i)^2}{n-2} = \frac{SSE}{n-2}$$

- **Varianza de los Estimadores:**

$$\mathbb{V}\text{ar}(\hat{\beta}_1) = \frac{S_e^2}{S_{xx}}, \qquad \mathbb{V}\text{ar}(\hat{\beta}_0) = S_e^2 \left(\frac{1}{n} + \frac{\bar{X}^2}{S_{xx}} \right)$$

- **Test de Significancia de la Pendiente ($H_0 : \beta_1 = 0$):**

$$T = \frac{\hat{\beta}_1 - 0}{\sqrt{S_e^2 / S_{xx}}} \sim t_{n-2}$$

- **Coeficiente de Determinación (R^2):**

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = r_{xy}^2$$

2.2. Intervalos en Regresión para un valor X_0

- **Intervalo de Confianza para el Valor Medio $\mathbb{E}[Y|X_0]$:**

$$IC_{1-\alpha}(\mathbb{E}[Y|X_0]) = \hat{Y}_0 \pm t_{n-2,\,1-\alpha/2} \cdot S_e \sqrt{\frac{1}{n} + \frac{(X_0 - \bar{X})^2}{S_{xx}}}$$

- **Intervalo de Predicción para una Observación Individual Y_0:**

$$IP_{1-\alpha}(Y_0) = \hat{Y}_0 \pm t_{n-2,\,1-\alpha/2} \cdot S_e \sqrt{1 + \frac{1}{n} + \frac{(X_0 - \bar{X})^2}{S_{xx}}}$$

---

3. Guía de Selección de Procedimiento en 3 Pasos

1. **Paso 1: Identificar el objetivo y las variables.**
   - ¿Estimar un parámetro o contrastar una afirmación?
   - ¿Variable cuantitativa (media/varianza) o cualitativa/categórica (proporción, tablas)?
2. **Paso 2: Evaluar la estructura muestral.**
   - ¿1 población o 2 poblaciones?
   - Si son 2 poblaciones: ¿Las muestras son independientes o pareadas?
   - ¿Se conoce el desvío poblacional σ o debe usarse el muestral S?
3. **Paso 3: Validar los supuestos.**
   - Si se comparan 2 medias independientes: **Primero correr el Test F de varianzas**.
   - Si hay $k \geq 3$ grupos: **Correr ANOVA (validar Levene y Shapiro-Wilk)**.

- Si se ajusta una regresión: **Examinar significancia de β_1, R^2 y gráficos de residuos**.

2.7 Unidad 4: Regresión Logística, Odds Ratios y Clasificación Naive Bayes

Unidad 4: Estadística Aplicada y Modelos de Regresión

---

1. Regresión No Lineal y Polinómica

Cuando la relación entre la variable predictora X y la respuesta Y no es lineal en los parámetros o variables:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_d X_i^d + \epsilon_i$$

- **Lineal en los parámetros:** Aunque X esté elevado a potencias, el modelo sigue siendo lineal respecto a los coeficientes β_j, por lo que se resuelve mediante MCO multivariado con transformación polinómica de características.
- **Modelos Intrínsecamente No Lineales:** Modelos exponenciales ($Y = \alpha e^{\beta X}$), logísticos o hiperbólicos que requieren transformaciones logarítmicas (linealización) o algoritmos de optimización no lineal (Levenberg-Marquardt / MCO No Lineal).

---

2. Regresión Logística para Clasificación Binaria

Para variables de respuesta binarias $Y \in \{0, 1\}$, predecir directamente $\mathbb{E}[Y|X]$ con un modelo lineal puede producir probabilidades fuera del rango $[0, 1]$. La **Regresión Logística** modela la probabilidad del evento $P(Y = 1|X) = p(X)$ mediante la **función sigmoide (función logística)**:

$$p(X) = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z}, \quad \text{donde } z = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

2.1. Odds y Logit (Transformación Log-Odds)

- **Odds (Razón de Probabilidades):**

$$\text{Odds} = \frac{p(X)}{1 - p(X)} = e^{\beta_0 + \sum \beta_j X_j} \in [0, \infty)$$

- **Logit (Log-Odds):**

$$\text{Logit}(p(X)) = \ln\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \in (-\infty, \infty)$$

2.2. Interpretación de Coeficientes y Odds Ratio (OR)

Un incremento de 1 unidad en X_j, manteniendo constantes los demás predictores, multiplica los Odds por el factor e^{β_j}:

$$\text{Odds Ratio (OR)} = e^{\beta_j}$$

* Si $\beta_j > 0 \implies OR > 1$: Aumenta la probabilidad de que $Y = 1$. * Si $\beta_j < 0 \implies OR < 1$: Disminuye la probabilidad de que $Y = 1$.

2.3. Estimación por Máxima Verosimilitud (MLE)

A diferencia de MCO, los parámetros se obtienen maximizando la función de log-verosimilitud mediante el algoritmo de Newton-Raphson o descenso por gradiente:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \ln(p(x_i)) + (1 - y_i) \ln(1 - p(x_i)) \right]$$

---

3. Clasificación Probabilística: Naive Bayes

El clasificador Naive Bayes se fundamenta en el **Teorema de Bayes** y asume la **hipótesis de independencia condicional** entre los atributos predictores dado el valor de la clase:

$$P(Y = c \mid X_1, \ldots, X_p) = \frac{P(Y = c) \cdot P(X_1, \ldots, X_p \mid Y = c)}{P(X_1, \ldots, X_p)}$$

3.1. Supuesto “Naive” (Ingenuo)

$$P(X_1, \ldots, X_p \mid Y = c) = \prod_{j=1}^{p} P(X_j \mid Y = c)$$

Regla de decisión MAP (Maximum A Posteriori):

$$\hat{y} = \arg\max_{c} \left[P(Y = c) \prod_{j=1}^{p} P(X_j \mid Y = c) \right]$$

3.2. Variantes Principales de Naive Bayes

1. **Gaussian Naive Bayes:** Asume que las variables continuas dentro de cada clase siguen una distribución Normal: $P(X_j = x \mid Y = c) = \frac{1}{\sqrt{2\pi\sigma_{jc}^2}} \exp\left(-\frac{(x - \mu_{jc})^2}{2\sigma_{jc}^2} \right)$.
2. **Multinomial Naive Bayes:** Adecuado para recuentos discretos de palabras / bolsa de palabras (análisis de texto).
3. **Bernoulli Naive Bayes:** Adecuado para variables binarias de presencia/ausencia.

---

4. Métricas de Evaluación para Clasificación

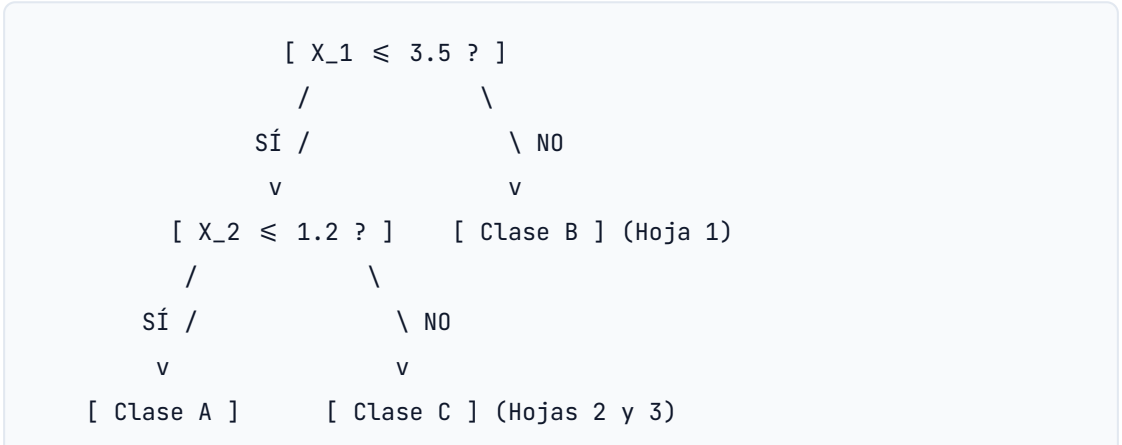
Métrica	Fórmula	Interpretación Práctica
Accuracy (Exactitud)	$\frac{TP+TN}{TP+TN+FP+FN}$	Proporción global de aciertos (inadecuada para clases desbalanceadas).
Precision (Precisión)	$\frac{TP}{TP+FP}$	De todos los clasificados como positivos, ¿cuántos son realmente positivos? (Minimiza FP).
Recall (Sensibilidad)	$\frac{TP}{TP+FN}$	De todos los casos positivos reales, ¿cuántos logró detectar el modelo? (Minimiza FN).
F1-Score	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	Media armónica entre precisión y sensibilidad.
AUC-ROC	Área bajo la curva TPR vs FPR	Capacidad discriminativa del modelo a lo largo de todos los umbrales de corte (0.5 = azar, 1.0 = perfecto).

2.8 Unidad 5: Árboles CART, Ensembles (Random Forest), K-Fold CV y Series de Tiempo

Unidad 5: Estadística Aplicada y Modelos de Regresión

1. Algoritmos Basados en Árboles de Decisión (CART)

Los árboles de decisión particionan recursivamente el espacio de características $\mathbb{R}^p$ en regiones rectangulares disjuntas $R_1, R_2, \ldots, R_M$, asignando a cada región una predicción constante.



