Ficha 4 - DC

João Nunes (A82300)
Luís Braga (A82088)

6/03/2020

Conteúdo

1 Descrição do Problema

O datasetusado neste exercício é o datasetde doenças cardíacas disponível noficheiroheart-c.arff, obtido no repositório da UCI. Este datasetdescreve fatores de risco para doenças cardíacas. O atributo numrepresenta o atributo da classe (binária): class <50 significa nenhuma doença; class> 50_1 indica aumento do nível de doença cardíaca. O principal objetivo deste exercício é prever doenças cardíacas a partir de outros atributos no dataset. Obviamente, trata-se de umproblema de classificação. O softwarea ser usado é o Weka. No entanto, sinta-se à vontade para tentar qualquer ideiaque possa ter para solucionar o problema com qualquer outro software. A descrição deste exercício é gradual. Portanto, espera-se que possa entender melhor os vários aspetos e questões envolvidos no processo de KDD.

2 Data Understanding

2.1 Para cada atributo, encontre as seguintes informações.

2.1.1 a) O tipo de atributo, p.e.nominal, ordinal, numérico.

Cada um dos atributos do dataset possui o seguinte tipo:

- Age – Numérico;
- Sex – Nominal;
- Cp – Nominal;
- Trestbps – Numérico;
- Chol – Numérico;
- Fbs – Nominal;
- Restecg – Nominal;
- Thalach – Numérico;
- Exang – Nominal;
- Oldpeak – Numérico;
- Slope – Nominal;
- Ca – Numérico;
- Thal – Nominal;
- Num – Nominal;

2.1.2 b) Percentagem de valores ausentes nos dados.

Cada um dos atributos anteriores possuirá a seguinte percentagem de valores ausentes.

- Age – 0%
- Sex – 0%
- Cp – 0%
- Trestbps – 0%

- Chol – 0
- Fbs – 0%
- Restecg – 0%
- Thalach – 0%
- Exang – 0%
- Oldpeak – 0%
- Slope – 0%
- Ca – 2%
- Thal – 1%
- Num – 0%

2.1.3 c) Máx, min, média e desvio padrão.

	Máximo	Mínimo	Média	Desvio padrão
age	77	29	54,66	9,082
sex	-	-	-	-
cp	-	-	-	-
trestbps	200	94	131,62	17,54
chol	564	126	246,26	51,83
fbs	-	-	-	-
restecg	-	-	-	-
thalach	202	71	149,65	22,91
exang	-	-	-	-
oldpeak	6,2	0	1,04	1,16
slope	-	-	-	-
ca	3	0	0,67	0,94
thal	-	-	-	-
num	-	-	-	-

2.1.4 d) Existem registos que tenham um valor para um atributo que nenhum outro registo tem?

Existem os seguintes atributos que possuem valores únicos: age, trestbps, chol, thalach, oldpeak.

2.1.5 e) Estude o histograma no canto inferior direito e descreva informalmente como o atributo parece influenciar o risco de doença cardíaca. O que significam as mensagens pop-up que aparecem ao arrastar o rato sobre o gráfico?

- Age – Ao aumentar a idade, maior é o atributo num pelo que indica que maior é o risco de doença cardíaca;
- Sex – Existe maior probabilidade de um macho ter doença quando comparado com o sexo feminino;
- Cp – Com uma dor no peito *asympt* há maior probabilidade de ter doença;
- Trestbps – Pelo histograma parece que no intervalo de *resting blood pressure* [111,7;147] existe uma maior probabilidade de ter doença cardíaca
- Chol – Pelo histograma parece que no intervalo de *serum cholestoral in mg/dl* [180,75;290,25] existe uma maior probabilidade de ter doença cardíaca

- Fbs – Relativamente ao atributo *fasting blood sugar* a proporção de ter doença é semelhante para o *true* e *false*
- Restecg – No atributo *resting electrocardiographic results* o *left vent hyper* e *normal* são os atributos mais prováveis de influenciar o risco de doença cardíaca
- Thalach – O *maximum heart rate achieved* parece ter maior probabilidade para valores menores
- Exang – O *exercise induced angina* aparenta ter mais probabilidade no *yes*
- Oldpeak – O *ST depression induced by exercise relative to rest* aparenta que quanto maiores os valores maior a probabilidade
- Slope – O *the slope of the peak exercise ST segment* tem mais probabilidade no *flat*
- Ca – O *number of major vessels (0-3) colored by flourosopy* tem mais probabilidade nos intervalos [0,5;1] e [1,5;2]
- Thal – O *thal* tem mais probabilidade no *reversable defect*

2.2 Mude para o separador Visualize,na parte superior da janela, para visualizar gráficos de dispersão 2D para cada par de atributos.

2.2.1 a) Que atributos parecem estar mais/menos associados a doenças cardíacas? Resuma numa tabela as suas descobertas sobre o valor preditivo de cada atributo.

Através da visualização de cada um dos gráficos no *Visualizer* foi possível estimar os seguintes intervalos de valor para cada um dos atributos onde se estima em que *range* existe a maior probabilidade de estar associado a uma doença cardíaca.

	Maior Probabilidade	Menor Probabilidade
age	[53;60]	[29;40]
sex	male	female
cp	asympt	atyp_angina
trestbps	[120:143]	[147;200]
chol	[200;260]	[126;200]
fbs	f	t
restecg	left_vent_hyper	normal
thalach	[136;180]	[180;202]
exang	yes	no
oldpeak	[2;3,1]	[0;1]
slope	flat	up
ca	[1;2]	[0;1]
thal	reversable_defect	normal

A partir deste momento foi utilizado o dataset data.arff

2.2.2 b) Algum par de atributos parece estar correlacionado?

Os atributos col e chol são correlacionados na medida em que quando um atributo aumenta o outro também aumenta.

