Departamento de Computação e Matemática
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
Universidade de São Paulo (USP)

Relatório Parcial (01/09/2019 – 31/02/2020) – Programa Institucional de Bolsas de Iniciação Científica (PIBIC)

Projeto 121814/2021-1

Vigência: 01/09/2021 – 31/08/2022

Classificação Baseada em K-Vizinhos Mais Próximos e no Grafo de Interações Nk

Gustavo Fernandes Carneiro de Castro
(Aluno)

Prof. Dr. Renato Tinós
(Orientador)

Ribeirão Preto, Março de 2021

O relatório está dividido da seguinte forma:

- Seção A: Resumo do Projeto;
- Seção B: Introdução;
- Seção C: Objetivos;
- Seção D: Materiais e Métodos;
- Seção E: Resultados e Discussão;
- Seção F: Conclusões;
- Seção G: Cronograma da etapa futura;
- Seção H: Referências;

SEÇÃO A.
RESUMO DO PROJETO

K-Vizinhos mais Próximos (K-Nearest Neighbors – KNN) é um algoritmo de classificação não-paramétrica utilizado bastante simples e intuitivo: um novo objeto é classificado com o rótulo da classe majoritária entre os *K* vizinhos mais próximos ao objeto. Neste projeto, propõe-se o uso do grafo de interações Nk para retornar os *K* vizinhos mais próximos no algoritmo KNN. O grafo de interações Nk, originalmente empregado em clusterização, é construído com base na distância e densidade espacial de objetos em pequenos grupos formados por *k* objetos. Ao usar a distância aliada à densidade espacial, possibilita-se a formação de aglomerados com formatos arbitrários, diferentes dos aglomerados hiper-elipsóides gerados pelo KNN original, devido ao uso de uma única métrica espacial, a distância euclidiana. Duas variações do método serão investigadas; os algoritmos diferem na forma em que os vértices associados aos *N* objetos da base de treinamento são visitados. Os $K = k$ objetos relacionados aos vértices visitados são retornados como vizinhos mais próximos. As variações propostas serão comparadas entre si e com o KNN original em experimentos com bases de dados com propriedades diversas para se observar a eficiência deste método de classificação original, apelidado de KNN modificado.

SEÇÃO B.
INTRODUÇÃO

Este é o relatório parcial do projeto de iniciação científica PIBIC de número 121814/2021-1 em desenvolvimento pelo bolsista Gustavo Fernandes Carneiro de Castro.

Classificação é um importante problema estatístico e de aprendizado de máquina [KOTSIANTIS et al, 2007]. Atualmente, classificação é requisito para diversas tecnologias aplicadas no nosso dia-a-dia, como reconhecimento de fala, identificação biométrica, visão computacional, sistemas de recomendação, entre outros.

Os algoritmos utilizados para classificação são normalmente empregados utilizando métricas de distância, especialmente a Euclidiana. Essa característica comum entre a maioria dos algoritmos facilita a formação de aglomerados hiper-elipsóides de objetos com uma mesma classe. Com isso, objetos pertencentes a aglomerados arbitrários, que não seguem este padrão de organização, definidos tanto pela distância como pela densidade espacial [RODRIGUEZ & LAIO, 2014], [ESTER et al., 1996], acabam por não serem classificados corretamente por estes algoritmos.

O algoritmo de classificação dos K-vizinhos mais próximos (K-nearest neighbors algorithm - KNN) é um destes algoritmos de base Euclidiana: um novo objeto é classificado com o rótulo da classe majoritária entre os *K* vizinhos mais próximos ao objeto, determinados de acordo com a distância até o novo objeto. Apesar deste problema, o algoritmo KNN é simples, intuitivo e um dos primeiros métodos eficientes de classificação não-paramétrica [FIX, 1985] , o que o torna uma ótima base para diversos outros algoritmos de classificação, regressão e, com algumas modificações, para outros problemas de aprendizado de máquina.

