LISTA 1

Luiz Henrique Tavares Caúla
lhtc@cin.ufpe.br
Aprendizagem de Máquina
Professor George Darmiton da Cunha Cavalcanti
Centro de Informática
Universidade Federal de Pernambuco

1. PARTE GERAL

1.1 Código-fonte

Geral às quatro questões propostas na Lista 1, mostrou-se a implementação do algoritmo de aprendizagem k-Nearest Neighbours, variando apenas na forma em que a distância entre as amostras é calculada. Portanto, para melhor modularização, o formato de cálculo da distância é passado como entrada e pouco interfere no restante da implementação do algoritmo k-NN.

O código-fonte utilizado neste relatório aceita as seguinte entradas:

- ***dataset*** - formatos aceitos: .data e .arff. *Default*: datasets/cm1.arff
- **k** - quantidade de vizinhos avaliados para determinação de classe. *Default*: 2;
- **w** - se deve-se ser considerado peso no cálculo das distâncias. *Default*: false;
- **distance** - algoritmo de distância utilizado. *Default*: euclidean;
- **kfold** - a quantidade de divisões para o *k-fold cross-validation*. *Default:* 5;
- **shuffle** - se o dataset deve ser reorganizado aleatoriamente. *Default*: false.
- **swap** - se o atributo classe estiver na primeira posição. *Default*: false.

1.1.1 Instruções

- Baixe o código-fonte, encontrado no GitHub;
- Instale Python em sua máquina;
- Instale os módulos `argparse` (para passagem de entradas direto do terminal) e `arff` (para leitura de arquivos .arff);
- Vá para o diretório da Lista 1 com `cd "Lista 1"`.

1.1.2 Uso

Você pode rodar o programa de diversas formas.

Ex.: Caso queira rodar para o dataset `/datasets/tic-tac-toe.data`, com **peso nas distâncias**, com a distância **VDM**, com um **k = 5** e realizando **shuffle**, rode:

```
python knn.py -d /datasets/tic-tac-toe.data -w -distance vdm -k 5 -shuffle
```

Ex.: Caso queira rodar para os valores *default*:

```
python knn.py
```

Obs.: A ordem das entradas não influencia, contanto que o valor venha logo após a flag.

Ex.:

```
python knn.py -d /datasets/tic-tac-toe.data -w
```

e

```
python knn.py -w -d /datasets/tic-tac-toe.data
```

rodarão com *dataset* tic-tac-toe e com distâncias ponderadas.

Obs.: Tenha cuidado na escolha do par *dataset*/distância. Se seus dados possuírem dados categóricos, a distância euclidiana não irá funcionar como esperado. Assim como para dados numéricos, VDM não funcionará. Para dados mistos, utilize a distância **HVDM**.

1.2 Pré-processamento

Como uma das ideias da implementação é dar suporte para qualquer *dataset*, poucas medidas de pré-processamento são possíveis. No entanto, foi possível tomar algumas medidas antes da execução do algoritmo principal que podem ter seu tempo calculado separadamente.

1.2.1 *Shuffle*

Como a maioria dos *datasets* disponíveis nos *websites* permitidos estavam organizados por classe e desbalanceados, o uso do *k-fold cross-validation* seria prejudicado, uma vez que alguns *folds* possuiriam a maioria dos dados de apenas uma classe. A opção ***shuffle*** reduz esse problema, tornando aleatória a ordem dos dados.

Isso implica, naturalmente, que os resultados finais não serão iguais para todas as execuções com a *flag shuffle*. Portanto, para as conclusões dessas execuções, a **média aritmética** será utilizada.

1.2.2 Cálculo de probabilidades, máximo e mínimo

A implementação da distância VDM requer o conhecimento das probabilidades de cada atributo para cada classe. Esses dados podem ser calculados previamente para algoritmos especializados, porém, no nosso caso, ele faz parte da execução pré-algoritmo principal.

De forma semelhante, na distância HVDM para atributos numéricos, o uso dos máximos e mínimos valores para cada atributo é necessário. Esses dados são calculados juntamente com as probabilidades.

