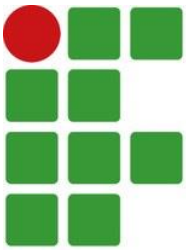

INSTITUTO FEDERAL
Ceará
Campus Fortaleza

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO - PPGCC
APRENDIZAGEM DE MÁQUINA
PROF. DR. AJALMAR REGO DA ROCHA NETO

ALUNO: DAVID WANDERSON DE ANDRADE NOGUEIRA

CLASSIFICAÇÃO COM KNN E DMC – IRIS E COLUNA VERTEBRAL 3C

FORTALEZA
2019.1

1. INTRODUÇÃO

Este relatório tem como objetivo mostrar os resultados obtidos na classificação da base de dados da Flor Iris e da Coluna 2C – 3C, utilizando os métodos KNN (*k-nearest neighbors)* e DMC (distância mínima ao centroide). A linguagem escolhida para a implementação do algoritmo foi o Python, por ser uma linguagem dinâmica e de alto nível, além de suas poderosas bibliotecas que fazem com que o Python seja hoje uma das principais linguagens de programação para machine learning.

2. BASE DE DADOS

2.1. IRIS

Para a primeira análise foi utilizada a base de dados da flor Iris, disponível em UC Irvine Machine Learning Repository[1]. A base de dados possui três classes de flores, são elas: Setosa, Versicolor e Virginica. Cada classe possui 4 parâmetros de atributos:

- Comprimento da sépala (sepal length)
- Largura da sépala (sepal width)
- Comprimento da pétala (petal length)
- Largura da pétala (petal width)

Cada classe possui um total de 50 amostras, totalizando 150 amostras.

2.2. COLUNA VERTEBRAL 3C

A segunda base de dados escolhida para análise, foi a coluna vertebral. disponível em UC Irvine Machine Learning Repository[2] Também é uma base de dados bastante conhecida, a base possui três classes: Hérnia, Espondilolistese e Normal. Cada classe possui seis atributos:

- Incidência Pélvica
- Inclinação Pélvica
- Ângulo de Lordose Lombar
- Inclinação Sacral
- Raio Pélvico
- Grau da Espondilolistese

Ao todo são 310 amostras, divididas para as três classes.

3. NORMALIZAÇÃO

Para melhor desempenho e análise de resultados, foi preciso normalizar os dados. O valor de Z (ou valor normalizado) é calculado para permitir que qualquer amostra dentro de um conjunto de dados tenha um valor definido entre o máximo e o mínimo de cada atributo. O cálculo consiste em encontrar a diferença do valor da amostra e do valor mínimo dos atributos e depois dividir o resultado pela diferença entre máximo e mínimo dos atributos, como mostra a formula a seguir:

[1] Base de dados retirado do site <https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data>, acesso em: 07/04/2019

[2] Base de dados retirado do site <http://archive.ics.uci.edu/ml/datasets/Vertebral+Column>, acesso em: 07/04/2019

$$Z = \frac{x - min}{\max - min}$$

Equação 1 - Normalização dos dados

onde: Z = Valor padronizado
x = Valor da amostra
max = Valor máximo de cada atributo
min = Valor mínimo de cada atributo

4. VISUALIZAÇÃO DOS ATRIBUTOS

Com base nos gráficos (figura 1) da base de dados da flor íris, é possível observar que uma das classes tem valores de atributos bem menores que as demais, é o caso da classe Setosa, ao aplicar o classificador observamos que é a classe com menor índice de erros, devido a sua diferença entre as outras duas classes. Os erros do classificador KNN e DMC se resumem em confundir Versicolor com Virgínica.

