INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Modelo de data mining para detecção de tumores em exames de rastreio

VITOR NUNO PATROCÍNIO DOS SANTOS

(Licenciado)

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática

Orientadores : Mestre Nuno Miguel Soares Datia
Doutora Matilde Pós-de-Mina Pato

Júri:

Presidente: Doutor Hélder Jorge Pinheiro Pita

Vogais: Doutora Cátia Luísa Santana Calisto Pesquita
Doutora Matilde Pós-de-Mina Pato

SETEMBRO, 2013

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

Modelo de data mining para detecção de tumores em exames de rastreio

VITOR NUNO PATROCÍNIO DOS SANTOS

(Licenciado)

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática

Orientadores : Mestre Nuno Miguel Soares Datia
Doutora Matilde Pós-de-Mina Pato

Júri:

Presidente: Doutor Hélder Jorge Pinheiro Pita

Vogais: Doutora Cátia Luísa Santana Calisto Pesquita
Doutora Matilde Pós-de-Mina Pato

SETEMBRO, 2013

Aos meus pais e irmã que sempre estiveram ao meu lado.
Tudo aquilo que sou e faço é reflexo dos valores e
sentimentos que sempre me transmitiram...

Agradecimentos

Os meus sinceros agradecimentos aos meus orientadores, Matilde Pós-de-Mina Pato e Nuno Datia, incansáveis no seu apoio e motivação, e inexcedíveis na sua dedicação e disponibilidade. A sua orientação precisa, organização exemplar e conselhos sensatos, foram essenciais para a concretização deste trabalho, contribuindo enormemente para o meu enriquecimento pessoal, académico e científico.

Aos meus colegas de mestrado, Carlos Abrantes e Leandro Nunes, pela amizade, companheirismo, motivação e espírito de entreajuda na realização dos inúmeros trabalhos das diferentes cadeiras.

A todos cujo o caminho se cruzou com o meu, durante todo o meu percurso académico. Entre os quais, o meu orientador de projecto de licenciatura, Carlos Guedes e os meus colegas de licenciatura, Bruno Fonseca, David Júlio, Hugo Cruz, Hugo Marçal, Jorge Graça e Ricardo Lourenço, pelo seu apoio em diversos momentos deste percurso.

Aos meus amigos, por continuarem a sê-lo apesar das minhas longas ausências. Pelos momentos de descontracção e partilha, tão importantes para o meu equilíbrio pessoal.

À minha família, mais concretamente, aos meus pais, Carlos e Vitória, à minha irmã Carla, à minha namorada Carmen e ao meu cunhado João, por toda a sua compreensão, paciência, incentivo, motivação e disponibilidade. Por todo o apoio que sempre prestaram nos momentos mais difíceis.

A todos, muito obrigado...

Resumo

O cancro da mama é uma das formas de cancro mais comum nas mulheres em todo o mundo. É actualmente o cancro, com excepção do cancro da pele, de maior incidência nas mulheres. A taxa de mortalidade que lhe está associada pode ser reduzida se a detecção ocorrer num estágio precoce da doença, normalmente, através de exames de rastreio designados por mamografias.

Existem algumas ferramentas que digitalizam esses exames e extraem algumas características que depois de tratadas, permitem ajudar os especialistas a classificar os pacientes como doentes de cancro ou não.

O objectivo deste trabalho é partir dessas características, construir e descrever um modelo de *Data Mining* para detecção do cancro da mama. É expectável que o modelo seja capaz de classificar correctamente todos os pacientes com cancro e, tenha um número reduzido de falsos positivos para evitar a realização de exames de diagnóstico invasivos em pacientes saudáveis.

Os dados provenientes de exames médicos contêm diversos desafios, dada a dimensão e características dos dados, pelo que se torna necessário adoptar diversas técnicas de redução do conjunto e posteriormente avaliar o seu impacto nos resultados. São usadas diversas técnicas de selecção de atributos e balanceamento dos dados.

São ainda comparados diversos algoritmos de aprendizagem, provenientes de diferentes famílias. É analisado e avaliado, o seu desempenho, face às diversas técnicas usadas na redução da dimensão dos dados. São usados meta-algoritmos como o *ensemble*, criado a partir da combinação de vários algoritmos base, tendo como objectivo a optimização da classificação.

Os resultados obtidos por combinação destas técnicas são então comparados e avaliados. Verifica-se que alguns algoritmos cumprem os objectivos propostos.

Também se mostra que o uso de PCA incrementa substancialmente a prestação do *Naive Bayes* ao contrário do *Random Forest* onde o desempenho é significativamente penalizado. O balanceamento também tem impacto na classificação embora menos significativo.

Um estudo de parametrização dos algoritmos analisados será um trabalho a desenvolver no futuro.

Palavras-chave: *data mining*, cancro da mama, selecção de atributos, balanceamento de dados, classificaçao, PCA ...

Abstract

Breast cancer is one of the most common cancer in women worldwide. Nowadays, breast cancer is a type of cancer with higher incidence in women, excluding skin cancer. The mortality rate can be reduced if detection occurs at an earlier stage of disease, generally by means of screening tests known as mammograms.

There are some tools in the market that digitize these exams, extract the features of the images and make that available to experts after treatment, helping them to classify the patients as cancer patients or not.

The aim of this work is to construct and describe a data mining model for the detection of breast cancer, based on these features. It is expected that the model will be able to correctly classify all patients with cancer and reduce the number of false positives, avoiding invasive diagnostic tests in healthy patients.

Data from medical exams contain many challenges, given the size and characteristics of the data, which makes it necessary to adopt several techniques to reduce the data set and then evaluate their impact on the results. Several techniques are used for feature selection and balancing the data.

There is also a comparison of different learning algorithms from different families. Is analyzed and evaluated its performance considering the various techniques used to reduce the size of data. Ensembles are used to combine several basic algorithms, with the aim to optimize the classification process.

The results obtained by combining these techniques are then compared and evaluated. It turns out that some algorithms meet their objectives. It is also shown that the use of PCA increases substantially the performance of Naive Bayes, unlike Random Forest where the performance is greatly penalized. The balancing also has impact on the classification, although that impact is less significant.

A study of parametrization of the studied algorithms shall be made in a future work.

Keywords: Data Mining, Breast Cancer, Feature Selection, Principal Component Analysis …

Lista de abreviaturas, acrónimos, siglas e símbolos

ANN	*Artificial Neural Network*
ARL	*Association Rule Learning*
AUC	Área sob a Curva
BN	*Bayesian Networks*
CAD	Detecção Assistida por Computador
CART	*Classification and Regression Trees*
CC	Crânio Caudal
CFS	*Correlation-based feature selection*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
CS	*Chi Squared*
DM	*Data Mining*
FN	Falsos Negativos
FP	Falsos Positivos
FR	*Feature Ranking*
FROC	*Free-response Receiver Operating Characteristic*
GR	*Gain Ratio*

IARC *International Agency for Research on Cancer*

IBL *Instance-Based Learning*

IG *Information Gain*

IL *Inductive Learning*

INE Instituto Nacional de Estatística

KDD *Knowledge Discovery from Databases*

KNN *K-nearest neighbors*

LDA *Linear Discriminant Analysis*

LR *Logistic Regression*

MARS *Multivariate Adaptive Regression Splines*

MLM *Multinomial Log-linear Models*

MLO Médio Lateral Oblíqua

NB *Naive Bayes*

PCA *Principal Component Analysis*

QDA *Quadratic Discriminant Analysis*

RBL *Rule-Based Learning*

RBF *Radial Basis Function*

RF *Random Forest*

ROC *Receiver Operating Characteristic*

RPART *Recursive Partitioning and Regression Trees*

SEMMA *Sample, Explore, Modify, Model and Assess*

SL *Statistical Learning*

SMOTE *Synthetic Minority Oversampling Technique*

SOM *Self-organizing Map*

SU *Symmetrical Uncertainty*

SVM *Support Vector Machines*

TN Verdadeiros Negativos

TP Verdadeiros Positivos

WHO *World Health Organisation*

Conteúdo

Lista de Figuras

Lista de Tabelas

Listagens

1

Introdução

O cancro é uma doença caracterizada por uma proliferação anormal de células, invade e destrói tecidos adjacentes, e pode-se espalhar pelo corpo, através de um processo chamado metástase. Estas propriedades malignas do cancro diferenciam-no dos tumores benignos, que muitas vezes regridem e não se "espalham", ou seja, não se disseminam para os tecidos em volta ou para outras partes do organismo. O cancro pode afectar pessoas de todas as idades, mas o risco aumenta com a idade [121]. Causa cerca de 13% de todas as mortes no mundo, sendo os cancros de pulmão, estômago, fígado, cólon e mama, no caso das mulheres, os que mais matam [68, 96]. A elevada taxa de mortalidade, a debilidade dos pacientes durante o tratamento, fase terminal da doença e a imunidade inexistente face à doença, causam muita sensibilidade nas pessoas, pelo que não é possível ser indiferente a esta doença e seus efeitos nefastos.

A investigação constante, numa área de intervenção tão importante como o cancro é, inquestionavelmente, necessária. O cancro da mama é actualmente o cancro (com excepção da pele) de maior incidência nas mulheres no mundo, quer nos países desenvolvidos quer em desenvolvimento [96]. A maioria das mortes por cancro de mama ocorrem em países com baixo e médio rendimentos, onde a maioria das mulheres são diagnosticadas em estágios avançados, principalmente por falta de conhecimentos e de barreiras de acesso aos serviços de saúde [96]. Deste modo pretende-se potenciar a análise dos exames de rastreio. O problema centra-se na dificuldade que os especialistas têm em interpretar os dados do exame de

rastreio, a mamografia. O objectivo deste trabalho, é dotar os especialistas de ferramentas de apoio à detecção de cancro com mais precisão, diminuindo o desafio, aumentando a eficácia e eficiência na detecção de tumores pelos especialistas. O *Data Mining* (DM) consiste numa das etapas do *Knowledge Discovery from Databases* (KDD) [1], onde o objectivo é encontrar padrões úteis a partir de dados já pré-processados. Talvez a definição mais importante tenha sido elaborada por Usama Fayyad [45]: "... o processo não-trivial de identificar, em dados, padrões válidos, novos, potencialmente úteis e ultimamente compreensíveis".

Na saúde, a grande quantidade de informação mantida pelas instituições faz aumentar cada vez mais a importância deste procedimento.

1.1 Contexto

O cancro da mama é uma das doenças com maior impacto na nossa sociedade, dada a sua elevada frequência e gravidade, além de consistir numa agressão a um dos órgãos com maior carga simbólica nas mulheres, simbolismo da maternidade e mais importante ainda, da sua feminilidade.

Em termos anatómicos, as mamas são glândulas secretoras de leite, assentes nos músculos peitorais que cobrem as costelas. A mama, figura 1.1, é constituída por lóbulos, ductos e estroma (tecido adiposo vulgarmente designado de gordura). Está dividida em 15 a 20 secções, designadas de lobos [33]. Os lobos são constituídos por lóbulos mais pequenos que por sua vez contêm grupos de pequenas glândulas com capacidade de produção de leite. O leite flui dos lóbulos até ao mamilo por finos canais designados de ductos. O espaço entre os ductos e os lóbulos é preenchido com estroma.

A mama tem ainda vasos sanguíneos e vasos linfáticos que transportam um líquido límpido, a linfa. Os vasos linfáticos terminam nos gânglios linfáticos, que fazem parte do sistema linfático, responsável por reter e eliminar substâncias estranhas ao organismo humano que circulam neste sistema, como por exemplo bactérias, vírus e células cancerígenas. Na região da mama e proximidades, encontram-se vários grupos linfáticos, nas axilas, acima da clavícula e no peito (atrás do esterno). Quando as células de cancro da mama entram no sistema linfático, podem ser encontradas nos gânglios linfáticos próximos da mama, designados por gânglios regionais, e detectadas através de exames específicos [33, 106].

[1] http://www.kdd.org/

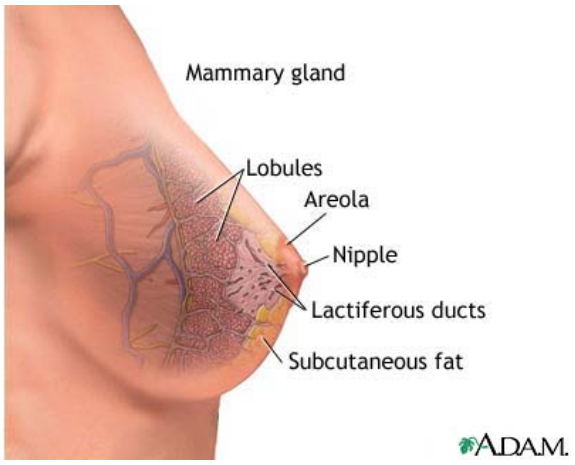


Figura 1.1: Anatomia da mama (Fonte: A.D.A.M.)

Os cancros podem ser classificados como não invasivos (*in situ*) ou invasivos (infiltrante). O cancro é classificado como não invasivo se este está confinado à área onde se desenvolveu inicialmente. É classificado como invasivo se este tem uma tendência de se espalhar para outras partes do tecido mamário ou outras regiões do corpo humano, processo designado de criação de metástases [124].

A grande maioria dos casos de cancro da mama ocorrem em mulheres em que não são identificados factores associados ao aparecimento da doença [71]. No entanto, foram identificados os seguintes factores:

Sexo O sexo feminino é mais propenso ao cancro na mama, sendo raros os casos deste tipo de cancro entre os homens

Idade Com o aumento da idade aumenta a probabilidade de adquirir cancro da mama [42]

História pessoal Existe maior risco de contrair cancro numa mama se existiu um cancro na outra [61]

História familiar O risco de ter cancro aumenta caso existam casos de cancro da mama na família [30, 48].

Alterações genéticas Alterações em certos genes tais como BRCA1, BRCA2 aumentam o risco de cancro [39]

Alterações hormonais Constituem factores de risco o aparecimento da menarca em idade precoce (primeira menstruação antes dos 12) ou uma entrada tardia na menopausa (após os 55 anos) [61, 62, 72]

Idade da primeira gravidez As mulheres que tiveram uma gravidez tardia ou que nunca tiveram filhos (Nuliparidade), apresentam um risco aumentado [72]

Terapêutica Hormonal de substituição Mulheres que tomam terapêutica hormonal para a menopausa durante 5 ou mais anos após a menopausa (apenas com estrogénios ou estrogénios e progesterona), parecem apresentar um maior risco de desenvolver cancro da mama [72, 81]

Exposição à radioterapia na região peitoral Mulheres que tenham efectuado radioterapia na região peitoral, antes dos 30 anos, apresentam um maior risco de desenvolver cancro da mama [37, 110]

Densidade da mama As mulheres com idade mais avançada que apresentam essencialmente um tecido denso em mais de 60-70% da mama, com pouco tecido adiposo, numa mamografia apresentam um risco 4 a 6 vezes superior de sofrer cancro [11]

Obesidade pós menopausa Uma mulher obesa apresenta uma produção adicional de estrogénio, hormona que está associada a um maior risco de desenvolvimento de cancro da mama [18, 64, 72]

Baixa actividade física Este factor está intimamente ligado ao anterior uma vez que surge como prevenção à obesidade e ao aumento de peso [9, 51]

Raça O cancro ocorre com mais frequência nas mulheres caucasianas, comparativamente a mulheres latinas, asiáticas ou afro-americanas [90]

Muito se fez desde que os cirurgiões começaram a manter os primeiros registos detalhados do cancro da mama, que datam de meados do século XIX. A análise desses documentos, permitiram a criação de estatísticas que indicam que os casos tratados por mastectomia apresentavam uma alta taxa de recorrência nos oito anos seguintes, especialmente quando a glândula ou os gânglios linfáticos foram afectados. Na altura, a remoção da mama e das glândulas era o tratamento comum, por forma a evitar qualquer desenvolvimento do tumor [38, 53, 70].