Aproveitando das vantagens do KNN, mas tentando solucionar os problemas advindos do uso de apenas uma única métrica (distância) para definição dos vizinhos próximos, propôs-se aqui o KNN modificado. Este é um algoritmo baseado no grafo de interações Nk, que é formado a partir de duas métricas: a distância Euclidiana e a densidade espacial. O grafo de interações Nk foi proposto inicialmente no NK Hybrid Genetic Algorithm - NKGA [TINÓS et al., 2018], utilizado para problema de *clustering*. O grafo de interações Nk consiste de *N* vértices, cada um para um dos *N* objetos na base de dados. Cada vértice é ligado a *k-1* objetos por meio de arestas definidas por densidade espacial e distância Euclidiana. No problema de

clustering, *N* grupos de *k* objetos definidos pelo grafo são utilizados para o cálculo da função de validação interna NKCV2, utilizada em *clustering*. Aqui, a função NKCV2 não é utilizada no KNN modificado: o grafo de interações Nk é utilizado para definir os *K* vizinhos de um objeto qualquer.

Em [MORAES & TINÓS, 2020], o grafo de interações Nk foi utilizado para o problema de busca por similaridade. Basicamente, o método proposto em [MORAES & TINÓS, 2020] retorna *K* objetos similares ao objeto consultado visitando *k* (*K* = *k*) vértices do grafo de interações Nk, vizinhos ao objeto a ser classificado. O método proposto se mostrou interessante para a consulta de objetos em bases de dados com aglomerados de formato arbitrário. Isso ocorre porque o método leva em consideração tanto a distância entre os objetos como a densidade espacial nos *N* pequenos grupos. Aqui, o método proposto em [MORAES & TINÓS, 2020] é adaptado para o KNN.

SEÇÃO C.
OBJETIVOS

Na Seção B, cita-se três observações relevantes: I) a existência de agrupamentos de dados em formatos diferentes de hiper-elipsóides, definidos não somente pela distância mas também pela densidade espacial; II) a simplicidade e eficiência do método KNN para o problema de classificação; III) e o processo de construção do grafo Nk ser intuitivo, de fácil alteração e o grafo ser fácil de se observar, ler e ser construído a partir da distância e densidade.

Dadas estas observações, pode-se então definir o objetivo principal do projeto: Investigar estratégias eficientes para a utilização do grafo de interações Nk para retornar os *K* objetos da base de treinamento que serão utilizados para a rotulação do novo objeto em KNN. Outros objetivos secundários são: Explorar as possibilidades oferecidas pela fusão de KNN e das propriedades do grafo de interações Nk; Modificar a maneira em que o grafo de interações Nk é construído; Comparar variantes do KNN modificado com o KNN original em bases de dados, com propriedades distintas, de repositórios públicos para diferentes valores de *K*; Comparar variantes do KNN modificado com o KNN original em um problema prático da área de Medicina para diferentes valores de *K*; e a capacitação do aluno nas áreas de Inteligência Artificial, Aprendizado de Máquina, Teoria dos Grafos e Classificação.

SEÇÃO D.
MATERIAIS E MÉTODOS

Essa seção está dividida no seguinte modo:

D.1. K-Vizinhos Mais Próximos (KNN)

D.2. Grafo de Interações Nk;

D.3. KNN baseado no Grafo de Interações Nk;

D.4. Bases de dados;

D.5. Implementação;

D.1. K-Vizinhos Mais Próximos (KNN)

O KNN é um algoritmo de classificação no qual um novo objeto *x* é rotulado examinando-se a classe majoritária entre os *K* objetos da base de treinamento mais próximos a x [AHA et al., 1991]. Existem algumas versões e modificações do algoritmo importantes de se mencionar: há uma versão para regressão; também é possível atribuir pesos aos *K* vizinhos analisados [ALTMAN, 1992]; e também existe uma abordagem de área, advinda da análise de uma hiper-esfera onde o parâmetro *K* é seu raio. Para ambas as abordagens, o parâmetro *K* possui um impacto significativo no resultado, desempenho e na definição das regiões de decisão do classificador. Estes *K* vizinhos são selecionados a partir da distância, geralmente Euclidiana, do objeto *x* a ser classificado.

D.2. Grafo de Interações Nk

O grafo de interações Nk é gerado para armazenar as informações dos grupos de objetos gerados pelo algoritmo NKGA [TINÓS et al., 2018]. Este algoritmo utiliza tanto a distância entre objetos como a densidade espacial para criar *N* pequenos grupos, cada um com *k* objetos, com o objetivo de agrupar exemplos de uma base de dados de tamanho *N*.

