

Machine Learning

DIPLOMADO | REMOTO

INTELIGENCIA ARTIFICIAL APLICADA: DEL MACHINE LEARNING A LOS MODELOS GRANDES DE LENGUAJE

CONTENIDO TEMÁTICO

Módulo 1.

Fundamentos de Python para Ciencia de Datos

- Sintaxis de Python: variables, tipos de datos, operadores (con ejercicios de aplicación).
- Estructuras de control: condicionales y bucles (con resolución de problemas).
- Funciones: definición, parámetros, retorno de valores (desarrollo de funciones personalizadas).
- Manejo de estructuras de datos: listas, tuplas, diccionarios, conjuntos (aplicación en escenarios de datos).
- Aplicación de Librerías Esenciales:
 - NumPy: Creación y manipulación de arrays, operaciones vectorizadas (ejercicios de álgebra lineal y manejo de datos numéricos).
 - Pandas: Creación y manipulación de Series y DataFrames, carga, limpieza y transformación de datos (implementaciones con diversos datasets).
 - Matplotlib y Seaborn: Creación de diferentes tipos de visualizaciones para exploración de datos (ejercicios de generación de gráficos).
- Configuración y uso de Google Colaboratory para desarrollo y ejecución de notebooks de Python, incluyendo importación de librerías y gestión de archivos.

Módulo 2.

Introducción a la Inteligencia Artificial y Machine Learning

- Conceptos clave de la IA y ML: discusión de casos de uso y aplicaciones reales.
- Tipos de Aprendizaje Automático: identificación y clasificación de problemas según el tipo de aprendizaje.
- Estudio de Caso y Aplicación del Flujo de Trabajo en ML: Desarrollo de un caso desde la definición del problema hasta la evaluación.
- Aplicación de Scikit-Learn: Introducción a la API de Scikit-Learn y preprocesamiento básico de datos (escalado, codificación).
- Regresión Lineal Simple y Múltiple.
- Fundamentos teóricos y supuestos.
 - Implementación: Desarrollo de modelos de regresión lineal, ajuste e interpretación de coeficientes con Scikit-Learn.
 - Ejercicios de predicción y evaluación.

Módulo 3.

Aprendizaje Supervisado: Modelado, Evaluación y Selección

- Aplicación de Gradiente Descendiente: Implementación conceptual y visualización del algoritmo para optimizar funciones simples. Discusión de variantes.
- Regresión Logística:
 - Fundamentos para clasificación, función sigmoide, función de coste.
 - Implementación Guiada: Desarrollo de modelos de regresión logística para clasificación binaria y multiclase con Scikit-Learn.
- Árboles de Decisión:
 - Construcción, criterios de división, poda (conceptual).
 - Implementación Guiada: Creación y visualización de árboles de decisión. Análisis de interpretabilidad.
 - Random Forests (Bagging):
 - Concepto de ensamblaje y bagging.
 - Implementación: de Random Forests y comparación con árboles individuales.
 - Análisis del Compromiso Sesgo-Varianza: (diagnóstico de subajuste y sobreajuste).
- Aplicación de Métricas de Evaluación de Modelos.
- Aplicación para Regresión (MAE, MSE, RMSE, R²).
- Aplicación para Clasificación (Matriz de confusión, Accuracy, Precision, Recall, F1-Score, Curva ROC y AUC).
- Implementación de Validación y Selección de Modelos:
- Implementación de Validación Cruzada.
- Ajuste de Hiperparámetros: Grid Search y Random Search con Scikit-Learn.
- Introducción a Otros Métodos de Ensamblaje: Conceptos y ejemplos de Boosting.

Módulo 4.

Redes Neuronales y Fundamentos de Deep Learning

- Exploración interactiva de las unidades básicas: Perceptrón y Neurona Logística.
- Introducción y Aplicación de TensorFlow y/o PyTorch: Manipulación de tensores, construcción de modelos básicos y uso de APIs (Keras, torch.nn).
- Redes Neuronales Feedforward (Multilayer Perceptron - MLP):
- Configuración de arquitecturas MLP en TensorFlow/Keras y PyTorch.
- Aplicación de Funciones de Activación: Experimentación con Sigmoid, Tanh, ReLU, Softmax, observando su impacto.
- Entrenamiento de Redes Neuronales:
 - Ejercicio de Comprensión y Aplicación del Algoritmo de Backpropagation: Seguimiento del flujo de error y actualización de pesos.
 - Aplicación de Optimizadores: Implementación y comparación de SGD, Adam, RMSprop, analizando curvas de aprendizaje.
 - Experimentación Dirigida: Pruebas con diferentes estrategias de Inicialización

de Pesos.

- Aplicación de Regularización: Implementación de técnicas (L1, L2, Dropout) para combatir el sobreajuste.
- Discusión con ejemplos de las diferencias entre ML clásico y Deep Learning.
- Construcción, entrenamiento y evaluación de Redes Neuronales Feedforward para regresión y clasificación (datasets como MNIST o Boston Housing).