De acordo com os dados disponibilizados no sítio da *International Agency for Research on Cancer* (IARC) , durante o ano de 2008, foram diagnosticados aproximadamente 12,7 milhões de novos casos e 7,6 milhões de pessoas morreram desta doença em todo o mundo [68]. Anualmente surgem, cerca de, 1,3 milhões de novos casos de cancro da mama, que causam 465 mil mortes [68, 94]. Este é um drama mundial e Portugal não é excepção. Os últimos dados, dizem-nos que

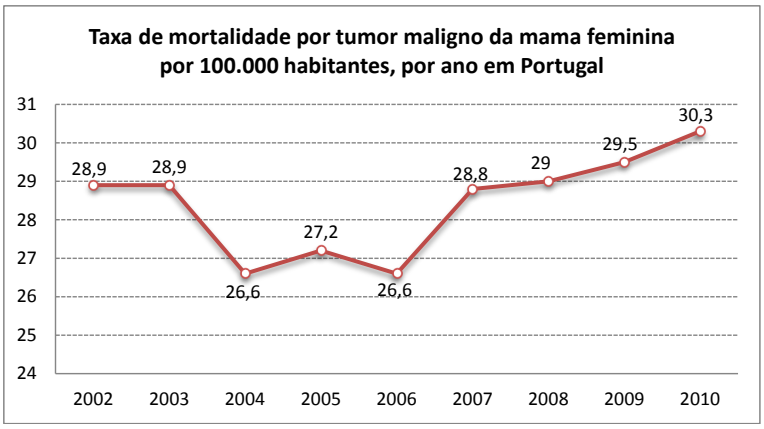


Figura 1.2: Taxa de mortalidade das mulheres portuguesas por cancro da mama. (Fonte: Instituto Nacional de Estatística (INE))

em 2010, morreram em Portugal aproximadamente 25 mil pessoas com cancro, 10 mil eram mulheres. Segundo os dados da *World Health Organisation* (WHO), aproximadamente 1,6 mil mulheres, morreram com cancro da mama [95], cerca de 30 mulheres em cada 100.000 (figura 1.2). Segundo as estatísticas, cerca de 67% das incidências de cancro da mama são tratadas com sucesso mas exigem a sua identificação em estágios iniciais do seu desenvolvimento [68].

Da história pode-se reafirmar a importância da detecção precoce do cancro da mama e muito tem sido feito nesse sentido desde então.

Em 1894, William Roentgen descobriu raios-X, o que permitiu a detecção de várias doenças, entre as quais o cancro da mama. Alguns anos mais tarde, em 1913, Albert Salomon, um patologista de Berlim, produziu imagens de 3000 espécimes de mastectomias, o que é considerado por muitos como o começo da mamografia apesar do seu uso só se ter tornado prática comum anos mais tarde [99]. Comparou raios-X de mamas com os tecidos obtidos pelas mastectomias e observou manchas negras nos centros de carcinomas na mama (microcalcificações). Estabeleceu assim as diferenças entre os tumores cancerígenos e não cancerígenos, providenciou uma quantidade substancial de informação sobre os tumores [119]. Entre 1930 e 1950 houve melhorias significativas na detecção e tratamento do cancro. O grande passo foi dado por Stafford L. Warren (Rochester Memorial Hospital, New York) quando desenvolveu um sistema estereoscópico para identificação de tumores, invenção da mamografia. Além disso, os médicos começaram a classificar o estágio e progressão do cancro. Raul Leborgne (Uruguai), em 1949, descobriu a importância de um melhor posicionamento e compressão da mama para a identificação de tumores. Esta compressão permitia uma melhoria significativa da qualidade da imagem obtida por raios-X, aumentando assim a eficácia

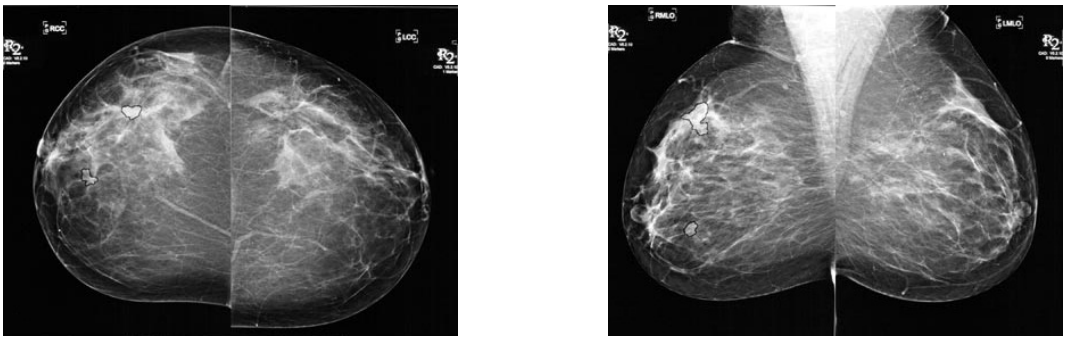
na detecção e diagnóstico do cancro da mama uma vez que se tornou possível a identificação de microcalcificações em fases precoces da doença. O auto-exame é recomendado nos anos 1940-50. Esta é uma das melhores formas de detecção de cancro da mama em fases precoces a par dos exames regulares através de mamografias, e recomendado ainda nos dias de hoje. Em 1960, o Dr. Robert Egan (Houston) recorre a um filme industrial de alta resolução para realizar a mamografia, com detalhes de imagem melhorada. Como resultado do primeiro estudo, 1963, de triagem aleatório conduzido pelo Plano de Seguro de Saúde de Nova York, os autores descobriram que o uso em 5 anos da mamografia, permitira a redução da taxa de mortalidade do cancro de mama em 30%.

A mamografia, uma das grandes responsáveis por essa evolução, permite a identificação de pequenos nódulos (ou caroço) na mama, mesmo antes que este possa ser sentido ou palpado. Pode também mostrar pequenas partículas de cálcio, designadas de microcalcificações. Quer os nódulos quer as microcalcificações podem ser sinais de cancro. Numa mamografia, se o especialista identificar uma área anormal pode pedir que esta seja repetida. No entanto, e apesar de ser o principal meio de diagnóstico, nos casos em que as dúvidas persistem é necessário efectuar exames complementares de despiste do cancro, tais como a ecografia ou a biopsia da área infectada. Esta última, é a única que apresenta um erro praticamente nulo.

Apesar de essencial, a mamografia, pode não detectar alguns cancros já presentes, os chamados "falsos negativos", pode no entanto detectar alguma coisa que, mais tarde, se verifique não ser um tumor, os denominados por "falsos positivos". Existem outras situações, onde as células cancerígenas podem "viajar" para outros órgãos, através do sistema linfático ou da corrente sanguínea sem serem detectadas. Quando o cancro metastiza, o novo tumor tem o mesmo tipo de células anormais do tumor primário. A evolução da mamografia não parou, e foi desenvolvido *software* de apoio à detecção de áreas suspeitas, onde o especialista é alertado para a existência de áreas potencialmente malignas. Com base na imagem e nas áreas identificadas, o especialista avalia os dados e pode validar a presença de tumor.

Dada a massificação da mamografia para a despistagem do cancro da mama, apenas em 5 a 10% dos exames são identificados tumores e destes, 1% são tumores malignos. A detecção de índicios de tumores através de mamografias constitui um desafio para os especilistas, pelo que as ferramentas de apoio à detecção de tumores se tornaram indispensáveis [94].

A mamografia é constituída por um exame de diagnóstico por imagem, que utiliza uma fonte de raio-X, com a finalidade de estudar o tecido mamário. Este exame permite obter imagens mais claras e detalhadas de qualquer área que pareça suspeita ou anormal. Normalmente são realizadas duas exposições para cada mama, uma perspectiva Crânio Caudal (CC) e outra designada como Médio Lateral Oblíqua (MLO). A segunda é mais eficaz porque permite a visualização de uma maior quantidade do tecido mamário e inclui estruturas do quadrante superior externo e do prolongamento axial. Por sua vez, a CC tem como objectivo incluir todo o material póstero medial [85]. A figura 1.3 mostra um exame completo às duas mamas de uma paciente, contendo as diferentes perspectivas de cada uma. De notar que apenas a mama direita contém massas suspeitas, identificadas no exame através de um contorno a negro para as evidenciar. O contorno é adicionado após a digitalização e análise das imagens obtidas no exame.



(a) Perspectiva Crânio Caudal (b) Perspectiva Médio Lateral Oblíqua

Figura 1.3: Imagens de uma mamografia com presença de massas suspeitas. (Fonte: Medical Health Imaging Hub)

A digitalização deste exame permite extrair as características que são alvo de análise pelo sistema de Detecção Assistida por Computador (CAD). A detecção assistida por computador consiste na avaliação de parâmetros tais como a forma, volume e densidade do tumor, e com base nestes, são localizadas na imagem as zonas potencialmente cancerígenas. Existem ainda autores que consideram que um sistema CAD não pode ser encarado como um auxílio ao diagnóstico, mas apenas como uma ajuda na detecção de lesões [23, 46, 118]. Este conceito de aplicação computacional foi desenvolvido com o único objectivo de colmatar dificuldades inerentes ao Homem quando efectua tarefas repetitivas. Isto é, pequenas distracções na avaliação pode ser fatal para um indivíduo ao não ser detectada uma lesão suspeita de cancro de mama. Num estudo realizado por Freer e Ulissey verificaram que o uso do CAD aumenta a taxa de detecção do tumor [49]. Deste modo, os sistemas CAD existem para serem aplicados como uma primeira

análise às imagens, permitindo ao sistema computacional assinalar as áreas que correspondem aos padrões previamente definidos. Seguidamente, o médico Radiologista, estuda o exame, analisando atentamente toda a extensão da mama, não descurando qualquer pormenor da imagem. À posteriori, o médico vai analisar novamente as regiões assinaladas pelo computador e tomar uma decisão sobre o potencial patológico de cada região assinalada.

1.2 Motivação

O crescente uso de ferramentas informáticas na área da saúde, aplicadas em exames de rastreio, e a rápida evolução dos sistemas de digitalização de exames, com elevada automatização de todo o processo, sob a forma de sistemas CAD, gera uma grande quantidade de informação que deve ser tratada e analisada. A informação recolhida nesses exames origina bases de dados médicas que possuem características únicas que as distinguem das demais e que tornam o processo de DM num desafio.

É possível identificar 3 características em grande parte dos conjuntos de dados de origem biomédica: (i) elevada dimensão, (ii) desequilíbrio entre classes e (iii) sensibilidade versus especificidade.

Do ponto de vista informático estas questões levantam alguns desafios. É necessário usar diversas técnicas que permitem lidar com cada uma destas características por forma a minimizar os problemas que lhes estão associados.

A elevada dimensão da base de dados deriva do extenso número de exames produzidos e do número de características que são analisadas em cada um deles. Por si só, um único exame pode gerar inúmeras regiões de interesse que têm de ser classificadas. A manipulação de grandes conjuntos de dados tem algum impacto na capacidade de processamento e desempenho, como por exemplo, lentidão de processos, falta de memória ou de espaço em disco. É por isso necessário utilizar técnicas de redução do conjunto de dados e de optimização do uso de recursos.

É ainda comum, existir muito mais informação sobre pacientes saudáveis do que pacientes doentes. Este desequilíbrio resulta do pequeno número de pacientes que são efectivamente diagnosticados como portadores de determinada doença face ao enorme número de pacientes que realizam os exames de rastreio. Este é um problema conhecido na comunidade ligada ao DM e que tem impacto na classificação dos doentes, dado que existe pouca informação sobre os pacientes que se pretende efectivamente classificar, ou seja, os pacientes doentes. Daí que

seja necessário utilizar técnicas que ultrapassem este problema, balanceando o conjunto de dados.

Por fim, e dado que se está a lidar com vidas humanas é necessário ter em atenção a relação de compromisso entre sensibilidade e especificidade, que se traduz na correcta identificação dos pacientes efectivamente doentes face à correcta classificação dos pacientes saudáveis.

Surge assim a necessidade de se fazer um estudo sobre a aplicação das diferentes técnicas disponíveis e avaliar o seu impacto na aplicação de determinado modelo sobre o conjunto de dados, determinando as que mais se adequam face aos objectivos propostos.

1.3 Objectivos

O objectivo principal deste trabalho é desenvolver um modelo de DM que permita classificar as áreas potencialmente cancerígenas em cada exame, com base na análise dos vários atributos, automatizando a detecção de zonas potencialmente cancerígenas. Dessa forma, o modelo assiste o radiologista, reduzindo o número de zonas em que este se precisa concentrar. Para que o modelo seja útil, é necessário reduzir o número de falsos positivos (zonas normais, identificadas como tumores) , mas também diminuir o número de falsos negativos (zonas tumorais, que não foram indentificadas). Como existe uma relação inversa entre eles, é necessário encontrar um compromisso. O modelo será desenvolvido a partir de um conjunto de dados disponibilizados pelo KDD Cup 2008 [94]. Estes dados foram disponibilizados para esta competição que decorreu em 2008. Os dados foram recolhidos através de *hardware* e *software* proprietário da Siemens.

1.4 Organização do documento

A tese está dividida em cinco capítulos principais. O presente capítulo faz o enquadramento dos temas abordados nesta tese. São expostos os conceitos fundamentais relativos à temática abordada no estudo, motivação, importância e descrição do problema.

No capítulo 2 são introduzidos alguns conceitos teóricos e abordados outros estudos relacionados. Os estudos mais recentes e adequados à detecção e previsão do tumor, assim como resultados mais relevantes.

O capítulo *3* descreve a metodologia, passo a passo, adoptada na análise dos dados. Refere um conjunto de informações teóricas indispensáveis para a compreensão e avaliação de todo o trabalho desenvolvido, tais como descrição dos dados e o seu tratamento.

No capitulo *4* é apresentado o modelo de detecção e previsão utilizado, a simulação realizada e descreve sucintamente os principais resultados obtidos através da realização deste trabalho.

Por fim, no capítulo *5* são apresentadas as conclusões e apontadas linhas para trabalho futuro.