O grafo de interações Nk é um grafo direcionado com *N* vértices, cada um com grau de saída *k*. Cada aresta (v_j, v_i) do grafo indica que o *j*-ésimo objeto é relacionado com o *i*-ésimo objeto. Para a construção deste grafo, dada uma base de dados com *N* objetos, cria-se um vértice com auto-loop para cada objeto y_i da base de dados. Depois, para a construção dos *k-1* vértices restantes, é calculada a densidade e distância dos objetos. A próxima aresta adicionada para cada vértice v_i será o objeto mais próximo a y_i, que possua uma densidade maior que a densidade de y_i. Para os vértices restantes, serão conectadas arestas de v_i com os objetos mais próximos a y_i. As adições de arestas são realizadas até que o grau de saída dos vértices atinja *k*. A Figura D.2.1. exemplifica o processo de criação do grafo de interações Nk.

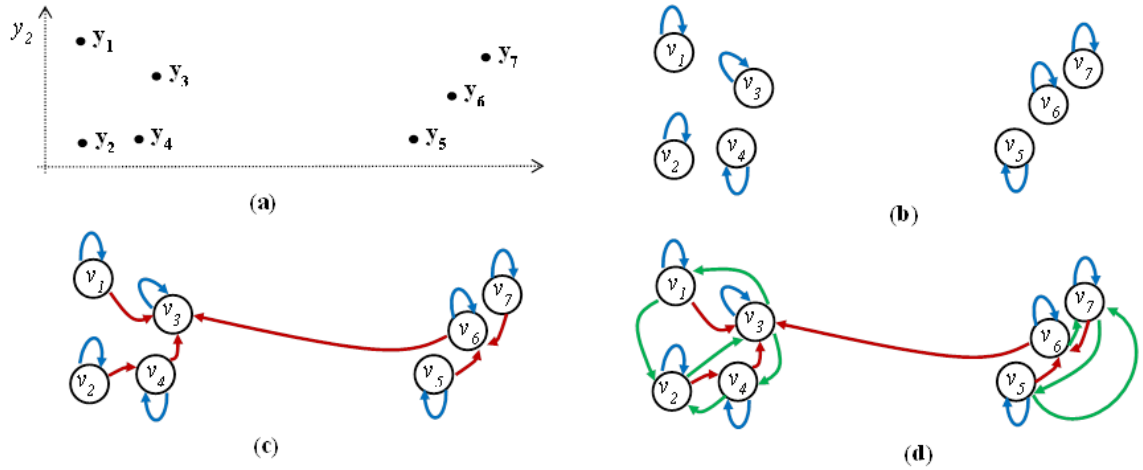

Figura D.2.1: Exemplo de construção do grafo de interações Nk com k = 3 para um conjunto com 7 objetos bidimensionais (N = 7, n = 2). Cada objeto da base de dados (a) é associado com um vértice com auto-loop (b). A densidade dos objetos é calculada e cada vértice é ligado ao vértice associado com o objeto mais próximo com maior densidade (c). O próximo passo é adicionar arestas para os vértices representando objetos mais próximos até que o grau de saída de cada vértice seja igual a k. O gráfico de interações (d) tem N = 7 vértices e Nk arestas.

D.3. KNN baseado no Grafo de Interações Nk

O KNN modificado utiliza do grafo de interações Nk para retornar os K vizinhos mais próximos a x no algoritmo KNN. O fato de o grafo ser montado a partir das métricas de distância e densidade faz com que os aglomerados em formatos diferentes de hiper-elipsoides sejam propriamente considerados e inseridos nas fronteiras de decisão de forma correta, ampliando os tipos de grupos e objetos que podem ser corretamente classificados pelo algoritmo KNN.

São utilizadas duas variações do método, que se diferem na forma em que os vértices associados aos *N* objetos da base de treinamento são visitados. Em ambas as variações, o grafo de interações é criado uma única vez para cada valor de *k*; os $K = k$ objetos relacionados aos vértices são retornados como os vizinhos mais próximos para rotular o novo objeto x, utilizando do KNN; e, similarmente ao que é feito no KNN original, um vértice v_x será selecionado como o representante de *x* no grafo de interações Nk (sendo v_x o objeto mais próximo de *x*, utilizando da distância Euclidiana como métrica).