1.1. Criterios de División para Clasificación

Para un nodo m con proporción de clases p_{mk} ($k = 1, \ldots, K$): * **Índice de Impureza de Gini:**

$$I_{\text{Gini}}(m) = \sum_{k=1}^{K} p_{mk}(1 - p_{mk}) = 1 - \sum_{k=1}^{K} p_{mk}^2$$

* **Entropía / Ganancia de Información:**

$$H(m) = -\sum_{k=1}^{K} p_{mk} \log_2(p_{mk})$$

1.2. Criterio de División para Regresión (MSE / Varianza)

Para un nodo con N_m observaciones y media $\bar{y}_m$:

$$\text{MSE}(m) = \frac{1}{N_m} \sum_{i \in R_m} (y_i - \bar{y}_m)^2$$

1.3. Control del Sobreajuste (Overfitting) y Poda

- **Hiperparámetros clave:** `max_depth` (profundidad máxima), `min_samples_split` (mínimo de muestras para dividir un nodo), `min_samples_leaf` (mínimo de muestras por hoja).
- **Cost-Complexity Pruning (Poda por Complejidad de Costos):** Minimiza $R_\alpha(T) = R(T) + \alpha|T|$, donde $|T|$ es el número de nodos hoja y $\alpha \geq 0$ es el factor de penalización.

---

2. Métodos de Ensamble: Bosques Aleatorios (Random Forest)

El algoritmo **Random Forest** es una técnica de Bagging (Bootstrap Aggregating) que entrena una colección de B árboles de decisión independientes y promedia sus predicciones para reducir drásticamente la varianza.

2.1. El Mecanismo de Descorrelación de Árboles

1. **Muestreo Bootstrap:** Cada árbol b se entrena sobre una muestra con reemplazo de tamaño n extraída del conjunto de entrenamiento.
2. **Selección Aleatoria de Características (Feature Subsampling):** En cada nodo del árbol, solo se evalúa un subconjunto aleatorio de $m \approx \sqrt{p}$ características (en clasificación) o $m \approx p/3$ (en regresión) para elegir la mejor división.
3. **Predicción Final:**
   - Clasificación: Voto por mayoría.
   - Regresión: $\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$.

2.2. Importancia de Características (Feature Importance)

- **Gini Importance / MDI (Mean Decrease Impurity):** Promedio de la reducción total de impureza atribuible a cada variable a lo largo de todos los árboles del bosque.
- **Permutation Importance:** Pérdida en el rendimiento del modelo cuando se permutan aleatoriamente los valores de una característica en el conjunto de validación.

---

3. Validación Cruzada (Cross-Validation) y Búsqueda de Hiperparámetros

```
Conjunto Total: [----------------- Dataset Completo -----------------]
                  |                                 |
Entrenamiento / Validación (80%)              Test Final (20%)
[ Fold 1 ][ Fold 2 ][ Fold 3 ][ Fold 4 ][ Fold 5 ]
  (Val)     (Train)   (Train)   (Train)   (Train)  ⟶ Iteración 1
  (Train)   (Val)     (Train)   (Train)   (Train)  ⟶ Iteración 2 ...
```

- **K-Fold Cross-Validation:** Divide el dataset en K bloques disjuntos; entrena en $K-1$ y valida en el restante, promediando los K puntajes.
- **GridSearchCV vs RandomizedSearchCV:** Exploración exhaustiva o probabilística de la grilla de hiperparámetros (`n_estimators`, `max_depth`, `max_features`).

---

4. Introducción Práctica a Series de Tiempo

Una serie temporal $\{Y_t\}$ es una secuencia de observaciones ordenadas cronológicamente.

4.1. Descomposición Clásica

$$Y_t = T_t + S_t + C_t + I_t \quad (\text{Aditiva: Tendencia} + \text{Estacionalidad} + \text{Ciclo} + \text{Residuo})$$

$$Y_t = T_t \times S_t \times C_t \times I_t \quad (\text{Multiplicativa})$$

4.2. Estacionariedad y Modelos ARIMA(p, d, q)

- **Estacionariedad en sentido débil:** Media y varianza constantes a lo largo del tiempo, y autocovarianza dependiente solo del rezago k.
- **Prueba Dickey-Fuller Aumentada (ADF):** H_0 : $\text{La serie posee raíz unitaria (No Estacionaria)}$.

- **Parámetros ARIMA:**
  - p: Orden del componente Autorregresivo (AR).
  - d: Grado de diferenciación para alcanzar la estacionariedad.
  - q: Orden de Medias Móviles (MA).

2.9 Unidad 6: Inferencia Causal, DAGs, Criterio Backdoor y Causal ML (DoWhy / EconML)

Unidad 6: Estadística Aplicada y Modelos de Regresión

---

1. La Escalera de la Causalidad de Judea Pearl

La inferencia causal busca responder preguntas más allá de la mera asociación estadística (correlación):

```
       [ Nivel 3: Contrafácticos ]      "¿Qué habría pasado si hubiera
tomado otra decisión?"
                    |                   P(Y_{X=x'} | X=x, Y=y)
                    ▼
       [ Nivel 2: Intervención ]        "¿Qué pasará con Y si ejecuto la
acción X?"
                    |                   P(Y | do(X = x))
                    ▼
       [ Nivel 1: Asociación / EDA ]    "¿Qué nos dice ver X sobre Y?"
                                        P(Y | X = x)  (Regresión y ML
Tradicional)
```

---

2. Grafos Acíclicos Dirigidos (DAGs) y Estructuras Fundamentales

Un **Grafo Acíclico Dirigido (DAG)** representa supuestos cualitativos sobre las relaciones causales entre variables:

2.1. Las Tres Estructuras Elementales (Tripletes)

1. **Cadena (Chain / Mediador):**

$$X \longrightarrow M \longrightarrow Y$$

   - X causa Y a través de M. X e Y están asociados.
   - Si se **condiciona (controla) por** M, el camino se bloquea y $X \perp\!\!\!\perp Y \mid M$.

2. **Bifurcación (Fork / Causa Común / Confounder):**

$$X \longleftarrow Z \longrightarrow Y$$

   - Z es un **confusor (confounder)** que genera una correlación espuria entre X e Y.
   - Para estimar el efecto causal puro de $X \to Y$, se **DEBE condicionar por** Z ($X \perp\!\!\!\perp Y \mid Z$).

3. **Colisionador (Collider / Efecto Común):**

$$X \longrightarrow C \longleftarrow Y$$

   - X e Y son marginalmente independientes ($X \perp\!\!\!\perp Y$).
   - ⚠ **Peligro de Sesgo de Selección:** Si se **condiciona por el colisionador** C, se abre un camino espurio y se induce una correlación artificial entre X e Y. **¡NUNCA controlar por colisionadores!**

---

3. Criterio de Ajuste Backdoor (Puerta Trasera)

Un conjunto de variables Z satisface el **Criterio Backdoor** relativo al tratamiento T y el resultado Y si: 1. Ningún nodo de Z es descendiente de T. 2. Z bloquea todos los caminos que entran a T y llegan a Y (caminos de puerta trasera).

$$\text{Fórmula de Ajuste: } P(Y = y \mid \text{do}(T = t)) = \sum_{z} P(Y = y \mid T = t, Z = z) P(Z = z)$$

---

4. Efectos Causales y Causal Machine Learning

4.1. Definiciones Fundamentales

Bajo el marco de Resultados Potenciales de Neyman-Rubin ($Y(1)$ si tratado, $Y(0)$ si control): * **ATE (Average Treatment Effect):** $\text{ATE} = \mathbb{E}[Y(1) - Y(0)]$. * **CATE (Conditional Average Treatment Effect / Heterogeneidad):** $\text{CATE}(x) = \mathbb{E}[Y(1) - Y(0) \mid X = x]$.

4.2. El Flujo de Trabajo en 4 Pasos con DoWhy

```
1. Modelar      ⟶ Definir el DAG causal (causal graph).
2. Identificar  ⟶ Aplicar criterios Backdoor/Frontdoor/IV para hallar la expresión matemática.
3. Estimar      ⟶ Ajustar modelos con Scikit-Learn / Statsmodels / EconML (Double ML, Causal Forests).
4. Refutar      ⟶ Tests de robustez (Placebo Treatment, Random Common Cause, Subset Removal).
```

PARTE III: GUÍAS DE EJERCITACIÓN PRÁCTICA Y PROBLEMAS DE LIBRO RESUELTOS

3.1 Guía de Ejercitación 01: Probabilidad y Procesos Estocásticos

Unidad 1: Estadística Aplicada y Modelos de Regresión

Ejercicio 1.1 (Probabilidad Condicional y Teorema de Bayes)

Inspirado en León-García, Cap. 2. En un sistema de detección de intrusiones en servidores de IA, el 2% de los paquetes entrantes son maliciosos (M). El sistema clasifica correctamente un paquete malicioso como amenaza (A) con probabilidad 0.98, pero tiene una tasa de falsas alarmas del 5% para paquetes legítimos. 1. Calcule la probabilidad total de que el sistema dispare una alarma ($P(A)$). 2. Si el sistema emite una alarma, ¿cuál es la probabilidad a posteriori de que el paquete sea verdaderamente malicioso ($P(M|A)$)? 3. Interprete el resultado desde la perspectiva de la toma de decisiones del analista.

