

CONTENIDO DE LA ASIGNATURA

Programa de Ingeniería de Sistemas y Telecomunicaciones



Asignatura	Programación para la Computación Científica en IA
Semestre	Electiva
Prerrequisito	Sistemas Operativos, Análisis de Algoritmos.
No. Créditos	3
Intensidad horaria	4 horas semanales
Código	CS-01
Período Académico	2010-I
Actualización	Octubre, 2019.

1. JUSTIFICACIÓN

Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos integradas de alto nivel, combinadas con tipeo dinámico y enlace dinámico, lo hacen muy atractivo para el Desarrollo rápido de aplicaciones, así como para usarlo como un lenguaje de scripting para conectar los componentes existentes. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo del mantenimiento del programa. El intérprete de Python y la extensa biblioteca estándar están disponibles en formato fuente o binario sin cargo para todas las plataformas principales, y se pueden distribuir libremente. Las bibliotecas pueden considerarse herramientas o programas que manejan gran parte de la codificación. Bibliotecas como Pandas, SciPy y NumPy permiten manipular y analizar conjuntos de datos masivos.

Por qué elegir Python:

Versatilidad de tareas: Python es un lenguaje más versátil y es más adecuado para manejar tareas relacionadas no estadísticas o no analíticas. Si se planea combinar tareas de ingeniería de software y ciencia de datos, Python es una mejor opción.

Integración de producción simple: Python funciona más fácilmente con otras partes de un sistema de producción. R puede requerir un poco más de trabajo para integrarse porque fue construido por estadísticos.

Múltiples bibliotecas útiles: Bibliotecas de ciencia de datos versátiles. Las bibliotecas de aprendizaje NumPy, Pandas y Scikit cubren gran parte de la funcionalidad para realizar ciencia de datos en Python.

R es un lenguaje de programación construido con la manipulación estadística en mente en el mundo de la ciencia de datos. Existe un ecosistema de paquetes para R para el análisis de datos, la visualización de datos y la ciencia de datos.

¿Por qué elegir R?

Funcionalidad estadística: en R, el análisis estadístico es más natural con un mayor enfoque en la interpretabilidad, no solo en la predicción.

Visualización de datos: la visualización de datos es más intuitiva y estéticamente agradable en R en comparación con las bibliotecas de visualización de datos disponibles en Python.

2. COMPETENCIAS A DESARROLLAR

Proporciona una comprensión del papel que desempeñar la computación en la resolución de problemas.

Conoce las características, funcionalidades y estructura de los lenguajes de programación Python, R y C++.

Identifica y analiza las ventajas para los usuarios en el uso de las bibliotecas y extensiones en la selección, creación, evaluación y administración en las diferentes técnicas de IA.

Conoce y aplica los procedimientos algorítmicos básicos para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

3. OBJETIVO GENERAL

Proporcionar fundamentos de los lenguajes de programación ampliamente empleados como herramientas en las diferentes técnicas de Inteligencia Artificial. Se presenta a los estudiantes un componente de aprendizaje para que puedan competir por proyectos de investigación y sobresalir en materias con componentes de programación. Al aprender tanto R, Python y C++, los estudiantes como analistas de datos ó científicos de datos pueden aprovechar las fortalezas de cada lenguaje de programación.

4. OBJETIVO ESPECÍFICOS

Conocer los fundamentos de la programación científica en R, Python y C++.

Distinguir las nociones de programación en lenguaje de programación.

Aprender técnicas para transformar datos como el análisis de componentes principales (PCA) y el análisis discriminante lineal (LDA).

Distinguir los principios básicos de visualización de datos y cómo aplicarlos usando ggplot2 de R y los paquetes matplotlib y seaborn de Python.

Aplicar la regresión lineal mediante problemas de regresión simple y múltiple.

Conocer los principios básicos de Aprendizaje de Máquina

Implementa y usa algoritmos de aprendizaje automático

Conocer los conceptos fundamentales de ciencia de datos a través de estudios de casos motivadores del mundo real.

Desarrollar lógica para la resolución de problemas.

Desarrollar habilidades de resolución de problemas y su implementación utilizando Python, C++ y R.

Comprender e implementar el concepto.

Comprender el concepto básico de la lógica informática.

Comprender los conceptos de código abierto.

5. CONTENIDO DE LA ASIGNATURA

ACTIVIDADES DE APRENDIZAJE		
Semana	Tema	Laboratorio/Talleres/exámenes
1	Metodología de la Programación <u>Conceptos generales</u> Enfoque modular, claridad y simplicidad de expresiones, uso de nombres propios para identificadores, comentarios, sangría. Documentación y mantenimiento del programa; Programas en Ejecución y depuración, errores de sintaxis, errores de tiempo de ejecución, errores lógicos.	
2	Ciclo de Procesamiento de Datos Presentación de Datos Procesamiento de Datos Almacenamiento de Datos	

