



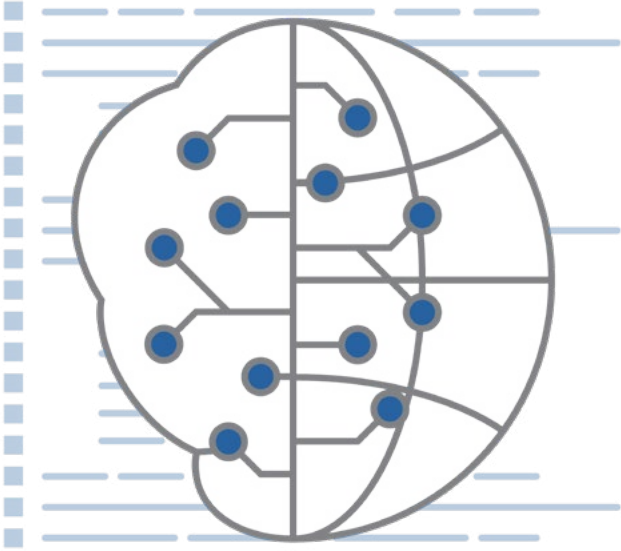
PROMiDAT
IBEROAMERICANO

Programa Iberoamericano de
Formación en Minería de Datos

PROGRAMA
**MÁSTER EJECUTIVO EN
CIENCIA DE DATOS E
INTELIGENCIA ARTIFICIAL¹**



¹Este programa no constituye programa de maestría académica universitaria, es más bien una certificación profesional ejecutiva.



MÁSTER EJECUTIVO EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

1. Introducción

El Máster Ejecutivo en Ciencia de Datos e Inteligencia Artificial está diseñado para capacitar a profesionales en técnicas avanzadas de ciencia de datos y “deep learning”. Los participantes aprenderán a aplicar estas habilidades en el análisis de datos, desarrollo de modelos predictivos y optimización de procesos en sus organizaciones. Este programa incluye desde fundamentos matemático-estadísticos de métodos supervisados y no supervisados hasta técnicas avanzadas en deep learning y un proyecto final de tesina.

¹Este programa no constituye programa de maestría académica universitaria, es más bien una certificación profesional ejecutiva.



2. Perfil de Entrada de los Estudiantes

Este programa está dirigido a profesionales de cualquier área que desean especializarse en ciencia de datos e inteligencia artificial, que tengan conocimientos previos en programación y analítica de datos.

Este programa es ideal para:

- Graduados de los Programas Experto en Ciencia de Datos y Experto en Inteligencia Artificial de PROMiDAT o personas que ya tienen al menos conocimientos intermedios en Programación R, Programación Python y en técnicas de la Ciencia de Datos e Inteligencia Artificial.
- Científicos de datos, analistas y profesionales de tecnología interesados en profundizar en inteligencia artificial y técnicas avanzadas de modelado.
- Profesionales en roles de análisis, gestión de datos y toma de decisiones, que deseen aplicar modelos predictivos y de optimización.
- Cualquier persona interesada en desarrollar proyectos en ciencia de datos e inteligencia artificial con un enfoque práctico y aplicado con conocimiento real de lo que está haciendo.



3. Objetivos Generales del Programa

El Máster Ejecutivo en Ciencia de Datos e Inteligencia Artificial tiene como objetivo principal formar a profesionales capaces de:

- Conocer los fundamentos matemático estadístico de la Ciencia de Datos y la Inteligencia Artificial.
- Desarrollar los conocimientos avanzados en matemática y en programación necesarios para la comprensión a fondo de la Ciencia de Datos, “machine learning” y “deep learning”.
- Implementar y optimizar modelos avanzados de “machine learning” y “deep learning”.
- Aplicar técnicas de análisis en datos reales y simulados, usando R y Python.
- Desarrollar y defender un proyecto de investigación (tesina) que demuestre su dominio en ciencia de datos e inteligencia artificial.

Este programa no constituye programa de maestría académica universitaria, es más bien una certificación profesional ejecutiva.



4. Posibilidades de Empleo

Los egresados podrán desempeñarse en roles como:

- Jefe de unidades especializadas en analítica de datos.
- Científico de Datos: Aplicar métodos avanzados de machine learning y deep learning en problemas complejos.
- Especialista en Deep Learning: Optimizar y desarrollar modelos avanzados para análisis predictivo.
- Consultor en Ciencia de Datos e IA: Asesorar a empresas en el uso efectivo de datos y algoritmos de IA para resolver problemas específicos.



5. Programa General

El Máster Ejecutivo en Ciencia de Datos e Inteligencia Artificial tiene una duración total de 10 meses y se compone de los siguientes cursos especializados:

Cursos Especializados:

1. ME-1000 Programación Paralela en R y Python.
2. ME-2000 Desarrollo Modelos Analíticos apoyados por herramientas de IA.
3. ME-3000 Álgebra Lineal Aplicada en Ciencia de Datos e Inteligencia Artificial.
4. ME-3002 Cálculo y Optimización Aplicado en Ciencia de Datos e Inteligencia Artificial.
5. ME-3004 Probabilidad y Estadística Aplicados a la Ciencia de Datos e Inteligencia Artificial.
6. ME-3008 Procesamiento de Lenguaje Natural y Agentes de Inteligencia Artificial.
7. ME-4000 Métodos No Supervisados Avanzados en R y Python.
8. ME-4002 Métodos Supervisados Avanzados en R y Python.
9. ME-9001 Artículo Final de Graduación.



6. Metodología

El máster se desarrollará en modalidad 100% online, con clases en vivo (presencial-remoto) a través de Zoom, las cuales se grabarán para que los estudiantes puedan acceder a ellas en cualquier momento. Los estudiantes contarán con:



Videoconferencias semanales:

Sesiones en tiempo real para interactuar con el tutor y compañeros.



Trabajos prácticos:

Ejercicios semanales aplicados a situaciones reales, enviados a través del campus virtual.



Foros de debate y consultas:

Espacio para resolver dudas y discutir temas del curso.



Campus virtual en Moodle:

Acceso 24/7 a materiales del curso, grabaciones de las clases, tareas y herramientas de comunicación.



7. Inversión

Duración:

10 meses

Precio por curso:

\$300 + 2% de IVA.

Precio Proyecto Final:

\$400 +2% de IVA.

Inversión total:

\$2800 + 2% de IVA

8. Información adicional:



Correo electrónico : info@promidat.com



Teléfono: +506 4030-1205



Web: www.promidat.com



WhatsApp: +506 8712-6978



Directo: +506 2271-0464



ME-1000

Programación Paralela en R y Python

Duración

Cuatro semanas

Descripción del Curso

La programación paralela es esencial en la ciencia de datos y el aprendizaje automático para el manejo eficiente de grandes volúmenes de datos y la ejecución de modelos computacionalmente intensivos. Este curso explora el uso de herramientas en R y Python para implementar programación paralela y optimizar el rendimiento de algoritmos. Se abordarán paquetes específicos como Snow en R y Joblib en Python, y se explorarán ejemplos prácticos de paralelización de tareas usando scikit-learn y Dask.