O primeiro método, o KNN modificado tipo A, consiste na aquisição da lista de adjacências de v_x. Os objetos relacionados a v_x são tidos como os vizinhos de *x* e, portanto, utilizados para rotulá-lo. Já o segundo método, o KNN modificado tipo B, consiste em encontrar e guardar o vértice v_j, da lista de adjacências de v_x, cujo objeto y_j seja o mais próximo (distância Euclidiana) ao objeto *x*. Após esta análise, faz-se $v_x = v_j$ e repete esta análise *k* vezes. A lista de vértices v_j visitados são tidos como os vizinhos de *x* e, portanto, utilizados para rotulá-lo.

Posteriormente serão criadas variações nas maneiras em que o gráfico de interações Nk é construído, variando a quantidade de vértices conectados através da densidade espacial (atualmente, somente uma aresta é definida por densidade).

D.4. Bases de dados

Para o teste de eficácia do novo algoritmo desenvolvido, esta sendo utilizado um conjunto de onze bases de dados, com características distintas, no qual possa-se analisar as propriedades do KNN modificado em diversas situações de aglomeração e disposição dos dados. Algumas destas bases foram utilizadas no projeto de mestrado de José Carlos Bueno de Moraes no Programa de Computação Aplicada [MORAES, 2020], mestrado no qual este projeto é derivado. Todas as bases são advindas dos repositórios públicos UCI [DUA & GRAFF, 2019] e Shape [FRÄNTI & SIERANOJA, 2018].

As bases de dados foram retiradas de repositórios gratuitos online. A maioria delas é criada artificialmente, com o propósito de treinar e testar algoritmos de classificação. Todas elas são bidimensionais, com um número de atributos variado, com um máximo de 34 (base de dados *ionosphere*); e um máximo de 3100 objetos distintos (base de dados *D31*). A Figura D.2.1. mostra a base apelidada de *spiral*, uma das onze utilizadas neste projeto.

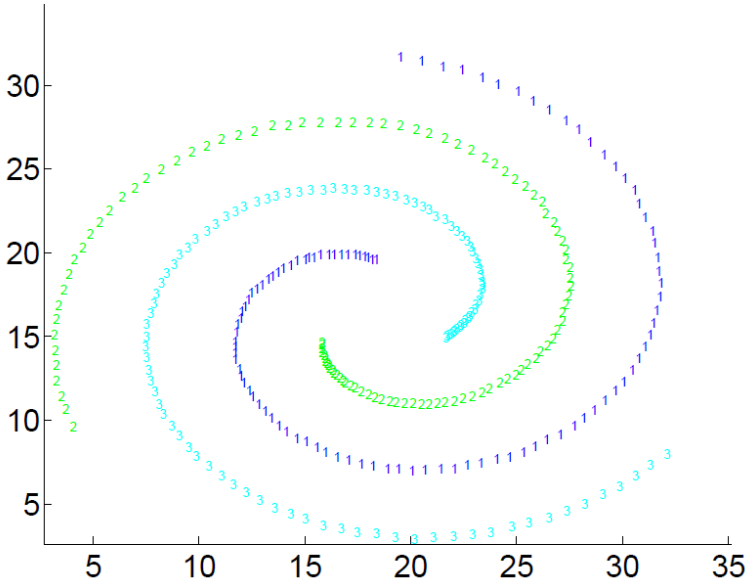

Figura D.4.1: Distribuição dos objetos da instância path-based 2 (spiral) do repositório Shape.

D.5. Implementação

Os programas para a implementação das duas variações do KNN modificado foram implementadas em Python. Diversas bibliotecas estão sendo utilizadas para auxiliar na manipulação dos dados, criação do grafo Nk, na implementação dos algoritmos, na classificação e nos testes.

Estre essas bibliotecas, podemos citar: *Pandas*, focado na manipulação de dados e criação de um tipo de estrutura de dado chamada *Data Frame*; *Scikit-learn*, uma biblioteca de aprendizado de máquina, com ferramentas para a aplicação de algoritmos, testes de eficácia, validação cruzada, entre outros; *SciPy* e *Numpy*, bibliotecas com diversos algoritmos e estruturas próprias para computação científica; e *NetworkX*, focada na criação, construção e manipulação de grafos.

