



# Programa de asignatura

Nombre de la asignatura	Métodos de Aprendizaje automático en Data Science		
Nombre de la asignatura en inglés	Machine Learning Methods in Data Science		
Unidad que dicta la asignatura	Facultad de Ingeniería y Ciencias		
Versión Año	2025	Código de asignatura	DIIO07
Nro. de Créditos SCT-Chile	6		
Horas Presenciales	Horas No Presenciales	Total de horas académicas	
45 = horas lectivas	90	135 = horas autónomas	
Área de conocimiento de la asignatura	Data Science		
Tipo de Asignatura	Asignatura obligatoria		
Pre-requisitos	Álgebra lineal, cálculo multivariado, probabilidades, estadística, programación en Python.		
Competencias del egresado que desarrolla la asignatura	De acuerdo al perfil de egreso declarado en la memoria, esta asignatura tributa a las siguientes competencias: C1. Analiza y resuelve problemas de naturaleza conceptual y teórica. C2. Identifica problemas prácticas o investigaciones con estrategias de solución. C3. Se desenvuelve en forma proactiva en equipo para desarrollar y proponer soluciones.		
Resultados de Aprendizaje	RA1 Analizar los fundamentos teóricos y matemáticos de algoritmos de aprendizaje automático mediante el estudio de teoría de optimización, estadística y álgebra lineal para comprender las bases conceptuales que sustentan métodos supervisados y no supervisados en diversos contextos de aplicación.  RA2. Implementar algoritmos de aprendizaje automático utilizando librerías especializadas para resolver problemas de clasificación, regresión y clustering mediante el desarrollo de pipelines completos que incluyan: preprocesamiento, entrenamiento, validación y evaluación de modelo.  RA3. Evaluar algoritmos de aprendizaje automático mediante el análisis comparativo de métricas de rendimiento, complejidad computacional y		

	<p>características del problema para optimizar la solución en dominios específicos.</p> <p>RA4. Aplicar algoritmos de aprendizaje no supervisado a través de librerías especializadas en Python (ej. scikit-learn) para analizar y encontrar agrupaciones de datos y representaciones de alto nivel de datos multidimensionales en diversos contextos de aplicación.</p>
<p>Descripción de la asignatura</p> <p>Estructura: Contexto →</p> <p>Desafío → Contenido →</p> <p>Metodología → Proyección</p>	<p>El curso de Aprendizaje Automático desafía a los estudiantes de doctorado y master a dominar tanto los fundamentos matemáticos como las implementaciones prácticas de algoritmos de machine learning supervisado y no supervisado.</p> <p>Los participantes explorarán desde la teoría de la optimización convexa en SVMs hasta la potencia computacional de las redes neuronales profundas, desarrollando una comprensión integral que abarca árboles de decisión, métodos de ensemble, modelos probabilísticos y técnicas de aprendizaje no supervisado. A través de implementaciones desde cero y el uso de frameworks modernos como TensorFlow y ScikitLearn, los estudiantes construirán pipelines de ML y desarrollarán la capacidad crítica para seleccionar y evaluar algoritmos apropiados según el contexto del problema, preparándolos para liderar investigación en ciencia de datos.</p>
Unidades de la asignatura	
<p><b>UNIDAD I: Fundamentos Teóricos del Machine Learning</b></p> <p>Contenido: Unidad 1 actual (Introduction to ML) Clases: 1-2</p> <p>RA Correspondiente: RA1 - <i>Analizar los fundamentos teóricos y matemáticos</i></p> <p>1.1 Definición formal de aprendizaje automático y tipos de problemas</p> <p>1.2 Marco teórico PAC (Probably Approximately Correct) learning</p> <p>1.3 Bias-variance tradeoff y su impacto en el rendimiento del modelo</p> <p>1.4 Complejidad computacional y de muestra en algoritmos de ML</p> <p>1.5 No Free Lunch Theorem y sus implicaciones</p> <p>Metodología y Evaluación</p> <p>1.6 Protocolo experimental: train/validation/test splits y cross-validation</p> <p>1.7 Métricas de evaluación para clasificación, regresión y clustering</p> <p>1.8 Técnicas de validación: k-fold, stratified, time series split</p> <p>1.9 Detección y manejo de overfitting y underfitting</p> <p>1.10 Curvas de aprendizaje y validación</p> <p><b>UNIDAD II: Aprendizaje No Supervisado</b></p> <p>Contenido: Unsupervised Learning. Clases: 3-7.</p> <p>RA Correspondiente: RA4 - <i>Aplicar algoritmos de aprendizaje no supervisado.</i></p> <p>Clustering</p> <p>2.1 K-means: algoritmo, convergencia, elección de K, fuzzy c-means, mean-shift.</p>	

2.2 Hierarchical clustering: linkage criteria, dendrogramas

2.3 DBSCAN: density-based clustering

2.4 Métricas de evaluación: silhouette, adjusted rand index

Reducción de Dimensionalidad

2.6 Principal Component Analysis (PCA): eigenvalues, eigenvectors

2.7 t-SNE: preservación de estructura local

2.8 Algoritmo Expectation-Maximization

Implementación Práctica:

Implementación de K-means y PCA desde cero

Uso de scikit-learn para clustering y dimensionality reduction

Visualización de clusters y componentes principales

Datasets sugeridos: Customer segmentation data, Single-cell RNA-seq, Document collections (Reuters),  
Image datasets para PCA/t-SNE.