Objetivos

1. Introducir los conceptos y técnicas básicas de programación paralela en R y Python.
2. Capacitar a los estudiantes en el uso de herramientas como Snow en R y Joblib en Python para mejorar el rendimiento de tareas computacionales.
3. Explorar la paralelización en el entrenamiento y evaluación de modelos de machine learning utilizando scikit-learn.
4. Enseñar a utilizar documentación técnica para profundizar en el uso de paquetes de programación paralela.

Metodología

El curso combina teoría y práctica para asegurar una comprensión profunda de los conceptos y habilidades prácticas en programación paralela. Se emplearán las siguientes herramientas y recursos:

1. Videoconferencias semanales:

Se ofrecerán sesiones en vivo en Zoom que serán grabadas para que los estudiantes puedan revisarlas cuando deseen.

2. Trabajos prácticos semanales:

Ejercicios y casos de estudio aplicados para implementar programación paralela.

3. Foros de discusión:

Un espacio en Moodle donde los estudiantes podrán plantear dudas y discutir con el tutor y compañeros.

4. Aula virtual en Moodle:

Aula virtual en Moodle: Acceso a materiales, enlaces a documentación, ejemplos de código y otros recursos complementarios.

Luego de este curso el estudiante será capaz de:

5. Implementar programación paralela en R usando el paquete Snow y en Python usando Joblib.
6. Aprovechar la paralelización en modelos de aprendizaje automático en scikit-learn.
7. Comprender y navegar la documentación técnica para aplicar programación paralela en proyectos de ciencia de datos.
8. Optimizar procesos computacionales para reducir tiempos de ejecución en tareas de ciencia de datos.

Contenido

- **Semana 1:**
Fundamentos de Programación Paralela en Ciencia de Datos
 - a. Introducción a la programación paralela y su importancia en ciencia de datos.
 - b. Conceptos básicos de concurrencia y paralelización.
 - c. Herramientas y paquetes principales para programación paralela en R y Python.
- **Semana 2:**
Programación Paralela en R con el Paquete Snow
 - a. Instalación y configuración de Snow en R.
 - b. Creación de clústeres y ejecución de tareas en paralelo.

- c. Ejemplos prácticos de paralelización en análisis de datos y simulaciones.
- d. Documentación: Snow Package Documentation.
- **Semana 3:**
Programación Paralela en Python con Joblib
 - a. Introducción a Joblib en Python.
 - b. Uso de joblib.Parallel para la ejecución de funciones en paralelo.
 - c. Casos prácticos de paralelización de tareas computacionales.
 - d. Documentación: Joblib.Parallel.
- **Semana 4:**
Paralelización en Modelos de Machine Learning con scikit-learn y Dask
 - a. Implementación de programación paralela en scikit-learn usando Joblib.
 - b. Uso de Dask para la escalabilidad en machine learning.
 - c. Ejemplo práctico: Paralelización de la Validación Cruzada en R y Python.
 - d. Ejemplo práctico: Paralelización de calibración y selección de modelos en R y Python.
 - e. Documentación: Scikit-learn con Joblib en Dask.

Evaluación

La evaluación consistirá en 4 tareas prácticas con un valor igual. Cada tarea estará enfocada en implementar y optimizar programación paralela en diferentes contextos de análisis de datos y aprendizaje automático.

Bibliografía

Vahid, F. (2020). Parallel Programming in Science and Engineering. Springer. Libro.

Joblib Documentation. (2023). Joblib.Parallel. <https://joblib.readthedocs.io/en/latest/generated/joblib.Parallel.html>

Dask Developers. (2023). Using Joblib with Dask. <https://ml.dask.org/joblib.html>

Scikit-Learn Developers. (2023). Parallelism in Scikit-Learn 1.5. <https://scikit-learn.org/1.5/computing/parallelism.html>

Scikit-Learn Developers. (2023). Parallelism in Scikit-Learn 1.0. <https://scikit-learn.org/1.0/computing/parallelism.html>

Downey, A. (2016). Think Python: How to Think Like a Computer Scientist. Green Tea Press.

Eddelbuettel, D. (2013). Seamless R and C++ Integration with Rcpp. Springer.

Wilkinson, B. & Allen, M. (2005). Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers. Pearson.



ME-2000

Desarrollo Modelos Analíticos apoyados por herramientas de IA

Duración

Cuatro semanas

Descripción del Curso

Este curso avanzado explora el desarrollo e implementación de modelos supervisados y métodos no supervisados, apoyándose en herramientas de inteligencia artificial como ChatGPT para optimizar el rendimiento y la precisión de los modelos. Los estudiantes aprenderán a integrar estas herramientas de IA para mejorar la eficiencia en la construcción y evaluación de modelos, desde la optimización de hiperparámetros hasta la generación de insights para preprocesamiento de datos y selección de características. Con proyectos prácticos en R y Python, el curso abarca aplicaciones de clustering, reducción de dimensionalidad, y modelos de clasificación y regresión.

Objetivos

1. Comprender y aplicar modelos supervisados avanzados para tareas de clasificación y regresión.
2. Utilizar métodos no supervisados como clustering y reducción de dimensionalidad en problemas reales.
3. Integrar herramientas de inteligencia artificial (IA) como ChatGPT para apoyar la construcción y optimización de modelos.
4. Implementar modelos en R y Python, aprovechando el soporte de IA para mejorar la eficiencia y precisión.
5. Desarrollar habilidades para seleccionar y evaluar modelos de acuerdo con el contexto específico del problema.

Metodología

El curso combina teoría avanzada y aplicaciones prácticas, utilizando R y Python para implementar y evaluar modelos. Se incentiva a los estudiantes a utilizar herramientas de IA para tareas como la optimización de hiperparámetros, generación de código, y asesoría en la elección de algoritmos. Las herramientas y recursos incluyen:

- 1. Videoconferencias semanales:**
Clases en vivo mediante Zoom, grabadas para revisión posterior.
- 2. Trabajos prácticos semanales:**
Ejercicios en R y Python para aplicar los conceptos en proyectos reales.
- 3. Foros de discusión:**
Espacio en Moodle para plantear dudas y compartir ideas.
- 4. Aula virtual en Moodle:**
Acceso a materiales, enlaces a bibliografía, ejemplos de código y recursos adicionales.

Luego de este curso el estudiante será capaz de:

1. Implementar modelos avanzados de clasificación y regresión con el apoyo de herramientas de IA.
2. Aplicar métodos no supervisados para análisis de datos.
3. Utilizar IA como ChatGPT para asistir en la optimización de modelos y generación de insights.
4. Evaluar y ajustar modelos en R y Python, mejorando la precisión y el rendimiento.
5. Resolver problemas de ciencia de datos utilizando técnicas de modelado avanzadas y asistencia de IA.
6. Integrar Motores de IA como ChatGPT en r y en Python.