2

Conceitos fundamentais e trabalho relacionado

Ao longo do trabalho foram surgindo algumas dúvidas e problemas, que carecem de uma base teórica para os compreender e encontrar as melhores soluções. No entanto, a generalidade dos problemas são conhecidos, pelo que existem diversas abordagens baseadas no conhecimento adquirido através da observação e resolução de situações idênticas.

Neste capítulo são apresentados, de forma orientada e não exaustiva, os conceitos essenciais para o contexto desta tese. Além disso, são descritos os métodos e técnicas utilizadas, assim como as justificações para as opções tomadas. São, também apresentados os algoritmos com maior relevância na resolução dos problemas identificados.

2.1 *Data Mining*

"*Data Mining*, é a extracção de informação implícita aos dados, previamente desconhecida, e potencialmente útil" [126]. A ideia subjacente ao processo de DM é criar um modelo informático, por exemplo, aplicação, que permita analisar e extrair padrões dos dados de forma automática, transformando-os numa estrutura compreensível e utilizável nos passos seguintes do processo de KDD. Estes

padrões são então utilizados na detecção de dependências entre os dados, detecção de casos particulares, explicação, compreensão, predição ou classificação de novos dados [126].

Este termo é muitas vezes usado como sinónimo, ou simplesmente como um passo essencial no processo de KDD, antecedido por uma fase designada de pré-processamento, onde é efectuada a limpeza e tratamento dos dados, e procedido por uma etapa de pós-processamento e avaliação dos resultados [55, 126]. Aproveitando esta última definição, pode-se assim afirmar que, DM é o conjunto de técnicas e estratégias usadas na extração dos padrões dos dados.

2.2 Dimensão dos dados

Um dos problemas abordado em diferentes etapas do processo de DM é a dimensão dos dados. No capítulo 3, em que são explorados os dados, surgem os primeiros problemas. Esta questão desperta dois pontos essenciais, fraco desempenho dos algoritmos devido à morosidade de execução dos algoritmos e a falta de recursos que impede essa mesma execução. Entende-se como desempenho dos algoritmos, a capacidade destes serem executados e produzirem resultados em tempo considerado útil. Esse desempenho tende a diminuir proporcionalmente com o tamanho dos dados, uma vez que é necessário mais processamento para produzir um resultado da aplicação de um mesmo algoritmo. Deste modo, este problema acaba por ser transversal às várias fases deste trabalho, com especial ênfase para a fase de modelação, ver capítulo 4.

A redução da dimensão dos dados pode ser feita essencialmente de três maneiras: (i) redução do número de atributos, vulgarmente designado como redução de colunas, (ii) redução do número de casos ou amostras, vulgarmente designado como redução de linhas, ou (iii) uma combinação destas duas técnicas. Algumas destas técnicas foram usadas e descritas no capítulo 3. Para além da redução da dimensão dos dados existe ainda uma outra possibilidade que passa por encontrar novas formas de representação dos dados que permitam treinar e classificar cada uma das amostras ou pacientes.

2.2.1 Selecção de atributos

Kira e Kendel, em [74], afirmam "O problema, selecção de atributos, é escolher um pequeno subconjunto de características *ideais* consideradas necessárias e suficientes para descrever o objectivo alvo." Dada a dimensão do conjunto de dados o objectivo é reduzir o número de atributos a um mínimo, o estritamente necessário, sem que isso interfira negativamente no processo de classificação e ao contrário, possa até potenciar os resultados evitando o *overfitting* (sobre-ajustamento) [113]. Para tal, é necessário que os atributos escolhidos contenham informação relevante e não redundante [73, 74, 100].

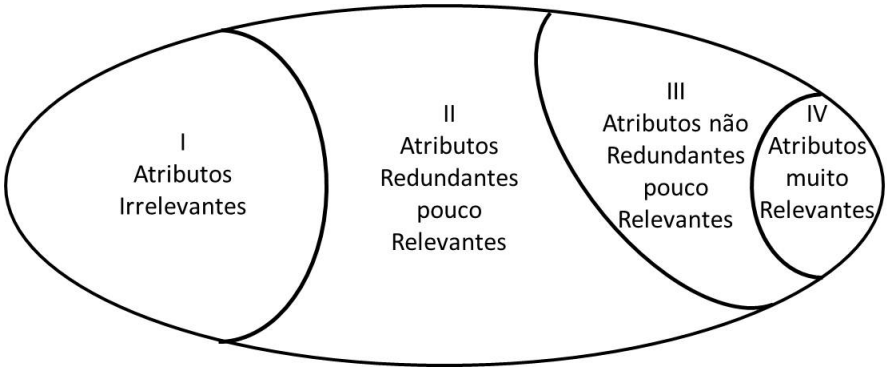


Figura 2.1: Relevância e redundância dos atributos.(Adaptado de Yu e Liu [129])

A figura 2.1 mostra a divisão dos atributos segundo a sua relevância e redundância. Atributos do tipo I e II, são irrelevantes ou pouco relevantes e redundantes, e não têm interesse para a classificação pelo que podem ser removidos do conjunto original. Na prática, um bom subconjunto de atributos é composto por atributos muito correlacionados, do tipo III e IV, com o atributo classificador da amostra, mas em que os atributos são pouco correlacionados entre si [54, 129]. Assim esta tarefa pode ser dividida em duas partes, remoção dos atributos correlacionados entre si e determinação do peso de cada um dos atributos na classificação de uma amostra.

2.2.1.1 Exclusão de atributos redundantes

A forma de reduzir o número de atributos de um conjunto de dados, mais usual, é obter o peso que cada atributo face ao atributo classificador, ou seja, é observada a relevância de cada atributo. No entanto, os conjuntos de dados que representam problemas concretos, por exemplo, bases de dados de imagens médicas, contêm

muitos atributos redundantes dado que alguns atributos são incluídos no conjunto original dada a impossibilidade de se saber previamente quais os atributos mais relevantes [73]. A remoção destes atributos é tanto ou mais importante pela existência destes e de informação adicional irrelevante, afectando o desempenho e precisão do algoritmo.

No entanto, os algoritmos que ordenam os atributos através do seu peso face ao atributo classificador não removem os atributos redundantes, uma vez que têm a mesma importância [34]. Este facto é explicado pela forte correlação entre si. Isto acontece somente onde os atributos, apesar de redundantes, são considerados relevantes para a classificação. Neste caso, um dos atributos, mas não os dois, pode ser excluído. O atributo escolhido contém toda a informação que os dois representam no conjunto original.

Um possível algoritmo é descrito por Hall [54] e denomina-se por *Correlation-based feature selection* (CFS). Este método faz uma ordenação dos atributos pelo seu peso e por essa mesma ordem, fixa cada um dos atributos e testa o grau de semelhança, entre o atributo e todos os restantes. Caso exista uma correlação elevada o atributo redundante é retirado. A forte correlação entre os atributos é um forte indicador usado na identificação dos atributos redundantes. Uma forma simplificada deste processo é a construção da matriz de correlação dos atributos (ver tabela 3.3). Esta matriz devolve a correlação entre todos os atributos que compõem o conjunto de dados. A remoção dos atributos redundantes é então feita, recorrendo à matriz, verificando os pares cujo valor de correlação é mais elevado e retirando um dos atributos do conjunto (ver subcapítulo 3.2.1).

2.2.1.2 Determinação dos atributos relevantes

Uma outra alternativa é remover os atributos irrelevantes. Existem vários métodos para o efeito, mas na sua generalidade o conceito é efectuar um *ranking* dos atributos tendo em conta o seu peso na classificação de uma amostra. Os algoritmos usados neste caso são, na maior parte, baseados na correlação ou no ganho de informação de cada um dos atributos (ver subcapítulo 3.2.2).

2.2.2 Balanceamento dos dados

Por sua vez, o balanceamento dos dados constitui outro problema pois interfere na qualidade da classificação. Nestes casos, os algoritmos de aprendizagem tendem a especializar-se na classificação da classe maioritária e ignoram os casos da

classe minoritária [26]. Três técnicas, possíveis, são: (i) *Undersampling*, (ii) *Oversampling* e (iii) *Synthetic Minority Oversampling Technique* (SMOTE). A adopção de diferentes técnicas de balanceamento têm impactos diferentes, quando usadas na ordenação dos atributos quanto, à sua importância e na aprendizagem, situação identificada no capítulo 3 e reforçada no capítulo 4.

A primeira técnica consiste no balanceamento através da redução das amostras da classe maioritária, escolhendo aleatoriamente amostras desta classe até perfazer o número de amostras da classe minoritária. Tem como desvantagem a possibilidade de remover casos da classe maioritária, que podem ser importantes para a classificação desta classe [60]. A segunda técnica consiste na replicação aleatória das amostras da classe minoritária até perfazer o número de amostras da classe maioritária. Tem como principal desvantagem a hipótese de se verificar *overfitting* do algoritmo. O algoritmo tende a especializar-se na classificação dos casos replicados, e diminui a precisão na classificação das amostras que não tem conhecimento prévio [60].

Segundo Chawla *et al.* [25], o erro na classificação de um caso da classe minoritária é frequente num conjunto de dados composto maioritariamente por casos de outra classe, também o custo associado é maior dado que se pretende que estes casos sejam correctamente classificados. A técnica de *Undersampling* é muitas vezes referida e proposta como uma possível solução que aumenta a sensibilidade da classe minoritária. Estes propõem uma solução que combina as técnicas de *Undersampling* da classe maioritária e *Oversampling* da classe minoritária, surge assim a técnica denominada por SMOTE. Este método permite aumentar o número de amostras do conjunto minoritário, gerando amostras sintetizadas a partir de um determinado número de amostras vizinhas de cada uma das amostras [25, 60, 123]. As amostras vizinhas consideradas para a sintetização da nova amostra também pertencem ao conjunto minoritário. A nova amostra é gerada tendo em conta a diferença entre a amostra e as amostras vizinhas, isto é, dado por:

$$S_i = X_i + \gamma\,(R_i - X_i)\,,$$

tal que **S**, **R** e **X** representam os conjuntos das novas amostras, das amostras vizinhas e das amostras da classe minoritária, respectivamente; γ define uma variável aleatória no intervalo $[0;1]$ e, independente das outras variáveis. O índice indica as variáveis da amostra [25]. O conjunto de treino fica assim com todas as amostras do conjunto minoritário, as n amostras geradas e as amostras escolhidas aleatoriamente do conjunto maioritário. A proporcionalidade entre os casos

de ambas as classes é configurável.

2.3 Conjuntos de treino e teste

Um dos passos do processo de DM é a realização de testes para validação dos resultados, por aplicação dos algoritmos no conjunto de dados. Relembra-se que o objectivo é criar um modelo usando técnicas de DM para detectar tumores em exames de rastreio, e esse modelo deve corresponder a alguns parâmetros como, detectar todos os pacientes com cancro e ter um número reduzido de Falsos Positivos (FP). Como tal a validação do modelo, incide na realização de testes e na comparação dos resultados obtidos com os resultados esperados. Pretende-se também que o modelo possa ser aplicado a novos casos, é desejável que não se especialize somente na classificação dos casos conhecidos, mas possa também classificar os novos casos de forma igualmente eficiente ou se possível com melhor desempenho ainda. Ou seja, trata-se de classificar os novos casos correctamente dado que os antigos foram usados no treino e previamente classificados [126]. Idealmente, seriam disponibilizados três conjuntos de dados. Um para uso na aprendizagem designado por conjunto de treino, um para optimização e validação dos algoritmos, designado por conjunto de validação, e outro para testes designado por conjunto de testes. Mas nem sempre é possível, visto que os dados disponíveis são escassos pelo que se tem que encontrar uma abordagem que permita simular estas condições. Aqui são descritas, de forma sucinta, três das técnicas mais usadas nestas situações e exemplificativas dos métodos e problemas encontrados neste processo.

2.3.1 *Hold-out*

O *hold-out* é uma das técnicas mais simples e geralmente aceite nos processos de DM. Consiste na divisão do conjunto de dados em dois conjuntos, um para treino e outro para teste. Existem alguns autores, por exemplo Mitchell *et al.* [91] e Witten*et al.* [126], que defendem uma subdivisão do conjunto de treino em dois conjuntos, um de treino e outro de validação e ajuste do algoritmo, ou seja, este subconjunto é usado na fase de treino. Esta situação é defendida em situações específicas, como por exemplo, a poda de árvores de decisão. O conjunto de teste serve então para validar os resultados numa fase posterior. Em qualquer uma das formas, os conjuntos de treino e teste devem ser disjuntos por forma a avaliar o seu desempenho em casos potencialmente distintos daqueles sob os quais se

efectua o treino. Essa disjunção é garantida pela escolha aleatória dos casos que fazem parte de cada um dos conjuntos. A percentagem que deve constar em cada um dos conjuntos também pode ser parametrizável tendo em conta a sua dimensão [79]. A questão que predomina, é saber qual a melhor divisão, mas esse é o dilema desta técnica. Na perspectiva da aprendizagem, na generalidade dos algoritmos de aprendizagem, quanto maior for o conjunto de treino melhor será o classificador, até um determinado tamanho, a partir do qual o classificador piora. E, quanto maior for o conjunto de testes maior será a estimativa de erro [126]. Isto é verdade desde que seja mantida a representatividade dos dados em ambos os conjuntos, ou seja,verifica-se uma relação de compromisso entre uma melhor aprendizagem, menor estimativa de erro e maior precisão. Na prática, uma distribuição geralmente aceite e usada, é de $1/3$ dos dados para teste e os restantes $2/3$ para treino, conforme figura 2.2 [75, 126]. Como a distribuição dos casos pelos dois conjuntos é feita aleatoriamente, é possível que alguma das classes esteja pouco ou nada representada num dos conjuntos. Deste modo, a aprendizagem pode viciar o resultado da classificação. Uma forma de evitar esta situação é criar dois conjuntos onde estão representadas todas classes e é preservada a proporcionalidade entre elas, por forma a que haja uma maior representatividade [126]. No entanto, não é possível assegurar que os conjuntos sejam efectivamente representativos. O problema é minimizado repetindo o processo de *holdout* diversas vezes de forma aleatória, obtendo vários pares de conjuntos de treino e teste, aos quais são aplicados os algoritmos de aprendizagem [126]. O erro global é obtido pela média ponderada dos erros dos conjuntos de treino e teste. Quantos mais testes forem realizados, menor será o erro global. Witten *et al.* [126] chama a este processo de *Repeated Holdout*.