Todas estas foram utilizadas como ferramentas para auxiliar na construção do algoritmo, desde a criação do grafo, até a criação do KNN e dos KNN modificados (do tipo A e B), até a aplicação dos métodos de verificação dos resultados, eficiência e eficácia.

SEÇÃO E.
RESULTADOS E DISCUSSÃO

As atividades realizadas estão divididas no seguinte modo:

E.1. Leitura e manipulação dos dados e bases de dados;

E.2. Criação do grafo de interações Nk;

E.3. Implementação dos algoritmos KNN e KNN modificado tipo A e tipo B;

E.4. Implementação da validação cruzada para os algoritmos de classificação;

E.5. Automação dos testes e impressão dos resultados;

De acordo com o cronograma proposto, houve levantamento bibliográfico, estudo das referências relacionadas ao tema (grafos, grafo de interações Nk, algoritmos de classificação, KNN, validação cruzada).

E.1. Leitura e manipulação dos dados e bases de dados

As bases de dados utilizadas nas análises foram lidas e armazenadas em estruturas de dados chamadas *Data Frames*, disponibilizadas pela biblioteca *Pandas*, utilizando um método desenvolvido em Python. Após a leitura e armazenamento, as manipulações realizadas no decorrer do código foram feitas utilizando dos *Data Frames* e das bibliotecas *Numpy* e *SciPy*, utilizando estruturas e algoritmos disponibilizados por estas.

E.2. Criação do grafo de interações Nk

Após a leitura, foi desenvolvido um método para a criação do grafo de interações Nk, que recebe o *Data Frame* com os dados; a distância dentre todos os objetos da base de dados; a densidade dos objetos; e o valor de *k* que representa a quantidade de conexões do grafo de in

terações. Para a aquisição da distância e da densidade, foram desenvolvidos rotinas e métodos próprios para tal, com auxílio das bibliotecas de computação científica. Para o armazenamento do grafo, e futuras manipulações nele, foi utilizada a biblioteca *NetworkX*.

A criação do grafo de interações segue a lógica explicada previamente em **D.2**. Um código em *C++*, utilizado para a criação do grafo em [TINÓS et al., 2018], também foi usado de base para a construção deste.

E.3. Implementação dos algoritmos KNN e KNN modificado tipo A e tipo B

Depois, foram implementados os algoritmos de classificação KNN (seção **D.1.**) e KNN modificado dos tipos A e B (seção **D.3.**). Ambos recebem um novo objeto, e a partir da distância, para o KNN, e do grafo, para o KNN modificado, o classificam conforme as classes dadas pela base de dados utilizada no treinamento do algoritmo.

Os métodos possuem parâmetros parecidos: o objeto *x* a ser classificado, o valor de *K* (ou *k*) para a quantidade de vizinhos ou conexões utilizadas na classificação, o *Data Frame* com os dados, e um *Data Frame* com a classificação dos objetos da base, além de dois parâmetros utilizados para ajustar a impressão dos dados. A diferença consiste no parâmetro da distância entre os objetos da base de dados analisada, para o KNN, e no parâmetro do grafo de interações Nk da base analisada, para os KNN modificados. Todos os métodos retornam a classificação de *x*.

E.4. Implementação da validação cruzada para os algoritmos de classificação;

Para confirmar a eficiência e eficácia dos algoritmos, é necessário aplicar um método de verificação. O método escolhido foi a validação cruzada, um método que consiste em: dividir a base de dados em *p* partes, pegar uma destas e utilizar de teste para as outras, utilizadas como treino para o algoritmo escolhido; realizar esta etapa *p* vezes, até todas as partes serem utilizadas de teste para o algoritmo.

Dois métodos foram criados para aplicar a validação cruzada nos algoritmos. Um para o KNN, e outro para os KNN modificados. Os parâmetros de ambos são parecidos: *K* (ou *k*) como o número de vizinhos ou conexões que serão utilizadas na classificação; *p* como o número de partes para a validação; e dois parâmetros utilizados para ajustar a impressão dos

dados. Porém o método para os KNN modificados possui um parâmetro utilizado para escolher qual tipo de algoritmo será utilizado: o A ou o B. Ambos os métodos retornam a matriz de confusão dos resultados obtidos.

