UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
COORDENADORIA DE ENGENHARIA ELÉTRICA

Inteligência Artificial Aplicada - Trabalho Prático 1

Ana Carolina Rodrigues Almeida 140950050
João Victor Galdino Bouzon 160950007
Luis Fernando Macêdo Innocêncio 160950019

Algoritmo K-means, Fuzzy C-means e CNN

São João del-Rei, Outubro de 2020.

1 OBJETIVOS

Iniciar o desenvolvimento das competências (conteúdos, habilidades e atitudes) neces-sárias para se tornar um profissional de IA a partir do estudo e da compreensão dos algortimos de classificação K-means, Fuzzy C-means e KNN por meio da plataforma Anaconda, a partir do uso de conjuntos de dados (dataset).

Os objetivos específicos são:

1. Identificar as características ou atributos referentes a um elemento que compõe um conjunto de dados;
2. Entender como selecionar as características que maximizam a taxa de acertos de algo-ritmo de agrupamento ou classificação;
3. Comparar métricas distintas usadas no cálculo da distância e comparação entre os dados e os vetores protótipos;
4. Compreender a matemática usado nas técnicas de exemplo (K-means, KNN e Fuzzy C-means);
5. Entender o efeito dos parâmetros de ajuste no resultado final de clusterização;
6. Implementar algoritmos a partir de exemplos da internet.

2 RECURSOS E PRÉ-REQUISITOS

Anaconda (Faça o download em https://www.scilab.org/ e instale antes do momento síncrono);

Spyder;

Bibliotecas: scikit, sklearn, skfuzzy e numpy.

Datasets

3 ATIVIDADES

A atividade a ser entregue consiste na implementação de quatro algoritmos de cluste-rização: K-means, KNN, Fuzzy C-means e um quarto a sua escolha (Sugestão: escolher um do endereço: https://scikit-learn.org/stable/modules/clustering.html Seção 2.3.1. Segue uma sugestão de atividades para que os objetivos sejam alcançados.

1. Instale a plataforma Anaconda (ou outro ambiente Python a sua escolha) e as bibliotecas;
2. Crie uma pasta e copie os códigos em exemplo enviados no portal didático;
3. Rode os códigos exemplo para verificar se tudo está funcionando adequadamente;
4. Divida o dataset iris em duas partes, um para treinamento e outro para verificar a porcen-tagem de acertos.
5. A partir dos algortimos de exemplo, crie uma forma de medir o índice de desempe-nho de cada técnica de classificação que mostre a porcentagem de acertos para cada grupo/classe;
6. Escolha um dataset diferente do iris e o divida em dois;
7. Implemente as 4 técnicas para esse nova dataset.
8. Crie uma tabela e coloque os resultados iniciais;
9. Tente modificar parâmetros de sintonia como o *m* no Fuzzy C-menas o número de vizinhos *k* no KNN e tente excluir ou combinar diferentes atributos/caratcterísticas presentes no dataset escolhido.
10. Acrescente uma coluna a mais na tabela para cada mudança que foi feita com o intuito de melhorar o índice de acertos. Anote como sua tentativa melhorou ou piorou os índices e compare os resultados, indicando vantagens e desvantagens de cada técnica.
11. Documente tudo conforme dividido nas próximas seções e lembre-se que plágio é ilegal e não traz aprendizado.
12. Crie um github ou gitlab e coloque os disponibilize os códigos comentandos.

4 REFERÊNCIAS

<https://scikit-learn.org/stable/modules/clustering.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier. html>

<https://scikit-learn.org/stable/modules/neighbors.html>

<https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_cmeans.html>

5 METODOLOGIA

Inicialmente importamos as bibliotecas numpy para aplicação de cálculos, a biblioteca panda para funções matriciais e a biblioteca Sklearn com os datasets.

Escolhemos o datasheet dos vinhos da biblioteca do sklearn. Este dataset possuiu 178 amostras, 3 classes e 13 parâmetros a serem analisados.