💡 Respuesta / Guía de solución: * $P(A) = P(A|M)P(M) + P(A|M^c)P(M^c) = (0.98)(0.02) + (0.05)(0.98) = 0.0196 + 0.0490 = 0.0686$ (6.86%). * $P(M|A) = \frac{0.0196}{0.0686} \approx 0.2857$ (28.57%). * Conclusión: A pesar de la alta sensibilidad (98%), debido a la baja prevalencia del ataque, más del 71% de las alarmas son falsos positivos (paradoja de la tasa base).

Ejercicio 1.2 (Cadenas de Markov: Modelado de Estados y Potencias de Matriz)

Inspirado en León-García, Cap. 11. Un nodo en un clúster distribuido puede encontrarse en tres estados: $S = \{1 : \text{Operativo Normal}, 2 : \text{Sobrecargado}, 3 : \text{Falla/Reinicio}\}$. La matriz de transición por minuto es:

$$P = \begin{pmatrix} 0.80 & 0.15 & 0.05 \\ 0.30 & 0.60 & 0.10 \\ 0.70 & 0.00 & 0.30 \end{pmatrix}$$

1. Verifique que la matriz P es estocástica. Dibuje el diagrama de transición de estados.
2. Si el nodo inicia en estado Operativo Normal ($\pi^{(0)} = [1, 0, 0]$), calcule la distribución de probabilidad tras 2 minutos ($\pi^{(2)}$).
3. Halle analíticamente el vector de distribución estacionaria $\pi = [\pi_1, \pi_2, \pi_3]$.

💡 Respuesta / Guía de solución: * $\pi^{(1)} = [0.80, 0.15, 0.05]$. * $\pi^{(2)} = \pi^{(1)}P = [0.80(0.8) + 0.15(0.3) + 0.05(0.7),\ 0.80(0.15) + 0.15(0.6),\ 0.80(0.05) + 0.15(0.1) + 0.05(0.3)] = [0.720, 0.210, 0.070]$. * Sistema estacionario: $\pi(P - I) = 0$ con $\pi_1 + \pi_2 + \pi_3 = 1 \implies \pi \approx [0.7188, 0.2109, 0.0703]$.

Ejercicio 1.3 (Simulación Monte Carlo de Trayectorias de un Agente en Python)

Puente pedagógico a Modelos Basados en Agentes (ABM). Escriba una función en Python que reciba la matriz P anterior, simule la trayectoria del nodo durante $T = 50.000$ pasos temporales y compute la proporción empírica de tiempo que el nodo pasa en cada estado. Compare el resultado empírico con la distribución estacionaria analítica.

3.2 Guía de Ejercitación 02: Inferencia Estadística, Test de Hipótesis y ANOVA

Unidad 2: Estadística Aplicada y Modelos de Regresión

Ejercicio 2.1 (Intervalo de Confianza y Tamaño Muestral para la Media)

Inspirado en Dopazo / UADE. Un centro de cómputos desea estimar el tiempo medio de respuesta μ (en ms) de una API de inferencia de LLMs. Por mediciones previas se asume una desviación estándar $\sigma = 45$ ms. 1. ¿Qué tamaño de muestra n se requiere para estimar μ con un margen de error máximo $e \leq 5$ ms y un nivel de

confianza del 95%? 2. Se toma una muestra piloto de tamaño $n = 16$ en una nueva versión del modelo, obteniéndose $\bar{X} = 124.5\,\text{ms}$ y $S = 38.2\,\text{ms}$. Asumiendo normalidad con σ desconocido, construya el intervalo de confianza del 99% para μ.

💡 Respuesta / Guía de solución: 1. $n = \left(\frac{1.96 \cdot 45}{5}\right)^2 = (17.64)^2 \approx 311.17 \implies \mathbf{n = 312}$ mediciones. 2. $t_{15,\,0.995} = 2.947$. $IC_{99\%}(\mu) = 124.5 \pm 2.947 \cdot \frac{38.2}{\sqrt{16}} = 124.5 \pm 28.14 \implies [\mathbf{96.36},\ \mathbf{152.64}]$ ms.

Ejercicio 2.2 (Comparación de Dos Poblaciones: Test F y Test t de Welch)

Inspirado en Chan, Badano & Rey, Cap. 2. Se evalúa la latencia de procesamiento de imágenes entre dos frameworks de Deep Learning (A y B) sobre muestras independientes: * Framework A: $n_1 = 12$, $\bar{X}_1 = 45.2\,\text{ms}$, $S_1^2 = 14.8$. * Framework B: $n_2 = 15$, $\bar{X}_2 = 52.8\,\text{ms}$, $S_2^2 = 42.1$.

1. Realice el test de igualdad de varianzas al nivel $\alpha = 0.05$. ¿Se cumple el supuesto de homocedasticidad?
2. En función del resultado anterior, contraste si existe diferencia significativa entre las latencias medias ($H_0 : \mu_1 = \mu_2$ vs $H_1 : \mu_1 \neq \mu_2$) al nivel $\alpha = 0.05$.

💡 Respuesta / Guía de solución: 1. $F_{\text{obs}} = \frac{S_2^2}{S_1^2} = \frac{42.1}{14.8} \approx 2.845$. $F_{\text{crítico}(14,11,0.975)} \approx 3.36$. Como $2.845 < 3.36$ (p-valor > 0.05), no se rechaza la igualdad de varianzas. 2. Varianza combinada: $S_p^2 = \frac{11(14.8) + 14(42.1)}{12 + 15 - 2} = \frac{162.8 + 589.4}{25} = 30.088 \implies S_p \approx 5.485$. $T_{\text{obs}} = \frac{45.2 - 52.8}{5.485\sqrt{\frac{1}{12} + \frac{1}{15}}} = \frac{-7.6}{5.485 \cdot 0.3873} = \frac{-7.6}{2.124} = \mathbf{-3.578}$. $t_{\text{crítico}(25,0.975)} = 2.060$. Como $|-3.578| > 2.060$, **se rechaza** H_0 (p-valor < 0.002). El Framework A es significativamente más veloz.

Ejercicio 2.3 (ANOVA de 1 Factor - DCA)

Inspirado en Dopazo / UADE. Se comparan 3 técnicas de compresión de embeddings (T1, T2, T3) midiendo el tiempo de consulta en $n = 5$ ejecuciones por grupo ($N = 15$): * $SSA = 156.4$, $SST = 212.8$. 1. Complete la tabla ANOVA (SS, GL, MS, F). 2. Concluya si existen diferencias significativas al nivel $\alpha = 0.05$ ($F_{2,12,0.95} = 3.89$).

💡 Respuesta / Guía de solución: * $SSE = SST - SSA = 212.8 - 156.4 = 56.4$. * $MSA = 156.4/2 = 78.2$, $MSE = 56.4/12 = 4.70$. * $F_{\text{obs}} = 78.2/4.70 = \mathbf{16.638}$. Como $16.638 > 3.89$, **se rechaza** H_0 ($p < 0.001$). Al menos una técnica difiere de las demás.

3.3 Guía de Ejercitación 03: Regresión Lineal Múltiple, Diagnóstico de Residuos y PCA

Unidad 3: Estadística Aplicada y Modelos de Regresión

Ejercicio 3.1 (Ajuste e Inferencia en Regresión Lineal Simple)

Inspirado en Szretter Noste, Cap. 1. Se analiza la relación entre el número de capas de un modelo neuronal (X) y el uso de memoria en VRAM en GB (Y) en $n = 10$ pruebas de laboratorio: * $\sum X_i = 120$, $\sum Y_i = 45.0$, $\sum X_i^2 = 1840$, $\sum Y_i^2 = 245.8$, $\sum X_i Y_i = 642.0$.

1. Calcule las sumas de cuadrados corregidas S_{xx}, S_{yy}, S_{xy}.
2. Halle los estimadores de MCO $\hat{\beta}_1$ y $\hat{\beta}_0$. Escriba la ecuación de la recta ajustada.
3. Calcule la varianza residual del modelo (S_e^2) y el coeficiente de determinación (R^2).
4. Realice el test de significancia de la pendiente ($H_0 : \beta_1 = 0$ vs $H_1 : \beta_1 \neq 0$) al nivel $\alpha = 0.01$ ($t_{8,\,0.995} = 3.355$).
5. Para un modelo de $X_0 = 15$ capas, halle el intervalo de predicción del 95% para la VRAM requerida.