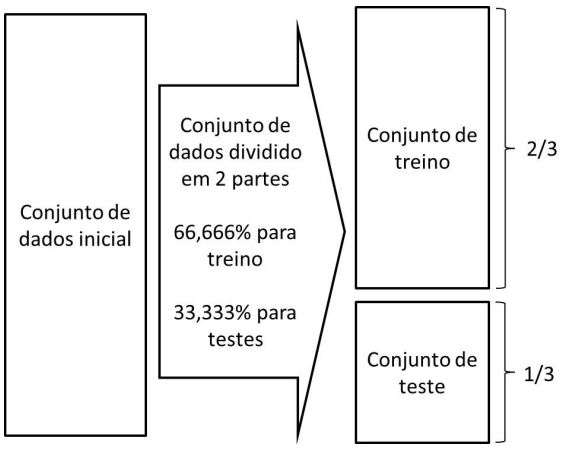


Figura 2.2: Descrição da técnica *Holdout*

2.3.2 *k-fold Cross Validation*

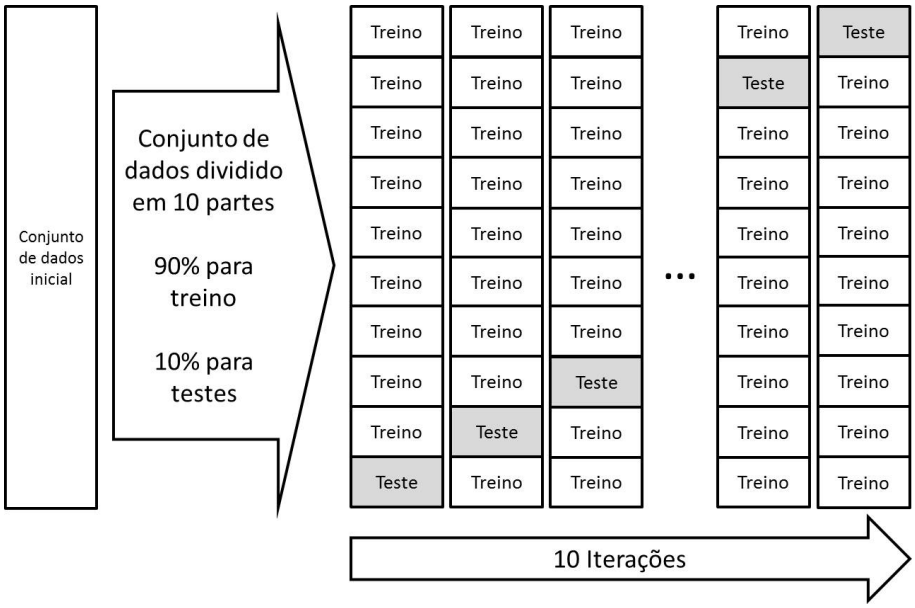


Figura 2.3: Descrição da técnica *10-Fold Cross Validation*, usada na divisão dos conjuntos em treino e teste

Como foi sugerido no subcapítulo 2.3.1, um dos problemas do *Holdout* é a não garantia de representatividade do conjunto de dados, apesar dos esforços que são feitos para obter dois conjuntos representativos do conjunto original. Não quer isto dizer, que não se consigam obter conjuntos representativos mas, não é garantido que a representatividade seja conseguida. Aproveitando, a técnica de *holdout* e a noção de repetição do processo para obter a representatividade dos dados, foi desenvolvida uma técnica conhecida por *k-Fold Cross Validation*. Uma das implementações mais conhecidas e geralmente aceite é o *10-Fold Cross Validation* [126]. Esta técnica consiste na divisão do conjunto de dados em dez partes iguais, mantendo a proporcionalidade entre os casos de cada uma das classes em cada uma das partes. São feitas dez iterações, sendo que em cada iteração será usada uma parte diferente do conjunto de dados para teste. Para uma melhor compreensão pode-se observar a figura 2.3. Cada uma das iterações é equivalente ao *holdout*, mas difere na divisão dos conjuntos, sendo que neste caso o conjunto de treino é constituido por 90% dos casos e o conjunto de teste é composto pelos restantes 10%. Todo o conjunto é testado, pelo que é necessário a realização de dez aprendizagens e correspondentes testes. A aplicação desta técnica pode não ser suficiente para garantir uma boa estimativa de erro, pelo que a repetição desta técnica por dez vezes, é um procedimento habitual [126]. Isto implica a realização de 100 testes, o que pode não ser viável dado o tempo envolvido no processo

e a escassez de recursos para o fazer. Num contexto em que o conjunto de dados é relativamente extenso, o processo pode ser moroso ou até irrealizável.

2.3.3 *Bootstrap*

Esta é outra técnica usada na criação de conjuntos de treino e teste. Difere das restantes abordagens por introduzir o conceito de escolha aleatória de amostras com reposição. Nas técnicas anteriores as amostras eram colocadas num único conjunto sem existir a possibilidade de repetição. Neste caso, e para um conjunto de dados com n amostras, são escolhidas n amostras que podem ser repetidas, o que implica a existência de amostras que não farão parte do conjunto de dados. Essas amostras farão parte do conjunto de teste [126]. Segundo dados estatísticos, uma distribuição normal das probabilidades origina um conjunto de treino com cerca de 0,632 amostras do conjunto original [55, 86, 126]. A aplicação desta técnica pode originar um *overfitting* do algoritmo dada a repetição das mesmas amostras. Além do mais também se deve tentar preservar a proporcionalidade de casos das várias classes no conjunto final, porque a escolha aleatória de amostras pode originar um conjunto de dados representado por apenas uma das classes ou com poucos exemplos das restantes. Num método puramente aleatório, nada nos garante que as amostras de uma determinada classe sejam escolhidas, porque a probabilidade de o ser é muito menor no caso do conjunto de dados alvo. Relembra-se que o conjunto é muito desequilibrado e contém poucas amostras da classe minoritária. Também neste caso, se deve proceder à repetição desta técnica uma série de vezes para se conseguir obter uma maior representatividade do conjunto de dados. Esta técnica é sobretudo adequada a pequenos conjuntos de dados.

2.4 Algoritmos de aprendizagem

Como veremos no capítulo 3, o processo de DM é composto por várias fases entre as quais se destaca a modelação. É nesta fase, que são aplicados algoritmos ao conjunto de dados e daí se possam identificar alguns padrões que ajudam na classificação. Pode-se assim dizer, que os algoritmos aprendem os padrões dos dados desde que correctamente parametrizados, criam uma função que então aplicam aos novos dados, tentando aproximar-se o mais possível da classificação correcta destes.

Os algoritmos podem ser classificados de diversas formas, diferentes autores agrupam os algoritmos de acordo com as características comuns entre os algoritmos que pretendem evidenciar. A mais básica das classificações tem em conta o modo como é realizada a aprendizagem, supervisionada ou não supervisionada. Na primeira situação, a aprendizagem é feita a partir de conhecimento prévio da classificação dos casos no conjunto de treino do algoritmo, classificação essa que posteriormente é utilizada no processo de aprendizagem. Situação contrária, a aprendizagem é efectuada tentando inferir correlações, ou agrupando os dados de acordo com algumas características comuns desconhecendo a classificação de cada um dos casos.

Os algoritmos podem ainda ser divididos quanto à forma de aprendizagem. Neste trabalho são considerados vários tipos de aprendizagem dentro do *Inductive Learning* (IL) entre os quais, árvores de decisão, regras de indução, modelos lineares, *Instance-Based Learning* (IBL) e *Statistical Learning* (SL). A caracterização de cada um destes tipos é um pouco difusa, pelo que podem surgir algumas diferenças entre autores de acordo com a interpretação de cada um, no entanto, verificam-se algumas características distintas entre os algoritmos agrupados em cada um destes conjuntos. O agrupamento destes algoritmos simplifica a sua descrição, uma vez que evidencia alguns aspectos essenciais e comuns a cada um deles, só descrevendo implementações específicas quando assim se justifica. Esta descrição é apresentada nos subcapítulos seguintes.

Neste processo deve-se ainda ter em atenção, além de outros aspectos, a capacidade de aprendizagem do algoritmo. O efeito de *overfitting* ou *overlearning* é um dos problemas que se pode observar. Situação em que o modelo gerado adapta-se bastante bem ao conjunto de treinos, utilizado na aprendizagem, no entanto apresenta fracos resultados nos casos de teste. Por outro lado, existe o *underfitting* ou *underlearning* em que o algoritmo generaliza muito e assim apresenta uma fraca capacidade para classificar correctamente os novos casos, tratando-os todos como pertencentes a uma determinada classe. Em suma, um bom algoritmo é aquele que não se especializa nem generaliza em demasia, estabelece uma relação de compromisso [19].

As primeiras famílias de algoritmos aqui descritas, árvores de decisão, regras de indução (*Rule-Based Learning* (RBL)) e modelos lineares, representam aquilo que se chama de inferência. Neste caso, os algoritmos tentam construir um qualquer conjunto de regras que permitem classificar um novo caso a partir dos atributos que constituem o conjunto de dados. Pode-se dizer que é o caso típico de qualquer algoritmo indutivo, mas neste caso dá-se ênfase ao termo *atributo*. Dentro

desta família, existe uma separação dos algoritmos em árvores de decisão ou regras de indução. Por exemplo, os nós das árvores podem ser encarados como um conjunto de regras, logo podem também fazer parte do RBL. Ou seja, as árvores podem ser encaradas como uma implementação particular de RBL. O contrário também é verdade mas a complexidade envolvida pode ser um pouco mais elevada. Segue-se a descrição de cada uma das famílias consideradas.

2.4.1 Árvores de decisão

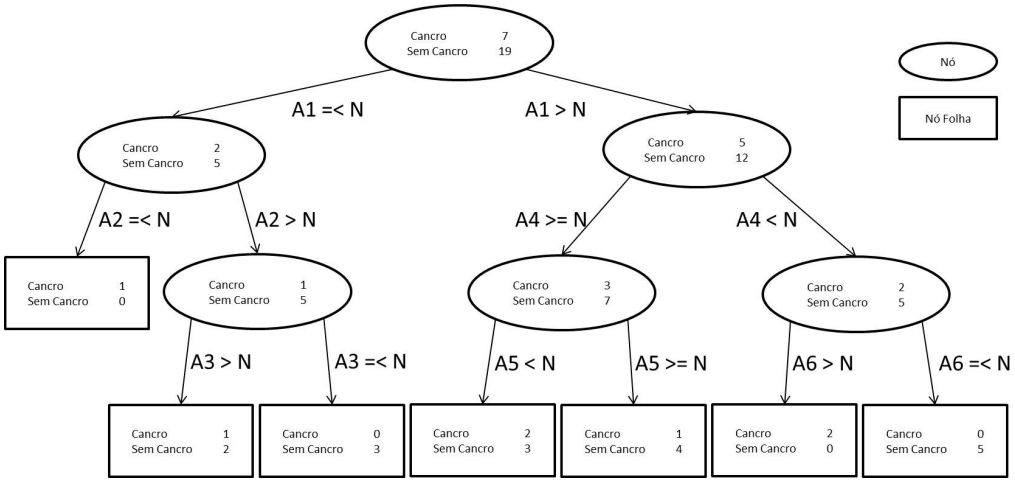


Figura 2.4: Exemplo de árvore de decisão

As árvores de decisão são uma família de algoritmos que se baseiam na classificação de uma variável, construindo para o efeito uma árvore a partir de regras deduzidas das variáveis preditivas [84]. O algoritmo consiste na construção de uma árvore tendo por base a partição recursiva dos casos pertencentes ao conjunto de treino, e cuja classificação e valores dos atributos são conhecidos. Cada partição é designada por nó. A árvore é construída até se atingir um nó folha onde, de acordo com ramo da árvore e o caminho percorrido, se classificam as amostras que venham a fazer parte desse nó, ver figura 2.4. De acordo com o número de casos conhecidos que façam parte desse nó, pode-se estipular a probabilidade de um determinado caso pertencer a uma dada classe. A grande dificuldade é escolher a melhor divisão em cada nó por forma a criar uma árvore que classifica correctamente e o mais eficiente possível cada nova amostra. O processo é simples se considerarmos que cada nó de uma árvore é o nó raiz de uma subárvore da árvore principal. Começa-se por escolher em cada nó o atributo mais significativo, com maior ganho de informação, ou seja, o atributo que separa o maior número de casos de classes diferentes. O processo repete-se até atingir o

critério de homogeneidade ou até que sejam satisfeitos outros critérios de paragem impostos à árvore. Assim, este algoritmo é um exemplo de um algoritmo de partição binária recursiva, pois é aplicado recursivamente a cada um dos subconjuntos gerados, até que não seja possível (ou necessário) efectuar mais nenhuma partição. A partição é gerada quer através da comparação do valor de um atributo com uma constante, comparação dos valores de vários atributos ou através de uma função que seja aplicada aos valores dos atributos [126]. A escolha da melhor partição depende do tipo de árvore, mas assentam sobretudo na redução da entropia. Uma desvantagem deste método é a especialização de uma árvore na detecção de casos similares aos presentes no conjunto de treino. Neste caso a generalização é mais difícil o que pode conduzir a uma incorrecta classificação dos novos casos. Existem algumas técnicas de "podar" a árvore, eliminando alguns nós e ramos, generalizando a aprendizagem. Algumas implementações que se destacam são o *CART* [17], o *ID3* [103] e o *C4.5* [104]. O *C4.5* é um exemplo da fronteira difusa entre as árvores de decisão e o RBL. O *C4.5* é uma especialização do método *ID3* que após a construção da árvore a partir do conjunto de treino, converte cada um dos caminhos do nó raiz até uma das folhas num conjunto de regras equivalente. Esse conjunto de regras é então generalizado removendo as regras que optimizam a precisão estimada do algoritmo e ordena cada conjunto de regras pela sua precisão. Essas regras são então aplicadas aos novos casos pela ordem definida [91].

2.4.2 Regras de indução

Como o próprio nome indica, os algoritmos desta família analisam o conjunto de treino e tentam encontrar um conjunto de regras que permitem classificar com o maior grau de certeza os novos casos. As regras podem-se decompor sob a forma de $\{\text{Condicao}\} \Rightarrow Y$, em que Y ocorre quando uma determinada condição é satisfeita. Isto permite classificar uma nova ocorrência que satisfaça uma condição previamente conhecida com um determinado grau de confiança, essencial na construção das regras. Uma das implementações possíveis é criar regras com um grau de confiança superior a X. Esta aprendizagem é também conhecida por *Association Rule Learning* (ARL). Algumas implementações conhecidas deste tipo de aprendizagem são os algoritmos Apriori [1] e Eclat [10]. Outro algoritmo conhecido é o *1-Rule* [65]. Este algoritmo destaca-se por usar apenas um atributo na classificação de um novo caso. Cada atributo é dividido em diversos intervalos nos quais são contabilizados os casos de cada classe. O atributo com menor erro

na classificação é escolhido para servir de base à classificação dos novos casos.

2.4.3 Modelos Lineares - Regressão Logística

A classificação de um modelo pode ser obtida através de modelos de regressão linear [86]. Os modelos de regressão são um dos mais clássicos algoritmos, mas ainda assim continua a ser bastante importante [59, 86]. Estes procuram obter uma função resultante da combinação linear dos atributos que permita obter um valor próximo do esperado. Para que tal seja possível, todos os atributos devem ser numéricos [86, 126]. No caso da classificação, é necessário estipular um limite a partir do qual os valores obtidos por regressão sejam considerados como pertencentes a uma das classes. O modelo linear mais simples resulta da adição de todos os atributos,

$$Y = w_0 + w_1 X_1 + \cdots + w_p X_p$$

tal que **Y** é a classe, **w** o peso de cada um dos atributos e **X** o valor do atributo [59, 86, 126]. Um dos passos é calcular os pesos a partir de informação existente em cada um dos atributos, com erro mínimo, entre as classificações, real e previsão [86, 126].