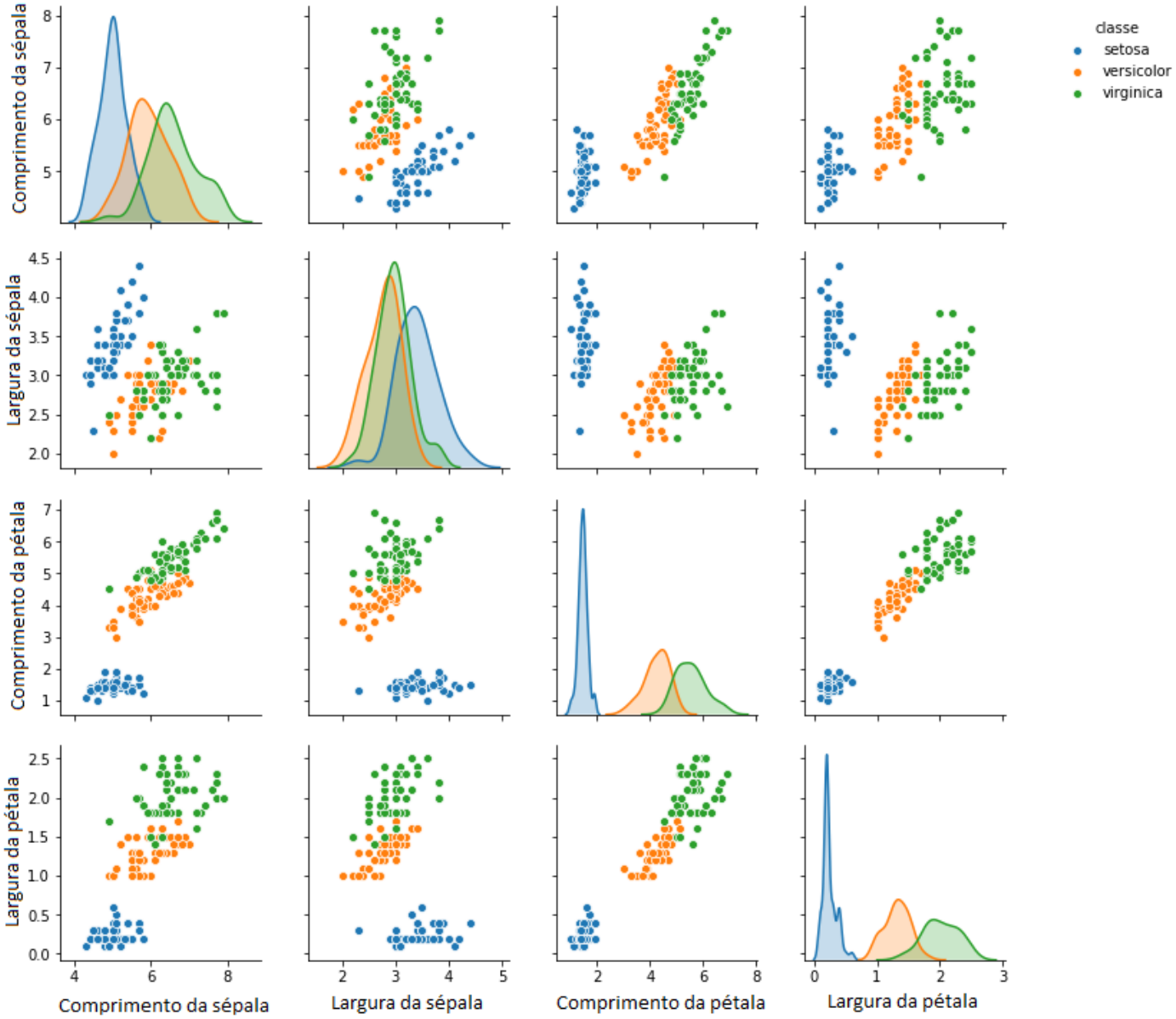

Figura 1 – Comparação dos atributos (Iris)

Conforme visualizado no gráfico da figura 2, os valores dos atributos da base de dados da coluna vertebral são mais próximos e tendem a confundir o algoritmo, diminuindo assim a acurácia do mesmo.

Figura 2 - Comparação dos atributos (Coluna)

Na figura 3, temos um gráfico entre os atributos Grau de Espondilolistese e Raio Pélvico para melhor visualização dos dados.