💡 Respuesta / Guía de solución: * $\bar{X} = 12.0, \bar{Y} = 4.50$. * $S_{xx} = 1840 - 10(12)^2 = 400$. * $S_{yy} = 245.8 - 10(4.5)^2 = 43.3$. * $S_{xy} = 642.0 - 10(12)(4.5) = 102.0$. * $\hat{\beta}_1 = \frac{102.0}{400} = \mathbf{0.255}$. $\hat{\beta}_0 = 4.50 - 0.255(12) = \mathbf{1.44}$. * $\hat{Y} = 1.44 + 0.255X$. * $SSE = S_{yy} - \hat{\beta}_1 S_{xy} = 43.3 - 0.255(102.0) = 43.3 - 26.01 = 17.29 \implies S_e^2 = \frac{17.29}{8} \approx \mathbf{2.161}$ ($S_e \approx 1.470$). * $R^2 = \frac{26.01}{43.3} \approx \mathbf{0.6007}$ (60.07% de la varianza explicada). * $T_{\text{obs}} = \frac{0.255}{\sqrt{2.161/400}} = \frac{0.255}{0.0735} = \mathbf{3.469} > 3.355 \implies$ **Se rechaza** H_0 al 1% ($p < 0.01$). * Para $X_0 = 15$: $\hat{Y}_0 = 1.44 + 0.255(15) = 5.265$ GB. $IP_{95\%} = 5.265 \pm 2.306(1.470)\sqrt{1 + \frac{1}{10} + \frac{(15-12)^2}{400}} = 5.265 \pm 3.390\sqrt{1.1225} = 5.265 \pm 3.592 \implies [\mathbf{1.67},\ \mathbf{8.86}]$ GB.

Ejercicio 3.2 (Regresión Múltiple y Diagnóstico de Colinealidad VIF)

Inspirado en Szretter Noste, Cap. 2 y 3; Müller & Guido, Cap. 2. En un modelo lineal múltiple con 3 predictores, la matriz de correlación entre los predictores X_1 y X_2 arroja un coeficiente $r_{12} = 0.94$. 1. Explique el impacto del fenómeno de multicolinealidad sobre los estimadores de MCO $\hat{\beta}_j$ y sus errores estándar. 2. Calcule el Factor de Inflación de la Varianza (VIF) aproximado entre X_1 y X_2: $\text{VIF}_j = \frac{1}{1 - R_j^2}$. ¿Supera el umbral de alerta habitual ($\text{VIF} > 5$ ó 10)? 3. Proponga dos estrategias metodológicas para mitigar este problema (ej. regularización Ridge o Análisis de Componentes Principales).

💡 Respuesta / Guía de solución: * Si $R_1^2 \approx (0.94)^2 = 0.8836 \implies \text{VIF}_1 = \frac{1}{1-0.8836} \approx \mathbf{8.59}$ (inflación severa de varianza). * Las varianzas de los estimadores se inflan, volviendo los coeficientes inestables y los p-valores artificialmente no significativos. Estrategias: eliminar una de las variables redundantes, aplicar regularización L_2 (Ridge) o transformar mediante PCA.

3.4 Guía de Ejercitación 04: Regresión Logística y Clasificación Naive Bayes

Unidad 4: Estadística Aplicada y Modelos de Regresión

Ejercicio 4.1 (Interpretación de Odds Ratio y Probabilidades en Regresión Logística)

Inspirado en Müller & Guido, Cap. 2; Han & Kamber, Cap. 8. Un modelo de regresión logística ajustado para predecir la deserción de usuarios en una plataforma ($Y = 1$: Deserta, $Y = 0$: Permanece) arrojó la siguiente ecuación:

$$\text{Logit}(p) = -1.80 + 0.65X_1 - 0.40X_2$$

Donde: * X_1: Número de incidentes técnicos reportados en el mes. * X_2: Horas semanales de uso de la plataforma.

1. Calcule el **Odds Ratio (OR)** para la variable X_1. Interprete su significado en términos de riesgo de deserción.
2. Calcule la probabilidad estimada de deserción $P(Y = 1)$ para un usuario que reportó $X_1 = 3$ incidentes y registra $X_2 = 2$ horas de uso semanal.
3. Si el umbral de decisión es $\tau = 0.50$, ¿qué predicción realiza el modelo para este usuario?

💡 Respuesta / Guía de solución: 1. $\text{OR}(X_1) = e^{0.65} \approx \mathbf{1.9155}$. Interpretación: Por cada incidente técnico adicional, los odds de deserción aumentan un 91.55%, manteniendo constante el uso de la plataforma. 2. $z = -1.80 + 0.65(3) - 0.40(2) = -1.80 + 1.95 - 0.80 = -0.65$.

$$P(Y=1) = \frac{1}{1+e^{-(-0.65)}} = \frac{1}{1+e^{0.65}} = \frac{1}{1+1.9155} = \frac{1}{2.9155} \approx \mathbf{0.3429}\ (34.29\%)$$

3. Como $0.3429 < 0.50$, el modelo clasifica al usuario como $\hat{Y} = 0$ **(No deserta)**.

Ejercicio 4.2 (Clasificador Naive Bayes Gaussian y Tablas de Probabilidad)

Inspirado en Han & Kamber, Cap. 8. Se desea clasificar correos electrónicos en Spam (C_1) o Legítimo (C_0). Las probabilidades a priori son $P(C_1) = 0.30$ y $P(C_0) = 0.70$. Se evalúan dos palabras clave binarias: X_1 (contiene "oferta") y X_2 (contiene "urgente"):

Condición	$P(X_1 = 1 \mid C_k)$	$P(X_2 = 1 \mid C_k)$
Spam (C_1)	0.80	0.70
Legítimo (C_0)	0.10	0.15

Llega un nuevo correo que contiene ambas palabras ($X_1 = 1, X_2 = 1$). 1. Aplique el supuesto de independencia condicional de Naive Bayes para calcular la probabilidad conjunta no normalizada para cada clase. 2. Normalice los valores y obtenga la probabilidad a posteriori $P(C_1 \mid X_1 = 1, X_2 = 1)$.

💡 Respuesta / Guía de solución: 1. $\text{Score}(C_1) = P(C_1) \cdot P(X_1 = 1|C_1) \cdot P(X_2 = 1|C_1) = 0.30 \times 0.80 \times 0.70 = \mathbf{0.168}$. $\text{Score}(C_0) = P(C_0) \cdot P(X_1 = 1|C_0) \cdot P(X_2 = 1|C_0) = 0.70 \times 0.10 \times 0.15 = \mathbf{0.0105}$. 2. Evidencia total: $P(X) = 0.168 + 0.0105 = 0.1785$.

$$P(\text{Spam} \mid X) = \frac{0.168}{0.1785} \approx \mathbf{0.9412}\ (94.12\%)$$

El correo se clasifica como **Spam** con un 94.12% de certeza.

3.5 Guía de Ejercitación 05: Árboles de Decisión, Random Forest y Series Temporales

Unidad 5: Estadística Aplicada y Modelos de Regresión

Ejercicio 5.1 (Cálculo Analítico del Índice de Gini y Mejor División en Árboles)

Inspirado en Han & Kamber, Cap. 8; Müller & Guido, Cap. 2. Un nodo raíz D contiene 100 instancias de entrenamiento: 60 de la Clase A y 40 de la Clase B. 1. Calcule la impureza de Gini del nodo padre $I_{\text{Gini}}(D)$. 2. Se evalúa dividir el nodo según la variable binaria $X_1 \in \{0, 1\}$, resultando en: * Rama izquierda ($X_1 = 0$, $N_L = 40$ instancias): 35 Clase A, 5 Clase B. * Rama derecha ($X_1 = 1$, $N_R = 60$ instancias): 25 Clase A, 35 Clase B. Calcule el índice de Gini ponderado de la división $I_{\text{Gini}}(D, X_1)$ y la reducción de impureza (ΔI_{Gini}).

💡 Respuesta / Guía de solución: 1. $I_{\text{Gini}}(D) = 1 - \left((0.60)^2 + (0.40)^2\right) = 1 - (0.36 + 0.16) = 1 - 0.52 = \mathbf{0.480}$. 2. Rama izquierda: $I_{\text{Gini}}(L) = 1 - \left((35/40)^2 + (5/40)^2\right) = 1 - (0.7656 + 0.0156) = 1 - 0.7812 = 0.2188$. Rama derecha: $I_{\text{Gini}}(R) = 1 - \left((25/60)^2 + (35/60)^2\right) = 1 - (0.1736 + 0.3403) = 1 - 0.5139 = 0.4861$. Impureza ponderada:

$$I_{\text{Gini}}(D, X_1) = \frac{40}{100}(0.2188) + \frac{60}{100}(0.4861) = 0.0875 + 0.2917 = \mathbf{0.3792}$$

Ganancia de pureza: $\Delta I = 0.480 - 0.3792 = \mathbf{0.1008}$ (división favorable).

Ejercicio 5.2 (Random Forest vs Árbol Simple: Reducción de Varianza)

Inspirado en Müller & Guido, Cap. 2. Suponga que cada árbol de decisión individual tiene una varianza de error $\sigma^2 = 12.0$, y que la correlación promedio entre las predicciones de dos árboles distintos es $\rho = 0.25$. 1. Aplicando la fórmula de la varianza del promedio de B estimadores correlacionados:

$$\mathbb{V}\text{ar}(\bar{T}) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Calcule la varianza esperada de un ensamble Random Forest con $B = 100$ árboles. 2. ¿A qué valor límite tiende la varianza si $B \to \infty$? ¿Por qué el mecanismo de subsampleo aleatorio de características en cada nodo es crucial para reducir ρ?

