

## Aprendizagem Automática II

### Trabalho prático – ano letivo 2018/2019

---

O trabalho prático a realizar nesta UC terá duas opções distintas:

- Opção 1: Análise de um conjunto de dados selecionado, usando a linguagem Python e *packages* disponíveis.
- Opção 2: Trabalho de desenvolvimento de algoritmos/ software no âmbito da Aprendizagem Máquina

Em ambos os casos, o trabalho será organizado em fases que são detalhadas em seguida.

#### Opção 1:

Neste caso, o grupo deverá começar por escolher o conjunto de dados que irá ser usado na análise de dados. Idealmente, estes dados deverão ser retirados de algum tipo de publicação (e.g. artigo científico). Não se recomenda a escolha de conjuntos de dados de competições onde existam já muitas soluções disponíveis por parte da comunidade, mas participar em competições a decorrer com soluções inovadoras é uma opção recomendada. Abaixo, são indicadas algumas possibilidades sugeridas pelo docente como ilustração. O conjunto de dados escolhido será validado pelo docente.

A análise dos dados deverá ser dividida em duas fases:

- Na fase 1, deverão ser incluídas as etapas de carregamento e preparação dos dados (incluindo se necessário a geração dos atributos) e sua exploração inicial, pré-processamento, aprendizagem não supervisionada, redução de dimensionalidade, seleção de atributos e aprendizagem supervisionada (usando métodos “tradicionais”) incluindo a otimização dos seus hiper-parâmetros. Deverão ser usados e avaliados (de forma robusta) diversos métodos de Aprendizagem Máquina mais tradicional, de acordo com as características dos dados.
- Na fase 2, deverá ser desenvolvido pelo menos um workflow que permita o uso, sobre os mesmos dados, de uma metodologia de *deep learning*, comparando os resultados com os obtidos na fase 1. Deverão ser exploradas alternativas em termos das configurações destes métodos, podendo ser testadas também classes distintas de modelos (e.g. *feedforward*, recorrentes, convolucionais).

Em ambos os casos, deverão ser criados *Jupyter Notebooks* contendo os *scripts* usados e os resultados obtidos, bem como incluindo a explicação das razões para cada um dos métodos e discutindo-se os resultados obtidos. Estes podem usar ficheiros de código adicional desenvolvido pelo grupo (módulos incluindo funções ou classes). Todo o código desenvolvido por terceiros que seja usado deve ser devidamente identificado. Opcionalmente, o grupo pode colocar o código desenvolvido num repositório público (e.g. github).

#### Opção 2:

Neste caso, o grupo deverá começar por escolher o tema para o desenvolvimento a efetuar, enviando na data abaixo indicada. Esta deve incluir os objetivos do desenvolvimento a atingir em cada uma das fases seguintes. Os temas sugeridos serão validados pelo docente.

No final de cada uma das fases, o grupo deve escrever um relatório sucinto do trabalho desenvolvido (máximo de 3 páginas), juntando os ficheiros relevantes (código, tabelas de resultados, etc). Nesta opção, não será imposta uma linguagem para o desenvolvimento de software. Opcionalmente, o grupo pode colocar o código desenvolvido num repositório público (e.g. github).

Prazos e condições de entrega:

- Constituição do grupo e escolha do conjunto de dados/ tema de desenvolvimento – indicar elementos do grupo, escolha realizada e suas razões (sucintas, máximo 1 página) – submissão no e-learning até dia **2 de abril**.
- Entrega material fase 1: submissão dos ficheiros no e-learning até dia **30 de abril**
- Entrega material fase 2: submissão dos ficheiros no e-learning até dia **28 de maio**
- Apresentação final: 7 de junho (a confirmar)

Sugestões de conjuntos de dados (opção 1):

- Kaggle: competições ativas - <https://www.kaggle.com/competitions>
- Competições do site DrivenData: DengAI , PumpItUp - <https://www.drivendata.org/competitions/>
- Numenta anomaly benchmark - <https://github.com/numenta/NAB>
- Datasets do artigo: “Deep Neural Nets as a Method for Quantitative Structure–Activity Relationships” (previsão da atividade de compostos) - [https://acs.figshare.com/articles/Deep\\_Neural\\_Nets\\_as\\_a\\_Method\\_for\\_Quantitative\\_Structure\\_Activity\\_Relationships/2193742/2](https://acs.figshare.com/articles/Deep_Neural_Nets_as_a_Method_for_Quantitative_Structure_Activity_Relationships/2193742/2)
- UCI Machine Learning Repository - <https://archive.ics.uci.edu/ml/datasets> (verificar algumas das entradas recentes)

Sugestões para a opção 2:

- Desenvolvimento de funções para exploração de ensembles de redes neuronais com complexidade heterogénea (usando python; keras)
- Algoritmos evolucionários para seleção de atributos usando modelo wrapper (usando python; package inspyred e scikit-learn)
- Exploração do package NEAT-Python para neurevolution
- Desenvolvimento de LSTMs para problemas de routing em redes de computadores