CONTENIDO DE LA ASIGNATURA

Programa de Ingeniería de Sistemas y Telecomunicaciones

3	Introducción a la computación con Python y R Jupyter Notebook Ramificación e iteración Descomposición, abstracciones, funciones. Tuplas, Listas, Alias, Mutabilidad, ClonaciónC++ Recursión, Diccionarios Pruebas, depuración, excepciones, afirmaciones Programación orientada a objetos	
4	Vectores, Matrices y Arreglos Multidimensionales Python/R: arreglos y matrices con Numpy, Indexing and Sliding, Expresiones Vectorizadas.	
5	Visualización de Datos Paquete ggplot2 Paquete matplotlib Representación gráfica con plot Parámetros de la función plot Añadir puntos y rectas a un gráfico Cómo agregar texto y curvas a un gráfico Las leyendas de los gráficos Vista previa Segmentos, flechas, símbolos y polígonos Introducción a la representación gráfica Introducción a matplotlib para Python Múltiples gráficos en una misma figura Figuras y ejes Añadir texto a los gráficos Cambios de ejes y escalas Ejercicio: un ejemplo de gráfico detallado en Python	
6	Resolución de ecuaciones e Interpolación -regresión Biblioteca Python: Scipy Sistemas de Ecuaciones Lineales Construyendo un regresor lineal simple usando Python. Algoritmo de descenso de gradiente para minimizar la función de costo. Análisis de regresión multiple. R-valor cuadrado.	
7	Ecuaciones diferenciales parciales Python:Scipy	
8	Optimización:Scipy Mínimos cuadrado Optimización con restricciones	
9	Introducción R Aritmética con R Variables Tipos de Datos Básicos Vectores Matrices Factors	

CONTENIDO DE LA ASIGNATURA

Programa de Ingeniería de Sistemas y Telecomunicaciones



	Data Frames Listas	
10	Computación Simbólica Python: SymPy R: rSymPy y Yacas.	
11	Matrices sparse y grafos Python: Scipy, Numpy, Networkx	
12	Procesamiento de Datos Python: Pandas	
13	Modelamiento Estadístico: con PyMC3 Aleatoriedad Regresión logística Modelos bayesianos de inferencia. Programación probabilista	
14	Introducción C++ Tipos de datos, variables y constantes Operadores y expresiones Control de flujo Funciones y requerimientos Programación Orientada a Objetos en C++ Constructor y destructor Herencia Manipulación de fichero de datos Estructura de datos	

6. ESTRATEGIAS METODOLÓGICAS

Cátedra Magistral: El docente hará inducción a los temas y desarrollará material complementario de consulta para el estudiante, sobre los temas expuestos.

Tareas: El estudiante complementará la información vista en clase, orientándola hacia el lenguaje de programación seleccionado.

Desarrollo de software a través de guías de trabajo: El docente elaborará guías que serán desarrolladas por los estudiantes en sesiones de laboratorio o en forma independiente.

Sesiones de laboratorio: donde el estudiante resolverá algunos interrogantes colocados en las guías de trabajo, con la presencia del docente.

Lecturas: El docente dejará temas para que sean indagados por los estudiantes, de tal manera que al llegar a clase puedan ser discutidos.

7. SOPORTE BIBLIOGRÁFICO

Robert Johansson (2019) - Numerical Python_ Scientific Computing-Apress.

Claus Fuhner, Jan Erik Solem, Olivier Verdier (2017) - Scientific Computing. Packt Publishing.

Peter R. Turner, Thomas Arildsen, Kathleen Kavanagh (2018) - Applied Scientific Computing - Springer.

Tilman M. Davies (2016). The Book of R: A First Course in Programming and Statistics

Norman Matloff (2011). The Art of R Programming: A Tour of Statistical Software Design 1st Edition

Richard Cotton (2013). Learning R: A Step-by-Step Function Guide to Data Analysis 1st Edition

Eric Matthes (2015). Python Crash Course: A Hands-On, Project-Based Introduction to Programming

Paul Barry (2016). Head First Python: A Brain-Friendly Guide

Allen B. Downey (2015). Think Python: How to Think Like a Computer Scientist 2nd Edition

Dr. Harvey Deitel, Paul Deitel (2005). C++ How to Program (5th Edition) 5th Edition.

8. ENLACES DE INFORMACIÓN COMPLEMENTARIA

<https://www.r-project.org/>

<https://www.guru99.com/r-programming-introduction-basics.html>

<https://www.python.org/>

https://www.w3schools.com/python/python_intro.asp

9. PREREQUISITOS

Arquitectura de Computadores

Paradigmas de Programación

Estructuras de Datos Lineales

10. APOYO TECNOLÓGICO

Conocimientos de Programación C.

Pc con linux Os - ubuntu 18.04

Metodología Práctica: **Sala de computadores**, para prácticas de programación y simulación.

11. SISTEMA DE EVALUACIÓN

- 1 Primer Corte 30%
 - 10% parcial
 - 5% Talleres
 - 7,5% Prácticas de Laboratorios.
 - 7,5% Informes escritos.
- 1 Segundo Corte 30%
 - 10% parcial
 - 5% Talleres
 - 7,5% Prácticas de Laboratorios.
 - 7,5% Informes escritos.
- 1 Tercer Corte 40%
 - 10% parcial
 - 5% Talleres
 - 7,5% Prácticas de Laboratorios.
 - 7,5% Informes escritos.
 - 10% Proyecto final