2. QUESTÕES

2.1 Questão 1: Distância euclidiana com e sem peso

2.1.1 Escolha dos *datasets*

Para esta questão, foram escolhidos os *datasets* CM1 (*k-fold 6)* e KC1 (*k-fold 4)* - ambos disponibilizados pela NASA para medidas de detecção de defeito em *softwares*. Ambos os *datasets* possuem **22** atributos que descrevem qualidades sobre os códigos dos *softwares* avaliados, tendo como classe **positiva** representativa de *softwares* que reportaram uma ou mais falhas e **negativa** como os que não reportaram nenhum erro.

Os k-folds escolhidos foram 6 e 4, respectivamente, para que cada divisão possa possuir a mesma quantidade de elementos para cada divisão.

	Positivos	Negativos	Total
CM1	49 (9,83%)	449 (90,16%)	498
KC1	326 (15,45%)	1783 (84,54%)	2108

Como pode-se observar, ambos *datasets* possuem dados desbalanceados para a classe **negativa**. Assim, espera-se que, para ambos conjunto de dados, tenhamos resultados semelhantes.

Vale a observação sobre o tamanho dos conjuntos, sendo o KC1 aproximadamente quatro vezes maior que o CM1. Esperamos que o tempo de execução mantenha a mesma proporção.

2.1.2 Resultados

Gráfico 2.1

CM1 (link para melhor visualização)