Contenido

- **Semana 1:**
Definición de los “Large Language Models” (LLMs) y su integración con el lenguaje R
 - a. ¿Qué son los LLMs?
 - b. Entrenamiento de los LLMs..
 - c. Técnicas avanzadas de “Prompting” con aplicación en ChatGPT.
 - d. Integración de R y ChatGPT con el paquete TheOpenAIR.
 - e. Integración directa de R y ChatGPT.
 - f. Ejemplos de casos reales de aplicación.

- **Semana 2:**
Modelos Supervisados Avanzados apoyados con IA
 - a. GitHub Copilot y CRISP-DM con ChatGPT.
 - b. Usando LLMs para proyectos de Ciencia de Datos basados en la metodología CRISP-DM.
 - c. Revisión de algoritmos avanzados: Árboles de decisión, Random Forest, y Gradient Boosting.
 - d. Validación cruzada y ajuste de modelos con soporte de herramientas IA.
 - e. Validación cruzada apoyados con IA y ChatGPT.
 - f. Ejemplo práctico en R: Optimización de hiperparámetros con traineR, Caret y ChatGPT.
- **Semana 3:**
Integración Directa de Python, ChatGPT y otros motores de IA
 - a. LLMs en Python.
 - b. Ejemplo práctico en Python: Optimización de modelos supervisados en Scikit-learn con ayuda de IA.
 - c. Uso de APIs para integrar Python y ChatGPT.
- **Semana 4:**
Automatización de Modelos Supervisados en Python con ChatGPT
 - a. Integración completa de modelos de clasificación y regresión en Python.
 - b. Uso de ChatGPT para generación de código y sugerencias en preprocesamiento de datos.
 - c. Creación de APPs con Python y ChatGPT.
 - d. Ejemplo de Análisis de Sentimientos en la WEB con Python y ChatGPT.

Evaluación

La evaluación consistirá en 4 tareas prácticas con igual peso, cada una enfocada en aplicar y optimizar métodos supervisados y no supervisados con el apoyo de herramientas de inteligencia artificial.

Bibliografía

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

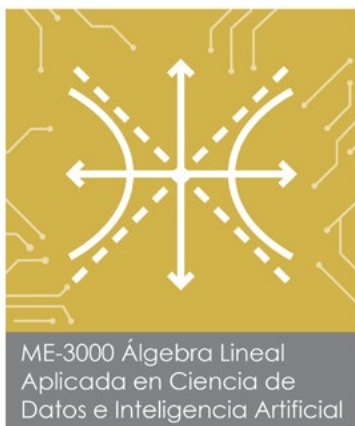
VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

Lantz, B. (2015). Machine Learning with R. Packt Publishing.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software.

Chollet, F. (2018). Deep Learning with Python. Manning Publications.

Wickham, H., & Grolemund, G. (2016). R for Data Science. O'Reilly Media.



ME-3000

Álgebra Lineal Aplicada en Ciencia de Datos e Inteligencia Artificial

Duración

Cuatro semanas

Descripción del Curso

El álgebra lineal es una herramienta fundamental en la ciencia de datos, la inteligencia artificial y el deep learning. Este curso introduce los conceptos esenciales de álgebra lineal necesarios para comprender y aplicar técnicas avanzadas de análisis de datos, reducción de dimensionalidad y modelos de aprendizaje automático. A través de ejercicios prácticos, los estudiantes aprenderán a utilizar vectores, matrices, y transformaciones lineales en el desarrollo de modelos predictivos y algoritmos de aprendizaje profundo. Los temas incluyen operaciones con vectores y matrices, descomposiciones en valores singulares, y sus aplicaciones en el Análisis en Componentes Principales, Máquinas de Soporte Vectorial y Redes Neuronales.

Objetivos

1. Familiarizar a los estudiantes con los conceptos básicos y avanzados de álgebra lineal aplicados a la ciencia de datos y el aprendizaje automático.
2. Dotar a los estudiantes de las habilidades para realizar operaciones con vectores y matrices, y aplicarlas en la construcción de modelos de aprendizaje supervisado y no supervisado.
3. Explicar la importancia de los valores y vectores propios, y su uso en técnicas de reducción de dimensionalidad y descomposición.
4. Comprender cómo el álgebra lineal sustenta el funcionamiento de algoritmos avanzados en deep learning, como redes neuronales convolucionales y redes de memoria a largo plazo (LSTM).
5. Desarrollar habilidades para aplicar técnicas de álgebra lineal en el análisis de datos reales y en el diseño de modelos de inteligencia artificial.

Metodología

La metodología del curso combina teoría y práctica, asegurando que los estudiantes puedan aplicar directamente los conceptos en problemas reales. Las herramientas y recursos incluyen:

1. Videoconferencias semanales:

Una sesión en Zoom que quedará grabada para que los estudiantes puedan revisarla en cualquier momento.

2. Trabajos prácticos semanales:

Ejercicios y casos de estudio enfocados en la implementación de algoritmos en Python.

3. Foros de discusión:

Espacio en Moodle donde los estudiantes pueden plantear preguntas, discutir temas del curso y colaborar con sus compañeros.

4. Aula virtual en Moodle:

Acceso a materiales de estudio, presentaciones, ejercicios, y recursos adicionales.

Luego de este curso el estudiante será capaz de:

1. Realizar operaciones complejas con vectores y matrices, incluyendo multiplicación, transposición, y cálculo de determinantes e inversas.
2. Aplicar valores y vectores propios en descomposiciones de matrices y reducción de dimensionalidad, como el Análisis en Componentes Principales (PCA).
3. Modelar rectas y planos en el espacio tridimensional y utilizar estas representaciones en visualización de datos y clustering.
4. Desarrollar e implementar algoritmos de Máquinas de Soporte Vectorial (SVM) y técnicas no supervisadas utilizando conceptos de álgebra lineal.
5. Comprender y construir redes neuronales simples aplicando operaciones matriciales, y entender el funcionamiento de capas convolucionales y de retropropagación en redes profundas.

Contenido

• **Semana 1:**

Fundamentos de Álgebra Lineal Aplicada a Ciencia de Datos

- a. Introducción a vectores y matrices en ciencia de datos.
- b. Operaciones con vectores (suma, producto escalar, producto cruz) y matrices (suma, multiplicación, determinantes, inversas).
- c. Aplicaciones en inteligencia artificial: álgebra lineal en transformaciones de datos y normalización.
- d. Caso práctico: Preprocesamiento de datos utilizando álgebra lineal.