Módulo 5.

Aprendizaje Profundo Avanzado y Arquitecturas Convolucionales

- Redes Neuronales Convolucionales (CNNs).
- Implementación Interactiva: Visualización y experimentación con operaciones de Capas Convolucionales y Capas de Pooling.
- Implementación Guiada: Diseño y construcción de una CNN básica para clasificación de imágenes (ej. CIFAR-10) en TensorFlow/Keras o PyTorch.
- Mejorando el Entrenamiento de Modelos Profundos:
 - Ejercicio de Implementación: Aplicación de Normalización por Lotes (Batch Normalization) en una CNN.
 - Aplicación de Data Augmentation: Implementación de técnicas de aumento de datos en datasets de imágenes.
 - Arquitecturas Profundas Notables (Visión general y discusión de innovaciones): LeNet, AlexNet, VGG, ResNet, Inception.
- Transferencia de Aprendizaje (Transfer Learning):
 - Implementación: Uso de redes pre-entrenadas (ej. VGG16, ResNet50) para extracción de características y fine-tuning en nuevas tareas.

Módulo 6.

Aplicaciones en Visión por Computadora

- Introducción a la Visión por Computadora (CV): discusión de tareas y desafíos con ejemplos.
- Aplicación de Herramientas de Procesamiento de Imágenes: Uso de OpenCV y Pillow para carga, manipulación y preprocesamiento de imágenes.
- Clasificación de Imágenes Avanzada:
 - Proyecto de Implementación: Ajuste fino de arquitecturas CNN complejas o modelos pre-entrenados para alta precisión.
 - Localización y Detección de Objetos:
 - Conceptos: Bounding boxes, IoU.
 - Aplicación de Detección de Objetos: Implementación con un modelo pre-entrenado de YOLO (o similar). Análisis de salidas.
- Segmentación de Imágenes:
 - Introducción a Segmentación Semántica y de Instancias.
 - Ejercicio de Aplicación/Demostración: Implementación con un modelo pre-entrenado de U-Net (o similar). Discusión de Mask R-CNN.

Módulo 7.

Procesamiento de Lenguaje Natural (NLP)

- Introducción al Procesamiento de Lenguaje Natural (NLP): discusión de desafíos y tareas comunes con ejemplos.
- Aplicación de Preprocesamiento de Texto: Implementación de Tokenización, stemming, lematización, eliminación de stop words (NLTK, spaCy).
- Representación de Texto (Embeddings):
 - Implementación: Visualización de Bag-of-Words, TF-IDF.
 - Ejercicio con Word Embeddings: Carga y exploración de embeddings pre-entrenados (Word2Vec, GloVe) para análisis de similitud.
- Redes Neuronales Recurrentes (RNNs):
 - Discusión del problema del gradiente desvaneciente/explosivo.
 - Implementación Guiada: Desarrollo de LSTMs y GRUs para clasificación de texto (análisis de sentimiento) en TensorFlow/Keras o PyTorch.
- Mecanismos de Atención y Arquitectura Transformer (Introducción conceptual y demostración).
- Mini-Proyecto de Aplicación en NLP: desarrollo de un clasificador de texto o sistema simple de análisis de sentimiento.

Módulo 8.

Modelos de Lenguaje Avanzados (LLMs) y Proyecto Final

- Introducción a los Modelos Lingüísticos Avanzados (LLMs): Evolución, arquitecturas, capacidades y limitaciones.
- Funcionamiento de los LLMs: Pre-entrenamiento y Fine-tuning (conceptual y ejemplos).
- Aplicación de LLMs:
- Interacción con APIs de LLMs (ej. OpenAI, Hugging Face) para generación de texto, Q&A, traducción, resumen.
- Ejercicio de Prompt Engineering: Aplicación de diseño de prompts efectivos.
- Aplicación de Herramientas y Plataformas: Uso de Hugging Face Transformers para cargar modelos pre-entrenados y realizar inferencia o fine-tuning básico.

• Todos nuestros programas incluyen un certificado de asistencia. Los participantes que cumplan con la asistencia mínima requerida y los requisitos académicos, recibirán su certificado.

• La Universidad del Rosario se reserva el derecho de modificar el equipo académico de los programas de Educación Continua que estén anunciados en la programación, así como podrá modificar el orden temático de desarrollo en este programa, garantizado que se abordará la totalidad de temas propuestos.

• La Universidad del Rosario podrá ordenar las fechas de desarrollo de los programas o de las participaciones de un programa en ejecución, en situaciones que así lo amerite. La notificación a inscritos o a los participantes de los programas se podrá hacer sin un plazo de tiempo previo estipulado, pero siempre intentando mantener un mínimo de horas previas para ello.

• Los programas de Educación Continua son educación informal, no conducen a título profesional.