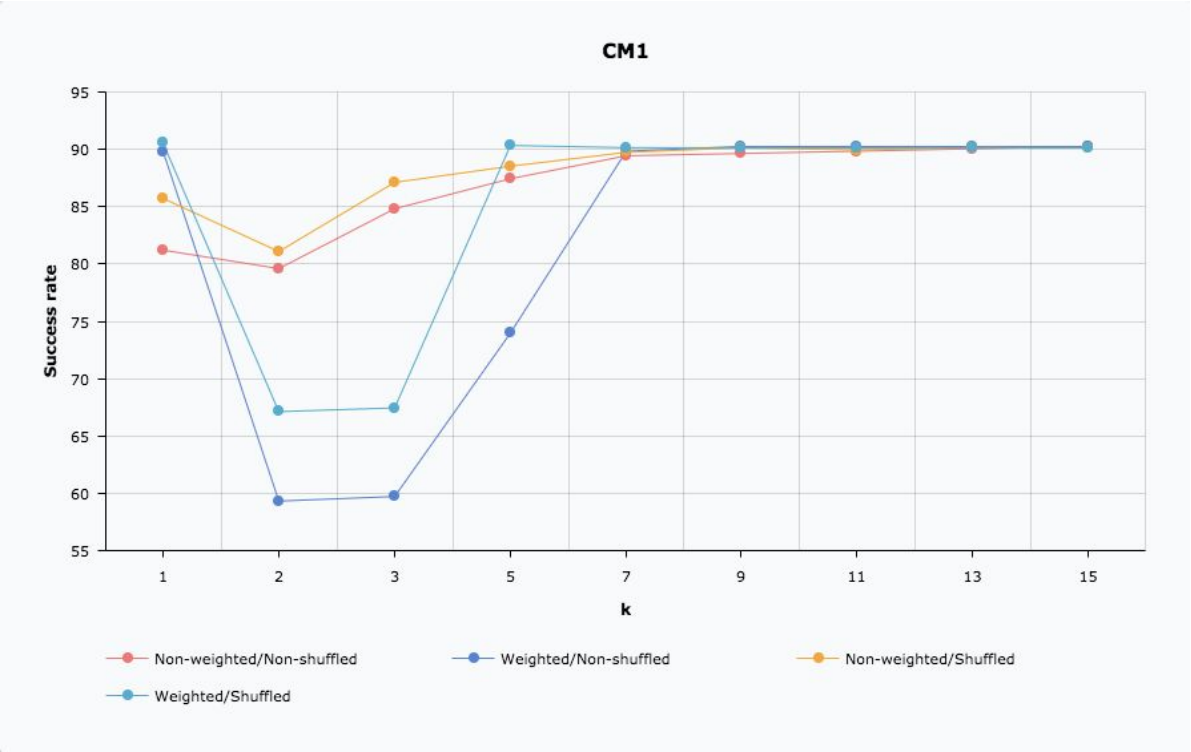
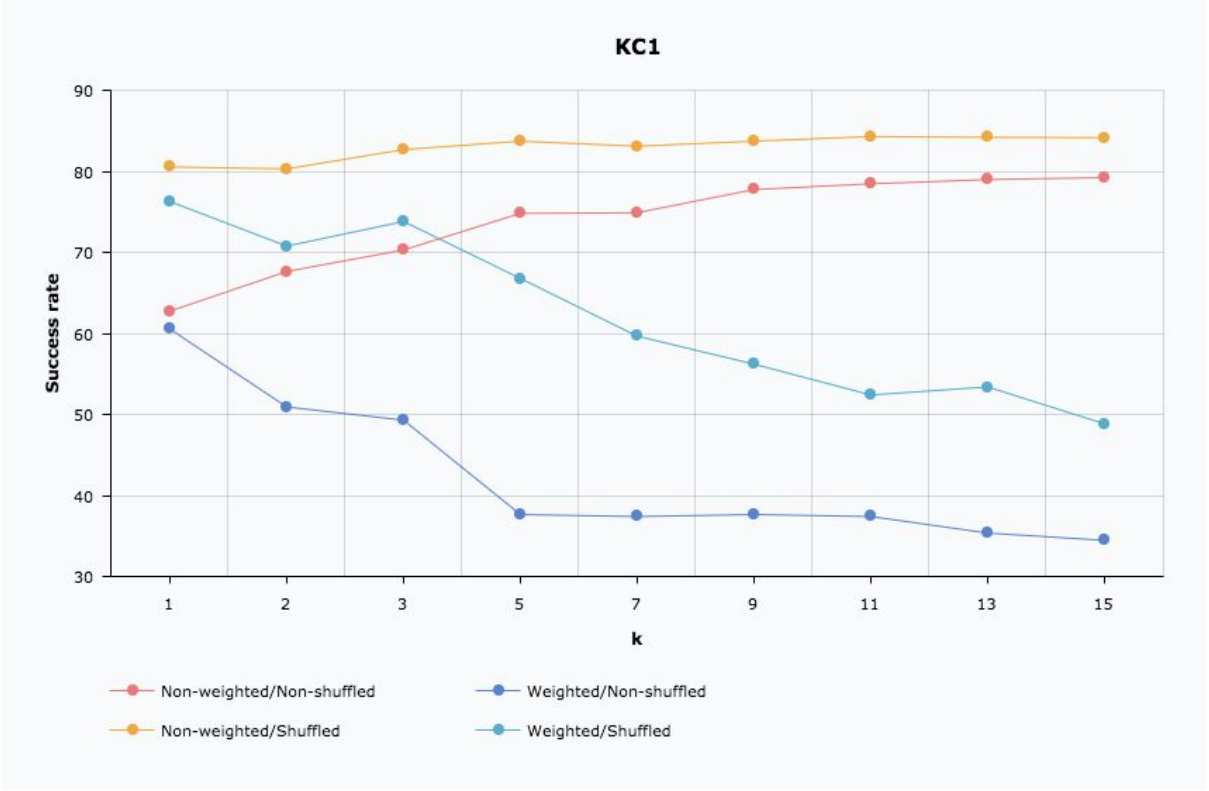


Gráfico 2.2

KC1 (link para melhor visualização)



2.2 Distância VDM

2.2.1 Escolha dos *datasets*

Para a segunda questão, foram escolhidos os conjuntos de dados:

- Balloons: Proveniente de uma pesquisa de Psicologia sobre se balões estarão cheios ou vazios dependendo de suas características, sendo **positivo** se cheio e **negativo** de vazio (*k-fold* 4);
- Tic-Tac-Toe: Classificador binário sobre as possíveis posições no "Jogo da Velha" e sobre seu vencedor, sendo **positivo** uma vitória para o "X" e **negativo** um derrota para o "X" (*k-fold* 4).