- **Semana 2:**
Determinantes y sus aplicaciones en Análisis de Datos
 - a. Definición e implementación en R y en Python.
 - b. Aplicaciones en: Análisis de colinealidad en regresión, cálculo de verosimilitud en modelos multivariados, detección de anomalías, cálculo de β , independencia en activaciones neuronales, reducción de dimensionalidad, optimización con restricciones, problemas de optimización convexa o bayesiana
 - c. Caso práctico: Implementación de la Regresión Clásica.
- **Semana 3:**
Valores y Vectores Propios y sus aplicaciones en Análisis de Datos
 - a. Introducción a valores y vectores propios y su importancia en ciencia de datos.
 - b. Descomposición en valores propios (Eigen-decomposition) y aplicaciones en reducción de dimensionalidad.
 - c. Introducción a descomposiciones de matrices (e.g., SVD) y su uso en deep learning.
 - d. Caso práctico: Implementación del Análisis en Componentes Principales (PCA) para reducción de dimensionalidad en un conjunto de datos.
- **Semana 4:**
Implementación del Análisis en Componentes Principales
 - a. Modelado de rectas y planos en el espacio.
 - b. Interpretación geométrica de datos en espacios multidimensionales y su aplicación en clustering.
 - c. Algoritmo e implementación del ACP en R y en Python.
 - d. Aplicación práctica en visualización y agrupación de datos.
 - e. Caso práctico: Programación del Algoritmo del Análisis en Componentes Principales.

Evaluación

La evaluación consistirá en 4 tareas con un valor igual. Cada tarea estará enfocada en aplicar los conceptos de álgebra lineal a problemas reales de ciencia de datos e inteligencia artificial.

Bibliografía

Strang, G. (2016). Introduction to Linear Algebra. Wellesley-Cambridge Press.

Shlens, J. (2014). A Tutorial on Principal Component Analysis. arXiv.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433-459.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Nielsen, M. (2015). Neural Networks and Deep Learning. Online book.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.



ME-3002

Cálculo y Optimización Aplicado en Ciencia de Datos e Inteligencia Artificial

Duración

Cuatro semanas

Descripción del Curso

Este curso explora los conceptos de cálculo y optimización aplicados en ciencia de datos e inteligencia artificial. Los temas abordan desde los fundamentos del cálculo diferencial hasta el uso de métodos de optimización como el gradiente para mejorar el rendimiento de modelos y algoritmos de Machine Learning. A lo largo del curso, los estudiantes aprenderán a calcular derivadas, aplicar la regla de la cadena, usar el vector gradiente, y encontrar máximos y mínimos en funciones, todo enfocado en aplicaciones prácticas en ciencia de datos.

Objetivos

1. Comprender los fundamentos del cálculo diferencial y su aplicación en la optimización de modelos de machine learning.
2. Aprender a calcular derivadas y aplicar la regla de la cadena en el contexto de funciones utilizadas en inteligencia artificial.
3. Utilizar el vector gradiente para identificar direcciones de crecimiento o decrecimiento de funciones.
4. Aplicar técnicas de optimización, como el método del gradiente, para encontrar mínimos en funciones de una o varias variables.
5. Desarrollar habilidades para optimizar modelos predictivos y funciones de costo en ciencia de datos.

Metodología

Este curso combina teoría y práctica en cálculo y optimización. Los estudiantes trabajarán con Python y R para aplicar los conceptos en problemas de ciencia de datos e inteligencia artificial. Las herramientas y recursos incluyen:

- 1. Videoconferencias semanales:**
Sesiones en Zoom que serán grabadas para su revisión posterior.
- 2. Trabajos prácticos semanales:**
Ejercicios aplicados en R y Python para reforzar los conceptos de cálculo y optimización.
- 3. Foros de discusión:**
Espacio en Moodle para discutir dudas y compartir conocimientos con el tutor y compañeros.
- 4. Aula virtual en Moodle:** Acceso a materiales, ejemplos de código y otros recursos.

Luego de este curso el estudiante será capaz de:

1. Calcular derivadas de funciones y aplicar la regla de la cadena en el contexto de Machine Learning.
2. Utilizar el vector gradiente para analizar el comportamiento de funciones en múltiples variables.
3. Encontrar máximos y mínimos de funciones de una o varias variables.
4. Implementar el método del gradiente en problemas de optimización para modelos de ciencia de datos.
5. Aplicar técnicas de optimización para mejorar el rendimiento de algoritmos de inteligencia artificial.

Contenido

- **Semana 1:**
Fundamentos del Cálculo Diferencial en Ciencia de Datos e Inteligencia Artificial
 - a. Introducción al cálculo de derivadas y su importancia en ciencia de datos.
 - b. Cálculo de derivadas de funciones básicas.
 - c. Aplicación de la regla de la cadena en funciones compuestas.
 - d. Ejemplo práctico en Python y R: Derivadas en funciones de costo de modelos de Machine Learning.

- **Semana 2:**
Vector Gradiente y sus Aplicaciones
 - a. Introducción al vector gradiente y su interpretación en funciones de varias variables.
 - b. Cálculo del gradiente en funciones multivariables.
 - c. El fenómeno de la desaparición de gradiente (vanishing gradient problem)
 - d. Ejemplos de aplicación del gradiente en la optimización de funciones de costo.
 - e. Ejemplo práctico en Python: Cálculo del gradiente en redes neuronales.
- **Semana 3:**
Máximos y Mínimos en Funciones de una y varias Variables
 - a. Identificación de máximos y mínimos en funciones de una variable.
 - b. Extensión a funciones de varias variables y uso del gradiente para identificar extremos.
 - c. Ejemplo práctico en R y Python: Optimización de funciones de error en modelos predictivos.
- **Semana 4:**
Otros métodos de optimización aplicados en la analítica de datos
 - a. Métodos: Nesterov Accelerated Gradient (NAG), AdaGrad, RMSProp, Adam, AdaMax, Nadam, Método de Newton, Quasi-Newton (BFGS, L-BFGS), Levenberg-Marquardt, Optimización Bayesiana, Dropout (en redes neuronales), Batch Normalization (en redes neuronales), Cyclical Learning Rates (CLR).
 - b. Algoritmos Genéticos, Enjambre de Partículas (PSO).
 - c. Ejemplo práctico en Python y R: Implementación del gradiente descendente en regresión lineal.

Evaluación

La evaluación consistirá en 4 tareas prácticas con igual valor. Cada tarea estará enfocada en aplicar y optimizar conceptos de cálculo y optimización en proyectos de ciencia de datos e Inteligencia Artificial.

Bibliografía

Strang, G. (2016). Calculus. Wellesley-Cambridge Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Nielsen, M. (2015). Neural Networks and Deep Learning. Online book.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

Chollet, F. (2018). Deep Learning with Python. Manning Publications.

Luenberger, D. G., & Ye, Y. (2016). Linear and Nonlinear Programming. Springer.