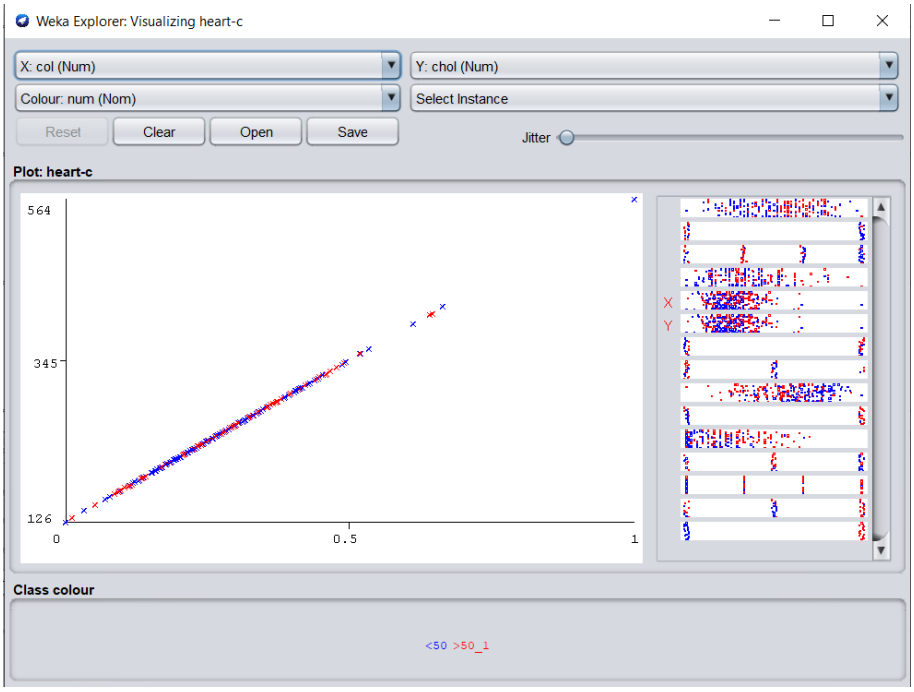

Figura 1: Correlação entre atributos col e chol.

2.3 Investigue também possíveis associações multi variadas de atributos com o atributo class, ou seja, estude gráficos de dispersão de dois atributos X e Y e tente identificar possíveis áreas "densas" de doenças cardíacas (se existirem).

No seguinte gráfico onde se relacionam os atributos thalach e col é possível identificar uma área de doença cardíaca bem como uma outra área de não doença cardíaca.

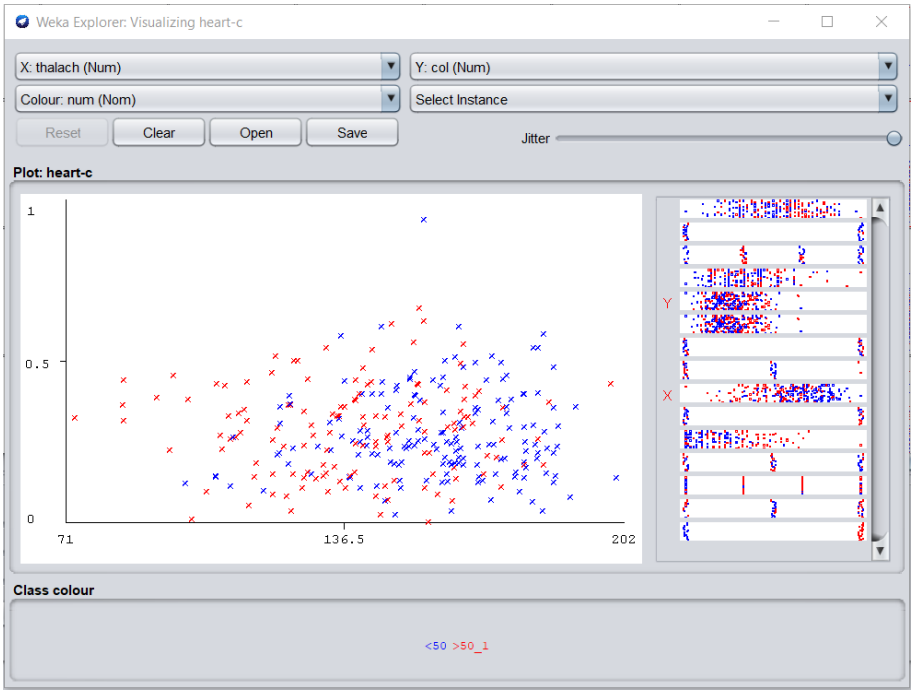

Figura 2: Correlação entre atributos thalach e col.

O mesmo atributo thalach na relação com o atributo oldpeak também possibilita a identificação de uma área densa de doença cardíaca.

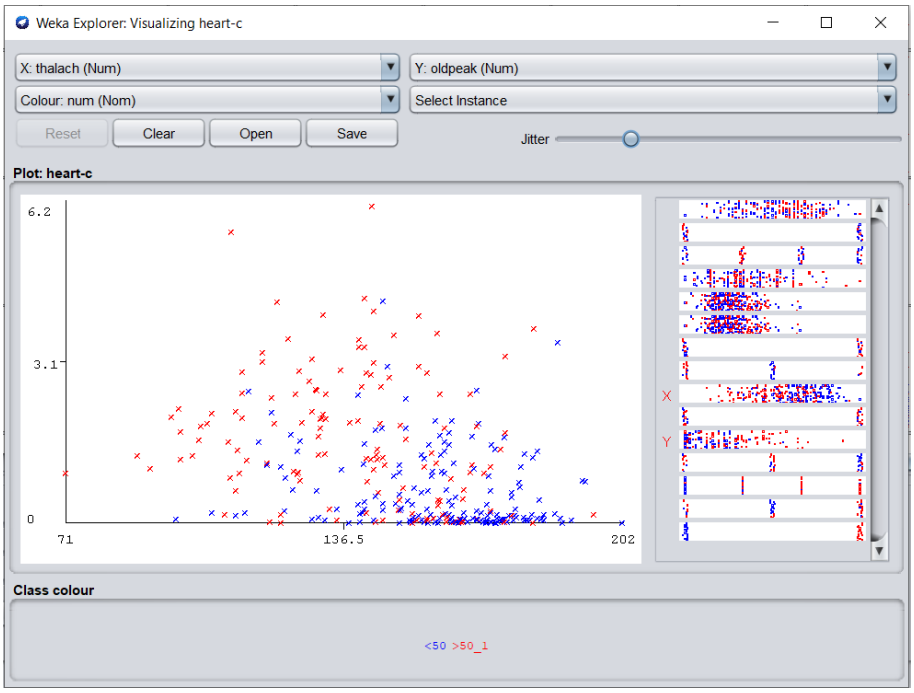

Figura 3: Correlação entre thalach e oldpeak.