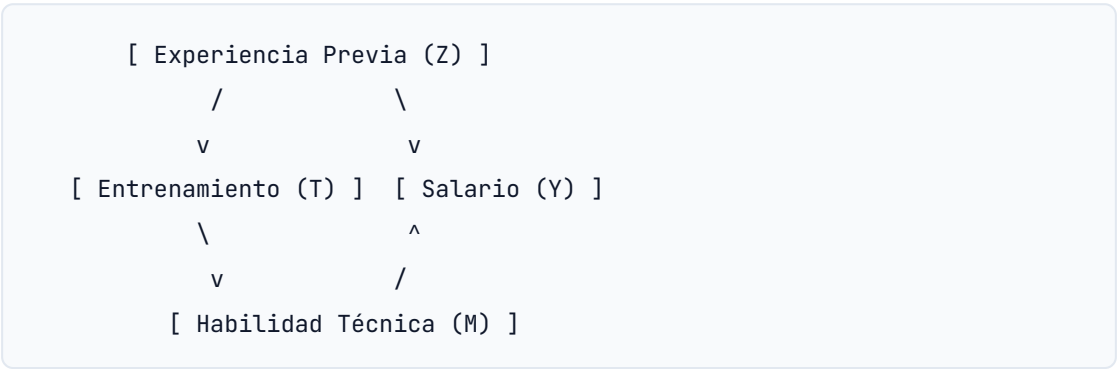
💡 Respuesta / Guía de solución: 1. $\mathbb{V}\text{ar}(\bar{T}) = (0.25)(12.0) + \frac{1-0.25}{100}(12.0) = 3.00 + \frac{0.75 \times 12.0}{100} = 3.00 + 0.09 = \mathbf{3.09}$ (reducción de varianza del 74.25% respecto a un árbol solo de 12.0). 2. Límite: $\lim_{B\to\infty} \mathbb{V}\text{ar}(\bar{T}) = \rho\sigma^2 = 3.00$. El subsampleo aleatorio de columnas evita que todos los árboles elijan siempre el mismo predictor dominante, reduciendo la correlación ρ y bajando aún más la cota mínima de varianza.

3.6 Guía de Ejercitación 06: Inferencia Causal, DAGs y Estimación con DoWhy

Unidad 6: Estadística Aplicada y Modelos de Regresión

Ejercicio 6.1 (Identificación de Caminos y Criterio de Ajuste Backdoor)

Inspirado en Ness, Cap. 2; Molak, Cap. 2 y 3. Considere el siguiente grafo causal que modela el impacto de un programa de entrenamiento intensivo en IA (T) sobre el salario final de los egresados (Y):



1. Identifique todos los caminos entre el tratamiento T y el resultado Y. Clasifique cada camino en causal o no causal (de puerta trasera).
2. ¿Cuál es el conjunto mínimo de variables de ajuste Z^* que satisface el **Criterio Backdoor** para estimar el efecto causal total $T \to Y$?
3. Explique qué ocurriría con la estimación si un analista incluye por error la variable M (Habilidad Técnica) en la regresión de control.

💡 Respuesta / Guía de solución: 1. Caminos: * Camino 1: $T \to M \to Y$ (Camino dirigido / causal hacia adelante). * Camino 2: $T \leftarrow Z \to Y$ (Camino de puerta trasera abierto por el confusor Z). 2. Conjunto de ajuste backdoor: $\mathbf{Z}^* = \{\mathbf{Z}\}$ (Experiencia Previa). Bloquea la causa común sin ser descendiente de T. 3. ⚠ **Sesgo de Bloqueo de Mediador (Overcontrolling Bias):** Si se controla por M, se bloquea el mecanismo causal natural a través del cual el entrenamiento impacta en el salario, subestimando gravemente el efecto total del programa.

Ejercicio 6.2 (El Peligro del Colisionador / Collider Bias)

Inspirado en Molak, Cap. 3; Ness, Cap. 3. Suponga que el talento innato (T) y la suerte (S) son independientes entre sí ($T \perp\!\!\!\perp S$). Ambos influyen en lograr el éxito en una startup (E): $T \to E \leftarrow S$. 1. Dibuje el DAG e identifique el rol del nodo E (Colisionador). 2. Un equipo de investigación decide analizar únicamente a los fundadores exitosos (condicionando en $E = 1$). ¿Qué correlación espuria se observará entre el talento y la suerte en esa submuestra? Justifique conceptualmente.

💡 Respuesta / Guía de solución: * E es un colisionador (collider). Marginalmente $T \perp\!\!\!\perp S$. * Al condicionar por $E = 1$, se abre un camino espurio que induce una **correlación negativa artificial** entre Talento y Suerte (Paradoja de Berkson): entre las personas exitosas, si alguien tiene poco talento necesariamente debió tener mucha suerte, y viceversa.

PARTE IV: TRABAJOS PRÁCTICOS OFICIALES DE EVALUACIÓN CONTINUA

4.1 Trabajo Práctico Nº 1: Exploración de Datos, Probabilidad y Simulación

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 4
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

1. Objetivos del Trabajo Práctico

- Implementar análisis exploratorio de datos (EDA) utilizando Pandas y Seaborn.
- Calcular medidas de posición y dispersión para describir conjuntos de datos.
- Modelar situaciones reales utilizando distribuciones de probabilidad continuas y discretas (Normal, Binomial, Poisson).
- Implementar simulaciones probabilísticas en Python con NumPy y Scipy para aproximar resultados teóricos.

2. Contexto del Problema: Análisis de Reclamos y Tiempos de Resolución en Soporte Técnico

Una empresa de servicios tecnológicos analiza su centro de soporte para entender el comportamiento de los tickets y reclamos de los clientes. Se cuenta con un dataset histórico (`tickets_soporte.csv`) con las siguientes variables: * `ticket_id`: Identificador único. * `tipo_incidencia`: Categoría (Software, Hardware, Red, Facturación). * `tiempo_resolucion_horas`: Tiempo continuo hasta el cierre del ticket. * `satisfaccion_cliente`: Puntaje discreto del 1 al 5.

3. Consignas a Desarrollar

Parte A: Análisis Exploratorio de Datos (EDA)

1. **Limpieza y Estructura:** Cargue el dataset con Pandas, describa su dimensionalidad y trate los valores nulos (si los hubiera).

2. **Medidas Descriptivas:** Calcule la media, mediana, desviación estándar, asimetría y curtosis para el `tiempo_resolucion_horas`, agrupando por `tipo_incidencia`.
3. **Visualización:** Realice un gráfico de cajas (box-plot) y un histograma para contrastar visualmente la distribución de tiempos de resolución por cada tipo de incidencia. ¿Qué categoría presenta mayor variabilidad?

Parte B: Modelado Probabilístico Teórico

Asuma los siguientes parámetros poblacionales históricos para responder analíticamente: 1. **Distribución Normal:** El tiempo de resolución en incidencias de "Software" sigue una distribución $N(\mu = 4.5, \sigma = 1.2)$ horas. Calcule la probabilidad de que un ticket demore más de 6 horas. 2. **Distribución de Poisson:** Los tickets de la categoría "Red" ingresan al sistema a una tasa de $\lambda = 3$ tickets por hora. Calcule la probabilidad de recibir exactamente 5 tickets en una hora determinada. 3. **Distribución Binomial:** Históricamente, el 85% de los clientes califican con 4 o 5 puntos (Alta Satisfacción). Si se toma una muestra aleatoria de 20 tickets cerrados, ¿cuál es la probabilidad de que al menos 18 tengan "Alta Satisfacción"?

Parte C: Simulación Estocástica (Monte Carlo)

1. Desarrolle una simulación en Python (usando `numpy.random`) para el escenario de "Software" (Distribución Normal). Simule $N = 10.000$ tickets.
2. Calcule la proporción empírica de tickets que demoraron más de 6 horas y compárela con el valor teórico obtenido en la Parte B (1).
3. Calcule el Error Cuadrático Medio (MSE) entre el valor empírico simulado y el analítico.

---

4. Criterios de Evaluación y Rúbrica (100 pts)

- **[30 pts]** Calidad del Análisis Exploratorio (EDA) y correctas interpretaciones de medidas descriptivas y gráficos.
- **[35 pts]** Precisión en los cálculos teóricos de probabilidad (NumPy / SciPy).
- **[20 pts]** Correcta implementación de la simulación de Monte Carlo y análisis comparativo empírico vs teórico.
- **[15 pts]** Claridad del código Python, presentación en Notebook y redacción técnica de conclusiones.

4.2 Trabajo Práctico Nº 2: Inferencia Estadística, Contraste de Hipótesis y ANOVA

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 4
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

---

1. Objetivos del Trabajo Práctico

- Formular y resolver ensayos de hipótesis paramétricos y no paramétricos sobre conjuntos de datos reales.
- Validar rigurosamente los supuestos inferenciales: Normalidad (Shapiro-Wilk, Q-Q Plot) y Homocedasticidad (Levene).
- Implementar e interpretar el Análisis de Varianza (ANOVA de 1 factor) y comparaciones post-hoc (Tukey HSD).
- Tomar decisiones basadas en datos controlando el nivel de significación α y evaluando la potencia del test.