ME-3004

Probabilidad y Estadística Aplicados a la Ciencia de Datos e Inteligencia Artificial

Duración

Cuatro semanas

Descripción del Curso

Este curso proporciona una introducción a los conceptos de probabilidad y estadística fundamentales en el análisis de datos y su aplicación en inteligencia artificial. Los temas cubren desde la definición de probabilidad, probabilidad condicional y el Teorema de Bayes hasta aplicaciones prácticas en árboles de decisión y métodos bayesianos. Al finalizar, los estudiantes habrán desarrollado una base sólida para aplicar conceptos estadísticos en el desarrollo y optimización de modelos de Machine Learning y análisis de datos.

Objetivos

1. Entender los conceptos básicos de probabilidad y estadística y su importancia en ciencia de datos.
2. Aplicar el Teorema de Bayes para inferencia en modelos de inteligencia artificial.
3. Utilizar probabilidades y estadísticas en la construcción y evaluación de árboles de decisión.
4. Introducir los fundamentos de los métodos bayesianos y su uso en machine learning.
5. Desarrollar habilidades para aplicar métodos probabilísticos en el análisis de datos y modelos predictivos.

Metodología

El curso combina teoría con aplicaciones prácticas en R y Python, asegurando que los estudiantes puedan implementar y analizar modelos probabilísticos. Las herramientas y recursos incluyen:

- 6. Videoconferencias semanales:**
Sesiones en Zoom que serán grabadas para su revisión posterior.
- 7. Trabajos prácticos semanales:**
Ejercicios aplicados en R y Python para reforzar los conceptos de probabilidad y estadística.
- 8. Foros de discusión:**
Espacio en Moodle para discutir temas del curso y resolver dudas.
- 9. Aula virtual en Moodle:**
Aula virtual en Moodle: Acceso a materiales, ejemplos de código y recursos adicionales.

Luego de este curso el estudiante será capaz de:

- 10.** Definir y calcular probabilidades en diferentes contextos.
- 11.** Aplicar el Teorema de Bayes para realizar inferencias y tomar decisiones en modelos de datos.
- 12.** Implementar árboles de decisión y analizar su rendimiento en problemas de clasificación.
- 13.** Comprender y aplicar métodos bayesianos en el desarrollo de modelos predictivos.
- 14.** Evaluar modelos de machine learning usando conceptos estadísticos fundamentales.

Contenido

- **Semana 1:**
Análisis Combinatorio (aprendiendo a contar)
 - a. Técnicas de Conteo.
 - b. Permutaciones con y sin repetición.
 - c. Combinaciones con y sin repetición.
 - d. Regla de la multiplicación.
 - e. Implementación en R y Python.
- **Semana 2:**
Introducción a la Probabilidad y Definición de Conceptos Básicos
 - a. Introducción a la probabilidad y su importancia en ciencia de datos.
 - b. Definición de probabilidad, eventos y espacio muestral.
 - c. Conceptos básicos de probabilidad condicionada.

- d. Teorema de Bayes.
- e. Ejemplo práctico en Python y R: Aplicación del Teorema de Bayes a la Analítica Predictiva.
- **Semana 3:**
Distribuciones de Probabilidad
 - a. Variables Aleatorias.
 - b. Distribución Normal.
 - c. Distribución Uniforme.
 - d. Otras distribuciones de probabilidad.
 - e. Esperanza matemática y desviación estándar.
 - f. Ejemplo práctico en R y Python: Implementación las distribuciones de probabilidad.
- **Semana 4:**
Distribuciones de Probabilidad discretas y continuas
 - a. Distribución Binomial y su aproximación por la normal.
 - b. Distribución Poisson y su aproximación por la normal.
 - c. Distribución hipergeométrica y su aproximación por la normal.
 - d. Otras distribuciones de probabilidad.
 - e. Esperanza matemática y desviación estándar.
 - f. Ejemplo práctico en R y Python: Implementación las distribuciones de probabilidad discretas y continuas.
 - g. Ejemplo práctico en R y Python: Aplicaciones en Árboles de Decisión y otros métodos basados en árboles, como Bosques Aleatorios.

Evaluación

La evaluación consistirá en 4 tareas prácticas con igual valor, enfocadas en aplicar conceptos de probabilidad y estadística en proyectos de ciencia de datos.

Bibliografía

Wasserman, L. (2004). All of Statistics: A Concise Course in Statistical Inference. Springer.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). Bayesian Data Analysis. CRC Press.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

Silver, N. (2012). The Signal and the Noise: Why So Many Predictions Fail—but Some Don't. Penguin Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Feller, W. (1968). An Introduction to Probability Theory and Its Applications. Wiley.

McGrayne, S. B. (2011). The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy. Yale University Press.

McGrayne, S. B. (2011). The Theory That Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, and Emerged Triumphant from Two Centuries of Controversy. Yale University Press.



ME-3008

Procesamiento de Lenguaje Natural y Agentes de Inteligencia Artificial

Duración

Cuatro semanas

Descripción del Curso

Este curso ofrece una formación integral y rigurosa en Procesamiento de Lenguaje Natural (NLP), abarcando desde los modelos vectoriales clásicos del lenguaje – como TF-IDF y análisis semántico – hasta las técnicas modernas de aprendizaje profundo basadas en redes neuronales, embeddings, CNNs, RNNs, LSTMs y Transformers. A lo largo del curso, el estudiantado comprenderá cómo las máquinas representan, procesan y generan lenguaje humano, y cómo estas representaciones se utilizan en aplicaciones reales como modelos de lenguaje de gran escala, extracción de información, grafos de conocimiento y motores de diálogo. El curso culmina con un enfoque práctico y aplicado, donde se diseña e implementa un chatbot inteligente utilizando TensorFlow y arquitecturas Transformer, incluyendo el preprocesamiento de datos, tokenización, mecanismos de atención y atención multi-cabeza, integrando fundamentos matemáticos, estadísticos y algorítmicos con soluciones reales de la industria.

Objetivos

1. **Comprender los fundamentos teóricos del Procesamiento de Lenguaje Natural**, incluyendo modelos vectoriales del lenguaje, representaciones basadas en conteos y análisis semántico, como base para el tratamiento computacional del lenguaje natural.
2. **Analizar y aplicar modelos de aprendizaje profundo para texto**, tales como embeddings, redes neuronales, CNNs, RNNs, LSTMs y arquitecturas Transformer, entendiendo su funcionamiento matemático y algorítmico.
3. **Diseñar e implementar soluciones prácticas de NLP**, integrando técnicas modernas para el desarrollo de aplicaciones reales como modelos de lenguaje, extracción de información, grafos de conocimiento y sistemas conversacionales.

4. **Construir e implementar un chatbot inteligente con TensorFlow**, dominando el flujo completo del proyecto: preparación de datos, tokenización, mecanismos de atención, atención multi-cabeza y despliegue del sistema.