O *Logistic Regression* (LR) é uma das extensões deste modelo, tem a capacidade de classificar uma amostra numa de duas classes [7]. *Linear Discriminant Analysis* (LDA) e *Quadratic Discriminant Analysis* (QDA) são outros modelos lineares. Estes diferem do LR por assumirem que os atributos têm uma distribuição normal [55]. Tentam encontrar uma ou mais funções lineares e quadráticas, respectivamente, para discriminar os elementos pertencentes a cada uma das classes [55]. Outra implementação é o *Multinomial Log-linear Models* (MLM), que no pacote `nnet` do *R* está implementado sobre *Artificial Neural Network* (ANN), entre outros aspectos generaliza o processo de classificação para várias classes em vez de duas [89]. O *Multivariate Adaptive Regression Splines* (MARS) [52] pode ser visto como uma generalização dos modelos lineares (*stepwise linear regression*) ou uma modificação ao *Classification and Regression Trees* (CART) dadas as suas similaridades, e é adequado para lidar com problemas multi-dimensionais [59].

2.4.4 Aprendizagem baseada em instâncias

Este caso especial de indução tem em conta as instâncias do conjunto de dados, e não um atributo ou conjunto de atributos em especial. A classificação de novas instâncias dependem do grau de verosimilhança entre os casos conhecidos e

os novos casos, esta é a característica que melhor distingue os algoritmos desta família dos restantes. Fazendo uma analogia com os outros algoritmos, onde a aprendizagem é feita tentando inferir regras que permitem classificar novas amostras, os algoritmos desta família guardam as instâncias pertencentes ao conjunto de treino e classificam as novas instâncias comparando-as com as anteriores [91, 126]. Têm como principais desvantagens o espaço necessário de armazenamento das instâncias que compõem o conjunto de treino, e servem de base para comparação com as novas instâncias; e, a passagem do custo de classificação para a fase de classificação, ao contrário da maioria dos algoritmos que fazem a maior parte do processamento em tempo de aprendizagem [91]. A comparação faz-se nesta fase, caso a caso, ao contrário de algoritmos de outras famílias, cuja generalização depende da nova instância e uma qualquer função inferida a partir do conjunto de treino. Um dos algoritmos mais conhecidos do IBL é o *K-nearest neighbors* (KNN). O *Support Vector Machines* (SVM) é outro dos algoritmos desta família. Também tem por base as instâncias do conjunto de treino, mas são escolhidas apenas algumas instâncias que representam as fronteiras entre as instâncias de cada classe. As novas instâncias são posteriormente comparadas com as armazenadas afim de verificar de que lado da fronteira é que se encontram. Reduz-se assim o número de instâncias que é preciso guardar o que implica uma redução dos recursos, em termos de espaço, necessários para a aprendizagem e teste deste algoritmo. Neste subcapítulo é feita uma introdução aos algoritmos considerados mais importantes desta família com especial destaque ao SVM, dado o sucesso da sua aplicação nos mais variados domínios. Outros algoritmos que também se podem enquadrar nesta classificação são o ANN e o *Self-organizing Map* (SOM).

2.4.4.1 *K-Nearest Neighbor*

O KNN é um dos algoritmos mais simples desta família. São guardadas todas as instâncias do conjunto de treino, posteriormente servem para comparação com as novas instâncias. A nova instância é classificada por maioria de votos dos seus vizinhos tendo em conta as k instâncias mais próximas da nova instância. As instâncias são obtidas por uma função de cálculo da distância entre a nova instância e cada uma das instâncias presentes no conjunto de treino. A distância é calculada de diversas formas entre as quais o cálculo da distância euclidiana entre duas instâncias. Trata-se de uma das formas de cálculo de distância mais usadas no KNN, a desvantagem deste método, deve-se ao igual peso que é dado a todos os atributos no processo de classificação. Como veremos mais tarde, nos

subcapítulos 3.2.2 e 2.2.1.2, existem atributos com maior relevância e outros absolutamente irrelevantes para a classificação de uma instância, e nem sempre é fácil determinar os atributos com maior relevância no conjunto de treino [126]. Tratar todos por igual pode levar a algumas incorrecções na classificação atribuída a uma nova amostra. Como resolução para este problema foram propostas diversas implementações do KNN onde se tenta minimizar este problema. Uma das formas é reduzir a dimensão dos atributos previamente à aplicação do KNN, outra é atribuir pesos a cada um dos atributos. Existe ainda algumas outras variantes como por exemplo, a atribuição de pesos às instâncias de acordo com a proximidade à nova instância [91].

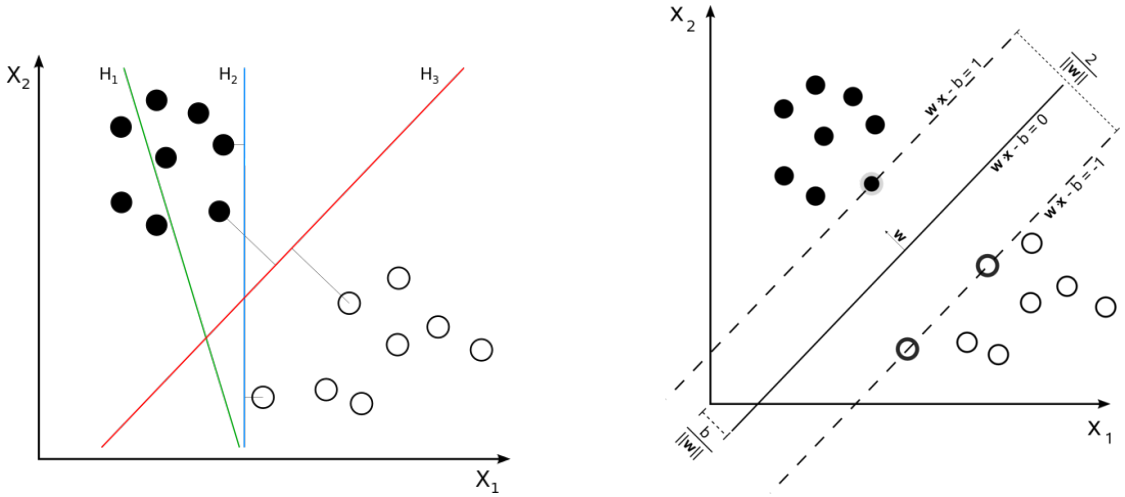
2.4.4.2 *Support Vector Machines*

O SVM é um dos algoritmos mais usados da actualidade, com aplicação nas mais diversas áreas, entre as quais a bioinformática, categorização de texto e hipertexto, classificação de imagens, em ciências médicas, marketing, entre outras. Trata-se de um algoritmo supervisionado e pertence ao grupo designado por *kernel methods*.

Inicialmente proposto por *Cortes* e *Vapnick* [32], este algoritmo tem sofrido várias evoluções ao longo dos anos até se ter tornado num dos algoritmos mais usados. O sucesso deste algoritmo é baseado na elegância matemática do método, no profundo trabalho teórico disponível e no sucesso do algoritmo em vários cenários [86].

A ideia é simples, considerando o exemplo mais básico de classificação de um caso em uma de duas classes, o objectivo consiste em mapear os casos para um espaço multidimensional, conhecido por espaço das características, e encontrar um hiperplano que separa os casos de cada uma das classes [32]. Podem existir vários hiperplanos que separam os casos das duas classes, ver figura 2.5(a), pelo que temos de encontrar o melhor. Assim, o hiperplano deve ter uma margem máxima entre os casos pertencentes a cada uma das classes para assim minimizar o erro de generalização [97]. Na figura 2.5(a) verifica-se que apesar das rectas *H2* e *H3* dividirem correctamente os casos de ambas as classes, a recta *H3* é a que melhor separa os dois conjuntos dado que a distância entre os casos marginais e o hiperplano são maiores. O hiperplano encontrado é equidistante aos casos marginais, casos que se encontram mais próximos do hiperplano, das duas classes. A figura 2.5(b) mostra as propriedades do hiperplano e das margens que separam os casos das duas classes. A margem, é o hiperplano paralelo ao hiperplano que

separa os dois conjuntos e que é definido pelos casos marginais. Os casos que definem as margens do hiperplano são também designados por vectores de suporte. Esses vectores definem a fronteira espacial da classe a que pertencem [41, 55, 126]. Os novos casos são classificados de acordo com o lado do hiperplano em que são mapeados.



(a) Escolha do melhor hiperplano (b) Definição do melhor hiperplano

Figura 2.5: SVM - Escolha do Hiperplano Óptimo.
(Fonte: http://en.wikipedia.org)

A figura 2.5 descreve o caso mais simples de separação linear dos casos de ambas as classes, no entanto existem SVM's que possuem *kernel's* com propriedades diferentes e permitem uma separação não linear dos casos [55, 126]. Entende-se como *Kernel*, o processo de transformação das características de um conjunto de dados para um outro espaço multidimensional onde as características podem ser separadas linearmente [55, 126]. Mas este mapeamento é feito implicitamente, na verdade não é realmente necessário efectuá-lo para o novo espaço, designa-se esse processo por *kernel trick* [115]. O mapeamento é evitado e é usada uma função que usa o produto vectorial para "simular" o novo espaço de características [114, 115].

Ao usar o SVM somos confrontados com alguns problemas entre os quais: (i) selecção dos atributos do conjunto de treino, (ii) escolha do *Kernel* e (iii) escolha dos parâmetros do *Kernel*. Estes problemas são cruciais para o bom desempenho do algoritmo uma vez que se influenciam mutuamente [67]. O problema relacionado com a selecção de atributos já foi abordado no subcapítulo 2.2.1. Neste caso pretende-se um conjunto de atributos o mais pequeno possível sem perder a representatividade do conjunto, por forma a obter um bom modelo de previsão e

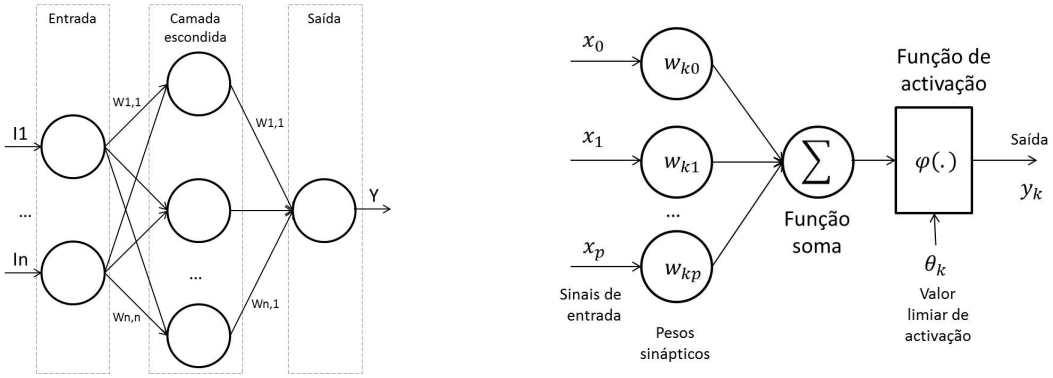
menos computacionalmente intensivo [67]. Quanto há escolha do *Kernel* existem várias possibilidades e não é possível identificar à partida qual o mais adequado para cada um dos problemas. Mas escolhido um *Kernel*, devem-se optimizar esses parâmetros. No entanto, existem casos em que o conjunto de treino não é linearmente separável pelo que pode-se optar por um outro *kernel*, como por exemplo o polinomial, *Radial Basis Function* (RBF) e sigmoidal [55, 126]. Nestes casos, também os parâmetros mudam. Os parâmetros que devem ser optimizadas incluem o parâmetro de penalidade, *C*, e os parâmetros da função de *kernel*, tais como o γ para o algoritmo RBF [67]. O parâmetro C é utilizado para ajustar a importância da maximização da margem relativamente aos erros no conjunto de treino [67]. Trata-se de uma relação de compromisso entre o erro na classificação dos casos de treino, e a escolha do hiperplano e suas margens. Tem especial relevância nos casos em que é impossível obter uma separação clara entre as diferentes classes pelo que neste caso é escolhido o hiperplano com menor erro, cujo peso é influenciado por *C*. No *kernel* polinomial é possível definir o grau do polinómio. No RBF, o parâmetro γ define a flexibilidade da fronteira entre os casos de ambas as classes, um γ elevado indicia uma fronteira menos linear entre os casos.

A optimização destes parametros pode ser feita de diversas formas, entre as quais, (i) manualmente, (ii) efectuando uma pesquisa exaustiva (*grid search*), sobre um intervalo de valores pré-definidos para cada parâmetro ou (iii) através de uma pesquisa aleatória num determinado intervalo de valores aceitáveis para cada um dos parâmetros [8]. Existe uma função disponibilizada pelo pacote *e1071*, *svm.tune*, que foi usada neste trabalho e que utiliza a técnica de pesquisa exaustiva dos parâmetros.

2.4.4.3 Redes Neuronais Artificiais

São vários os algoritmos implementados sob este paradigma, entre os quais se inclui o próprio SVM. As redes neuronais também se incluem neste subconjunto, IBL, dado que a aprendizagem é feita instância a instância adaptando os pesos de cada um dos nós que constituem as redes neuronais. As redes neuronais são modelos matemáticos inspirados no funcionamento do cérebro e nas suas ligações internas através das redes de neurónios [91]. Os neurónios biológicos são substituídos por nós e a ligação entre neurónios, sinapses, é substituída por conexões, ver figura 2.6. A cada conexão é atribuído um peso aleatório no início da aprendizagem e que é ajustado durante a fase de treino quando surge uma nova instância [41]. A cada nó cabe o cálculo dos pesos das conexões que entram

nesse nó e activam a saída caso seja ultrapassado um determinado valor [91], ver figura 2.6(b). A rede neuronal é usualmente constituída por três camadas, uma com as entradas no sistema, outra com as saídas e uma camada intermédia, também designada por camada oculta. Existem vários modelos de ANN entre os quais modelos compostos por várias camadas intermédias. Na figura 2.6(a) está representada uma rede neuronal simples, três camadas, cujos círculos representam os nós da rede e as setas entre círculos representam as ligações entre os nós.



(a) Rede neuronal simples (b) Detalhe do nó

Figura 2.6: Exemplo de rede neuronal
(Adaptado de http://www.learnartificialneuralnetworks.com)

A técnica SOM é uma implementação específica dentro desta classe de algoritmos pertencentes às redes neuronais [76]. A SOM consiste numa projecção de um espaço multidimensional numa dimensão inferior, usualmente 2D. Esse espaço é dividido em forma de grelha, e as instâncias são classificadas de acordo com a grelha onde são colocadas e das instâncias que estão presentes nessa grelha [76].

2.4.5 Aprendizagem probabilística

Este tipo de aprendizagem é baseado no estudo probabilístico do conjunto de dados. Em vez de inferir relações entre os atributos ou instâncias, a aprendizagem é feita através de um estudo estatístico com base nos atributos do conjunto de dados, determinando a probabilidade de uma nova instância pertencer a uma classe. Os algoritmos baseados no teorema de *Bayes* são uma das famílias mais representativas deste tipo de aprendizagem, sendo o *Naive Bayes* (NB) um dos algoritmos mais simples e mais usado no processo de DM.

