

Projeto Final: Padrões de Projeto em Aprendizado de Máquina

Descrição geral

- Cada equipe deverá codificar uma solução computacional usando orientação a objetos e os padrões de projeto abaixo (definido por sorteio).
- Além da codificação dos algoritmos em si, cada equipe precisará entregar um relatório (até 15 páginas) e preparar uma apresentação do trabalho (entre 10 e 15 minutos). A avaliação do trabalho também compreende uma arguição oral dos autores pelo professor/alunos.
- Os fontes e o relatório do trabalho devem ser enviados em uma única pasta compactada em formato “.zip” via plataforma Teams até a data de entrega definida acima.

Descrição específica

- O trabalho contempla o uso de padrões de projeto para o desenvolvimento de soluções fictícias inspiradas em pipelines de aprendizado de máquina e inteligência artificial.
- Três algoritmos de aprendizado precisam ser considerados na solução: rede neural artificial, k-vizinhos mais próximos e árvore de decisão. De acordo com a saída desses algoritmos, existem dois tipos de problemas: classificação no qual a saída do algoritmo é um valor discreto; e regressão no qual a saída do algoritmo é um valor contínuo.
- Destaque para os seguintes atributos e métodos de cada algoritmo:
 - Árvore de decisão possui como atributo o número de árvores, a medida de informação e o tipo de tarefa (regressão ou classificação). Entre os métodos principais estão *fit* e *predict*, os quais realizam respectivamente o treinamento e teste dos modelos de árvore de decisão, bem como o método *imprimeModelo* que informa os valores dos atributos da árvore e o método *avaliaModelo*¹ que informa se o desempenho de predição de um algoritmo é insatisfatório, regular, bom ou muito bom;
 - K-vizinhos mais próximos (KNN) possui como atributo número de vizinhos, medida de proximidade e o tipo de tarefa (regressão ou classificação). Os métodos principais são os mesmos informados para árvore de decisão.
 - Rede neural artificial possui como atributo o número de camadas, o número de neurônios por camada, a função de ativação, a taxa de aprendizado e o tipo de tarefa (regressão ou classificação). Os métodos principais são os mesmos informados para os demais algoritmos.
- A execução de um método *fit* implica em carregar os dados, normalizá-los e então ajustar o modelo de aprendizado;
- A execução de um método *predict* implica em carregar um novo dado, carregar o modelo de aprendizado ajustado anteriormente e então prever a saída correspondente ao novo dado;
- Considere que cada algoritmo pode funcionar em três modos: modelo desajustado (*unfitted*), modelo ajustado (*fitted*) e modelo iterativo (*iterFit*). No cenário *unfitted*, o modelo realiza apenas o ajuste sobre os dados disponíveis, não podendo prever novos dados; no cenário *fitted*, o modelo realiza a predição de novos dados apresentados a ele, não podendo ser ajustado aos dados disponíveis; e no cenário *iterFit*, o modelo realiza a predição de novos dados (*predict*) e em seguida, automaticamente, realiza um novo ajuste do modelo (*fit*). As seguintes transições precisam ser configuradas entre os modos:
 - Todos os modelos são inicializados no modo *unfitted*;

¹Por razões práticas, recomenda-se o uso de uma função aleatória no método.

- Após a primeira chamada do método `fit`, o modo passa a ser *fitted*;
- Caso o desempenho do modelo seja insatisfatório, o modo dele retorna ao estado *unfitted* para novo ajuste;
- Caso o desempenho do modelo seja regular, ele passa para o modo *iterFit*;
- Caso o desempenho do modelo seja bom ou ótimo, ele permanece no estado *fitted*.

Sobre a implementação das soluções

- Recomenda-se fortemente que as soluções sejam codificadas em linguagem Java.
- Para envio dos fontes, crie uma pasta que contenha os fontes necessários para execução das soluções em ambiente linux/Fedora 31-36, além de um arquivo README explicando passo a passo a instalação, compilação e execução dos fontes.
- Reitera-se o caráter autoral do trabalho. Logo, as duplas não devem usar ou se aproveitar de “soluções prontas” na realização do trabalho.

Sobre o relatório

O relatório de até 15 páginas deve conter uma explicação sobre os padrões de projeto investigados, além de apresentar os diagramas de classes correspondentes à codificação bem como discutir a composição de pelo menos dois padrões, entre outras decisões de projeto. É importante que a equipe também considere o desenvolvimento do trabalho priorizando a divisão da solução em camadas, tal como o padrão de arquitetura MVC.

Apresentação do trabalho

- A equipe deverá preparar uma apresentação de slides cobrindo os pontos solicitados para o relatório.
- A equipe deverá apresentar o trabalho de modo que o tempo de apresentação de cada integrante seja similar.
- A apresentação deverá ter duração entre 10 e 15 minutos.

Tópicos

Os padrões de projeto abaixo precisam ser considerados por cada equipe sorteada. A equipe poderá usar outros padrões de projeto adicionais, conforme as necessidades identificadas.

1. Strategy, template, facade
2. Strategy, state, facade
3. Strategy, template, state
4. Strategy, factory, facade
5. Strategy, factory, state
6. Abstract Factory, state, singleton
7. Abstract Factory, strategy, singleton