---

2. Contexto del Problema: Benchmark de Motores de Inferencia para LLMs

Una empresa de tecnología evalúa 4 plataformas de inferencia local para desplegar Small Language Models (SLMs de 7B parámetros): * **Motor 1:** Ollama (Default C++ backend) * **Motor 2:** vLLM (PagedAttention) * **Motor 3:** TGI (Text Generation Inference) * **Motor 4:** Llama.cpp (Metal/CUDA optimized)

Se diseñó un experimento aleatorizado midiendo la velocidad de generación (Tokens por Segundo, tok/s) sobre $n = 30$ prompts idénticos de benchmarking por cada motor ($N = 120$ mediciones en total).

---

3. Consignas a Desarrollar

Parte A: Análisis Exploratorio y Validación de Supuestos

1. **Estadística Descriptiva:** Genere un DataFrame con las mediciones y calcule media, desvío estándar, mediana, rango intercuartílico (IQR) e intervalos de confianza del 95% para la media de cada motor.

2. **Pruebas de Normalidad:** Para cada uno de los 4 grupos, ejecute el test de Shapiro-Wilk y grafique el Q-Q Plot. Concluya si se sostiene el supuesto de normalidad ($\alpha = 0.05$).
3. **Prueba de Homocedasticidad:** Ejecute el test de Levene para verificar la igualdad de varianzas entre los 4 tratamientos.

Parte B: Comparaciones de 2 Poblaciones

1. **Comparación Directa Ollama vs vLLM:** Realice un test t de Student para muestras independientes para determinar si vLLM supera significativamente a Ollama en velocidad de generación. Seleccione la variante adecuada (Pooled o Welch) según el resultado del test de varianzas.
2. **Análisis de Potencia:** Si se quisiera detectar una diferencia mínima de $2.0\ \mathrm{tok/s}$ con una potencia del 90% ($\beta = 0.10$) al nivel $\alpha = 0.01$, ¿cuál debería haber sido el tamaño de muestra n?

Parte C: ANOVA de 1 Factor y Contrastes Post-Hoc

1. **Tabla ANOVA:** Ejecute el Análisis de Varianza global ($H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4$). Calcule SSA, SSE, SST, MSA, MSE y el estadístico F.
2. **Decisión e Interpretación:** Determine si hay diferencias estadísticamente significativas con un nivel de significación $\alpha = 0.01$.
3. **Comparaciones Múltiples de Tukey HSD:** Si el ANOVA resulta significativo, ejecute el test de Tukey HSD para identificar cuáles pares de motores difieren entre sí. Elabore una recomendación técnica justificada para la empresa.

---

4. Rúbrica de Evaluación (100 pts)

- **[20 pts]** Rigor en la validación de supuestos (Shapiro-Wilk, Q-Q Plot, Levene).
- **[30 pts]** Correcta formulación de hipótesis estadísticas y ejecución de tests de 2 poblaciones.
- **[30 pts]** Tabla ANOVA completa, contrastes de Tukey e interpretación del p-valor.
- **[20 pts]** Calidad del código en Python (`scipy.stats`, `statsmodels`) y claridad en las conclusiones.

4.3 Trabajo Práctico Nº 3: Regresión Lineal Múltiple, Residuos e Ingeniería PCA

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 8
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

---

1. Objetivos del Trabajo Práctico

- Estimar, evaluar e interpretar modelos de regresión lineal múltiple con Scikit-Learn y Statsmodels.
- Realizar un diagnóstico exhaustivo de los residuos: normalidad, homocedasticidad, valores atípicos (outliers) y puntos influyentes (Distancia de Cook).
- Diagnosticar y mitigar problemas de multicolinealidad mediante el Factor de Inflación de la Varianza (VIF).
- Aplicar Análisis de Componentes Principales (PCA) como técnica de reducción de dimensionalidad y preparación de características.

---

2. Contexto del Problema: Tasación Predictiva y Valuación Inmobiliaria

Se dispone de un dataset de propiedades inmobiliarias (`properati_caba.csv` o similar) con variables cuantitativas y cualitativas: * Variable dependiente (Y): `price_usd` (Precio en dólares). * Predictoras (X): `surface_total` (m^2 totales), `surface_covered` (m^2 cubiertos), `rooms` (ambientes), `bathrooms` (baños), `antiquity` (antigüedad en años), `distance_subway_meters` (distancia al subte).

---

3. Consignas a Desarrollar

Parte A: Regresión Lineal Simple y Diagnóstico Preliminar

1. Ajuste un modelo de regresión lineal simple $Y = \beta_0 + \beta_1 \cdot \text{surface_total} + \epsilon$.
2. Interprete el valor e intervalo de confianza del 95% de $\hat{\beta}_1$. Calcule el coeficiente R^2.
3. Grafique los residuos vs valores ajustados ($\hat{Y}$) para inspeccionar visualmente posibles violaciones de homocedasticidad o no linealidad.

Parte B: Regresión Múltiple, Outliers y Multicolinealidad

1. **Modelo Múltiple:** Ajuste el modelo lineal completo con todas las variables numéricas disponibles.
2. **Evaluación de Multicolinealidad (VIF):** Calcule el VIF para cada variable predictora. Si detecta $\text{VIF} > 5$, justifique qué variables presentan redundancia informativa y proponga una solución.
3. **Detección de Outliers e Influyentes:** Calcule los residuos estandarizados/estudentizados y la **Distancia de Cook**. Identifique y justifique el tratamiento de observaciones con apalancamiento (leverage) o influencia desmedida.
4. **Validación de Residuos:** Aplique el test de Shapiro-Wilk a los residuos y el test de Breusch-Pagan / White para heterocedasticidad.

Parte C: Reducción de Dimensionalidad con PCA

1. Estandarice las variables predictoras (`StandardScaler`).
2. Aplique PCA sobre las variables explicativas y grafique la curva de **Varianza Explicada Acumulada** (Scree Plot).
3. Determine cuántos componentes principales son necesarios para retener al menos el 85% de la varianza total.
4. Ajuste una **Regresión por Componentes Principales (PCR)** y compare su rendimiento (R^2 ajustado y RMSE en test) frente al modelo múltiple original.

---

4. Rúbrica de Evaluación (100 pts)

- **[25 pts]** Ajuste e interpretación rigurosa de coeficientes MCO y métricas de bondad de ajuste (R^2, RMSE).
- **[25 pts]** Diagnóstico exhaustivo de supuestos (Residuos, Breusch-Pagan, VIF, Distancia de Cook).
- **[30 pts]** Correcta implementación de PCA, scree plot y regresión sobre componentes principales.
- **[20 pts]** Estructura del código, reproducibilidad (`random_state`) y conclusiones técnicas.

4.4 Trabajo Práctico Nº 4: Regresión Logística y Clasificación Naive Bayes

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 8
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

---

1. Objetivos del Trabajo Práctico

- Implementar y evaluar modelos de **Regresión Logística** para problemas de clasificación binaria y multiclase.
- Interpretar coeficientes, funciones logit y **Odds Ratios (OR)** en el contexto de negocio / clínico.
- Implementar y comparar el clasificador probabilístico **Gaussian y Multinomial Naive Bayes**.
- Analizar matrices de confusión, curvas ROC y métricas de rendimiento en escenarios con balance y desbalance de clases.

---

2. Contexto del Problema: Predicción de Riesgo Crediticio / Default Financiero

Se analiza una cartera de clientes bancarios (`credit_risk_dataset.csv`) con el objetivo de predecir la probabilidad de incumplimiento de pago (`loan_status`: $1 = \text{Default}, 0 = \text{Al día}$).

Variables disponibles: * `person_age`: Edad del solicitante. * `person_income`: Ingreso anual. * `person_home_ownership`: Tipo de vivienda (RENT, OWN, MORTGAGE). * `loan_amnt`: Monto del préstamo solicitado. * `loan_int_rate`: Tasa de interés asignada. * `loan_percent_income`: Porcentaje del ingreso comprometido con el préstamo. * `cb_person_default_on_file`: Historial crediticio previo de default (Y/N).

---

3. Consignas a Desarrollar

Parte A: Preprocesamiento y Regresión Logística

1. **Tratamiento de Datos:** Realice imputación de valores faltantes (si existieren), codifique variables categóricas (`OneHotEncoder` o

`pd.get_dummies(drop_first=True)`) y divida en Train (80%) y Test (20%) de forma estratificada.

2. **Ajuste del Modelo con Statsmodels:** Ajuste el modelo de regresión logística usando `statsmodels.api.Logit` para obtener la tabla completa con errores estándar, z-scores y p-valores de cada predictor.
3. **Interpretación de Odds Ratios:** Calcule los Odds Ratios ($e^{\hat{\beta}_j}$) y sus respectivos intervalos de confianza del 95%. Interprete detalladamente el impacto de `loan_percent_income` y `cb_person_default_on_file` sobre el riesgo de default.

Parte B: Clasificador Naive Bayes

1. **Ajuste con Scikit-Learn:** Entrene un clasificador `GaussianNB` sobre el conjunto de entrenamiento estandarizado.
2. **Evaluación de Probabilidades a Posteriori:** Obtenga las probabilidades predichas (`predict_proba`) para las primeras 5 observaciones de test y explique cómo opera el principio del producto de verosimilitudes condicionales.