2.4.5.1 *Naive Bayes*

O NB é um algoritmo probabilístico baseado no teorema de *Bayes*. Sendo um dos algoritmos mais simples da família dos algoritmos Bayesianos, tem como pressuposto de que todos os atributos são independentes entre si face ao atributo classificador [87]. Esta assunção não é realista, dado que conjuntos de dados reais contêm na sua generalidade, alguns atributos dependentes. Apesar disso, é esta assunção que torna este algoritmo praticável de forma eficiente, especialmente nos casos cujos atributos não são fortemente relacionados [29, 78]. É um algoritmo que simplifica a aprendizagem e que na prática compete frequentemente com algoritmos mais sofisticados [105]. Trata-se ainda de um algoritmo supervisionado. O seu princípio de funcionamento tem por base o cálculo das probabilidades de uma amostra de acordo com os valores dos seus atributos. Para isso, e no caso das variáveis categóricas, é calculada a frequência de um determinado valor de um atributo para cada classe. No caso das variáveis numéricas, a probabilidade é dada por uma função de densidade que depende da média (μ) e do desvio padrão (σ), dos valores do atributo para determinada classe. Matematicamente, a função é dada por

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

Como referido inicialmente, este algoritmo tem por base a aplicação do teorema de *Bayes* para o cálculo da probabilidade de uma determinada amostra. O teorema consiste no princípio de que a probabilidade de uma determinada classe C, é dada pelo cálculo das probabilidades condicionadas de cada um dos atributos (A_i), matematicamente definida por

$$p(C|A_1,\ldots,A_n) = \frac{p(C)\ p(A_1,\ldots,A_n|C)}{p(A_1,\ldots,A_n)}.$$

No entanto, o algoritmo NB parte da presunção de que os atributos são independentes pelo que não é necessário calcular a probabilidade condicionada de cada atributo. Isto permite simplificar a fórmula final de cálculo da probabilidade de uma determinada classe C,

$$p(C|A_1,\ldots,A_n) = \frac{p(C)\prod_{i=1}^{n} p(A_i|C)}{p(A_1,...,A_n)}.$$

Com $p(C)$ a probabilidade de uma dada amostra pertencer a uma determinada classe. Por exemplo, dado o conjunto inicial (102.294 amostras), a probabilidade

de uma amostra ser positiva é mínima,

$$p(cancro) = \frac{n\ amostras\ positivas}{n\ total\ amostras} = \frac{623}{102.294} = 0,0061,$$

ao contrário da probabilidade de não ter cancro,

$$p(\neg cancro) = \frac{101.671}{102.294} = 0,9939.$$

Num conjunto em que o balanceamento seja corrigido, usando *Undersampling* por exemplo, as probabilidades serão idênticas, $\approx 0,5$. Assim, classificar uma nova amostra é calcular a probabilidade de uma determinada amostra pertencer a cada uma das classes. A classe que apresenta uma probabilidade maior será a classe da nova amostra.

2.5 Optimização

Até agora, foram descritos vários algoritmos de diferentes famílias que classificam as novas instâncias de acordo com um modelo singular que foi construído para o efeito. Embora possam produzir bons resultados, o objectivo do processo de modelação consiste na optimização de algoritmos procurando obter o melhor desempenho possível. Existem diferentes formas de o fazer, como por exemplo, (i) procurar obter os conjuntos de dados mais representativos, (ii) alterar parâmetros do algoritmo de aprendizagem, (iii) maximizar a Área sob a Curva (AUC) (ver subcapítulo 2.6), (iv) utilizar diferentes limiares de probabilidades na distinção de classes na fase de classificação ou (v) associar vários algoritmos. Este subcapítulo foca essencialmente este último ponto. Entre algumas dessas técnicas estão incluídas o *bagging*, *boosting* e o *ensemble* de algoritmos. A base destas três técnicas está na combinação de diversos algoritmos para tentar ultrapassar algumas limitações dos algoritmos de aprendizagem através da geração de um conjunto de modelos alternativos e combinando posteriormente as suas predições [59, 120], mas adoptando estratégias diversas na combinação ou agregação dos algoritmos. A figura 2.7 ilustra a base comum destas técnicas na geração de um novo modelo através da combinação de outros modelos mais simples. O uso destas técnicas é similar à tomada de decisões dos membros de comités ou conselhos de administração onde estes suportam as suas decisões na opinião de outros técnicos de uma ou várias áreas de interesse [126]. Essas opiniões podem ser diversas e ter diferentes pesos de acordo com, por exemplo, a experiência,

capacidade técnica ou área de conhecimento do técnico. São essas diferenças que são transpostas para as diferentes abordagens dos diversos modelos de agregação de algoritmos. Diferem essencialmente entre si no modo em que é feita a classificação e na forma como são obtidos ou compostos os modelos [120]. A classificação pode ser feita, por exemplo, por maioria de votos, média simples ou média ponderada. Parte-se assim da presunção e esperança de que a probabilidade de classificar incorrectamente uma instância é muito menor, dado que a probabilidade da maioria dos modelos errar na classificação de uma mesma amostra é menor [112].

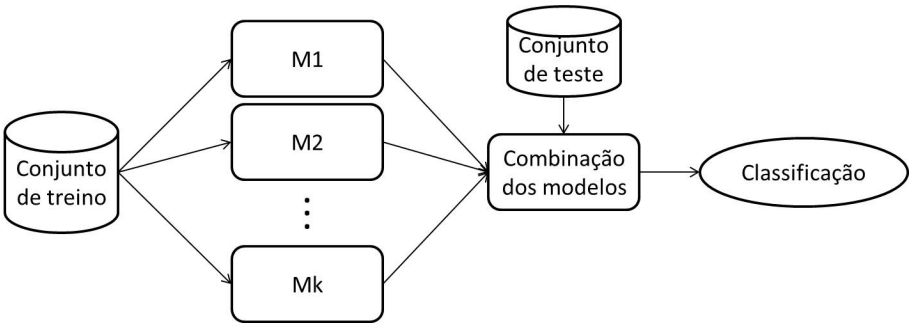


Figura 2.7: Modelo resultante da combinação de vários modelos mais simples. Adaptado de [55]

2.5.1 *Bagging*

A técnica *bagging*, nome derivado de *bootstrap aggregation*, distingue-se por na sua génese, estar a geração de n modelos usando o mesmo algoritmo mas com distribuições de dados diferentes [41]. Cada modelo é construído apenas com uma parte dos dados de treino, usando a técnica de *bootstrap* que consiste na escolha aleatória de amostras com reposição [41] (ver subcapítulo 2.3.3). A classificação final do modelo é feita por maioria de votos [41]. Por exemplo, se o *bagging* for composto por três modelos simples e dois deles classificarem uma instância como positiva e outra como negativa, a resposta do modelo final indicará que essa instância é classificada como positiva. Embora surja associado à composição de modelos baseados em árvores [15], este método pode ser aplicado a outros algoritmos. Uma implementação com bastante sucesso derivada desta técnica é o *Random Forest* (RF), descrito em [16]. RF é resultante da combinação de um largo número de modelos baseados em árvores de decisão. A obtenção destes modelos é feito usando *bagging*. Existe uma particularidade na criação das árvores, especificamente na criação dos nós dessa árvore. Em cada nó é escolhido um pequeno

conjunto de atributos aleatoriamente a partir dos quais se escolhe a melhor partição dos dados desse nó. Esta técnica permite obter um conjunto aleatório de árvores, cada uma especializada na detecção de determinadas especificidades do conjunto de dados [120, 125]. Outra implementação conhecida é o *double-bagging* que consiste na aplicação do *bagging* e no aproveitamento das instâncias que não foram usadas na aprendizagem, para optimizar os resultados, aplicando algoritmos de outras famílias como por exemplo o LDA [66]. Para compreender melhor esta técnica é necessário analisar mais ao pormenor o processo de *bootstrap* usado na geração dos modelos do *bagging*, ver subcapítulo 2.3.3. Simplificando, e dada a escolha aleatória das instâncias de cada um dos conjuntos usados na aprendizagem, cerca de 36,8% das amostras não são usadas neste processo, esta porção dos dados designa-se por amostras *Out-of-bag* [66]. Estas são então usadas na aprendizagem de um outro algoritmo. No final é feita a votação tendo em conta o *bagging* das árvores de decisão e o *bagging* de outro algoritmo usado na aprendizagem das amostras *out-of-bag* [66].

2.5.2 *Boosting*

A técnica *boosting* tem como motivação a combinação de vários modelos com um fraco desempenho para produzir um modelo com um bom desempenho [59]. O meta-algoritmo consiste na criação de diversos modelos de forma iterativa, e na atribuição de um peso a cada uma das instâncias do conjunto de dados. O peso de cada instância é tido em conta na probabilidade de uma determinada instância constar no conjunto de dados seguinte. Quer isto dizer que, na primeira iteração todas as instâncias têm igual peso e como tal igual probabilidade de pertencer ao conjunto de treino dessa iteração. O peso de uma instância varia de acordo com a correcta ou incorrecta classificação em cada iteração. Se a classificação for correcta o seu peso diminui, caso contrário aumenta, aumentando a probabilidade dessa amostra ser incluída na iteração seguinte. Pretende-se que cada iteração preste mais atenção aos casos mal classificados nas iterações anteriores, complementando o modelo anterior [55, 126]. No final das iterações, é dado um peso aos algoritmos usados em cada iteração de acordo com a sua precisão [55]. Uma das implementações com maior sucesso e representativas desta técnica é o *adaboost*, descrito em [50].

2.5.3 *Ensemble*

Na realidade tanto o *bagging* como o *boosting* descritos anteriormente são implementações específicas de *ensembles*. O conceito de *ensemble* é mais vasto. A ideia por trás deste conceito é construir um modelo de predição combinando a força de vários modelos mais simples [59]. Mesmo que estes modelos isoladamente tenham uma prestação fraca, quando em conjunto podem formar um modelo com melhor desempenho. Com base nestas premissas este conceito é menos restrictivo, permitindo a associação de algoritmos de diferentes famílias e diversificar a forma de combinação dos modelos.

2.6 Métricas para aferição e avaliação

A escolha de um bom método de avaliação é essencial para a obtenção do modelo mais adequado e com melhor desempenho na classificação das amostras e pacientes com cancro.

São usadas três métricas para avaliação e comparação dos modelos, matriz confusão, curva *Receiver Operating Characteristic* (ROC) e AUC.

A matriz confusão, ilustrada na tabela 2.1, permite verificar a precisão de um modelo. Com base nesta matriz pode-se ainda determinar a exactidão, precisão e sensibilidade de um algoritmo. A exactidão, não consta da tabela 2.1 mas é obtida a partir da matriz, é dada por:

$$\frac{TP + TN}{(TP + TN + FP + FN)}.$$

Relembra-se que o objectivo deste trabalho é classificar correctamente todos os casos Verdadeiros Positivos (TP), ou seja, aumentar a sensibilidade, mas reduzindo ao máximo a classificação errónea de casos Falsos Negativos (FN), isto é, aumentar especificidade, evitando assim a reavaliação de pacientes através de novos exames.

De acordo com Zweig *et al.* [131], as curvas ROC são uma ferramenta fundamental na avaliação dos modelos. Além de permitir uma visualização rápida, a partir desta conseguimos obter a matriz confusão que no caso da classificação é uma métrica de precisão de um modelo. Pode-se obter ainda o AUC que se trata de um índice de qualidade da curva ROC [13].

		Predição		
		Positivo	Negativo	
Actual	Positivo	TP	FN	Sensibilidade $= \frac{TP}{(TP+FN)}$
	Negativo	FP	TN	Especificidade $= \frac{TN}{(FP+TN)}$
		Positivos Correctamente Identificados $= \frac{TP}{(TP+FP)}$	Negativos Correctamente Identificados $= \frac{TN}{(TN+FN)}$	

Tabela 2.1: Matriz Confusão

A curva ROC é um dos métodos mais usados em técnicas de DM em casos médicos para comparação de modelos. A curva ROC consiste na representação gráfica dos pares, sensibilidade e especificidade, resultantes da variação de diferentes limites de corte [13]. Isto é, para traçar a curva ROC são necessários dois vectores, um com a classificação real de uma amostra, neste caso positiva ou negativa, e um outro com a probabilidade de uma determinada amostra ser positiva, estimada por um modelo. A imposição de um valor de corte a diferentes valores de probabilidades e comparando o resultado com a classificação, permite a construção da curva. Assim, é possível identificar visualmente o desempenho do modelo além de dar indicações sobre a forma de optimizar o modelo considerando os objectivos propostos, redução de FN aumentando assim a eficácia e redução do número de FP para aumentar a eficiência, através da alteração do valor de corte. Pode-se explorar desta forma a diferença na confiança de uma predição por parte de um modelo e mudar o valor limiar a partir do qual é feita uma determinada classificação. A figura 2.8 ilustra três tipos de curvas ROC. O desempenho de cada modelo é medido por proximidade da predição perfeita, ou seja, quando todos os casos são classificados correctamente, graficamente as curvas de resposta estão representadas junto ao canto superior esquerdo. A curva 1 distingue os casos de cada uma das classes pelo que se aproxima da classificação perfeita, contrariamente à curva 3 onde os casos estão sobrepostos e como tal a classificação é aleatória. A curva 2 ilustra a situação mais habitual onde a maioria dos casos são distinguíveis e existe apenas uma faixa onde estes se sobrepõem. É a redução desta

faixa que optimiza um classificador. O resultado final trata-se de uma relação de compromisso entre estas duas vertentes, sensibilidade e especificidade [44, 58].

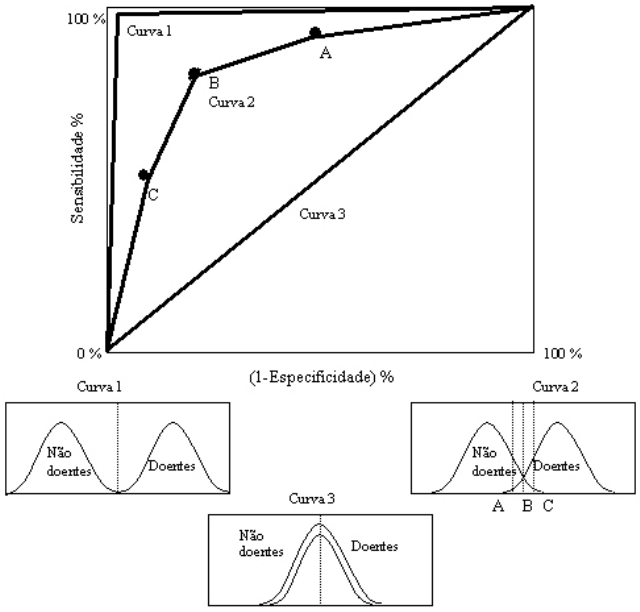


Figura 2.8: Exemplos de Curvas ROC
(Fonte: `http://medstatweb.med.up.pt`)

A AUC é obtida a partir da análise da curva ROC, e usada como medida de qualidade do desempenho de um modelo, numa relação directa [12, 14, 47, 58] e, como tal serve de comparação entre os diferentes modelos. Apesar de Hand, recentemente, ter defendido uma medida alternativa [56, 57]. A AUC representa a probabilidade de um paciente doente escolhido aleatoriamente, ser correctamente classificado, em termos efectivos ou probabilísticos, com maior probabilidade de ter cancro que um paciente saudável escolhido aleatoriamente [58]. Trata-se da curva ROC a partir das percentagens de TP e FP. Usados em conjunto estes três indicadores permitem a comparação dos diferentes modelos.

