Universidade Federal de São Carlos
Pós-Graduação em Ciência da Computação
Aprendizado de Máquina
Prof. Tiago A. Almeida <talmeida@ufscar.br>

Exercício 10
Análise de Agrupamentos

Objetivo

Ao contrário do aprendizado supervisionado, no qual existe um atributo-alvo que deseja-se ser previsto, o aprendizado não-supervisionado tem como característica um conjunto de amostras sem atributos-alvos associados, sobre o qual busca-se descobrir relações ou padrões entre os dados.

A subárea mais popular do aprendizado não-supervisionado é o da análise de agrupamentos. O objetivo principal em problemas de agrupamento é dividir as amostras da base de dados em grupos (também chamados de *clusters*) que possuam características em comum. Existem muitas aplicações que essa técnica pode ser empregada: separar clientes por padrão de compra, detectar fraudes em seguros, identificar expressões gênicas, detectar epicentros de acidentes naturais, entre outros inúmeros exemplos.

Uma enorme quantidade de algoritmos de agrupamentos foi proposta ao longo dos anos, cada um com suas estratégias e características específicas. Devido ao fato de o problema não possuir um atributo-alvo, a avaliação de algoritmos de agrupamento é um processo complexo, podendo cada algoritmo ser indicado para um problema específico.

Neste exercício, será implementado um dos algoritmos de agrupamento mais tradicionais da área, o K-Médias. Por ser de fácil aplicação e apresentar bons resultados em problemas reais, o algoritmo é bastante utilizado até hoje, servindo como um dos principais algoritmos para *baseline* de novas propostas.

Após concluir o exercício, espera-se que você consiga aplicar o algoritmo K-Médias em qualquer problema de análise de agrupamento que tentar resolver. Através da implementação de cada etapa do algoritmo, é esperado que você consiga entender perfeitamente o seu funcionamento.

O exercício

Ao longo do exercício, você deverá completar 3 funções:

- `findClosestCentroids`: responsável por encontrar os centroides mais próximos de cada amostra;

- `calculateCentroids`: responsável por calcular os novos centroides baseado no agrupamento realizado; e
- `calculateCost`: responsável por calcular o custo obtido para um determinado agrupamento.

Preencha o código apenas nos espaços delimitados por comentários, normalmente iniciados por um comentário "`COMPLETE O CÓDIGO AQUI`" e instruções para a implementação.

As implementações devem genéricas e funcionar para qualquer conjunto de dados. Na avaliação, as funções serão testadas em bases com quantidade de amostras e atributos diferentes das fornecidas com o exercício, tendo em comum apenas o nome da coluna que contém a classe das amostras. Não adicione comandos do tipo `print` ou `display` dentro das funções que serão completadas, apenas o código da função..

Aproveite para interagir com o *notebook*: teste valores diferentes para os números de *clusters* e procure entender como o método funciona.

Os casos de teste

Este exercício possui **5 casos de teste** que buscam avaliar cada uma das funções implementadas. A distribuição de tarefas avaliadas por cada caso é feita da seguinte forma:

- **Casos de teste 1 e 2:** testa a função `findClosestCentroids`;
- **Caso de teste 3 e 4:** testa a função `calculateCentroids`;
- **Caso de teste 5:** testa a função `calculateCost`.