### **UNIDAD III: Métodos Basados en Árboles**

Contenido: Decision Trees + Ensembles. Clases: 8-9.

Correspondiente: RA2 - Implementar algoritmos con librerías Python

Decision Trees Fundamentos Teóricos

2.1.1 Teoría de la información: entropía, ganancia de información, índice Gini

2.1.2 Algoritmos de construcción: ID3, C4.5, CART (de preferencia CART)

2.1.3 Criterios de división y stopping conditions

2.1.4 Pruning: pre-pruning y post-pruning strategies

2.1.5 Interpretabilidad vs. complejidad del modelo

2.1.6 Árboles de regresión y métricas de error

2.1.7 Regularización en árboles de decisión

Ensembles of Decision Trees

2.2.1 Bagging: Bootstrap Aggregating y Random Forest: selección aleatoria de features y muestras

2.2.2 Boosting: Gradient Boosting, XGBoost, LightGBM (optimización secuencial )

2.2.3 Stacking y blending techniques

2.2.4 Análisis teórico de reducción de varianza y bias (regularización y optimizaciones computacionales )

2.2.5 Diversidad en ensembles y correlation among learners

### **UNIDAD IV: Métodos Avanzados de Aprendizaje Supervisado**

Contenido: Support Vector Machine + Neural Networks. Clases: 10-14.

RA Correspondiente: RA3 - Evaluar y seleccionar algoritmos apropiados

Support Vector Machines

4.1.1 Teoría de optimización convexa aplicada a SVM

4.1.2 repaso: Problema primal y dual: Lagrange multipliers, Condiciones KKT (Karush-Kuhn-Tucker)

4.1.3 Margen máximo y vectores de soporte

4.1.4 Kernel trick y funciones kernel válidas

4.1.5 Kernels comunes: linear, polynomial, RBF, sigmoid

Neural Networks

Fundamentos Teóricos

4.1.1 Perceptron y multilayer perceptrons

4.2.2 Universal approximation theorem

4.3.3 Deep feedforward networks

- 4.3.4 Backpropagation algorithm: derivación matemática
- 4.4.5 Regularización: dropout, batch normalization, weight decay
- 4.5.6 Gradient descent variants: SGD, Adam, RMSprop
- 4.6.7 Análisis de funciones de pérdida y convergencia.

#### Estrategias de enseñanza y aprendizaje

##### 1. Clase lectiva (cátedra)

El profesor se enfoca en la presentación de conceptos clave mediante ejemplos ilustrativos. Incluye preguntas y discusiones guiadas para promover la participación. Se utiliza Python para realizar análisis de datos en tiempo real. Adicionalmente se hace un análisis guiado de papers seminales con discusión de aportes teóricos fundamentales. Si es necesario, se pueden realizar Debugging colaborativo de código con errores intencionales.

##### 2. Trabajo autónomo

El estudiante hace una revisión de presentaciones y recursos adicionales que ha indicado el profesor. Además resuelve de ejercicios prácticos en Python a modo de autoevaluación, donde se presenta un caso (en clase) y los estudiantes deben resolverlo e interpretarlo. Autoevaluación mediante ejercicios no evaluados y lectura crítica de capítulos específicos de libros de referencia.

##### 3. Simulaciones

Se diseñan casos guiados (pre-fabricados) para que sean aplicables dentro de un período corto de tiempo (durante la clase o de una clase a otra) para que los estudiantes elaboren paso a paso la construcción de árboles de decisión, k-means, perceptrón y clustering. Luego desarrollan mini-presentaciones de los resultados de las simulaciones.

##### 4. Aprendizaje basado en problemas

Casos de estudio con datos reales no balanceados (puede ser de empresa o investigación) que involucran más tiempo de análisis del equipo de estudiantes. Son desafíos más exigentes que pueden acercarse a las competencias tipo Kaggle con enfoque en metodología rigurosa. Incluye la necesidad de code reviews colaborativos entre pares.

#### Procedimientos de Evaluación de aprendizajes

1. Dos pruebas individuales de preguntas de conceptos y teóricas, abiertas relativas a los contenidos de las unidades de la asignatura. Esto representará un  $\frac{2}{3}$  de la nota final ( $\frac{1}{3}$  cada prueba).
2. Trabajos aplicado: Crear y presentar la aplicación de algoritmos a problemas específicos o Casos de estudios dados por el profesor de empresa o investigación. Incluye la presentación o exposición de resultados, justificaciones de elección de algoritmo y resultados de desempeño del aprendizaje supervisado/no supervisado. El promedio de estos trabajos representará un  $\frac{1}{3}$  de la nota final.

Nota final mínima para aprobar el curso : 4.0.

#### Recursos de aprendizaje

##### Teórico:

- Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- The Elements of Statistical Learning: Data Mining, Inference, and Prediction. (2nd edition) by Trevor Hastie, Robert Tibshirani, Jerome Friedman

- Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville
- An Introduction to Statistical Learning (with Applications in R) by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
- Gaussian Processes for Machine Learning by Carl Edward Rasmussen and Christopher K. I. Williams

**Práctico:**

- Apuntes entregados por el profesor.
- Introduction to Machine Learning with Python by Andreas Mueller and Sarah Guido.
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow (2nd edition) by Aurélien Géron.
- Deep Learning with Python by François Chollet.
- Python machine learning by Sebastian Raschka
- Python for Data Analysis by Wes McKinney