Metodología

El curso combinará teoría y práctica aplicada en Python, permitiendo a los estudiantes crear agentes de Inteligencia Artificial. Las herramientas y recursos incluyen:

1. **Videoconferencias semanales:**
Sesiones en Zoom que serán grabadas para su revisión posterior.
2. **Trabajos prácticos:**
Ejercicios aplicados en Python para aplicar el Procesamiento de Lenguaje Natural en problemas de la industria.
3. **Foros de discusión:**
Espacio en Moodle para la resolución de dudas y discusión de temas del curso.
4. **Aula virtual en Moodle:**
Acceso a materiales, ejemplos de código y recursos adicionales.

Luego de este curso el estudiante será capaz de:

1. Representar texto mediante modelos vectoriales del lenguaje, construir matrices basadas en conteos (TF, TF-IDF) y aplicar técnicas de análisis semántico para el procesamiento computacional de datos textuales reales.
2. Entrenar, evaluar y comparar modelos de aprendizaje profundo para texto, utilizando embeddings, redes neuronales, CNNs, RNNs, LSTMs y arquitecturas Transformer, interpretando su comportamiento desde un punto de vista matemático y algorítmico.
3. Diseñar e implementar aplicaciones completas de Procesamiento de Lenguaje Natural, tales como modelos de lenguaje, sistemas de extracción de información, grafos de conocimiento y motores de diálogo, integrando técnicas modernas de PLN en contextos reales.
4. Construir, entrenar y desplegar un chatbot inteligente con TensorFlow, realizando el preprocesamiento de datos, la tokenización, la implementación de mecanismos de atención y atención multi-cabeza, y la integración final del sistema.

Contenido

- **Semana 1:**
Máquinas de palabras: Modelos vectoriales del lenguaje natural
 - a. Máquinas que leen y escriben: una visión general del procesamiento del lenguaje natural (Natural Language Processing – NLP).

- b. Tokens del pensamiento: palabras del lenguaje natural.
 - c. Matemáticas con palabras: vectores de frecuencia de término–frecuencia inversa de documento (TF-IDF).
 - d. Encontrar significado en los conteos de palabras: análisis semántico.
 - e. Programación en Python de los temas anteriores.
- **Semana 2:**
Aprendizaje profundo (Deep Learning): Redes neuronales
 - a. Cerebro de palabras: redes neuronales.
 - b. Razonamiento con incrustaciones (embeddings) de palabras.
 - c. Encontrar núcleos de conocimiento en texto con CNNs.
 - d. Reducir, reutilizar y reciclar tus palabras: RNNs y LSTMs.
 - e. Implementación en Python.
- **Semana 3:**
Pasarse a lo real: Aplicaciones de procesamiento del lenguaje natural (Natural Language Processing – NLP) en el mundo real
 - a. Aprendizaje profundo apilable: Transformers.
 - b. Modelos de lenguaje grandes en el mundo real.
 - c. Extracción de información y grafos de conocimiento.
 - d. Conversar sin parar: motores de diálogo.
- **Semana 4:**
Construcción de un chatbot con TensorFlow y tecnología Transformer
 - a. Chatbot de IA en pocas palabras
 - ¿Qué es un chatbot?
 - ¿Qué es una palabra de activación (wake word) en un chatbot?
 - Componentes de NLP en un chatbot
 - b. Construcción de un chatbot de películas usando TensorFlow y tecnología Transformer
 - El conjunto de datos del chatbot
 - Preprocesamiento de diálogos de películas
 - Tokenización de conversaciones de películas
 - Proceso de filtrado y padding

- Creación del objeto del conjunto de datos de películas en TensorFlow (mDS)
 - Cálculo de pesos de aprendizaje de atención
 - Atención multi-cabeza (Multi-Head Attention, MHAttention)
- c. Implementación del sistema en Python.

Evaluación

La evaluación consistirá en cuatro entregas prácticas.

Bibliografía

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Prentice Hall.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Tunstall, L., von Werra, L., & Wolf, T. (2022). *Natural Language Processing with Transformers*. O'Reilly Media.

Rothman, D. (2021). *Transformers for Natural Language Processing* (2nd ed.). Packt Publishing.

Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (3rd ed.). O'Reilly Media.

Bengfort, B., Bilbro, R., & Ojeda, T. (2018). *Applied Text Analysis with Python*. O'Reilly Media.

Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer.



ME-4000 Métodos NO Supervisados Avanzados en R y Python

Duración

Cuatro semanas

Descripción del Curso

Este curso avanzado cubre técnicas de métodos no supervisados en el análisis de datos, aplicadas a través de R y Python. Los estudiantes explorarán temas como clustering avanzado, reducción de dimensionalidad y análisis factorial de correspondencias, además de técnicas especializadas como el Análisis Factorial de Correspondencias Simples y Múltiples y el Método STATIS para tablas múltiples. Este conocimiento permitirá a los estudiantes aplicar métodos no supervisados en proyectos complejos de ciencia de datos y resolver problemas de agrupamiento y segmentación.

Objetivos

1. Profundizar en técnicas avanzadas de métodos no supervisados en análisis de datos.
2. Aprender a implementar y evaluar técnicas de clustering avanzado y reducción de dimensionalidad.
3. Aplicar el Análisis Factorial de Correspondencias Simples y Múltiples en R y Python.
4. Explorar y aplicar el Método STATIS para el análisis de tablas múltiples.
5. Desarrollar habilidades para aplicar estos métodos en ciencia de datos, análisis de mercado y estudios multidimensionales.

Metodología

El curso combinará teoría y práctica aplicada en R y Python, permitiendo a los estudiantes implementar y analizar métodos no supervisados en problemas reales de ciencia de datos. Las herramientas y recursos incluyen:

- 1. Videoconferencias semanales:**
Sesiones en Zoom que serán grabadas para su revisión posterior.
- 2. Trabajos prácticos semanales:**
Ejercicios aplicados en R y Python para fortalecer la implementación de métodos no supervisados avanzados.
- 3. Foros de discusión:**
Espacio en Moodle para la resolución de dudas y discusión de temas del curso.
- 4. Aula virtual en Moodle:**
Acceso a materiales, ejemplos de código y recursos adicionales.

Luego de este curso el estudiante será capaz de:

1. Aplicar técnicas avanzadas de clustering y reducción de dimensionalidad en R y Python.
2. Realizar Análisis Factorial de Correspondencias Simples y Múltiples.
3. Implementar el Método STATIS para el análisis de tablas múltiples.
4. Utilizar métodos no supervisados avanzados para resolver problemas de segmentación, agrupación y análisis exploratorio de datos.
5. Evaluar y comparar métodos no supervisados en el contexto de aplicaciones en ciencia de datos.