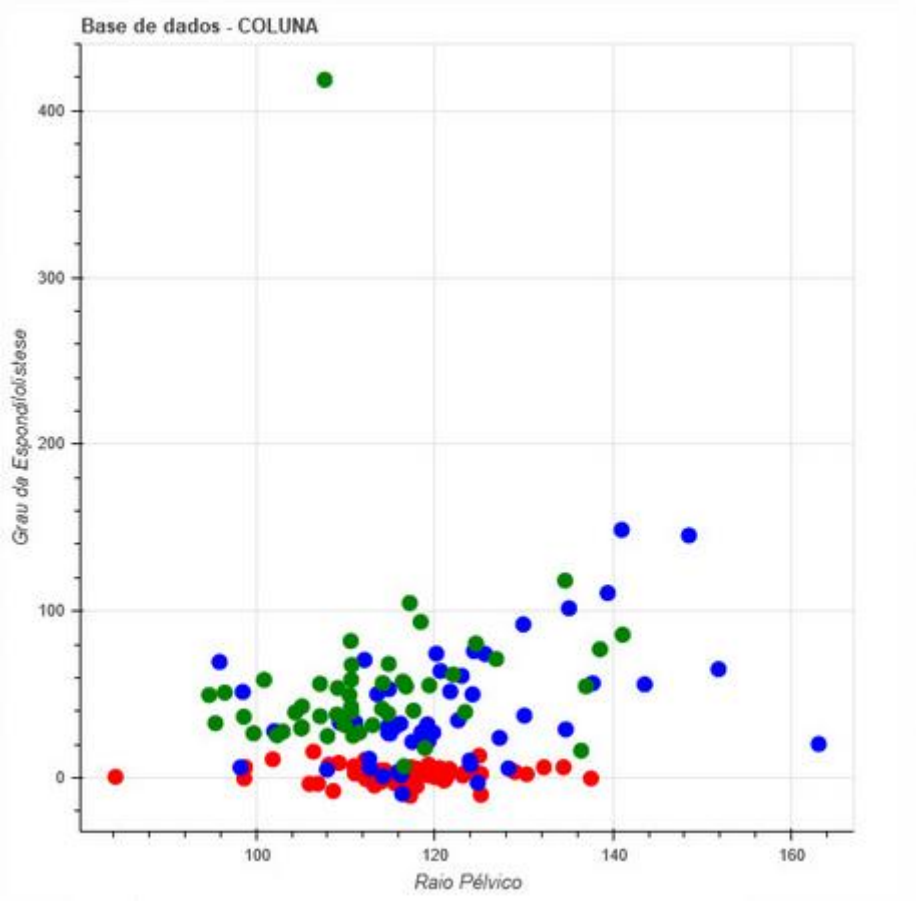

Figura 3 – Grau de Espondilolistese x Raio Pélvico

5. KNN - *k-nearest neighbors*

O algoritmo foi implementado para uma aplicação com holdout de 80% para treino e 20% para teste. Para otimizar o valor de K foi utilizado o método de validação K-fold, que retorna o melhor valor de K para cada modelo, dessa forma podemos escolher a melhor acurácia média. Uma variação para o KNN de 1 a 70 vizinhos mais próximos, para o número de 20 realizações, a seguir temos o demonstrativo dos resultados obtidos:

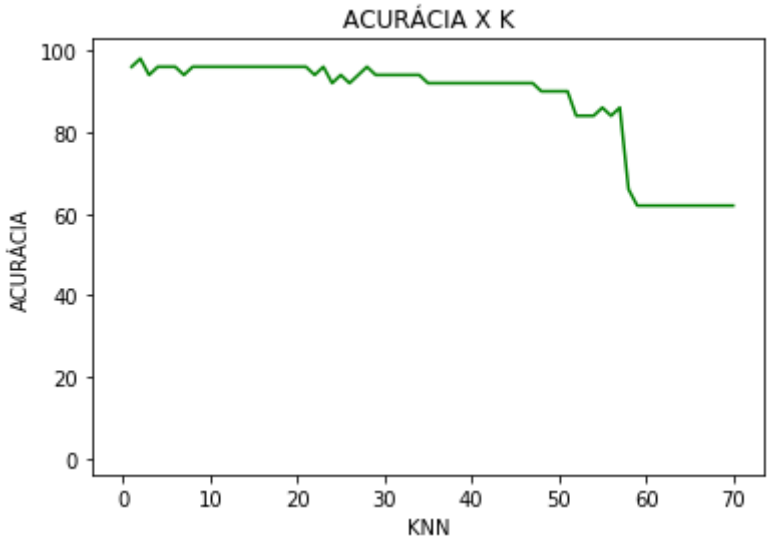

Figura 4 – Gráfico: Acurácia média x K (Iris)