A matriz de confusão é uma estrutura que permite a visualização do desempenho do algoritmo. Ela mostra a taxa de acertos e erros para cada uma das classes, além da proporção de cada tipo de erro.

E.5. Automação dos testes e impressão dos resultados;

Para realizar os testes, um script Python foi desenvolvido, que itera sobre todas as bases de dados, realiza a execução de todos os métodos de validação cruzada e armazena todos os resultados para um intervalo de valores para *K*. Este script utiliza dos métodos citados previamente, pertencentes a objetos do tipo NKGraph. Estes objetos representam uma base de dados, e possuem os *Data Frames* com os dados, as classes, e possuem as distâncias e densidades armazenadas.

A princípio, o intervalo de *K* utilizado foi de 1 a 10, para os algoritmos do KNN e dos KNN modificados do tipo A e B. Abaixo seguem os resultados obtidos:

Agregation							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	787	0.998731	787	0.998731	787	0.998731	788
2	783	0.993655	783	0.993655	786	0.997462	
3	786	0.997462	785	0.996193	786	0.997462	
4	784	0.994924	784	0.994924	784	0.994924	
5	786	0.997462	783	0.993655	786	0.997462	
6	786	0.997462	784	0.994924	786	0.997462	
7	786	0.997462	784	0.994924	786	0.997462	
8	787	0.998731	784	0.994924	787	0.998731	
9	786	0.997462	786	0.997462	786	0.997462	
10	786	0.997462	784	0.994924	786	0.997462	

Compound							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	389	0.974937	389	0.974937	389	0.974937	399
2	384	0.962406	384	0.962406	389	0.974937	
3	382	0.957393	381	0.954887	382	0.957393	
4	387	0.969925	381	0.954887	387	0.969925	
5	381	0.954887	373	0.934837	381	0.954887	
6	383	0.9599	372	0.932331	383	0.9599	
7	379	0.949875	372	0.932331	379	0.949875	
8	379	0.949875	372	0.932331	379	0.949875	
9	377	0.944862	369	0.924812	377	0.944862	
10	377	0.944862	369	0.924812	377	0.944862	

D31							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	2981	0.961613	2981	0.961613	2981	0.961613	3100
2	2965	0.956452	2965	0.956452	2991	0.964839	
3	2992	0.965161	2997	0.966774	2989	0.964194	
4	2991	0.964839	2998	0.967097	2989	0.964194	
5	3001	0.968065	3006	0.969677	3000	0.967742	
6	3004	0.969032	3001	0.968065	3004	0.969032	
7	3000	0.967742	2997	0.966774	3000	0.967742	
8	3000	0.967742	2997	0.966774	3000	0.967742	
9	2997	0.966774	3003	0.96871	2997	0.966774	
10	2999	0.967419	3007	0.97	2999	0.967419	

ecoli							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	274	0.815476	274	0.815476	274	0.815476	336
2	265	0.78869	265	0.78869	274	0.815476	
3	285	0.848214	288	0.857143	285	0.848214	
4	285	0.848214	287	0.854167	286	0.85119	
5	285	0.848214	283	0.842262	286	0.85119	
6	286	0.85119	282	0.839286	286	0.85119	
7	287	0.854167	286	0.85119	287	0.854167	
8	291	0.866071	285	0.848214	290	0.863095	
9	287	0.854167	287	0.854167	288	0.857143	
10	288	0.857143	278	0.827381	289	0.860119	

flame							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	240	1	240	1	240	1	240
2	237	0.9875	237	0.9875	238	0.991667	
3	239	0.995833	238	0.991667	239	0.995833	
4	238	0.991667	240	1	238	0.991667	
5	239	0.995833	240	1	239	0.995833	
6	239	0.995833	237	0.9875	239	0.995833	
7	238	0.991667	236	0.983333	238	0.991667	
8	238	0.991667	236	0.983333	238	0.991667	
9	238	0.991667	237	0.9875	238	0.991667	
10	238	0.991667	236	0.983333	238	0.991667	