3 Data Preprocessing

3.1 Seleção de atributos. Investigue a possibilidade de usar o filtro Weka Attribute Selection para selecionar um subconjunto de atributos com boa capacidade de previsão. Em seguida, descreva brevemente o(s) filtro(s) usado(s) e compare os resultados obtidos com as conclusões obtidas na seção anterior.Guarde o conjunto de dados com os atributos selecionados no ficheiro heart-c1.arff.

Ainda na tab do *Preprocess* é possível escolher o filtro AttributeSelection que tal como a aplicação indica é um filtro supervisionado de seleção de atributos. No mesmo filtro é possível definir também algoritmos distintos de avaliação e de procura. O *filter* anterior selecionou as seguintes classes.

Figura 4: Atributos selecionados pelo filtro AttributeSelection.

Como é possível verificar o algoritmo de filtragem anterior seleccionou mesmo assim os atributos thalach e col que possuem áreas densas de doença. O que está de acordo com o que foi verificado anteriormente

3.2 Lidar com valores ausentes. Considere os seguintes métodos para lida com valores ausentes e investigue cada possibilidade no Weka. Os registos não deverão ser eliminados e é aconselhável atribuir valores onde faltam dados, usando um método adequado.

Ainda no filter, foi utilizado o filtro unsupervised de *Replace Missing Values* cujo objetivo é substituir os dados em falta pela média, no caso dos numéricos, e a moda, no caso dos nominais.

3.3 Investigue a possibilidade de usar regressão (linear) para estimar os valores ausentes para cada atributo. Guarde o conjunto de dados que obteve sem valores ausentes no ficheiro heart-c3.arff.

De modo a utilizar a regressão linear foi necessário primeiro eliminar todos os atributos nominais, de seguida após os eliminar é possível na *tab classify* selecionar o algoritmo de regressão linear e aí é possível selecionar o *target attribute* dos quais é possível consultar as fórmulas geradas pelo algoritmo para estimar o valor de cada um dos *target attributes* consoante os outros atributos também presentes no *dataset*. Por exemplo, a seguinte fórmula com o *target attribute thalach* é possível estimar os missing values partindo dessa fórmula gerada.

```
Linear Regression Model

thalach =

     -0.9679 * age +
      0.1398 * trestbps +
     14.4078 * col +
     -5.6872 * oldpeak +
    185.8236

Time taken to build model: 0 seconds
```

Figura 5: Fórmula gerada pelo modelo de regressão linear.

3.4 Elimine os registos discrepantes e guarde o conjunto de dados obtido sem outliers no ficheiroheart-c34.arff. Investigue a possibilidade de usar o filtro Weka Unsupervised Attribute InterquartileRange para detetar outliers e o filtro Weka Unsupervised Instance RemoveWithValues para eliminar outliers(não se esqueça de configurar os parâmetros attributeIndex, que diz respeito ao índice do outlier, e nominalIndices, que corresponde à localização (first ou last) do valor nominal do atributo que se pretende remover)

Depois de aplicar um filtro para substituir os missing values, tendo sido utilizado o algoritmo *Replace Missing Values*, é agora possível detetar os outliers. Para tal utilizou-se os parâmetros standard do algoritmo interquartil para detetar os *outliers*. Este algoritmo gerou dois atributos adicionais para o dataset, o atributo *outlier* que conta quantos outliers existem no dataset e o atributo *extreme value* que conta se existem valores extremos no *dataset*. Através da visualização dos gráficos nos atributos na tab *Visualize* é possível encontrar um outlier no *dataset*.

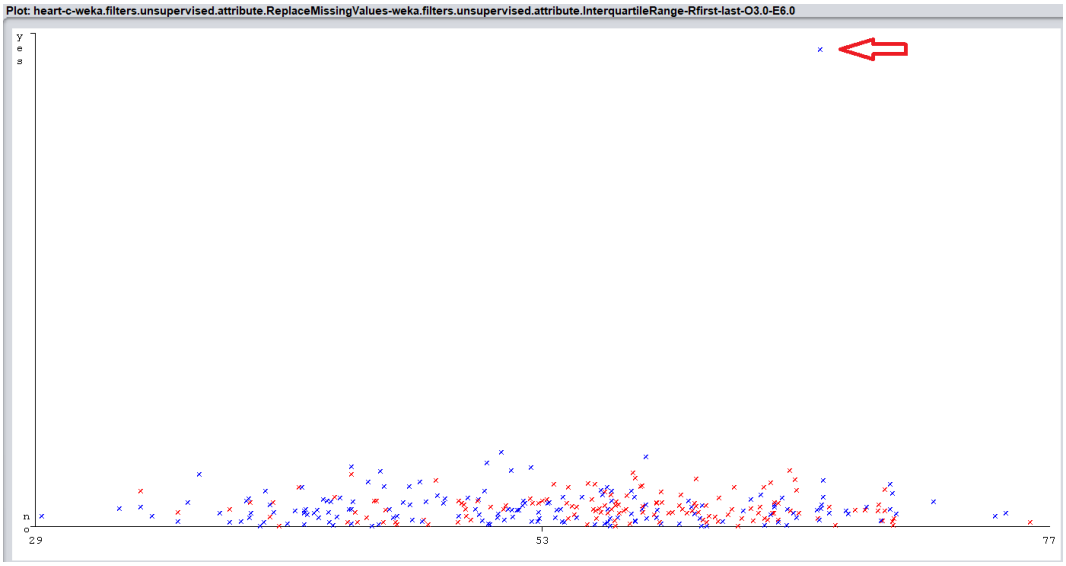

Figura 6: Outlier na idade.