Parte C: Comparación de Modelos y Curvas ROC / AUC

1. **Matrices de Confusión:** Grafique las matrices de confusión de ambos modelos en el conjunto de test para el umbral estándar $\tau = 0.50$. Calcule Accuracy, Precision, Recall y F1-Score.
2. **Curva ROC y AUC:** Grafique en un mismo gráfico las curvas ROC de la Regresión Logística y de Naive Bayes. Calcule el área bajo la curva (AUC-ROC) de cada modelo.
3. **Análisis de Umbral Óptimo:** En un problema de riesgo crediticio donde un Falso Negativo (prestarle a alguien que no paga) es 5 veces más costoso que un Falso Positivo (rechazar a un buen pagador), proponga y justifique un nuevo umbral de corte τ^* que optimice la función de pérdida del banco.

---

4. Rúbrica de Evaluación (100 pts)

- **[30 pts]** Ajuste de Regresión Logística, cálculo e interpretación rigurosa de los Odds Ratios.
- **[25 pts]** Implementación y comprensión teórica del algoritmo Naive Bayes.
- **[25 pts]** Análisis comparativo de métricas (ROC, AUC, Precision, Recall) y optimización de umbrales.
- **[20 pts]** Claridad del informe, visualizaciones de curvas y conclusiones orientadas a la toma de decisiones.

4.5 Trabajo Práctico Nº 5: Árboles de Decisión, Random Forest y Series Temporales

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 11
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

---

1. Objetivos del Trabajo Práctico

- Entrenar y visualizar **Árboles de Decisión** (CART) para regresión y clasificación.
- Implementar ensambles **Random Forest**, evaluando la reducción de varianza y la importancia de características (Feature Importance).
- Aplicar **Validación Cruzada (K-Fold)** y optimización de hiperparámetros con `GridSearchCV`.
- Realizar un análisis introductorio de **Series de Tiempo** (descomposición y modelo ARIMA).

---

2. Contexto del Problema: Predicción de Demanda Energética y Consumo

Se dispone de un registro horario de consumo de energía eléctrica (`energy_consumption_timeseries.csv`) junto con variables climáticas y de calendario: * Variable dependiente: `energy_demand_mw` (Demanda en Megawatts). * Predictoras: `temperature_celsius`, `humidity_percent`, `wind_speed_kmh`, `is_weekend`, `hour_of_day`, `month`.

---

3. Consignas a Desarrollar

Parte A: Árbol de Regresión Simple y Poda

1. **Ajuste del Árbol:** Entrene un `DecisionTreeRegressor` sin límite de profundidad (`max_depth=None`). Evalúe el R^2 y RMSE tanto en Train como en Test. Identifique la presencia de sobreajuste (overfitting).
2. **Visualización e Interpretación:** Grafique la estructura de los primeros 3 niveles del árbol con `plot_tree`. Explique cómo se seleccionaron los puntos de corte en el nodo raíz.

3. **Poda y Regularización:** Evalúe el impacto de limitar `max_depth` (de 2 a 15) y `min_samples_leaf` sobre las métricas en test.

Parte B: Ensamble Random Forest y Búsqueda de Hiperparámetros

1. **Random Forest Regressor:** Entrene un ensamble `RandomForestRegressor` con $B = 100$ árboles. Compare su rendimiento en test frente al árbol individual podado y a una regresión lineal múltiple estándar.
2. **GridSearchCV:** Diseñe una grilla de hiperparámetros con Validación Cruzada de 5 pliegues (`KFold(n_splits=5, shuffle=True)`):
   - `n_estimators` : [50, 100, 200]
   - `max_depth` : [5, 10, 15, None]
   - `max_features` : ['sqrt', 'log2', 0.5]
3. **Feature Importance:** Extraiga y grafique la importancia de variables por Gini/ MDI y mediante Permutation Importance. ¿Cuáles son los 3 factores que más condicionan la demanda energética?

Parte C: Modelado de Series de Tiempo (ARIMA)

1. **Descomposición Temporal:** Convierta la serie temporal a frecuencia diaria y realice la descomposición en Tendencia, Estacionalidad y Residuo (`seasonal_decompose`).
2. **Prueba de Estacionariedad:** Aplique la prueba de Dickey-Fuller Aumentada (`adfuller`). Si la serie no es estacionaria, aplique la primera diferencia ($d = 1$).
3. **Ajuste ARIMA:** Ajuste un modelo $\mathrm{ARIMA}(p, d, q)$ sobre la serie temporal y genere la predicción para los siguientes 14 días con intervalos de confianza del 95%.

---

4. Rúbrica de Evaluación (100 pts)

- **[25 pts]** Construcción, visualización e interpretación del árbol de decisión y diagnóstico de sobreajuste.
- **[35 pts]** Optimización con Random Forest, K-Fold CV y análisis riguroso de Feature Importance.
- **[25 pts]** Descomposición de series de tiempo, test ADF y predicción con modelo ARIMA.
- **[15 pts]** Calidad del código, reproducibilidad y recomendaciones técnicas finales.

4.6 Trabajo Práctico Nº 6: Inferencia Causal, DAGs y Causal ML

Asignatura: Estadística Aplicada y Modelos de Regresión (2026)
Fecha de Entrega: Semana 11
Modalidad: Grupal (hasta 3 integrantes) / Formato Jupyter Notebook (`.ipynb`) y Reporte Ejecutivo.

---

1. Objetivos del Trabajo Práctico

- Construir e interpretar **Grafos Acíclicos Dirigidos (DAGs)** para formalizar supuestos causales.
- Identificar variables confusoras (confounders), mediadores y colisionadores (colliders).
- Aplicar el **Criterio de Ajuste Backdoor** para eliminar sesgos en modelos de regresión.
- Estimar efectos causales (ATE y CATE) utilizando las librerías `DoWhy` y `EconML`.
- Ejecutar pruebas de refutación estocástica (Refutation Tests) para evaluar la robustez del efecto causal estimado.

---

2. Contexto del Problema: Evaluación de Impacto de un Nuevo Algoritmo de Recomendación

Una plataforma de e-commerce implementó un nuevo motor de recomendación personalizado ($T = 1$: Algoritmo Nuevo, $T = 0$: Algoritmo Clásico) sobre una muestra de $N = 10.000$ usuarios. Se busca medir el **Efecto Causal Promedio (ATE)** del nuevo algoritmo sobre el monto total gastado en el mes (Y, en USD).

Variables observadas: * `treatment` (T): Asignación del nuevo algoritmo ($0/1$). * `revenue` (Y): Gasto mensual del usuario. * `user_tenure_months` (Z_1): Antigüedad del usuario en la plataforma. * `monthly_active_days` (Z_2): Días de actividad mensual previa. * `device_type` (Z_3): Tipo de dispositivo (Móvil vs Desktop). * `clicks_on_recommendations` (M): Cantidad de clics en las recomendaciones (Mediador).

---

3. Consignas a Desarrollar

Parte A: Modelado Causal y Construcción del DAG

1. **Especificación del Grafo:** Construya el DAG causal utilizando la librería `networkx` o el formato DOT de `DoWhy`.
   - Incluya los arcos causales teóricos: $Z \to T$, $Z \to Y$, $T \to M \to Y$.
2. **Análisis de Caminos:** Identifique analíticamente:
   - Los caminos causales directos e indirectos.
   - Los caminos de puerta trasera (backdoor paths).
   - El conjunto mínimo de ajuste suficiente para estimar el ATE sin sesgo de mediador ni de colisionador.

Parte B: Estimación Causal en 4 Pasos con DoWhy

1. **Paso 1 (Model):** Cree el objeto `CausalModel` en DoWhy asociando el DataFrame y el grafo DAG.
2. **Paso 2 (Identify):** Ejecute `model.identify_effect(proceed_when_unidentifiable=True)` mediante el criterio Backdoor. Muestre la expresión identificada.
3. **Paso 3 (Estimate):**
   - Estime el ATE mediante una **Regresión Lineal con Ajuste de Covariables** (`method_name="backdoor.linear_regression"`).
   - Estime el ATE mediante **Propensity Score Matching / Weighting** (`method_name="backdoor.propensity_score_weighting"`).
   - Compare los estimadores causales frente a una regresión ingenua (no controlada) $Y \sim T$. Explique la magnitud del sesgo de confusión eliminado.

Parte C: Causal Machine Learning y Heterogeneidad de Efectos (EconML)

1. **Double Machine Learning (DML):** Utilice el estimador `LinearDML` de EconML para modelar la no linealidad en los confusores usando Random Forests como modelos base.
2. **Heterogeneidad de Efecto (CATE):** Evalúe si el efecto del nuevo algoritmo varía según la antigüedad del usuario (`user_tenure_months`). Grafique la curva de CATE estimado vs antigüedad.

Parte D: Pruebas de Refutación (Sensibilidad y Robustez)

Ejecute al menos dos pruebas de refutación en DoWhy: 1. **Añadir Causa Común Aleatoria (Random Common Cause):** ¿El efecto estimado cambia significativamente al agregar ruido blanco como confusor? 2. **Tratamiento Placebo (Placebo Treatment Refuter):** Al reemplazar T por una variable aleatoria, ¿el efecto causal estimado cae a cero?

