

PROGRAMA DE CURSO

Código	Nombre				
MA6202	Laboratorio de Ciencia de Datos				
Nombre en Inglés					
Data Science Laboratory					
SCT	Unidades Docentes	Horas de Cátedra	Horas Auxiliar	Docencia	Horas de Trabajo Personal
6	10	3	1,5		5,5
Requisitos			Carácter del Curso		
CC5206 / EL4106 / IN6531 / MA5204 / AUTOR			Electivo de Carrera, Magister y Doctorado.		
Resultados de Aprendizaje					
<p>Este curso entrega las herramientas necesarias, tanto teóricas como prácticas, en el modelamiento, resolución y puesta en marcha en desafíos en ciencia de datos. Según este diseño, el estudiante:</p> <ul style="list-style-type: none"> • Adquiere conocimientos que le permiten formular, implementar e interpretar modelos matemáticos avanzados en análisis de datos. • Profundiza en herramientas teóricas de aprendizaje de máquinas, considerando de manera simultánea los aspectos relativos a su implementación eficiente. • Es capaz de enfrentar problemas reales de ciencia de datos, contemplando análisis exploratorios, limpieza, implementación eficiente de modelos matemáticos y despliegue de una solución en la nube. • Puede diseñar nuevas herramientas computacionales para resolver problemas de análisis de datos en caso de ser necesario. 					

Metodología Docente	Evaluación General
<p>La metodología del curso comprende:</p> <ul style="list-style-type: none"> • Clases expositivas de 90 minutos • Demostraciones de los métodos aprendidos • Clases auxiliares enfocadas al trabajo práctico, con base en las herramientas aprendidas durante el curso 	<p>La herramientas y técnicas proporcionadas por el curso son evaluadas por medio de:</p> <ul style="list-style-type: none"> • Ejercicios de programación resueltos en clase. • Tareas • Despliegue de una aplicación

Unidades Temáticas

Número	Nombre de la Unidad	Duración en Semanas
1	Introducción a Python en ciencia de datos	2
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> ● Introducción a ciencia de datos. ● Sistema de control de versiones git. ● Elementos de programación en Python: estructuras y métodos base. ● Elementos programación orientada a objetos en Python. ● Librerías base: functools, itertools, Numpy, Scipy. 	<p>Los estudiantes se contextualizan y aprenden un sistema de control de versiones para implementar sus proyectos, se familiarizan con el lenguaje de programación Python, sus métodos generales y la programación orientada a objetos y aprenden a utilizar los paquetes básicos del stack científico.</p>	[1],[2],[3]

Número	Nombre de la Unidad	Duración en Semanas
2	Manejo de datos y visualización	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> ● Exploración y manipulación usando Pandas y SQL. ● Visualización de datos usando Matplotlib y Seaborn. ● Técnicas de clustering y reducción de dimensionalidad utilizando Scikit-Learn. 	<p>Los estudiantes aprenden a manejar, analizar y visualizar datos. Son capaces de limpiar datos corruptos de manera eficiente, además de explorar y obtener visualizaciones claras y concordantes con el fenómeno estudiado. Los estudiantes revisitan e implementan algoritmos clásicos de aglomeración</p>	[2],[4],[5],[6]

Número	Nombre de la Unidad	Duración en Semanas
3	Computación de alto rendimiento en modelos de aprendizaje automático	4
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Teoría en modelos de aprendizaje automático e implementación usando Scikit-Learn. • Teoría en modelos gráficos probabilísticos e implementación usando Pomegranate. • Elementos de computación paralela y distribuida. • Profiling. • Herramientas de computación de alto rendimiento en Python: Multiprocessing y Dask. 	<p>Los estudiantes comprenden las fortalezas y limitaciones de distintos modelos teóricos y son capaces de implementarlos con la finalidad de evaluarlos en distintas aplicaciones. Los estudiantes aprenden técnicas de paralelización y vectorización, son capaces de implementar modelos de aprendizaje automático en datos reales, pudiendo evaluar y mejorar su eficiencia computacional.</p>	<p>[6],[5],[7],[8],[13]</p>

Número	Nombre de la Unidad	Duración en Semanas
4	Deep Learning, programación probabilística y explicabilidad	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> • Implementación de modelos de aprendizaje profundo en GPU usando Pytorch. • Programación probabilística profunda basada en GPU usando Pyro. • Métodos de explicabilidad de modelos. 	<p>Los estudiantes manejan herramientas de Deep Learning y programación probabilística basadas en GPU. Comprenden distintas arquitecturas de red y son capaces de implementar técnicas de explicabilidad agnósticas al modelo. Los estudiantes evalúan distintas arquitecturas de redes de forma experimental y confirman la teoría aprendida.</p>	<p>[9],[10],[11]</p>

Número	Nombre de la Unidad	Duración en Semanas
5	Despliegue en la Nube	3
Contenidos	Resultados de Aprendizajes de la Unidad	Referencias a la Bibliografía
<ul style="list-style-type: none"> ● Visualización interactiva usando Bokeh ● Gestión de contenedores usando Docker ● Aplicaciones web minimalistas en Python mediante Flask ● Despliegue en la Nube 	Los estudiantes aprenden a construir aplicaciones autocontenidas e interactivas sobre desafíos de ciencia de datos. Despliegan dichas aplicaciones en la Nube siendo capaces, por sí mismos, de llevar a cabo un proyecto desde prueba de concepto a producción.	[5],[12]

Bibliografía General
<p>[1] Joel Grus, Data Science from Scratch, First Principles with Python, O'Reilly, 2015</p> <p>[2] Wes McKinney, Python for Data Analysis, Data Wrangling with Pandas, NumPy, and IPython, O'Reilly, 2017</p> <p>[3] Jake VanderPlas, Python Data Science Handbook Essential Tools for Working with Data, O'Reilly, 2016</p> <p>[4] Tim Grobmann, Mario Dobler, Data Visualization with Python, O'Reilly, 2019</p> <p>[5] Sebastian Raschka, Python Machine Learning, Packt, 2019</p> <p>[6] Thomas Nield, Getting Started with SQL A Hands-On Approach for Beginners, O'Reilly, 2016</p> <p>[6] Giancarlo Zaccone, Python Parallel Programming Cookbook, Packt 2019</p> <p>[7] Schreiber, Pomegranate: Fast and Flexible Probabilistic Modeling in Python, Journal of Machine Learning Research, 2018</p> <p>[8] Dask: Library for dynamic task scheduling, Dask Development Team, 2016</p> <p>[9] Christoph Molnar, Interpretable Machine Learning A Guide for Making Black Box Models Explainable, 2019.</p> <p>[10] Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca, Python Deep Learning Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and Tensorflow, Packt, 2019</p> <p>[11] Bingham. Chen, Jankowiak, Obermeyer, Pradhan, Karaletsos, Theofanis, Rohit ,Szerlip, Horsfall, Goodman, Pyro: Deep Universal Probabilistic Programming, Journal of Machine Learning Research, 2018.</p> <p>[12] Joshua Cook, Docker for Data Science, Building Scalable and Extensible Data Infrastructure Around the Jupyter Notebook Server, Apress, 2017</p> <p>[13] K. Murphy, Machine learning: A probabilistic perspective, MIT, 2012.</p>

Vigencia desde:	Otoño 2020
Elaborado por:	Nicolás Caro, Felipe Tobar
Revisado por:	Daniel Remenik – Jefe Docente