2.7 Trabalhos relacionados

Neste subcapítulo são apresentados alguns trabalhos, que servem de base à compreensão e realização desta tese, no âmbito do DM e sua aplicação em dados médicos, com especial ênfase à detecção de tumores.

Um dos trabalhos com especial relevância foi desenvolvido por Perlich *et al.* [98] no âmbito do KDD Cup 2008. O relatório apresentado pelos vencedores deste

concurso descreve as etapas e os conceitos envolvidos na realização de um trabalho nesta área. Descreve resumidamente o pré-processamento, modelação e pós processamento. Na etapa de pré-processamento começam por um processo exploratório onde descobriram algumas informações pertinentes quanto aos identificadores dos pacientes, identificando que estes atributos contêm informação relevante para a classificação das amostras. Além disso experimentaram retirar os *outliers* (valores extremos) dos atributos e colocar atributos adicionais para melhorar o desempenho dos algoritmos, mas cujo resultado tinha impactos negativos na classificação. No pós-processamento tentaram maximizar a AUC e a taxa de TP. Usam ainda o meta-algoritmo *bagging* na composição de modelos SVM simples para optimização dos resultados. Na sua análise, apresentam o *Bagged Linear* SVM *with Post Processing* como o modelo com melhor desempenho nos testes que efectuaram [98]. Existe um segundo relatório dentro do âmbito desta competição escrito por Lo *et al.* [83], os segundos classificados. São descritos os modelos utilizados e alguns métodos que usaram para optimizar os modelos. Realça-se, uma vez mais, o uso de *ensemble* de modelos e o uso do SVM, se bem que desta vez o *Adaboost* foi o meta-algoritmo utilizado na composição dos algoritmos mais simples [83].

Como foi dito no capítulo 1, a competição do KDD Cup 2008 consistiu em duas tarefas distintas: (i) aumento da AUC da *Free-response Receiver Operating Characteristic* (FROC), na região clinicamente relevante entre 0,2 e 0,3 falsos positivos por imagem, e (ii) redução da carga de trabalho dos radiologistas diminuindo o número de exames que devem ser revistos garantindo que todos os pacientes doentes são reavaliados, ou seja, sensibilidade máxima a estes casos. Foram usadas diferentes métricas em cada uma das tarefas. Na primeira tarefa foi usada a AUC da FROC para avaliar os resultados dos competidores, tendo os vencedores atingido uma AUC de 0,093 e os três primeiros classificados obtiveram uma AUC superior a 0,089. A segunda tarefa foi avaliada segundo dois parâmetros, sensibilidade e especificidade. Neste caso, os três primeiros classificados conseguiram obter a sensibilidade máxima com especificidade acima dos 0,168, sendo que o primeiro classificado conseguiu os três primeiros resultados da competição com valores de especificidade entre 0,624 e 0,681, muito distantes do segundo classificado que se quedou com uma especificidade de 0,174[1].

Fora desta competição, existem vários trabalhos nesta área mas com conjuntos de dados diferentes, mas também provenientes de dados médicos existentes em

[1]Resultados disponíveis em http://www.sigkdd.org/kdd-cup-2008-breast-cancer. Competição KDD CUP 2008

diversas bases de dados. Ressalva-se que apesar de serem dados médicos, as características dos conjuntos de dados são diferentes pelo que se tem de adoptar estratégias diferentes conforme os casos, mas não deixam de ser um bom ponto de partida.

Lavrač, apresenta um estudo onde selecciona um conjunto de técnicas de DM adequadas ao uso em medicina, dada a sua especificidade. São apresentados alguns algoritmos e métricas para a sua avaliação [80].

Delen *et al.* [35] e Bellachia e Guven [6] apresentam em cada um dos artigos, três modelos para classificar a sobrevivência ao cancro da mama. Os artigos são coincidentes na análise de dois dos algoritmos ANN e *C4.5*, uma especialização das árvores de decisão, além de usarem o mesmo método, *10-fold cross validation* para validação de resultados. Os autores diferenciam-se ao usar um terceiro algoritmo diferente nas suas análises, Bellachia e Guven usam o NB e Delen *et al.* usam a LR. Dada a proximidade de objectivos, estes algoritmos fizeram parte de uma avaliação preliminar deste trabalho. Burke *et al.* [20] é um dos trabalhos de base referido por Delen *et al.*, em cujo estudo é referido o uso de *Principal Component Analysis* (PCA) na redução do conjunto de dados médicos [20]. Ayer *et al.* apresentam uma interessante compilação de modelos CAD usados na detecção de cancro da mama em mamografias desde 1996 a 2010, entre os quais surge o LDA e *Bayesian Networks* (BN), o primeiro fez parte do conjunto de testes já o segundo não foi testado neste trabalho [4]. Por sua vez, Bellazzi e Zupan indicam um conjunto de famílias de modelos de DM utilizados em medicina clínica [7]. Além dos modelos já referidos introduz o KNN como um dos modelos passíveis de utilização em modelações de casos clínicos. Além da referência a alguns algoritmos, descreve também de forma sucinta e sistemática todo o processo de DM aplicado à área médica. Por outro lado, Meyer *et al.* [89] fizeram uma das mais completas comparações entre algoritmos enumerando quinze algoritmos como adequados para a classificação. Estes algoritmos fazem parte de diversas famílias pelo que se torna interessante a sua análise para perceber a sua aplicabilidade neste trabalho. Foram incluídos nesta tese apenas alguns dos algoritmos citados por Meyer *et al.*, tais como o RF, o MARS, o *Bagging*, o *Double-Bagging* e o QDA. Adicionalmente foram estudados os modelos supervionados e não supervisionados do SOM, utilizados por exemplo em ecografias [28], mas cuja aplicação na detecção de cancro não é muito conhecida. Daí a curiosidade em perceber o desempenho deste algoritmo nestes casos.

3

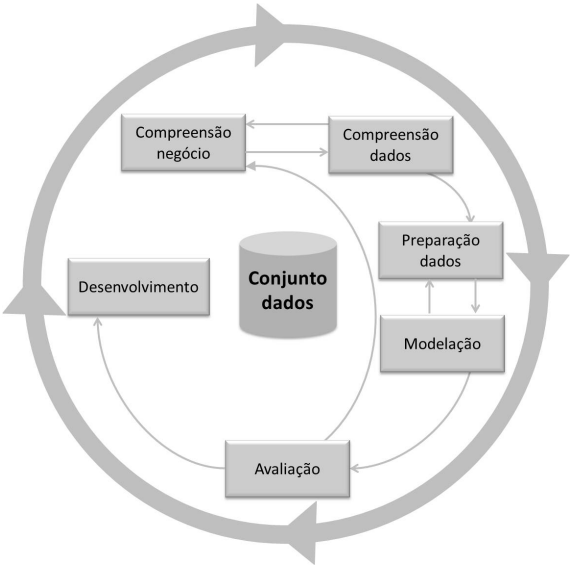
Exploração dos dados

"O termo KDD foi formalizado em 1989 como uma referência ao conceito mais amplo de procura de conhecimento em dados e, é um processo que envolve a identificação e o reconhecimento de padrões numa base de dados de uma forma automática" [5]. Assim, podemos definir KDD como o processo de descoberta de novas correlações, padrões e tendências significativas, por meios de análises minuciosas a um conjunto de dados de elevada dimensão e/ou complexidade [5, 45].

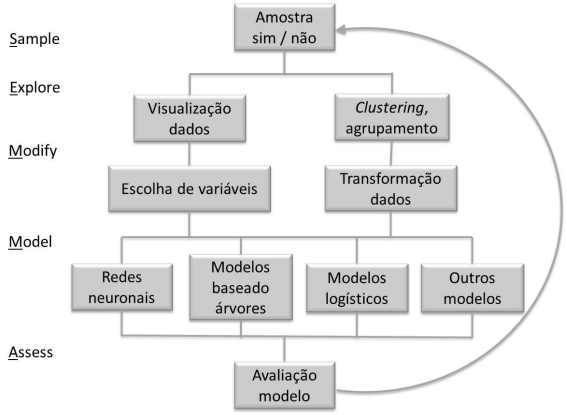
O processo KDD está dividido em três grandes grupos: pré-processamento, processo de DM e por fim, pós-processamento. Deste modo, o processo inicia-se com a compreensão do estudo do domínio e objectivos a atingir. Segue-se então, a recolha e transformação de dados. Neste passo, é necessário proceder à escolha das técnicas e métodos a aplicar. Por fim, é gerado um conjunto de padrões para analisar, validar e interpretar.

Quando pretendemos aplicar o KDD a uma base de dados deve-se ter especial atenção: (i) à dimensão, se for muito grande pode dar origem a uma enorme variedade de padrões, combinações e hipóteses; (ii) à existência de atributos com valores nulos ou omissos; (iii) e, *outliers* [5, 77].

Para que um processo de KDD se torne mais fácil de compreender, implementar e de desenvolver, este deve ser enquadrado numa metodologia. Poderão surgir alguns problemas, especialmente, na fase inicial: (i) extracção; (ii) preparação; (iii) e/ou validação dos dados.



(a) CRISP-DM



(b) SEMMA

Figura 3.1: Processo DM - Metodologias.
(Adaptado de http://www.informationbuilders.com)

O processo de KDD é interactivo, uma vez que as fases estão muito relacionadas e qualquer alteração numa delas irá afectar as fases seguintes e provavelmente o sucesso de todo o processo.

No final do século XX, entidades envolvidas nestas áreas seguiram as suas próprias estratégias e métodos, com produção de resultados distintos. Pelo que, surgiu a necessidade de definir uma metodologia que servisse de referência para o desenvolvimento de projectos de KDD. Foram propostas várias metodologias, das quais se destacam actualmente duas. A *Sample, Explore, Modify, Model and Assess* (SEMMA), desenvolvida pelo SAS Inc, usualmente interpretada como uma metodologia refere-se a um processo central de DM. A partir de uma amostra estatisticamente representativa, SEMMA aplica de uma forma simples, técnicas

estatísticas e de visualização, selecciona e transforma as variáveis preditivas mais importantes, modela as variáveis para prever resultados e valida a exactidão do modelo, ver figura 3.1(b) [5, 45].

A segunda conhecida como metodologia *Cross Industry Standard Process for Data Mining* (CRISP-DM), foi desenvolvida através de um consórcio entre Daimler-Chrysler, NCR[1] e SPSS[2], em finais de 1996, e desde então tem vindo a ser utilizada para dar respostas mais fidedignas e fáceis de gerir [24]. A última versão é orientada para aspectos práticos, com base em princípios académicos e em aplicações práticas de DM.

A metodologia CRISP-DM apresenta-se em 6 fases, ver figura 3.1(a), nomeadamente: (i) compreender os objectivos e condições necessárias do projecto; (ii) estudo dos dados; (iii) preparação dos dados; (iv) modelação; (v) avaliação e, (vi) implementação. Sendo pensadas para ser aplicadas em qualquer área de trabalho [24]. O objectivo desta metodologia é tornar os projectos de DM mais rápidos, baratos e simples de gerir, independentemente da sua dimensão e grau de complexidade.

As metodologias SEMMA[3] e CRISP-DM apresentam-se da mesma forma, estruturando o processo de KDD em fases que se encontram relacionadas entre si, convertendo o processo de descoberta de conhecimento num processo iterativo.

A metodologia SEMMA encontra-se mais direccionada nas características do desenvolvimento das técnicas e do processo, enquanto que a CRISP-DM mantém uma perspectiva mais ampla em relação aos objectivos do projecto. Esta diferença verifica-se, desde logo na primeira fase, isto é, SEMMA preocupa-se com a recolha de uma amostra de dados. Por sua vez, CRISP-DM realiza uma análise do problema para o transformar num problema técnico. Assim, podemos concluir está mais dirigida para uma concepção real do projecto, podendo ser integrada numa metodologia de gestão de projectos. Outra diferença significativa é a sua relação com as ferramentas comerciais. A metodologia SEMMA está muito ligada a produtos SAS, enquanto que a metodologia CRISP-DM foi desenhada como uma metodologia neutra em relação à ferramenta que se utiliza, sendo que a sua distribuição é livre e gratuita.

Deste modo, optou-se pela metodologia CRISP-DM durante este trabalho, por ser

[1] http://www.ncr.com
[2] http://www.spss.com
[3] Neste documento, por questões de simplificação, iremos referir como uma metodologia, apesar da imprecisão do termo.

mais completa e neutra, se adequar melhor ao problema apresentado e, em especial por ser aquele que tem maior expressão entre as metodologias existentes [5].

Neste capítulo são descritos os procedimentos usados para a compreensão e exploração dos dados, inserido na etapa 2 do CRISP-DM:

1. Importação dos dados para o R
2. Visualização e descrição dos dados
3. Tratamento de dados
4. Correlação entre atributos e remoção de redundâncias
5. Balanceamento do conjunto de dados

3.1 Descrição dos dados

Neste trabalho utilizou-se a ferramenta R [4], uma linguagem e ambiente de desenvolvimento para computação estatística. As suas funcionalidades, juntamente com os inúmeros pacotes disponíveis [5], fazem deste ambiente uma excelente ferramenta para efectuar os passos necessários num processo de KDD.

Os dados são disponibilizados em dois ficheiros CSV [6]. O primeiro, "info.txt", contém a informação genérica relativa à identificação do exame, paciente e localização das lesões, cujos atributos são descritos na tabela 3.1.

O segundo ficheiro, "features.txt", contém informação de 102.294 candidatos, cada um descrito por 117 atributos que caracterizam cada lesão.

O código listado em 3.1 mostra a importação dos dois ficheiros para o *R*. Foi efectuada a junção destes (linha 25) para produzir o conjunto de dados a utilizar nas etapas seguintes.

Sobre os atributos dos ficheiros foi efectuada uma análise dos dados, recorrendo a alguns indicadores de estatística descritiva, nomeadamente, mínimo e máximo, mediana, média, primeiro e terceiro quartil. Os resultados do ficheiro "info.txt" podem ser observados na tabela 3.2. A mediana do atributo `Malignant.Mass` é -1 e a média é próxima deste valor. Isso indicia um desequilíbrio entre o número

[4] http://www.r-project.org/
[5] http://cran.r-project.org/web/packages/
[6] Dados disponíveis em http://www.sigkdd.org/. Usados na competição KDD CUP 2008.