A partir da visualização do gráfico da figura 4, que mostra a curva da acurácia em relação ao valor de K, podemos observar que a partir de um certo valor a curva da acurácia tem uma característica de descida. Até um K=20 o valor da acurácia tem um certo equilíbrio, após o K=30

temos uma baixa na acurácia, que se repete até K=50. A partir de um K=55 temos uma queda brusca na acurácia, o que prova que essa faixa tem um rendimento baixo. Após vários testes podemos dizer que apesar da curva mudar em cada teste, temos uma faixa para um K entre 8 e 20 com valores mais altos de acurácia e equilíbrio.

6. DMC

Para a aplicação do DMC a base de dados foi dividida em 80% para treino e 20% para teste do algoritmo. Para fins de análise dos resultados, os atributos do treino foram normalizados (0 à 1) e embaralhados para não viciar o treino, da mesma forma que o exemplo do KNN. Foi calculado o centroide de cada classe dentro do treino, o classificador classifica o teste com base na menor distância Euclidiana entre teste e os centroides. Comparado com o KNN, a acurácia média foi inferior.

Utilizando um holdout de 50 a 85% das amostras para o treinamento do algoritmo, cada teste considerando 20 realizações, temos os seguintes resultados demonstrados na tabela abaixo:

AMOSTRAS TREINO (%)	ACURÁCIA MÉDIA (%)
50	90.2
55	91.4
60	92.5
65	91.6
70	92.8
75	93.1
80	93.3

Tabela 1 – Treinamento e Acurácia média, em 20 realizações (íris)

De acordo com os dados observados na Tabela 1, temos uma alteração no valor da acurácia ao aumentar o número de amostras para treino, como os dados são embaralhados, a cada teste temos novos valores, mas que seguem praticamente o mesmo padrão.

AMOSTRAS TREINO (%)	ACURÁCIA MÉDIA (%)
50	70.5
55	71.2
60	71.4
65	71.5
70	71.1
75	74.4
80	74.2

Tabela 2 – Treinamento e Acurácia média, em 20 realizações (Coluna)

Observando a Tabela 2, temos uma alteração no valor da acurácia ao aumentar o número de amostras para treino, como os dados são embaralhados, a cada teste temos novos valores, mas que seguem praticamente o mesmo padrão.

7. MATRIZ DE CONFUSÃO

A matriz de confusão mostra de maneira ordenada e clara, o cruzamento dos dados entre o resultado obtido pelo classificador e a classificação real. É possível observar que o classificador teve melhor desempenho na Iris, a razão disto é que a base de dados da Iris tem um espaçamento maior entre os atributos, ou seja, são melhores para separar. Já os da coluna vertebral são mais próximos, o que pode confundir o classificador.

		PREVISTO		
		SETOSA	VERSICOLOR	VIRGINICA
CLASSE REAL	SETOSA	12	0	0
	VERSICOLOR	0	10	0
	VIRGINICA	0	1	7

Tabela 3 – Matriz de confusão KNN Iris, 20 realizações k=9.

		PREVISTO		
		SETOSA	VERSICOLOR	VIRGINICA
CLASSE REAL	SETOSA	9	0	0
	VERSICOLOR	0	10	3
	VIRGINICA	0	2	6

Tabela 4 – Matriz de confusão DMC Iris, 20 realizações.

		PREVISTO		
		HÉRNIA	ESPONDILOLISTESE	NORMAL
CLASSE REAL	HÉRNIA	15	4	3
	ESPONDILOLISTESE	0	13	1
	NORMAL	11	5	10

Tabela 5 – Matriz de confusão KNN Coluna 3c, 20 realizações k=9.