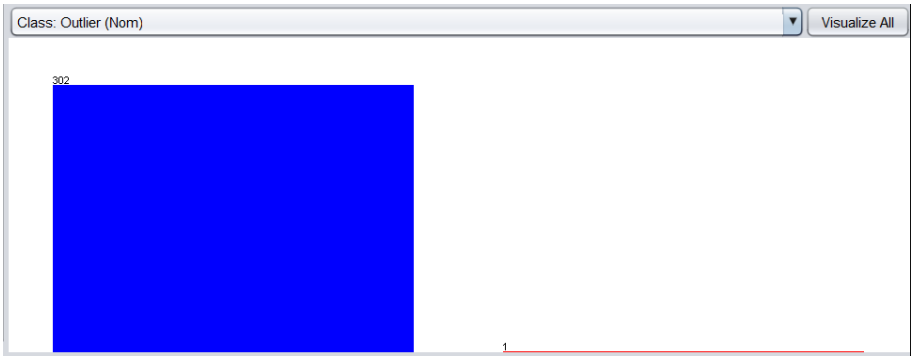

Figura 7: Número de outliers no dataset.

Ao aplica o algoritmo *RemoveWithValues* foi necessário configurar este ao apontar o *attributeIndex* para a coluna 16, ou seja a coluna *Outlier* e o *nominalIndices* para *last* para remover o último valor nominal que corresponde ao *"yes"* (de ser outlier). Após configurar o algoritmo com os parâmetros anteriores após o correr foi possível verificar

que foi removida uma instância do dataset (que correspondia ao outlier) tendo agora o dataset apenas 302 instâncias e sem outliers como se poderá verificar no seguinte gráfico.

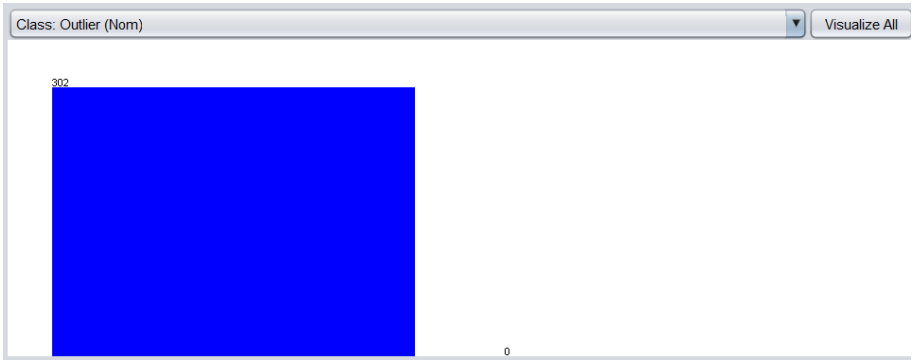

Figura 8: Dataset sem outliers.

4 Data Mining

Foram utilizados os datasets provenientes do 2.4, 2.1 e o dataset original.

4.1 Comece com o classificador OneR.

4.1.1 O que pode concluir? Compare as suas conclusões com as conclusões que obteve na seção 1.1.

O algoritmo OneR quando aplicado no dataset original permitiu que 71.62% das instâncias fosse corretamente classificadas.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         217               71.6172 %
Incorrectly Classified Instances        86               28.3828 %
Kappa statistic                          0.4305
Mean absolute error                      0.2838
Root mean squared error                  0.5328
Relative absolute error                 57.2125 %
Root relative squared error            106.9685 %
Total Number of Instances              303
```

Figura 9: Algoritmo OneR no dataset original.

Utilizando o algoritmo OneR no dataset proveniente da questão 2.1 foi possível obter uma boa percentagem igual de instâncias corretamente classificadas com 71.62% das instâncias corretamente classificadas.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         217               71.6172 %
Incorrectly Classified Instances        86               28.3828 %
Kappa statistic                          0.4305
Mean absolute error                      0.2838
Root mean squared error                  0.5328
Relative absolute error                 57.2125 %
Root relative squared error            106.9685 %
Total Number of Instances              303
```

Figura 10: Algoritmo OneR no dataset 2.1.

Passando para o dataset proveniente da questão 2.4, através da eliminação de *outliers* foi possível aumentar um pouco a percentagem de instâncias corretamente classificadas para 72.19%.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         218               72.1854 %
Incorrectly Classified Instances        84               27.8146 %
Kappa statistic                          0.4402
Mean absolute error                      0.2781
Root mean squared error                  0.5274
Relative absolute error                 56.0372 %
Root relative squared error            105.8636 %
Total Number of Instances              302
```

Figura 11: Algoritmo OneR no dataset 2.4.

Portanto como é possível verificar a técnica mais vantajosa provem de eliminar os outliers de modo a aumentar a percentagem de casos corretamente avaliados. O algoritmo *OneR* também identifica o atributo que mais contribui para o fato de ter doença cardíaca, sendo que em ambos os casos identificou o atributo *thal* como sendo o mais influente. O mesmo também foi verificado na secção 1, onde foi dito que com *reversable defect* maior é a probabilidade de ter doença cardíaca.

```
thal:
        fixed_defect    -> >50_1
        normal  -> <50
        reversable_defect       -> >50_1
(232/302 instances correct)
```

Figura 12: Atributo com mais influência.

4.1.2 Compare a precisão do classificador obtida no conjunto de treino (training set) com a estimativa de precisão obtida através do método 10 fold-cross validation. Como explica esta diferença (se existir)?

Na pergunta anterior foi utilizado o método 10 fold-cross validation, pelo que nesta questão foi necessário utilizar o *training set*. Começando pelo dataset original, no mesmo é possível verificar um aumento do número de instâncias corretamente classificadas pelo que aumenta também a precisão do algoritmo.

```
=== Summary ===

Correctly Classified Instances         232               76.5677 %
Incorrectly Classified Instances        71               23.4323 %
Kappa statistic                          0.5268
Mean absolute error                      0.2343
Root mean squared error                  0.4841
Relative absolute error                 47.2373 %
Root relative squared error             97.2006 %
Total Number of Instances              303
```

Figura 13: Precisão do modelo no dataset original.

Passando para o dataset proveniente da questão 2.1 mais uma vez a precisão é precisamente a mesma que no dataset original.