---

4. Rúbrica de Evaluación (100 pts)

- **[25 pts]** Correcta formulación del DAG, identificación de confusores y aplicación del criterio Backdoor.
- **[30 pts]** Flujo completo de estimación en DoWhy (Linear Regression, Propensity Scores).
- **[25 pts]** Estimación de CATE con Double Machine Learning en EconML e interpretación de heterogeneidad.
- **[20 pts]** Pruebas de refutación, rigor en las conclusiones y distinción explícita entre asociación y causalidad.

PARTE V: TRABAJO PRÁCTICO INTEGRADOR (PROYECTO CAPSTONE)

5.1 Enunciado del Proyecto Integrador End-to-End

Cátedra: Estadística Aplicada y Modelos de Regresión (2026)
Profesor: Ivan Gonzalo Vilaboa
Entrega y Defensa Oral: Semana 13
Modalidad: Grupal (hasta 3 integrantes) / Repositorio GitHub, Notebook reproducible y Defensa Oral con Presentación Ejecutiva.

---

1. Fundamentación y Propósito del Proyecto

El **Trabajo Práctico Integrador (TPI)** constituye la instancia evaluativa final y articuladora de la materia. Los equipos deberán abordar un problema real y complejo de ciencia de datos, integrando de manera end-to-end: 1. El **rigor analítico de la inferencia estadística clásica** (contrastes de hipótesis, intervalos de confianza, validación de supuestos y ANOVA). 2. El **poder predictivo del Machine**

Learning moderno (regresión lineal múltiple, regularización, PCA, regresión logística, árboles de decisión y Random Forest con validación cruzada). 3. La **lucidez metodológica de la Inferencia Causal** (construcción de DAGs, criterio backdoor, estimación de efectos causales y pruebas de refutación).

---

2. Dominio y Selección del Dataset

Cada equipo seleccionará un conjunto de datos tabular real (con un mínimo de $N \geq 1.000$ observaciones y al menos 8 variables cuantitativas/cualitativas) perteneciente a uno de los siguientes dominios sugeridos: * **Salud y Epidemiología:** Factores determinantes en la progresión clínica de pacientes / impacto de tratamientos. * **Finanzas y Riesgo:** Modelado de solvencia crediticia, fijación de precios y análisis de fraude. * **Economía Urbana e Inmobiliaria:** Valuación hedónica de propiedades (ej. dataset Properati) y accesibilidad a servicios. * **Energía y Sustentabilidad:** Predicción y descomposición de demanda energética / impacto de políticas ambientales.

---

3. Estructura Obligatoria del Trabajo (5 Hitos)

```
[ Hito 1: EDA & Inferencia Clásica ]  ⟶ Test de normalidad, hipótesis y ANOVA sobre grupos.
                 |
[ Hito 2: Preparación & PCA ]         ⟶ Imputación, outliers, estandarización y reducción PCA.
                 |
[ Hito 3: Modelos de Regresión & ML ] ⟶ Regresión lineal vs Árboles vs Random Forest con K-Fold CV.
                 |
[ Hito 4: Inferencia Causal (DoWhy) ] ⟶ DAG causal, ajuste de confusores y estimación de ATE/CATE.
                 |
[ Hito 5: Comunicación y Defensa ]    ⟶ Reporte ejecutivo reproducible y presentación oral ante el tribunal.
```

Hito 1: Análisis Exploratorio e Inferencia Estadística (30% del contenido)

- Limpieza de datos, imputación y análisis univariado/bivariado.
- Formulación y ejecución de al menos **dos ensayos de hipótesis paramétricos/no paramétricos** justificados según las preguntas de negocio.

- Aplicación de un **ANOVA de 1 Factor** para comparar la variable principal a través de distintos subgrupos poblacionales, validando supuestos de normalidad y homocedasticidad (Levene).

Hito 2: Ingeniería de Características y Reducción de Dimensionalidad

- Detección y tratamiento de outliers e influyentes (Distancia de Cook).
- Diagnóstico de multicolinealidad (VIF).
- Implementación de **PCA** con scree plot e interpretación de la varianza explicada.

Hito 3: Modelado Predictivo Comparativo y Validación Cruzada

- Ajuste de al menos 3 familias de modelos:
  1. Modelo Lineal / Regularizado (Ridge / Lasso / Logística).
  2. Árbol de Decisión Podado (CART).
  3. Ensamble Random Forest.
- Optimización de hiperparámetros mediante `GridSearchCV` con K-Fold Cross-Validation (5 pliegues).
- Comparación rigurosa de métricas de generalización en el conjunto de Test (R^2, RMSE, MAE o AUC-ROC, Precision, Recall, F1).
- Análisis de Importancia de Variables (Feature Importance y Permutation Importance).

Hito 4: Inferencia Causal y Evaluación de Políticas (Causal ML)

- Formulación del **Grafo Acíclico Dirigido (DAG)** que exprese los supuestos estructurales del dominio.
- Identificación de una relación de tratamiento $\rightarrow$ resultado ($T \rightarrow Y$) de interés estratégico.
- Estimación del efecto causal promedio (ATE) aplicando el **Criterio Backdoor** en `DoWhy`.
- Pruebas de refutación (Random Common Cause y Placebo Treatment).
- Discusión reflexiva: ¿Cómo difiere la conclusión puramente predictiva (Hito 3) de la conclusión causal (Hito 4)?

Hito 5: Entregables y Defensa Oral

1. **Repositorio GitHub:** Código modular, limpio, documentado y reproducible (`requirements.txt`).
2. **Notebook Principal (`TP_Integrador_Final.ipynb`):** Estructurado según los 4 hitos técnicos con narrativa fluida.
3. **Presentación Ejecutiva (Slides PDF):** 10-12 diapositivas para la defensa oral de 15 minutos en clase.

4. Rúbrica de Calificación (100 puntos totales)

Dimensión de Evaluación	Puntaje	Criterios de Excelencia (Sobresaliente)
1. Inferencia Clásica y Diseño	**20 pts**	Hipótesis bien formuladas, validación rigurosa de supuestos (Levene, Shapiro) y ANOVA correctamente interpretado.
2. Preparación de Datos y PCA	**15 pts**	Tratamiento justificado de valores faltantes y outliers; cálculo de VIF y reducción PCA coherente.
3. Modelado Predictivo y ML	**25 pts**	Comparación metodológica impecable entre modelos lineales, árboles y ensembles con K-Fold CV y métricas de test.
4. Inferencia Causal y DoWhy	**25 pts**	DAG conceptualmente sólido, aplicación estricta del criterio backdoor, estimación de ATE y pruebas de refutación superadas.
5. Comunicación y Defensa Oral	**15 pts**	Código limpio y reproducible en GitHub; defensa oral clara, dominio técnico individual y respuestas fundamentadas.

5.2 Rúbrica Detallada de Calificación y Criterios de Evaluación

Cátedra: Estadística Aplicada y Modelos de Regresión (2026)

Esta rúbrica establece los estándares de corrección para el informe técnico, el código fuente y la defensa oral del proyecto final.

Escala de Ponderación por Niveles

Criterio	Insuficiente (< 60%)	Aprobado (60-74%)	Notable (75-89%)	Sobresaliente (90-100%)
Inferencia Estadística y ANOVA (20 pts)	Omite supuestos o aplica tests incorrectos sin justificación.	Aplica tests estándar pero no valida supuestos (Levene / normalidad).	Valida supuestos y aplica ANOVA con leves errores de interpretación.	Diseño inferencial impecable, validación de supuestos, ANOVA y contrastes post-hoc con conclusiones claras.
Preparación de Datos y PCA (15 pts)	No detecta outliers ni multicolinealidad.	Tratamiento básico de datos sin analizar multicolinealidad (VIF).	Diagnóstico de VIF y outliers correcto; aplica PCA de forma mecánica.	Limpieza exhaustiva, diagnóstico de VIF/Cook's distance y PCA con fundamentación de varianza explicada.
Modelado Predictivo y ML (25 pts)	Entrena un único modelo sin división Train/ Test o con sobreajuste evidente.	Compara modelos con división simple pero sin validación cruzada sistemática.	Aplica K-Fold CV y compara métricas estándar en Test adecuadamente.	Comparación rigurosa de 3 familias con GridSearchCV, K-Fold CV, análisis de feature importance y curvas de aprendizaje.
Inferencia Causal y Causal ML (25 pts)	Confunde correlación con causalidad; no formula DAG.	Define un DAG elemental pero comete errores al elegir variables de ajuste.	Aplica criterio backdoor y estima ATE con DoWhy adecuadamente.	DAG conceptualmente robusto, criterio backdoor estricto, estimación en DoWhy/EconML y pruebas de refutación completas.
Código, Reproducibilidad	Código desordenado, no	Código ejecutable pero	Repositorio limpio con `requirements.txt`;	Código modular y profesional en

Criterio	Insuficiente (< 60%)	Aprobado (60-74%)	Notable (75-89%)	Sobresaliente (90-100%)
y Defensa (15 pts)	reproducible; defensa deficiente.	poco documentado; defensa aceptable.	exposición clara del equipo.	GitHub; defensa oral brillante con dominio técnico de todos los integrantes.