Para a técnica do C-Means nós agrupamos o dataset em 3 clusters usando o código base da técnica e aplicamos a métrica **metrics.adjusted_mutual_info_score** para contagem de acertos. Visto que o código tem como objetivo apenas agrupar os dados por similaridade entre si, a reposta da rotina será depois classificada pelo operador tendo este a responsabilidade de identificar cada conjunto gerado pela rotina.

Com relação à técnica do KNN é preciso destacar inicialmente que não é possível trabalhar por meio de clusterização visto que esta não se utiliza deste método e sim do que classificação, ou seja, o diferente da técnica do C-Means. A métrica que utilizamos para determinar a exatidão do resultado dado pela técnica foi a **knn.score.** Alem das bibliotecas já importadas anteriormente, para esta técnica também fizemos uso da biblioteca matplotlib.pyplot que nos dá uma visualização gráfica.

Escolhemos ainda uma outra técnica chamada DBSCAN - *Density Based Spatial Clustering of Applications with Noise.* Esta consiste na criação de clusters em regiões de alta densidade de concentração de dados ignorando regiões de baixa densidade de concentração de dados. Como o próprio nome da técnica aponta, ela é indicada para datasets que contém ruído por ignorar outliers do banco de dados. Diferente do C-Means esse método pode gerar clusters irregulares enquanto o C-Means gera clusters esféricos. Entretanto o DBSCAN não é indicado para datasets que tenham dados espalhados de forma uniforme dada a forma que este separa os dados dos outliers. Fizemos uso da mesma métrica usada no C-Means, a **metrics.adjusted_mutual_info_score.**

Entretanto para a técnica em fuzzy nosso grupo não conseguiu material para criação da rotina de medição de acertos. Sendo assim poderemos apenas entregar nosso trabalho realizado com as técnicas acima descritas.

6 RESULTADOS

Após compilar os códigos de cada técnica para cada dataset nos obtivemos:

C-Means Iris 0,75512
C-Means Wine……..…….0,42268
KNN Iris…….……..…….0,97777
KNN Wine.....0,77777
DBSCAN Iris.....0,59899
DBSCAN Wine.....0,41932

7 DISCUSSÃO DOS RESULTADOS

De acordo com os resultados expostos acima é possível verificar que o a técnica do KNN apresenta um maior índice de acertos para os dois datasets. Quanto aos datasets podemos também notar que o numero de acertos é maior para o Iris, isso provavelmente se dá pelo banco de dados do Wine ter 13 variáveis sendo levadas em consideração enquanto o Iris tem apenas 4, o número maior de variáveis implica numa menor precisão por levar em conta muito mais informação.

8 CONCLUSÃO

Este trabalho pratico teve como ótimo ponto positivo o incentivo de aplicar as técnicas de classificação e agrupamento de dados abordadas nas aulas. Pudemos ver na pratica as diferenças entre as técnicas e suas eficácias para determinado tipo de banco de dados. Já na parte negativa como já foi descrito na metodologia não fomos capazes de fazer o código para a técnica do Fuzzy C-Means ficando para nós depois pesquisarmos e com auxílio do professor em breve conseguir por em pratica o algoritmo.

SABER	INDAGAR	APRENDER	REFLETIR
O que eu sabia antes de fazer a atividade?	O que eu ainda preciso apren-der?	O que eu aprendi?	Como eu aprendi? Quais estratégias utilizei e como posso usá-las em outras situações de aprendizagem?
Básico de python	Fuzzy C-means	Métricas	Tutoriais do Sklearn
Básico das técnicas	Python avançado	Clusterização	
Bibliotecas python		Classificação	

Tabela 8.1 – Tabelar SIAR: Simulação de algortimos de classificação e agrupamento.

9 NOVAS REFERÊNCIAS

<https://towardsdatascience.com/k-means-vs-dbscan-clustering-49f8e627de27>