```
=== Summary ===

Correctly Classified Instances         232               76.5677 %
Incorrectly Classified Instances        71               23.4323 %
Kappa statistic                          0.5268
Mean absolute error                      0.2343
Root mean squared error                  0.4841
Relative absolute error                 47.2373 %
Root relative squared error             97.2006 %
Total Number of Instances              303
```

Figura 14: Precisão do modelo no dataset da questão 2.1.

No dataset proveniente da questão 2.4, pelo que também aconteceu na questão anterior foi possível aumentar um pouco a precisão utilizando o *training set* para os 76.82%.

```
=== Summary ===

Correctly Classified Instances         232               76.8212 %
Incorrectly Classified Instances        70               23.1788 %
Kappa statistic                          0.5319
Mean absolute error                      0.2318
Root mean squared error                  0.4814
Relative absolute error                 46.7015 %
Root relative squared error             96.6476 %
Total Number of Instances              302
```

Figura 15: Precisão do modelo no dataset da questão 2.4.

Ao escolher a opção de *training set* é normal que o número de instâncias corretamente classificadas aumente uma vez que se está a testar no mesmo dataset que o algoritmo escolhido está a treinar, o que pode conduzir a situações de *overfit*.

4.2 Use o classificador JRip, ou seja, a versão Weka do classificador de regras RIPPER.

4.2.1 Crie um classificador com e sem rule pruning. Qual é o melhor? Justifique a sua resposta.

Para esta questão é necessário alterar os parâmetros do algoritmo, sendo necessário mexer no *usePruning* alternando de verdadeiro para falso em cada dataset e anotando o número de instâncias corretamente classificadas, tendo sido utilizado em todos o método de *Cross validation* com 10 *folds*. Como tal, foi possível elaborar a seguinte tabela.

	Dataset original	Dataset pergunta 2.1	Dataset pergunta 2.4
Precisão (True/False)	0,812/0,778	0,803/0,786	0,769/0,783

Como é possível verificar é sempre preferível o uso de pruning ao invés de não usar o pruning uma vez que com o uso de pruning conduz sempre a melhores precisões do modelo.

4.3 Use o classificador J48, ou seja, a versão Weka do classificador C4.5 da árvore de decisão.

4.3.1 Explore o uso de diferentes parâmetros de J48, como pruning("unpruned") e número mínimo de registos nas folhas("minNumObj").

Mais uma vez com o algoritmo J48 foi necessário alterar os parâmetros do mesmo, tendo sido alterado o fato de usar pruning ou não em cada um dos datasets, o que conduziu as seguintes precisões do modelo. Convém realçar também que em cada um dos datasets o modelo foi executado com *Cross Validation* com *10 folds*.

	Dataset original	Dataset pergunta 2.1	Dataset pergunta 2.4
Precisão (True/False)	0,779/0,779	0,774/0,786	0,768/0,765

Como é possível verificar na tabela anterior, o facto de parâmetro *unpruned* estar a *true/false* tem pouco impacto na performance (precisão) do algoritmo em geral.

4.3.2 Descreva os padrões que obteve e compare com as conclusões obtidas nas questões anteriores.

Portanto, utilizando este algoritmo *J48* em geral no que toca à análise da precisão este algoritmo produz piores resultados que o *JRip*, pelo que o facto de a árvore estar *pruned* ou não, não impacta no que toca à precisão do algoritmo.

5 Clustering Tendency

5.1 Investigue se existe uma tendência de clustering no dataset. Pode começar por agrupar os dados com o algoritmo SimpleKMeans, para k a variar de 2 a 10.

Para responder às seguintes questões foi necessário alterar o parâmetro *numClusters (k)* de modo a variar consoante o intervalo dado. Para além disso foi utilizado o *dataset* original para responder a esta pergunta.

5.1.1 Não use o atributo class, num, para o clustering.

k=2

Ignorando o atributo num, é possível verificar à partida que a distribuição de clustering foi de 56%/44%.