		PREVISTO		
		HÉRNIA	ESPONDILOLISTESE	NORMAL
CLASSE REAL	HÉRNIA	11	2	3
	ESPONDILOLISTESE	3	20	2
	NORMAL	5	5	11

Tabela 6 – Matriz de confusão DMC Coluna 3c, 20 realizações.

8. SUPERFICIE DE DECISÃO

O gráfico da superfície de decisão mostra os limites de cada classe, observamos a separação bem definida na classe Setosa. Variando um o K de 3 a 20, foi observado que o K=9 tem uma boa média de acurácia, com cerca de 96% e desvio padrão de 2,30.

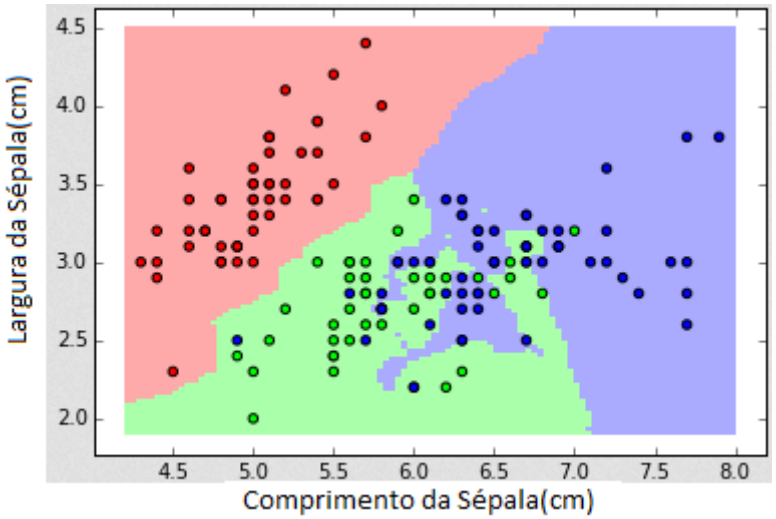

Figura 5 – Superfície de decisão KNN(Iris)

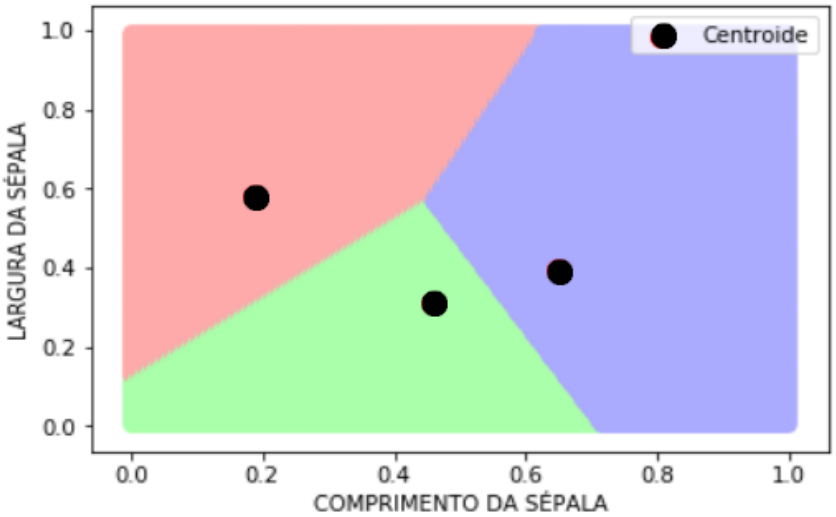

Figura 6 – Superfície de decisão DMC (Iris)

9. CONCLUSÃO

Podemos concluir que o classificador KNN teve um melhor rendimento para essas duas aplicações, com o valor do K otimizado através do K-fold é possível ter uma acurácia equilibrada e de valor considerado alto. Enquanto o DMC tem uma maior variação, pois a base de comparação são os centroides. No entanto a classificação com DMC é mais fácil de implementar e exige menos esforço computacional, dependendo da aplicação pode ser escolhida para classificação.