ionosphere							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	306	0.871795	306	0.871795	306	0.871795	351
2	310	0.883191	310	0.883191	307	0.874644	
3	291	0.82906	295	0.840456	291	0.82906	
4	298	0.849003	300	0.854701	298	0.849003	
5	293	0.834758	296	0.843305	293	0.834758	
6	297	0.846154	296	0.843305	297	0.846154	
7	293	0.834758	298	0.849003	293	0.834758	
8	295	0.840456	292	0.831909	295	0.840456	
9	293	0.834758	290	0.826211	293	0.834758	
10	293	0.834758	291	0.82906	293	0.834758	

iris							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	144	0.96	144	0.96	144	0.96	150
2	139	0.926667	139	0.926667	144	0.96	
3	143	0.953333	144	0.96	144	0.96	
4	142	0.946667	142	0.946667	142	0.946667	
5	143	0.953333	143	0.953333	143	0.953333	
6	142	0.946667	142	0.946667	143	0.953333	
7	143	0.953333	144	0.96	143	0.953333	
8	140	0.933333	145	0.966667	141	0.94	
9	141	0.94	146	0.973333	141	0.94	
10	142	0.946667	143	0.953333	143	0.953333	

jain							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	373	1	373	1	373	1	373
2	370	0.991957	370	0.991957	373	1	
3	373	1	373	1	373	1	
4	373	1	373	1	373	1	
5	373	1	373	1	373	1	
6	373	1	373	1	373	1	
7	373	1	373	1	373	1	
8	373	1	373	1	373	1	
9	373	1	373	1	373	1	
10	373	1	373	1	373	1	

pathbased							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	300	1	300	1	300	1	300
2	299	0.996667	299	0.996667	300	1	
3	298	0.993333	298	0.993333	298	0.993333	
4	297	0.99	296	0.986667	297	0.99	
5	298	0.993333	297	0.99	298	0.993333	
6	297	0.99	297	0.99	297	0.99	
7	297	0.99	295	0.983333	297	0.99	
8	296	0.986667	297	0.99	296	0.986667	
9	297	0.99	296	0.986667	297	0.99	
10	296	0.986667	295	0.983333	296	0.986667	

R15							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	597	0.995	597	0.995	597	0.995	600
2	589	0.981667	589	0.981667	597	0.995	
3	598	0.996667	598	0.996667	598	0.996667	
4	597	0.995	596	0.993333	597	0.995	
5	598	0.996667	597	0.995	598	0.996667	
6	598	0.996667	596	0.993333	598	0.996667	
7	598	0.996667	596	0.993333	598	0.996667	
8	598	0.996667	596	0.993333	598	0.996667	
9	598	0.996667	596	0.993333	598	0.996667	
10	598	0.996667	596	0.993333	598	0.996667	

spiral							
K	A		B		KNN		Total
	Acertos	%	Acertos	%	Acertos	%	
1	312	1	312	1	312	1	312
2	312	1	312	1	312	1	
3	312	1	312	1	312	1	
4	312	1	312	1	312	1	
5	312	1	312	1	312	1	
6	312	1	311	0.996795	312	1	
7	312	1	311	0.996795	312	1	
8	311	0.996795	307	0.983974	311	0.996795	
9	311	0.996795	304	0.974359	311	0.996795	
10	308	0.987179	302	0.967949	308	0.987179	

SEÇÃO F.
CONCLUSÕES

Até o momento, mesmo com a aquisição de alguns dados, não é possível afirmar com certeza o desempenho dos algoritmos. Apesar disso, é possível pontuar alguns fatos observados: Ambas as bases não artificiais (*íris* e *ionosphere*) possuem seus melhores resultados adquiridos pelo algoritmo do KNN modificado do tipo B; Para valores de $K = k$ mais altos, há uma tendência à convergência de resultados dentre o algoritmo do KNN e do KNN modificado do tipo A. Isso se deve pelo fato de que a única diferença entre eles é o primeiro vizinho, sendo o tipo A definido por densidade; Bases de dados com aglomerados com formato hiper-elipsóides possuem uma tendência a serem melhor classificados pelo KNN e pelo KNN modificado do tipo A, com algumas exceções, provavelmente relacionadas a distância desses aglomerados (aglomerados muito próximos acabam por serem melhor classificados pelo tipo B, um exemplo é o *D31* para alguns valores de *K*).