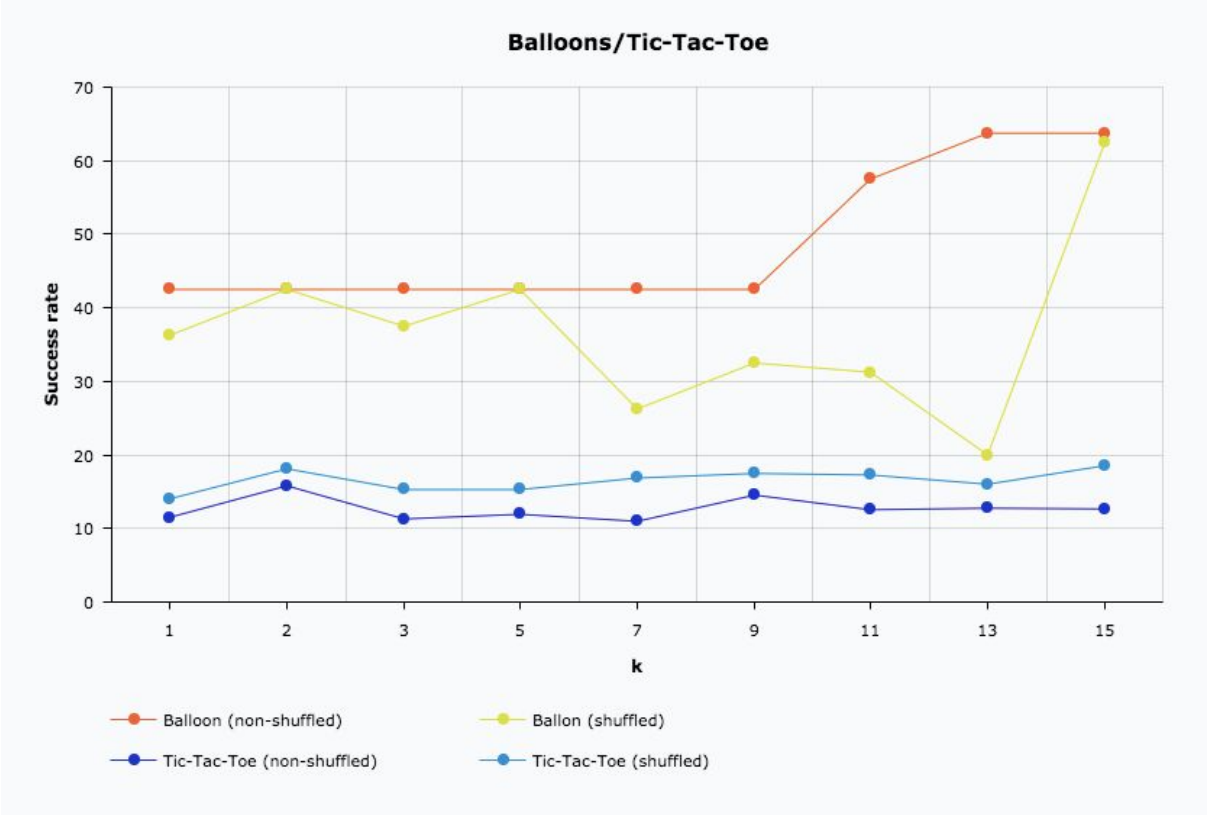
	Qtd. Atributos	Negativos	Positivos	Total
Balloons	4	25%	75%	16
Tic-Tac-Toe	9	34,7%	65,3%	956

O conjunto dos balões é notável por possuir poucos dados, portanto, valores altos para K devem prejudicar os resultados. Já o Tic-Tac-Toe possui atributos binários ("x" e "o"), e vejamos como essa característica se comporta dentro do VDM.

2.2.2 Resultados

Gráfico 2.3

Balloons/Tic-Tac-Toe (link para melhor visualização)



2.3 Distância HVDM

2.3.1 Escolha dos *datasets*

Para os *datasets* mistos, foram escolhidos os seguintes conjuntos:

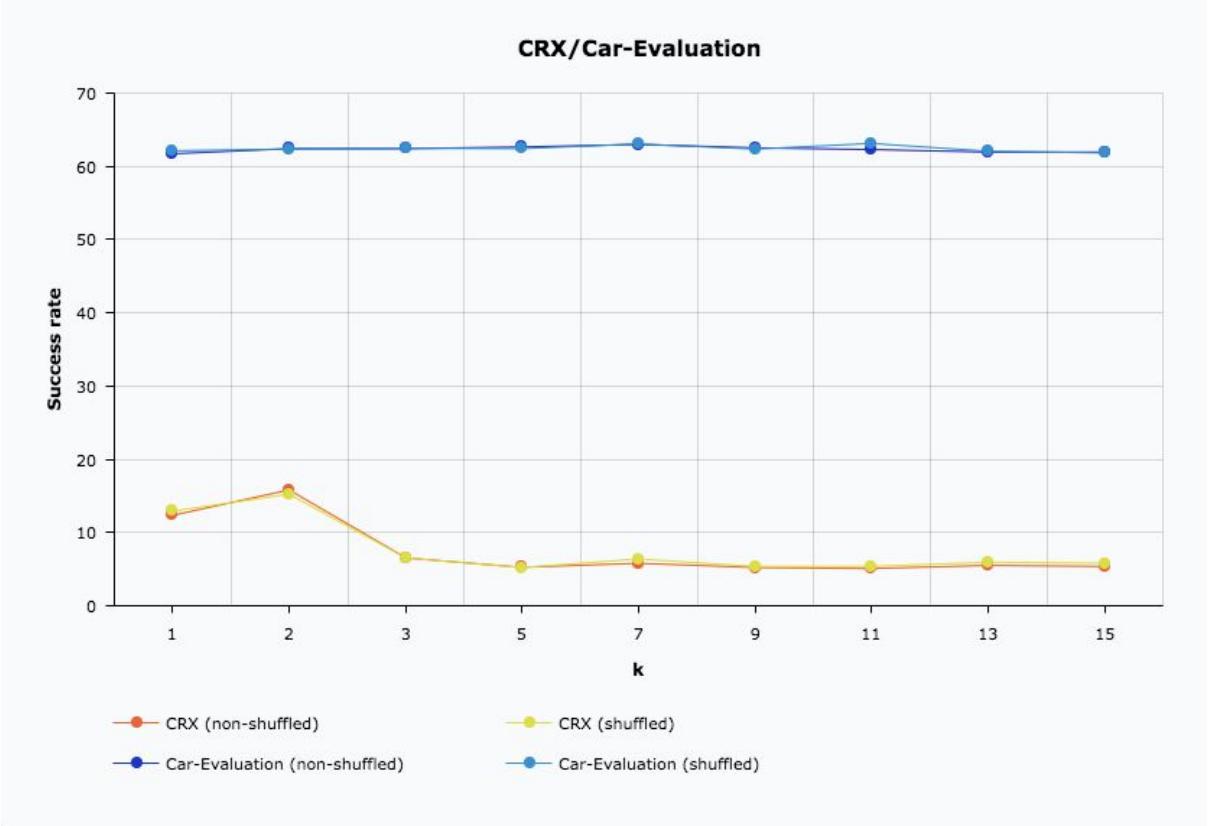
- CRX: Um banco de dados para aprovação de crédito (*k-fold* 5);
- Car Evaluation: Um *dataset* para determinação do valor de compra de um carro. As classes são **não-aceitável**, **aceitável**, **bom** e **muito bom** (*k-fold* 6).

	Qtd. Atributos	Numérico vs. Categórico	Total
CRX	15	40/60	690
Car Evaluation	6	33/66	1728

2.3.2 Resultados

Gráfico 2.4

CRX/Car-Evaluation (link para melhor visualização)



2.4 Conclusões

De forma geral, podemos observar que os *datasets* avaliados por **distância euclidiana** retornaram taxas de acertos melhores do que o VDM e HVDM.

2.4.1 Aplicação do *shuffling*

Quando *shuffling* foi aplicado como pré-processamento, o desempenho pareceu melhorar, com exceção do *dataset* Balloons - onde possuiu piores taxas - e dos avaliados por HVDM - onde possuiu taxas similares ao que não sofreram *shuffling*.

Além disso, ao observarmos as tabelas do capítulo 3, é possível notar que o **desvio padrão** entre as taxas de cada *k-fold* foi **significativamente melhor**, tornando os resultados mais confiáveis.