Contenido

- **Semana 1:**
Análisis Factorial de Correspondencias Simples y Múltiples
 - a. Introducción al Análisis Factorial de Correspondencias (AFC) para tablas simples.
 - b. Extensión al Análisis Factorial de Correspondencias Múltiples (AFCM) para tablas múltiples.
 - c. Interpretación de los resultados del AFC y AFCM en contextos de ciencia de datos y segmentación de mercado.
 - d. Ejemplo práctico en Python: Implementación de AFC y AFCM.
 - e. Ejemplo práctico en R: Análisis factorial de correspondencias en datos categóricos.

- **Semana 2:**
Método STATIS y STATIS Dual para el Análisis de Tablas Múltiples
 - a. Introducción al Método STATIS y su aplicación en ciencia de datos.
 - b. Análisis de estructuras comunes entre tablas múltiples y su interpretación.
 - c. Aplicaciones de STATIS en estudios multidimensionales y análisis de tendencias.
 - d. Ejemplo práctico en Python y R: Implementación del Método STATIS para tablas múltiples.
- **Semana 3:**
Técnicas Avanzadas de Reducción de Dimensionalidad y Visualización
 - a. Métodos t-SNE, y UMAP.
 - b. Selección de técnicas de reducción de dimensionalidad para visualización y agrupación.
 - c. Ejemplo práctico en Python: Implementación de t-SNE y UMAP para visualización de datos.
 - d. Ejemplo práctico en R: PCA aplicado a datos multidimensionales.
- **Semana 4:**
Modelos no euclidianos de reducción de la dimensión y técnicas de agrupación basadas en densidades
 - a. Análisis de Datos Topológicos.
 - b. Estadística Riemanniana.
 - c. Introducción a métodos de clustering avanzados: DBSCAN, clustering jerárquico, y clustering basado en densidad.
 - d. Evaluación de modelos de clustering y métricas de validación.
 - e. Ejemplo práctico en Python: Implementación de la Estadística Riemanniana.
 - f. Ejemplo práctico en R: Agrupación de datos utilizando clustering basado en densidad.

Evaluación

La evaluación consistirá en 4 tareas prácticas con igual valor, enfocadas en aplicar métodos no supervisados avanzados en análisis de datos.

Bibliografía

Murtagh, F., & Legendre, P. (2014). Ward's Hierarchical Clustering Method: Clustering Criterion and Agglomerative Algorithm. *Journal of Classification*.

Greenacre, M. (2007). *Correspondence Analysis in Practice*. CRC Press.

Abdi, H., & Williams, L. J. (2010). *Principal Component Analysis*. Wiley Interdisciplinary Reviews: Computational Statistics.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.

Lebart, L., Morineau, A., & Warwick, K. M. (1984). *Multivariate Descriptive Statistical Analysis*. Wiley.

Pagès, J. (2015). *Multiple Factor Analysis by Example Using R*. CRC Press.

Husson, F., Josse, J., & Pagès, J. (2010). Principal Component Methods - Hierarchical Clustering - Partitional Clustering: Why Would We Need to Choose for Visualizing Data?. *Journal of Statistical Software*.

Gower, J. C., & Dijksterhuis, G. B. (2004). *Procrustes Problems*. Oxford University Press.

Lê, S., Josse, J., & Husson, F. (2008). FactoMineR: An R Package for Multivariate Analysis. *Journal of Statistical Software*.

VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.



ME-4002

Métodos Supervisados Avanzados en R y Python

Duración

Cuatro semanas

Descripción del Curso

Este curso avanzado se centra en el estudio y la implementación de métodos supervisados avanzados en el análisis de datos, utilizando R y Python. Los estudiantes explorarán algoritmos de clasificación y regresión de última generación, optimización de hiperparámetros, técnicas de ensamblado y métodos de validación para mejorar la precisión y eficiencia de los modelos. Al final del curso, los estudiantes serán capaces de implementar y ajustar modelos supervisados avanzados para resolver problemas complejos en ciencia de datos y machine learning.

Objetivos

1. Comprender y aplicar la técnicas de hiperparametrización en algoritmos avanzados de clasificación y regresión.
2. Aprender técnicas de hiperparametrización en ensamblado, como bagging, boosting y stacking, para mejorar la precisión de los modelos.
3. Optimizar los hiperparámetros de modelos supervisados en R y Python.
4. Aplicar hiperparametrización en métodos de validación cruzada y evaluación de rendimiento en modelos de machine learning.
5. Desarrollar habilidades para seleccionar y ajustar modelos supervisados en función de diferentes aplicaciones en ciencia de datos.

Metodología

El curso combina teoría y práctica aplicada en R y Python, permitiendo a los estudiantes implementar y evaluar modelos supervisados en problemas reales de ciencia de datos. Las herramientas y recursos incluyen:

- 1. Videoconferencias semanales:**
Sesiones en Zoom que serán grabadas para su revisión posterior.
- 2. Trabajos prácticos semanales:**
Ejercicios aplicados en R y Python para fortalecer la implementación de métodos supervisados avanzados.
- 3. Foros de discusión:**
Espacio en Moodle para resolver dudas y discutir temas del curso.
- 4. Aula virtual en Moodle:**
Acceso a materiales, ejemplos de código y otros recursos.

Luego de este curso el estudiante será capaz de:

1. Implementar y evaluar algoritmos avanzados de clasificación y regresión en R y Python.
2. Optimizar modelos supervisados mediante técnicas de ajuste de hiperparámetros.
3. Aplicar técnicas de ensamblado, como bagging y boosting, para mejorar el rendimiento de los modelos.
4. Utilizar métodos de validación y métricas de evaluación para medir la eficacia de los modelos.
5. Seleccionar e implementar el modelo supervisado más adecuado según el contexto y los datos disponibles.

Contenido

- **Semana 1:**
Hiperparametrización de Modelos Analíticos en R
 - a. HPT y Optimización de Hiperparámetros (HPO).
 - b. Medidas de desempeño o calidad.
 - c. Grid Search vs. Random Search.
 - d. Optimización basada en modelos sustitutos (SMBO).
 - e. Ejemplo práctico en R: Implementación de Métodos de hiperpametrización en R.

- **Semana 2:**
Hiperparametrización en R usando Validación Cruzada Modelos Supervisados para medir la calidad
 - a. Hiperparametrización en técnicas de validación cruzada y métodos de evaluación.
 - b. Ejemplo práctico en R: Validación cruzada y evaluación de un modelo de clasificación.
 - c. Ejemplo práctico en R: Hiperparametrización y evaluación de rendimiento en modelos de regresión.
- **Semana 3:**
Hiperparametrización mediante Validación cruzada estratificada en Modelos Supervisados en Python
 - a. Métodos de optimización de hiperparámetros: Grid Search, Random Search, optimización bayesiana, entre otros.
 - b. Técnicas de regularización para mejorar la generalización de los modelos.
 - c. Ejemplo práctico en Python: Optimización de hiperparámetros a pie y usando Scikit-learn.
- **Semana 4:**
Técnicas avanzadas de hiperparametrización en Python.
 - a. Hiperparametrización mediante búsqueda gruesa a fina (Coarse-to-Fine Search, CFS).
 - b. Hiperparametrización mediante búsqueda por reducción sucesiva (Successive Halving, SH).
 - c. Hiperparametrización mediante Optimización Bayesiana (BO).
 - d. Relación entre la Optimización Bayesiana y el Teorema de Bayes.
 - e. Ejemplo práctico en Python: Implementación en Python de todos los métodos anteriores.