O próximo passo será alterar a quantidade de vizinhos analisados pela densidade, visando verificar a importância e desempenho desta métrica na classificação. Após a aquisição desses novos resultados, será realizada uma análise mais minuciosa para se afirmar com mais certeza o desempenho e qualidade dos algoritmos.

SEÇÃO G.
CRONOGRAMA DA ETAPA FUTURA

No semestre que se inicia, haverá o desenvolvimento da alteração proposta para o algoritmo de criação do grafo de interação Nk, a partir da mudança na quantidade de vizinhos selecionados a partir da densidade. Após essa etapa, será realizada uma análise dos dados obtidos, e depois um novo conjunto de testes em problemas de classificação na área de Medicina.

A seguir, há o cronograma proposto para projeto:

	1º Bimestre		2º Bimestre		3º Bimestre		4º Bimestre		5º Bimestre		6º Bimestre	
Fase 1	Ok	Ok	Ok	Ok								
Fase 2		Ok	Ok	Ok								
Fase 3			Ok	Ok	Ok							
Fase 4				Ok	Ok	Ok						
Fase 5					Ok	Ok						
Fase 6												
Fase 7												
Fase 8												
Fase 9					Ok	Ok						

Figura G.1.: Cronograma.

Fase 1) Levantamento Bibliográfico: estudo das referências relacionadas aos temas da pesquisa.
Fase 2) Implementação do grafo de interações Nk e do KNN original na linguagem Python. Esta será a linguagem utilizada no desenvolvimento do projeto.
Fase 3) Desenvolvimento da primeira variação do KNN baseado no grafo de interações Nk (ver Seção 3.3) .
Fase 4) Desenvolvimento da segunda variação do KNN baseado no grafo de interações Nk (ver Seção 3.3)
Fase 5) Testes com os métodos desenvolvidos em instâncias das bases *UCI* e *Shape*.
Fase 6) Desenvolvimento e testes da variação do KNN baseado no grafo de interações Nk modificado.
Fase 7) Testes com os métodos desenvolvidos em um problema de classificação da área de Medicina. O problema ainda será definido
Fase 8) Análise dos resultados obtidos.
Fase 9) Confecção de Relatórios e artigos científicos.

SEÇÃO H.
REFERÊNCIAS

ALTMAN, N. S. (1992). “An introduction to kernel and nearest-neighbor nonparametric regression”, The American Statistician, 46(3): 175-185.

AHA, D. W.; KIBLER, D. & ALBERT, M.K. (1991). “Instance-based learning algorithms”, Machine Learning, 6(1): 37-66.

DUA, D. & GRAFF, C. (2019). “UCI Machine Learning Repository”, [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J. & XU, X. (1996), “A density-based algorithm for discovering clusters in large spatial databases with noise”, In the Proc. of the 2nd ACM Int. Conf. Knowl. Discovery Data Min. (KDD), 226–231.

FIX, E. (1985). “Discriminatory analysis: nonparametric discrimination, consistency properties”, Technical Report, USAF School of Aviation Medicine.

FRÄNTI, P. & SIERANOJA, S. (2018). “K-means properties on six clustering benchmark datasets”, Applied Intelligence, 48 (12): 4743-4759.

KOTSIANTIS, S. B.; ZAHARAKIS, I. & PINTELAS, P. (2007). “Supervised machine learning: A review of classification techniques”, Emerging artificial intelligence applications in computer engineering, 160(1): 3-24.

MORAES, J. C. B. (2020). “Busca por similaridade utilizando grafo de interações NK”, Dissertação de Mestrado em Computação Aplicada, Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo.

MORAES, J. C. B., & TINÓS, R. (2020). “Busca por Similaridade usando o Gráfico de Interação NK”, Nos Anais do XVII Encontro Nacional de Inteligência Artificial e Computacional, 222-233.

RODRIGUEZ, A. & LAIO, A. (2014). “Clustering by fast search and find of density peaks,” Science, 344(6191): 1492–1496.

TINÓS, R.; ZHAO, L.; CHICANO, F. & WHITLEY, D. (2018). “NK hybrid genetic algorithm for clustering”, IEEE Transactions on Evolutionary Computation, 22(5): 748-761.