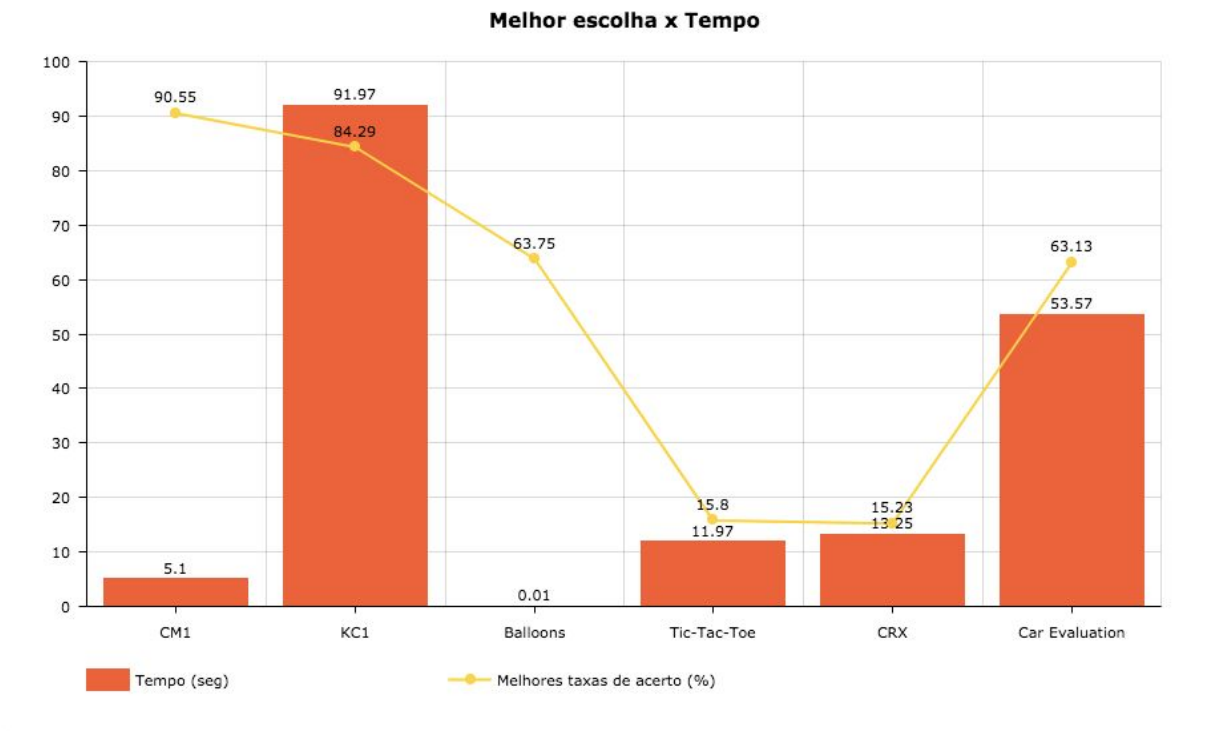
2.4.2 Desempenho x Tempo

Agora, vamos analisar a melhor escolha de cada *database* sobre o tempo de execução para poder traçar paralelos e tirar conclusões.

Obs.: O tempo de treinamento, se considerado como o de pré-processamento, foi irrelevante em todos os casos, onde não conseguiu atingir a marca dos 0,01 segundos.

Gráfico 2.5

Melhores escolhas x Tempo (link para melhor visualização)



2.4.3 CM1 | K = 1 | Com peso | Com *shuffle*

Dentre todos os *datasets* avaliados, CM1 foi o que obteve a melhor taxa de acerto/tempo de execução com uma boa folga. O conjunto de dados possui 498 elementos com um bom tempo.

Analisando o gráfico das taxas do CM1, podemos observar uma queda considerável de desempenho com escolhas para o *k* entre 2 e 7 e estabilizam-se com valores próximos de 90% para valores maiores. Isso pode significar que as classes estão espalhadas em distância. Ademais, as altas taxas para k = 1 mostram que, para este *dataset*, elementos possuem pelo menos uma vizinha próxima de mesma classe.

2.4.4 KC1 | K = 11 | Sem peso | Com *shuffle*

Para o KC1, a escolha do K para execuções sem consideração de distâncias ponderadas e utilizando *shuffle* pouco importava, pois as taxas de acerto se mantiveram entre 80% e 84%.