```
=== Model and evaluation on training set ===

Clustered Instances

0       170 ( 56%)
1       133 ( 44%)
```

Figura 16: Distribuição clustering dataset original.

Para além disso e analisando graficamente cada um dos atributos é possível vetrificar algumas tendências de clustering, como por exemplo no atributo restecg onde está agrupado ou no left_vent_hyper ou no normal.

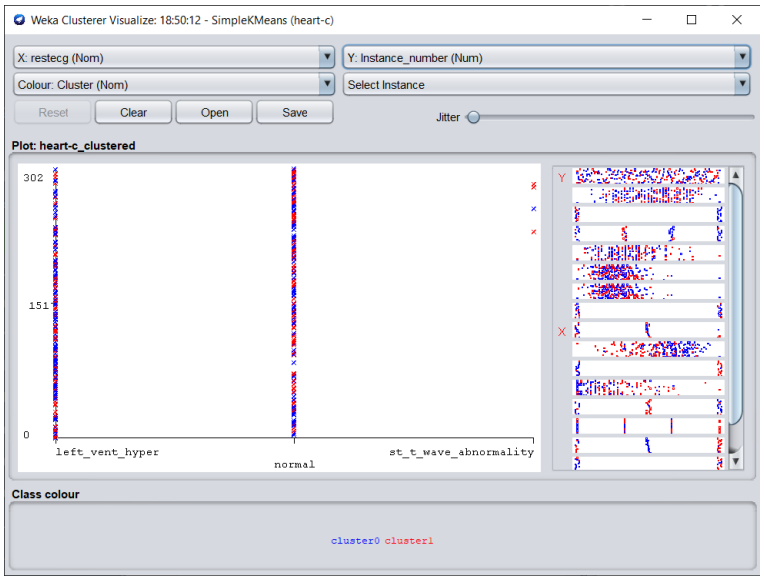

Figura 17: Distribuição clustering no atributo restecg.

Para além deste atributo o mesmo pode ser verificado no *thal (normal e reversable_defect)*, no *cp (asympt e non_-anginal)*, no *slope (flat e up)*.

k=3

Com k=3 a distribuição de *clustering* foi 36%/33%/31%, onde se verificou resultados semelhantes aos anteriores, onde, por exemplo, no caso do atributo *thal* verifica-se cluster nos mesmos campos que no anterior, ou seja, normal e reversable_defect.

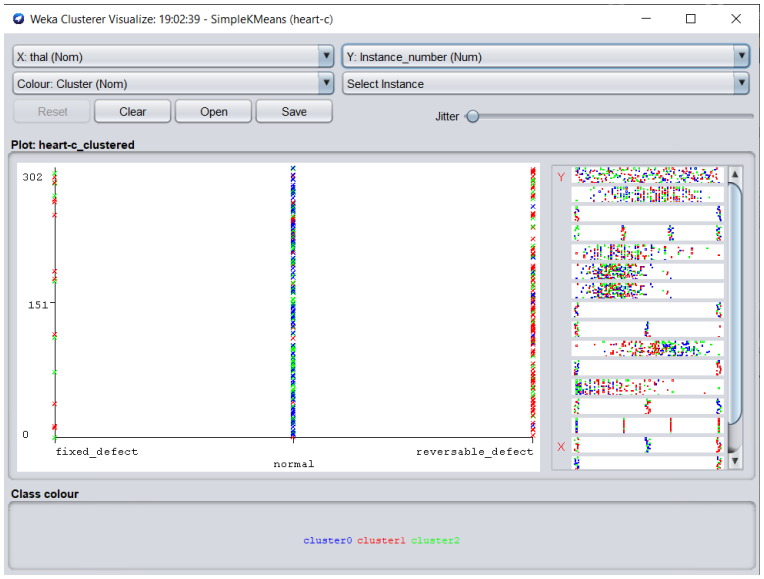

Figura 18: Distribuição clustering no atributo thal com k=3.

No próprio atributo restecg também se verifica os mesmos resultados que anteriormente.

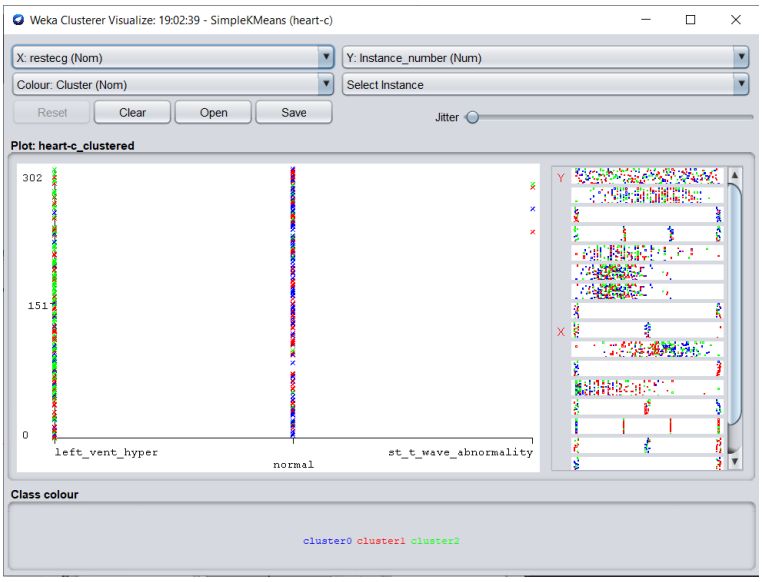

Figura 19: Distribuição clustering no atributo restecg com k=3.

Como os resultado são ainda pouco diferenciados com o anterior *numClusters* passará para k=5 de modo a obter resultados mais expressivos.

k=5

Com o número de clusters a cinco, a distribuição de clusters é a seguinte.

```
=== Model and evaluation on training set ===

Clustered Instances

0       53 ( 17%)
1       55 ( 18%)
2       46 ( 15%)
3       69 ( 23%)
4       80 ( 26%)
```

Figura 20: Distribuição clustering com k=5.

Analisando desta vez o relatório produzido pelo algoritmo, é possível verificar que mesmo com o k=5 os clusters possuem pouca variabilidade e são pouco diferenciados, possuindo uma soma de erros quadrados de 499,19.

Final cluster centroids:

Attribute	Full Data (303.0)	Cluster# 0 (53.0)	1 (55.0)	2 (46.0)	3 (69.0)	4 (80.0)
age	54.3663	58.4151	56.7273	55.1522	56.7536	47.55
sex	male	female	male	male	male	male
cp	asympt	non_anginal	asympt	asympt	asympt	non_anginal
trestbps	131.6238	133.4906	132.2364	129.2391	138	125.8375
col	0.2746	0.3305	0.2641	0.2707	0.2895	0.2341
chol	246.264	270.7547	241.6727	244.5652	252.7971	228.5375
fbs	f	f	f	f	f	f
restecg	normal	left_vent_hyper	normal	left_vent_hyper	left_vent_hyper	normal
thalach	149.6469	152.7925	134.9636	160.0217	135.5652	163.8375
exang	no	no	yes	no	yes	no
oldpeak	1.0396	0.617	1.5673	0.6217	1.7826	0.5563
slope	up	up	flat	up	flat	up
ca	0.6745	0.3396	0.885	0.9783	1.0967	0.2128
thal	normal	normal	reversable_defect	normal	reversable_defect	normal

Figura 21: Relatório de clustering k=5.

k=7

Com o número de clusters a sete, a distibuição de clustering é a seguinte:

```
=== Model and evaluation on training set ===

Clustered Instances

0       44 ( 15%)
1       10 (  3%)
2       32 ( 11%)
3       50 ( 17%)
4       73 ( 24%)
5       15 (  5%)
6       79 ( 26%)
```

Figura 22: Distribuição clustering com k=7.

Para além disso foi também verificado o relatório de clustering onde se verificou uma maior variabildade dentro dos clusters, sendo que mesmo assim em certos atributos manteve-se alguma tendência nos mesmos campos que aqueles apresentados anteriormente como no *restecg* com o campo normal e left_vent_hyper. A soma dos erros quadrados com este número de clusters é de 487,8.

Full Data (303.0)	Cluster# 0 (44.0)	1 (10.0)	2 (32.0)	3 (50.0)	4 (73.0)	5 (15.0)	6 (79.0)
54.3663	58.9318	59.9	51.7813	57.64	47.3562	58	55.8861
male	female	female	male	male	male	male	male
asympt	non_anginal	asympt	asympt	asympt	non_anginal	atyp_angina	asympt
131.6238	134.6591	154.6	124.875	129.88	126.0822	140	134.3924
0.2746	0.3339	0.3589	0.2602	0.2784	0.2343	0.2604	0.2742
246.264	272.25	283.2	239.9688	247.94	228.6027	240.0667	246.1013
f	f	f	f	f	f	t	f
normal	left_vent_hyper	left_vent_hyper	left_vent_hyper	left_vent_hyper	normal	left_vent_hyper	normal
149.6469	154.3636	144.4	160.75	139.02	163.7671	160.0667	134.8861
no	no	no	no	no	no	no	yes
1.0396	0.5886	3.47	0.375	1.372	0.4808	0.3333	1.6924
up	up	down	up	flat	up	up	flat
0.6745	0.3409	2.2	1	0.8	0.2058	0.6667	0.8905
normal	normal	reversable_defect	normal	normal	normal	normal	reversable_defect

Figura 23: Relatório de clustering k=7.

k=8

Com o número de clusters a oito, a distribuição de clustering é a seguinte:

```
=== Model and evaluation on training set ===

Clustered Instances

0       41 ( 14%)
1        9 (  3%)
2       32 ( 11%)
3       44 ( 15%)
4       71 ( 23%)
5       15 (  5%)
6       68 ( 22%)
7       23 (  8%)
```

Figura 24: Distribuição clustering com k=8.

Analisando a distribuição de clusters o cluster número um, cinco e 7 possui uma distribuição baixa de instâncias quando comparados com os outros, para além disso e analisando também o relatório produzido verifica-se uma maior variabilidade como seria expectável, sendo que alguns campos possuem de novo alguma tendência como aquelas apresentadas anteriormente. A soma dos erros quadrados com este número de clusters é de 464,7, sendo que diminui bastante quando comparado com o número de clusters a sete.

k=9

Com o número de clusters a nove, a distribuição de clustering é a seguinte:

```
=== Model and evaluation on training set ===

Clustered Instances

0       35 ( 12%)
1        9 (  3%)
2       31 ( 10%)
3       28 (  9%)
4       66 ( 22%)
5       13 (  4%)
6       67 ( 22%)
7       23 (  8%)
8       31 ( 10%)
```

Figura 25: Distribuição clustering com k=9.

Mais uma vez existem clusters com baixa distribuição como o 1, o 3, o 5 e o 7, sendo que a soma dos erros quadrados diminiu para 452,9. Contudo verifica-se que o *cluster* está a tender para os mesmos valores, como se poderá verificar analisando os campos nominais.

k=10

Com o número de clusters a dez, a distribuição de clustering é a seguinte:

```
=== Model and evaluation on training set ===

Clustered Instances

0       35 ( 12%)
1        9 (  3%)
2       31 ( 10%)
3       28 (  9%)
4       66 ( 22%)
5       13 (  4%)
6       67 ( 22%)
7       23 (  8%)
8       31 ( 10%)
```

Figura 26: Distribuição clustering com k=10.

Portanto, cada vez mais existem clusters com menor distribuição uma vez que se está a aumentar o número de clusters totais, o que é concordante. A soma de erros quadrados diminui mais uma vez para 444,4 e tal como foi dito anteriormente aumentando o número de clusters para 10 é possível verificar uma convergência.

5.1.2 Encontre um valor adequado para k, ou seja, o número de clusters que vai construir. Justifique a sua escolha de k.

Portanto, com valores demasiado pequenos de clusters não existe variabildade suficiente de clusters para dispersar os valores, e com um número demasiado elevado de clusters existe demasiada variablidade de clusters e dispersão para agrupar os valores. Como tal, o número ideal de clusters não deverá ser demasiado grande nem demasiado pequeno sendo que com um número de clusters igual a sete existe um número adequado de variabiliade e dispersão de valores.

5.1.3 Use o atributo class para o clustering e certifique-se de que os desvios padrão também são computados para atributos numéricos(displayStdDevs)

Foi utilizado o número de clusters indicado na pergunta anterior, ou seja, com sete clusters, como tal utilizando o atributo class para o clustering, foi gerado o seguinte relatório.

Final cluster centroids:

Attribute	Full Data	Cluster# 0	1	2	3
	(303.0)	(44.0)	(10.0)	(32.0)	(50.0)
age	54.3663	58.9318	59.9	51.7813	57.64
	+/-9.0821	+/-8.3147	+/-3.035	+/-10.5608	+/-7.7664
sex	male	female	female	male	male
male	207.0 (68%)	1.0 (2%)	2.0 (20%)	30.0 (93%)	40.0 (80%)
female	96.0 (31%)	43.0 (97%)	8.0 (80%)	2.0 (6%)	10.0 (20%)
cp	asympt	non_anginal	asympt	asympt	asympt
typ_angina	23.0 (7%)	4.0 (9%)	1.0 (10%)	4.0 (12%)	6.0 (12%)
asympt	143.0 (47%)	6.0 (13%)	9.0 (90%)	22.0 (68%)	32.0 (64%)
non_anginal	87.0 (28%)	25.0 (56%)	0.0 (0%)	3.0 (9%)	9.0 (18%)
atyp_angina	50.0 (16%)	9.0 (20%)	0.0 (0%)	3.0 (9%)	3.0 (6%)
trestbps	131.6238	134.6591	154.6	124.875	129.88
	+/-17.5381	+/-16.3465	+/-24.0933	+/-14.738	+/-17.3307
col	0.2746	0.3339	0.3589	0.2602	0.2784
	+/-0.1183	+/-0.1603	+/-0.1775	+/-0.0914	+/-0.104
chol	246.264	272.25	283.2	239.9688	247.94
	+/-51.8308	+/-70.2041	+/-77.7529	+/-40.0278	+/-45.558
fbs	f	f	f	f	f
t	45.0 (14%)	5.0 (11%)	2.0 (20%)	2.0 (6%)	6.0 (12%)
f	258.0 (85%)	39.0 (88%)	8.0 (80%)	30.0 (93%)	44.0 (88%)

Figura 27: Parte do relatório gerado para k=7.

5.1.4 Estude as medidas numéricas apresentadas pelo Weka para cada cluster. O que pode concluir?

Para cada atributo numérico é possível observar o desvio padrão de cada um deles, como tal é possível verificar que em geral não existe grande variabilidade nos valores de cada um dos atributos, o que indica a qualidade de *clustering* efetuado.

5.1.5 Selecione a opção "Visualize cluster assignments" e tente descobrir uma descrição para cada cluster.

Analisando graficamente a distribuição de *clustering* para o atributo thal é possível verificar a distribuição de dados para cada um dos *clusters* não sendo evidente uma descrição para cada um dos sete clusters.

Figura 28: Distribuição clustering no atributo thal usando o atributo num.

5.1.6 Investigue a possibilidade de utilizar as informações do cluster para construir um classificador para num. Compare os resultados com o que obteve na seção 1.3. Obteve um classificador melhor? Pista: na janela "Visualize cluster assignments" seleccione no eixo Y "Cluster" e guarde como um novo dataset

Através da pista dada foi possível guardar o dataset num novo ficheiro, sendo agora possível verificar o a distribuição das instâncias por cada um dos clusters como se segue.

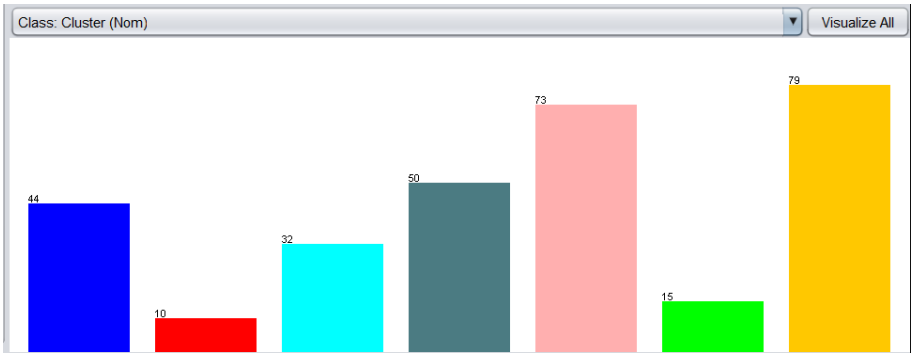

Figura 29: Count plot do novo atributo cluster.

Utilizando este novo dataset, foram aplicados com os parâmetros standard os classificadores OneR, JRip e J48 tendo sido obtido os seguintes resultados.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         236               77.8878 %
Incorrectly Classified Instances        67               22.1122 %
Kappa statistic                          0.5618
Mean absolute error                      0.2211
Root mean squared error                  0.4702
Relative absolute error                 44.5725 %
Root relative squared error             94.4157 %
Total Number of Instances              303
```

Figura 30: Sumário gerado a partir do novo dataset para o algoritmo OneR.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         246               81.1881 %
Incorrectly Classified Instances        57               18.8119 %
Kappa statistic                          0.6187
Mean absolute error                      0.2622
Root mean squared error                  0.398
Relative absolute error                 52.8565 %
Root relative squared error             79.908  %
Total Number of Instances              303
```

Figura 31: Sumário gerado a partir do novo dataset para o algoritmo JRip.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         238               78.5479 %
Incorrectly Classified Instances        65               21.4521 %
Kappa statistic                          0.5636
Mean absolute error                      0.26
Root mean squared error                  0.4318
Relative absolute error                 52.4173 %
Root relative squared error             86.7027 %
Total Number of Instances              303
```

Figura 32: Sumário gerado a partir do novo dataset para o algoritmo J48.

Quando comparado com os resultados obtidos anteriormente pelos três algoritmos anteriores, é possível verificar que no algoritmo OneR e J48 este novo dataset produz melhores resultados uma vez que melhorou o número de instâncias classificadas corretamente com os novos dados provenientes do cluster. Para o algoritmo JRip o número de instâncias classificadas corretamente (precisão) é aproximadamente igual.

6 Predicting Performance

6.1 Na etapa anterior construiu vários modelos. Por fim, é necessário comparar os diferentes modelos e apresentar as suas conclusões.

6.1.1 Weka oferece várias medidas de avaliação de desempenho. Escolha algumas medidas de desempenho e justifique a sua escolha.

Tal como foram utilizados ao longo do relatório medidas como a número de instâncias corretamente classificadas e incorretamente classificadas bem como a precisão são importantes para definir o bom ou mau desempenho do algoritmo. Uma vez que são essas as medidas que ditam o quão bem é que o algoritmo classifica para os dados que possui.

6.1.2 Resuma numa tabela as medidas de desempenho para cada classificador e cada dataset.

As medidas anteriores foram sempre apresentadas e discutidas em cada uma das questões anteriores. Seria também interessante utilizar o erro para acrescentar à tabela contudo não foram guardadas todas as medições. Na tabela seguinte apresenta-se os três algoritmos utilizados com a percentagem de instâncias classificadas corretamente e incorretamente para o dataset original, o dataset proveniente da pergunta 2.1, 2.4 e por fim o dataset gerado através do clustering (respetivamente). Os parâmetros dos algoritmos de classificação permaneceram standard.

	OneR	JRip	J48
Percentagem correta	71.62/71.62/72.19/77.90	81.2/80.3/76.9/81.19	77.9/77.4/76.8/78.55
Percentagem incorreta	28.38/28.38/27.81/22.10	18.8/19.7/23.1/18.81	22.1/22.6/23.2/21.45

6.1.3 O que pode concluir?

Portanto, é possível verificar que o dataset que possui melhores resultados em geral é o dataset proveniente do *clustering* sendo que o algoritmo que produz também maiores instâncias classificadas corretamente é o algoritmo *JRip*.