Evaluación

La evaluación consistirá en 4 tareas prácticas con igual valor, enfocadas en aplicar y optimizar métodos supervisados avanzados en proyectos de ciencia de datos.

Bibliografía

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). Additive Logistic Regression: A Statistical View of Boosting. The Annals of Statistics.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. arXiv preprint arXiv:1603.02754.

Chollet, F. (2018). Deep Learning with Python. Manning Publications.

Brownlee, J. (2016). Master Machine Learning Algorithms. Machine Learning Mastery.



ME-9001

Artículo Final de Graduación

Duración

Ocho semanas

Descripción del Curso

En este curso Artículo Final de Graduación, el estudiante aplicará todos los conocimientos adquiridos en los cursos previos de métodos avanzados en ciencia de datos, deep learning y optimización. El curso está diseñado para guiar al estudiante en el desarrollo de una investigación original y su aplicación en datos reales, desde la revisión de técnicas de investigación hasta la formulación de conclusiones y propuestas de trabajo futuro. Las primeras dos semanas estarán dedicadas a revisar técnicas de investigación y los elementos clave de un artículo (paper) científico. Las siguientes seis semanas se dedicarán a sesiones individualizadas de asesoría y trabajo para orientar al estudiante en el desarrollo de su artículo.

Objetivos

1. Capacitar al estudiante en técnicas de investigación y en la estructura de un artículo en Ciencia de Datos e Inteligencia Artificial.
2. Guiar al estudiante en la construcción de un marco teórico, revisión del estado del arte, y desarrollo de una propuesta de investigación.
3. Orientar el análisis comparativo entre datos reales y simulados y su presentación en tablas y gráficos.
4. Facilitar la redacción de conclusiones y la identificación de áreas de trabajo futuro.
5. Preparar al estudiante para la defensa y presentación final de su artículo.

Metodología

El curso combina sesiones teóricas iniciales, trabajo individual, y asesorías personalizadas. En las primeras dos semanas, se cubrirán temas fundamentales de técnicas de investigación y los elementos de un artículo científico. Luego, el estudiante trabajará de forma independiente en su proyecto, recibiendo orientación en sesiones de asesoría individualizadas cada semana para asegurar el progreso adecuado del artículo.

Las herramientas y recursos incluyen:

- 1. Sesiones teóricas:**
Durante las dos primeras semanas, se presentarán técnicas de investigación y los elementos estructurales de un artículo científico.
- 2. Asesoría individual:**
Se realizarán sesiones semanales de asesoría personalizada para apoyar el desarrollo de un artículo.
- 3. Foro de consulta:**
Espacio en Moodle para plantear dudas y recibir retroalimentación continua.
- 4. Aula virtual en Moodle:**
Acceso a materiales de apoyo, ejemplos de estructura un artículo científico y bibliografía complementaria.

Luego de este curso el estudiante será capaz de:

1. Aplicar técnicas de investigación y los elementos clave de un artículo en su trabajo final de graduación.
2. Realizar una revisión exhaustiva del estado del arte y desarrollar un marco teórico sólido.
3. Comparar y analizar datos reales y simulados, y presentar los resultados en tablas y gráficos.
4. Redactar conclusiones claras y proponer áreas de trabajo futuro en su investigación.
5. Defender y presentar de manera efectiva los resultados de su proyecto de graduación.

Contenido

- **Semana 1:**
Técnicas de Investigación y Metodología
 - a. Revisión de metodologías de investigación en Ciencia de Datos e Inteligencia Artificial.
 - b. Estrategias para la búsqueda y revisión de literatura científica.
 - c. Definición y formulación de preguntas de investigación.
 - d. Ejercicio práctico: Formulación de preguntas de investigación y objetivos para un artículo científico.

- **Semana 2:**
Elementos de un artículo
 - a. Estructura de un artículo: marco teórico, estado del arte, desarrollo de la propuesta, análisis de resultados, conclusiones y trabajo futuro.
 - b. Construcción del marco teórico y revisión del estado del arte.
 - c. Comparación en tablas de datos reales y simulados.
 - d. Redacción preliminar de conclusiones y propuesta de trabajo futuro.
 - e. Ejercicio práctico: Desarrollo del esquema de un artículo científico y redacción preliminar de las secciones.
 - f. Edición profesional de textos científicos en L^AT_EX y uso de Overleaf.
- **Semanas 3 a 8:**
Asesoría Individualizada y Trabajo de Desarrollo

Semana 3: Desarrollo de la propuesta de investigación y asesoría en la estructura de la tesis.

Semana 4: Revisión del marco teórico y orientación en la metodología y análisis de datos.

Semana 5: Implementación del análisis comparativo entre datos reales y simulados.

Semana 6: Presentación preliminar de resultados en tablas y gráficos; retroalimentación y ajustes.

Semana 7: Redacción de conclusiones y discusión del trabajo futuro.

Semana 8: Revisión final, preparación para la defensa y presentación del proyecto del artículo.

Evaluación

La evaluación consistirá en la entrega y defensa del artículo final, que incluirá:

1. **Propuesta de Investigación:** Planteamiento claro de la problemática, objetivos y justificación del proyecto.
2. **Marco Teórico y Estado del Arte:** Revisión exhaustiva de la literatura y contextualización del problema de investigación.
3. **Análisis de Datos:** Implementación de métodos avanzados y comparación de resultados entre datos reales y simulados.
4. **Conclusiones y Trabajo Futuro:** Redacción de conclusiones fundamentadas y propuestas de continuidad en la investigación.
5. **Defensa y Presentación:** Defensa oral por Zoom del proyecto ante el comité evaluador.

Bibliografía

Creswell, J. W., & Creswell, J. D. (2018). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. SAGE Publications.

Booth, W. C., Colomb, G. G., & Williams, J. M. (2008). The Craft of Research. University of Chicago Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning. Springer.

Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Murphy, K. P. (2012). Machine Learning: A Probabilistic Perspective. MIT Press.

Lencioni, P. (2002). The Five Dysfunctions of a Team: A Leadership Fable. Jossey-Bass.

Silver, N. (2012). The Signal and the Noise: Why So Many Predictions Fail—but Some Don't. Penguin Press.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv.

Material adicional proporcionado por el asesor, específico al proyecto del estudiante.



Correo electrónico : info@promidat.com



Web: www.promidat.com



Teléfono: +506 4030-1205



WhatsApp: +506 8712-6978



Directo: +506 2271-0464



Programa Iberoamericano de
Formación en Minería de Datos