É possível observar um padrão entre as execuções com e sem distâncias ponderadas. Para as com peso, as taxas parecem decrescer com o aumento do valor de *k*, e o inverso acontece para as não ponderadas.

Vale notar que o tempo de execução mostrou-se bastante alto para este *dataset*, ultrapassando a casa dos 1 minuto e 30 segundos. Supõe-se, naturalmente, que o tamanho do conjunto de dados influenciou no tempo exagerado, mas a escolha da linguagem - Python - pode também ter prejudicado.

2.4.5 Balloons | K = 13 | Sem *shuffle*

Balloons foi um *dataset* interessante - possui dados bastante escassos e atributos que não parecem influenciar na escolha das classes, que, por si só, já parecem imprevisíveis.

O tamanho do *dataset* pareceu influenciar bastante nas taxas, pois, para valores de *k* entre 1 e 9, os mesmos resultados foram encontrados. Após esses valores, a taxa sobe - porém, o valor de *k* fica maior que a quantidade de elementos do *k-fold* (8). O código-fonte tratava desses casos, pedindo para considerar apenas os vizinhos que estivessem dentro da divisão, mas o resultado pareceu ser influenciado.

Como esperado, o tempo de execução foi muito próximo de nulo, devido a pouca quantidade de elementos. Porém, isso não teve grande influência no desempenho da taxa de acertos.

2.4.6 Tic-Tac-Toe | K = 15 | Com *shuffle*

Os atributos binários do Tic-Tac-Toe não tiveram boa aceitação no algoritmo de distância VDM, fazendo com que este *dataset* possuísse o segundo pior desempenho entre os conjuntos avaliados.

2.4.7 CRX | K = 2 | Sem *shuffle*

Assim como o outro *dataset* avaliado pela distância HVDM, CRX não apresentou mudanças significativas pelo uso do *shuffle*. Porém, apresentou os piores resultados.

2.4.8 Car Evaluation | K = 13 | Com *shuffle*

A escolha do valor de *k* pouco pareceu influenciar nos resultados do Car Evaluation, mantendo-se entre 62% e 63%.

Vale salientar que os valores numéricos deste *dataset* (quantidade de portas e capacidade de pessoas) poderiam ser considerados categóricos, como sugere a

classificação do UCI. Porém, o uso da distância euclidiana pareceu influenciar positivamente nas taxas de acerto.

3. TABELAS

Tabela 3.1

CM1 | Distância euclidiana | sem peso | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	81,21	79,60	84,82	87,44	89,44	89,64	89,84	90,05	90,25
Std. Dvt.	17,45	19,92	19,12	20,14	20,98	21,07	21,13	21,19	21,27
Tempo (s)	5,31	5,01	5,02	5,02	5,39	5,15	5,01	4,95	5,06

Tabela 3.2 - CM1

CM1 | Distância euclidiana | com peso | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	89,84	59,32	59,72	73,98	89,84	90,25	90,25	90,25	90,25
Std. Dvt.	21,09	42,48	41,98	37,67	21,09	21,27	21,27	21,27	21,27
Tempo (s)	5,07	5,15	5,17	5,41	5,34	5,11	5,18	5,34	5,46

Tabela 3.3

CM1 | Distância euclidiana | sem peso | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	85,72	81,09	87,12	88,53	89,74	90,14	89,95	90,15	90,14
Std. Dvt.	3,39	2,15	2,94	1,66	1,79	2,91	3,77	3,49	2,81
Tempo (s)	5,00	5,09	5,04	5,08	5,69	5,19	5,20	5,08	5,05

Tabela 3.4

CM1 | Distância euclidiana | com peso | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	90,55	67,12	67,45	90,34	90,15	90,14	90,13	90,15	90,14
Std. Dvt.	2,89	29,61	30,3	3,10	3,05	2,91	3,10	1,73	2,46
Tempo (s)	5,10	5,23	5,19	5,16	5,18	5,19	5,05	5,17	5,10

Tabela 3.5

KC1 | Distância euclidiana | sem peso | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	62,75	67,64	70,30	74,81	74,90	77,75	78,46	78,98	79,22
Std. Dvt.	14,55	17,47	18,89	21,36	21,42	23,14	23,39	23,79	23,91
Tempo (s)	84,87	84,40	83,24	84,52	84,02	82,97	88,17	89,87	86,01

