

ECO MS 308510: Métodos numéricos e modelos computacionais em economia

Daniel Cajueiro

7 de novembro de 2025

Primeiro Semestre de 2026 – Departamento de Economia
Horário: Terças e Quintas 10 as 12.

1 Objetivos

Introduzir diferentes paradigmas de computação como programação estruturada, orientação a objeto e programação funcional.

Introduzir diversas estratégias de projeto de algoritmos como, por exemplo, divida e conquiste, backtracking, branch and bound e programação dinâmica.

Introduzir algoritmos clássicos e atuais que aparecem em várias situações diferentes, suas aplicações e os custos computacionais associados.

Explorar soluções de problemas numéricos que aparecem com frequência em aplicações computacionais.

Introduzir abordagens computacionais que naturalmente aparecem em economia como modelos baseados em agentes, modelos baseados em redes e simulações monte carlo.

Discutir problemas computacionais em teoria da decisão.

Desenvolver várias aplicações em dados usando abordagens baseadas em econometria e aprendizagem de máquinas.

Discutir questões centrais do aprendizado de máquina que podem impactar significativamente a sociedade devido ao uso crescente da inteligência artificial como risco algorítmico, viés algorítmico, falta de interpretabilidade, questões de privacidade, impactos sobre o mercado de trabalho, desigualdade e concentração de poder econômico, interação homem máquina e regulação.

Explorar a discussão atual de computação quântica e seus impactos.

2 Ementa

Programação estruturada. Algoritmos. Análise numérica. Otimização numérica. Aprendizado por reforço, programação dinâmica e métodos Monte Carlo. Aprendizagem de Máquinas. Aplicações em Economia, Econometria e Teoria da Decisão.

3 Dinâmica

O curso será composto por aulas expositivas e muitos exercícios computacionais. No caso de haver falta de tempo para dar todos os tópicos, uma pequena parte do curso poderá ser compartilhada através de aulas gravadas.

As aulas gravadas devem ser usadas apenas pelos estudantes devidamente matriculados no curso e nunca serem divulgadas em redes sociais ou compartilhadas de alguma forma com indivíduos não matriculados no curso.

4 Linguagens computacionais

O objetivo do curso não é necessariamente ensinar uma linguagem de programação. A grande maioria dos conceitos estudados nesse curso podem ser implementados na maioria das linguagens. Sugere-se que o estudante escolha uma que atenda os seus objetivos e se torne fluente nela. Entretanto, todos os exemplos do curso serão feitos em Python e alguns exercícios do curso exigirão o uso dessa linguagem. Mas existe espaço suficiente para cada aluno trabalhar com a linguagem que desejar, embora possa não ser a escolha mais eficiente, se ele já não dominar muito bem a linguagem de sua escolha ou se ele não tiver bibliotecas que ajudem no desenvolvimento da parte aplicada do curso.

5 Avaliação

Não haverá prova escrita formal.

30% da nota:

Avaliação oral individual, baseada em um tema escolhido em conjunto com o estudante.

35% da nota:

Apresentação de um artigo acadêmico selecionado a partir de uma lista previamente disponibilizada. A apresentação ocorrerá, preferencialmente, nas últimas aulas do curso.

35% da nota:

Apresentação de uma aula didática sobre um tópico relacionado ao conteúdo do curso, escolhido a partir de uma lista complementar de temas. Essa avaliação pode ser em dupla ou trio, dependendo do tamanho da turma, se o estudante preferir.

6 Referências

Existe uma lista longa de referências que podem ser usadas no curso. Elas estão listadas no PRorum:

<http://prorum.com/index.php/2028/referencias-principais-computacionais-lecionado-doutorado>

Referências principais:

Introduction to Algorithms - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein

Introduction to the Design and Analysis of Algorithms - Anany Levitin

Reinforcement Learning: An Introduction - Richard S. Sutton (Author) e Andrew G. Barto

Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning - Alan J. Izenman.

Deep Learning - Yoshua Bengio, Ian Goodfellow e Aaron Courville