Tabela 3.6

KC1 | Distância euclidiana | com peso | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	60,53	50,92	49,29	37,64	37,40	37,64	37,40	35,36	34,46
Std. Dvt.	28,05	22,15	25,79	5,38	4,56	4,60	4,46	6,36	5,67
Tempo (s)	89,19	87,93	95,17	96,98	85,32	84,75	90,37	90,61	82,23

Tabela 3.7

KC1 | Distância euclidiana | sem peso | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	80,54	80,30	82,68	83,72	83,10	83,72	84,29	84,20	84,15
Std. Dvt.	1,65	1,66	1,57	1,67	0,36	1,22	1,97	0,83	1,04
Tempo (s)	89,48	89,71	83,74	84,55	95,16	95,99	91,97	91,84	84,21

Tabela 3.8

KC1 | Distância euclidiana | com peso | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	76,27	70,77	73,81	66,78	59,66	56,20	52,40	53,39	48,84
Std. Dvt.	10,04	11,13	12,24	12,05	3,57	5,66	2,47	4,48	2,22
Tempo (s)	91,70	92,25	88,90	89,31	96,09	96,28	92,01	93,67	87,33

Tabela 3.9

Balloons | VDM | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	42,50	42,50	42,50	42,50	42,50	42,50	57,50	63,75	63,75
Std. Dvt.	4,33	4,33	4,33	4,33	4,33	4,33	4,33	6,49	6,49
Tempo (s)	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01

Tabela 3.10

Balloons | VDM | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	36,25	42,50	37,50	42,50	26,25	32,50	31,25	20,00	62,50
Std. Dvt.	6,49	14,79	17,85	14,79	21,61	19,20	21,90	24,49	17,85
Tempo (s)	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01

Tabela 3.11

Tic-Tac-Toe | VDM | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	11,51	15,80	11,30	11,93	10,98	14,54	12,55	12,76	12,66
Std. Dvt.	6,20	8,83	5,80	6,03	6,43	8,47	7,55	7,45	7,40
Tempo (s)	12,51	11,97	11,75	12,68	11,59	11,76	12,12	12,04	11,88

Tabela 3.12

Tic-Tac-Toe | VDM | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	14,03	18,11	15,29	15,29	16,86	17,49	17,28	16,02	18,54
Std. Dvt.	1,62	1,81	0,61	3,26	1,93	2,50	1,17	2,78	1,71
Tempo (s)	11,36	11,87	11,94	12,18	11,98	11,90	12,24	11,86	12,43

Tabela 3.13

CRX | HVDM | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	12,33	15,82	6,53	5,25	5,80	5,22	5,08	5,51	5,37
Std. Dvt.	6,15	8,87	3,78	2,53	3,10	2,80	2,67	3,32	3,12
Tempo (s)	14,32	13,49	16,97	12,91	12,94	12,93	13,86	13,24	13,33

Tabela 3.14

CRX | HVDM | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	12,92	15,23	6,53	5,22	6,38	5,37	5,37	5,95	5,81
Std. Dvt.	2,69	4,35	2,28	1,85	1,47	1,48	1,86	1,85	1,78
Tempo (s)	13,27	13,25	13,36	13,34	12,57	12,86	12,73	13,09	12,78

Tabela 3.15

Car Evaluation | HVDM | sem *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	61,72	62,42	62.42	62,65	62,99	62.53	62,30	61,95	61,95
Std. Dvt.	10,03	10,63	9,63	10,52	10,73	11,00	10,54	10,48	10,48
Tempo (s)	51,18	55,26	56,26	56,15	55,21	56,14	55,60	56,87	55,34

Tabela 3.16

Car Evaluation | HVDM | com *shuffle*

K	1	2	3	5	7	9	11	13	15
Acerto (%)	62,13	62,36	62.48	62,42	63,06	62,36	63,13	62,07	61,84
Std. Dvt.	2,05	2,50	2,53	2,88	2,55	2,24	1,58	2,08	2,96
Tempo (s)	54,95	55,41	56,11	48,92	56,82	57,78	53,57	57,04	56,10