Atributo	Descrição	Domínio
Malignant Mass	Classificação do tumor	Cancerígeno (-1) / Não Cancerígeno (1)
Image-Finding-ID	Identificação da lesão. Ambas, as mamas, são visualizadas em imagens MLO e CC	Inteiro não negativo
Study-Finding-ID	Identificação, exclusiva, da lesão através das imagens MLO e CC	Inteiro
Image-ID	Identificação da imagem da zona suspeita	Inteiro
Study-ID	Identificação do paciente	Inteiro
LeftBreast	Mama examinada	1 se for mama esquerda, 0 se for mama direita
MLO	Tipo de exame	1 se for obtida por imagem MLO, 0 se for obtida por imagem CC
X-location	Localização do tumor - coordenada em X	Inteiro
Y-location	Localização do tumor - coordenada em Y	Inteiro
X-nipple-location	Localização do mamilo - coordenada em X	Inteiro
Y-nipple-location	Localização do mamilo - coordenada em Y	Inteiro

Tabela 3.1: Informações adicionais para cada região da mama suspeita de tumor maligno (info.txt)

Malignant.Mass	Left.Breast	MLO	X.location	Y.location	X.nipple.location	Y.nipple.location
Min. :-1,0000	Min. :0,0000	Min. :0,0000	Min. : 25	Min. : 54	Min. : 14	Min. : 601
1st Qu.:-1,0000	1st Qu.:0,0000	1st Qu.:0,0000	1st Qu.: 816	1st Qu.:1676	1st Qu.:1350	1st Qu.:1999
Median :-1,0000	Median :1,0000	Median :0,0000	Median :1619	Median :2085	Median :1636	Median :2213
Mean :-0,9878	Mean :0,5059	Mean :0,4791	Mean :1611	Mean :2091	Mean :1639	Mean :2195
3rd Qu.:-1,0000	3rd Qu.:1,0000	3rd Qu.:1,0000	3rd Qu.:2390	3rd Qu.:2513	3rd Qu.:1910	3rd Qu.:2411
Max. : 1,0000	Max. :1,0000	Max. :1,0000	Max. :3299	Max. :4023	Max. :3324	Max. :3107

Tabela 3.2: Sumário dos dados do ficheiro info.txt

de incidências em que não foram detectados indícios de tumores malignos e o número de incidências em que esta identificação foi positiva. Verifica-se que pelo menos 75% dos casos não são malignos dado que os primeiros três quartos dos dados deste atributo têm valor -1. É de salientar que o conjunto de dados de treino contém 102.294 incidências, das quais apenas 623 são malignas. Trata-se de um conjunto desequilibrado, onde existem 164 vezes mais casos negativos que positivos. Esta situação pode ter impacto na modelação dos dados e será abordada adiante. Uma vez que os dados estavam completos, isto é, sem valores nulos ou omissos, não foi feito nenhum tratamento especial a estes.

```
# Carregar dados do ficheiro Info.txt
# Informacoes sobre cada um dos pontos analisados em Features.txt
info <- read.table('info.txt',
       header=F,
       dec='.',
       col.names=c('Malignant Mass','Image Finding ID','Study Finding ID',
       'Image ID','Study ID','Left Breast','MLO','X location',
       'Y location','X nipple location','Y nipple location'),
       na.strings=c('XXXXXXX'))

# Carregar caracteristicas para o R Features.txt
x <- 1:117
fNames <- paste('F',x, collapse = NULL)
features <- read.table('features.txt',
       header=F,
       dec='.',
       col.names=fNames,
       na.strings=c('XXXXXXX'))

# Informacao sobre as variaveis
summary(info)
summary(features)

# Juntar Dataset
allDataTogether <- cbind(info, features)
```

Listagem 3.1: Importação dos dados

Note-se que os atributos identificadores de cada uma das incidências, tais como `Image.Finding.Id`, `Study.Finding.ID`, `Image.ID` e `Study.ID` não têm, normalmente, utilidade para a detecção de tumores malignos.

No processo de exploração e análise de dados, é importante efectuar uma visualização destes, afim de extrair alguma informação [31].

A figura 3.2 representa a localização de cada incidência de cancro em relação ao mamilo, onde os casos positivos são identificados a vermelho. Esta, é obtida através do cálculo da diferença entre a localização da incidência numa imagem (*Location*) e a localização do mamilo (*NippleLocation*).

$$P_{(x,y)} = Location_{(x,y)} - NippleLocation_{(x,y)}$$

A figura está divida por tipo de exame (MLO ou CC) e por mama. Da análise da imagem não se observa uma área onde seja mais provável a ocorrência de casos positivos de cancro. No entanto, verifica-se que existem muito poucos casos positivos nas zonas de fronteira do seio. A maioria dos casos encontra-se no interior do mesmo. Assim, não é espectável uma forte relação entre localização da incidência e malignidade do tumor.

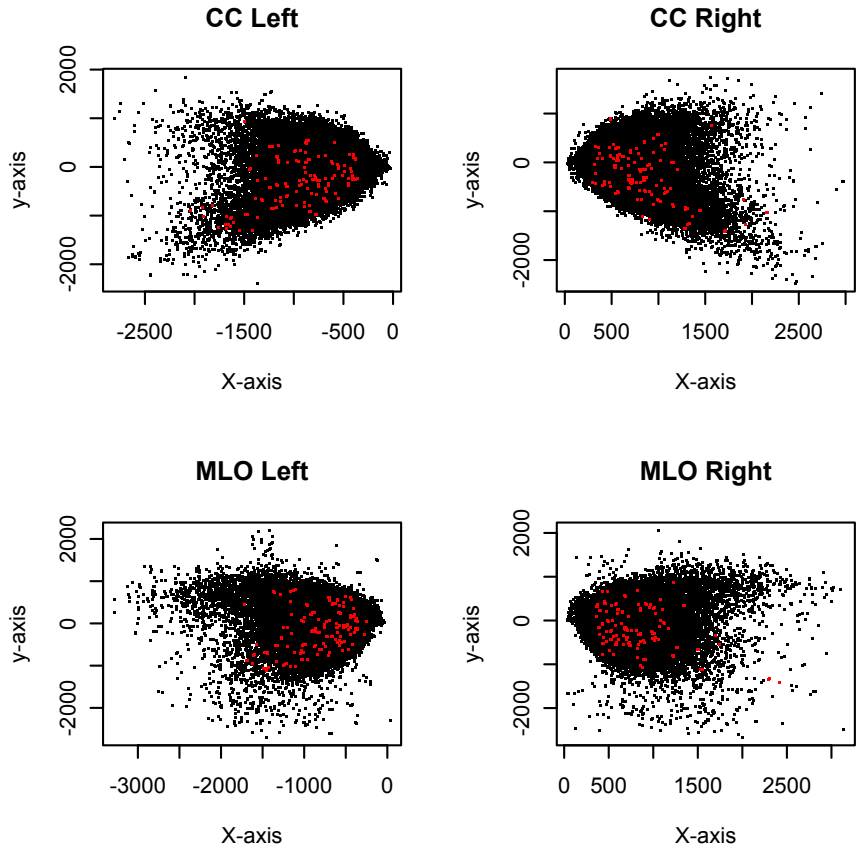


Figura 3.2: Localização das incidências de cancro por mama e exame

3.2 Correlação entre atributos e remoção de redundâncias

O conjunto de dados em análise é extenso e a sua manipulação exige a alocação de muitos recursos que podem inviabilizar o seu processamento ou ter perdas de desempenho. Este é um problema conhecido, que se dá o nome de *"Curse of dimensionality"* [122]. Como o conjunto pode conter redundâncias que não acrescentam valor para classificar uma incidência, foi feita uma análise dos atributos mais relevantes para essa tarefa (ver subcapítulo 2.2.1). Através da análise do impacto que cada atributo tem no modelo, identifica-se e remove-se toda a informação considerada irrelevante ou redundante. A selecção de atributos é um pré-processamento utilizado em aprendizagem automática, com o objectivo de reduzir a dimensão dos dados e optimizar todo o processo de classificação [128].

	M.M	L.B	MLO	X.l	Y.l	X..	Y..	F.1	F.2	F.3	F.4	F.5	F.6	F.7	F.8	F.9	F.10	F.11	F.12	F.13	F.14	F.15
M.M	1																					
L.B		1																				
MLO			1																			
X.l		+		1																		
Y.l					1																	
X.n						1																
Y.n					.		1															
F.1								1														
F.2									1													
F.3										1												
F.4									.	+	1											
F.5										-	.	1										
F.6										-	.	*	1									
F.7														1								
F.8														*	1							
F.9														.	.	1						
F.10														.	.	*	1					
F.11																		1				
F.12																		-	1			
F.13																				1		
F.14														.	.						1	
F.15														.	.						B	1
F.16														.	.						.	.
F.17														-	-					.		.
F.18																						
F.19												.	.									
F.20																-	-		.			
F.21																-	-					
F.22																						
F.23			.																			
F.24														-	-			.			-	-
F.25														-	-							
F.26														.	-							
F.27		+		B																		
F.28					B		.															
F.29		+		B																		
F.30					B		.															

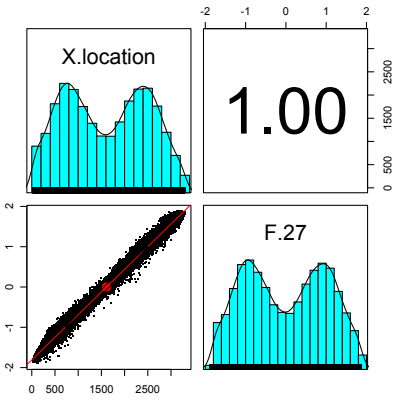
Tabela 3.3: Correlação entre os atributos. A representação é dada por $[0;0,3[$:' ', $[0,3;0,6[$:'.', $[0,6;0,8[$:'-', $[0,8;0,9[$:'+', $[0,9;0,95[$:'*', $[0,95;1[$:'B'.
(Legenda: M.M - Malignant.Mass, L.B - Left.Breast, X.l - X.location, Y.l - Y.location, X.n - X.nipple.location, Y.n - Y.nipple.location, F1 a F117 - Atributos não discriminados)

3.2.1 Correlação entre atributos

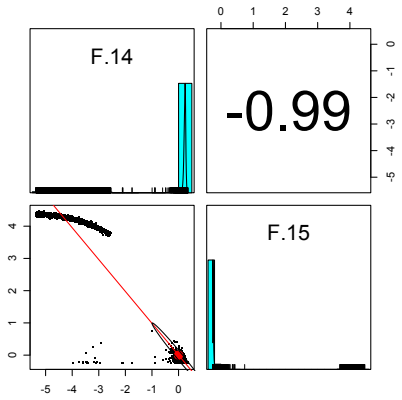
Uma abordagem para determinar a redundância existente nos atributos que compõem o conjunto de dados, consiste na determinação da correlação entre estes. Desta forma obtém-se uma primeira visão sobre a possível redundância dos diversos atributos. Uma correlação alta indica uma ligação forte entre os atributos, isto é, estamos perante atributos que representam características semelhantes ou directamente proporcionais [54].

O *R* disponibiliza um conjunto de funções que permite obter essa correlação. A tabela 3.3 apresenta de forma reduzida, a correlação entre alguns atributos dos dados. Para cada par de atributos é calculado o seu coeficiente de correlação, também designado por "ρ de Pearson". O valor do coeficiente está compreendido no intervalo $[-1;1]$. Um $\rho = 1$ significa que existe uma correlação perfeita entre

duas variáveis, se contrariamente ao $\rho = 0$ que indica a inexistência de relação entre estas. Se $\rho = -1$, significa que as duas variáveis são inversamente proporcionais [101, 107]. Tomemos como exemplo os atributos `F.14` e `F.15`. O coeficiente de correlação pertence ao intervalo $[0,95; 1[$, isto é, apresenta uma correlação quase perfeita. Para confirmar a dependência linear entre os atributos e visualizar de forma rápida fez-se uma representação gráfica. Por vezes, valores altos na correlação não correspondem a uma relação forte entre as variáveis, como no exemplo do quarteto de Anscombe [2]. Como se pode observar na figura 3.3(b), existe uma correlação inversa. No entanto, os dados não têm uma distribuição uniforme sob a diagonal do gráfico, o que indica uma relação fraca. Comparando esse resultado com o obtido com outros dois atributos, `X.location` e `F.27`, pode-se observar, quer na tabela 3.3 e na figura 3.3(a), que são variáveis relacionadas. A figura 3.3 mostra a distribuição de frequências de cada um dos atributos analisados.



(a) Correlação entre X Location e F.27



(b) Correlação entre F.14 e F.15

Figura 3.3: Exemplos de correlações

Foram detectadas 57 correlações lineares elevadas, ou seja, compreendidas no intervalo $[0,96; 1[$. Geralmente, duas variáveis são consideradas como fortemente correlacionadas se o valor da correlação for superior a 0,9 [120], mas dado que existe um número considerável de correlações (57 em 124 atributos), optou-se por não alargar o intervalo.

Obtidas as correlações dos atributos é necessário seleccionar os atributos que são redundantes e que se podem omitir nas análises futuras (ver listagem A.3). Para encontrar os atributos redundantes, o primeiro passo é obter a matriz com os coeficientes de correlação entre cada um dos atributos (listagem A.1, linha 20) usando a função *cor* disponibilizada pelo *R*. Aliás, é esta matriz que dá origem

à tabela 3.3. Dessa matriz escolhem-se apenas os pares de atributos com coeficientes mais elevados, superiores a 0,96 (listagem A.1, linhas 24-27). Todos os atributos são colocados num único vector (listagem A.3, linha 5). Compara-se cada par de atributos correlacionados com o vector e, caso estes estejam contidos no vector é retirado um deles, (listagem A.3, linhas 7-10). No final é feita a diferença entre o conjunto inicial e o conjunto obtido (listagem A.3, linha 12).

Deste procedimento, conclui-se que existem 36 atributos redundantes no conjunto, e que por isso são fortes candidatos para serem retirados. No entanto, e caso se opte pela remoção destes, é necessário avaliar o impacto desta na obtenção do modelo de classificação.

3.2.2 Selecção de atributos

Outra forma de reduzir a dimensão do conjunto de dados é efectuar a selecção dos atributos [126] que mais contribuem para a classificação de cada caso. O objectivo é obter o peso de cada atributo na classificação e ordená-los. A escolha do algoritmo para obtenção dos pesos pode influenciar os resultados obtidos, pelo que foram usados diferentes algoritmos e métodos para a selecção dos atributos.

A dimensão do conjunto de dados impossibilitou a utilização directa dos algoritmos de selecção de atributos. A solução adoptada foi executar o mesmo algoritmo determinístico inúmeras vezes com um conjunto de dados aleatório na entrada da função, até emergir um padrão ou tendência. Esta técnica é inspirada no algoritmo de Monte Carlo, onde se afirma que um resultado pode ser obtido pela combinação de sucessivas aproximações aleatórias a esse mesmo resultado [40, 92]. Para que o impacto na determinação do peso do atributo seja mínimo, o conjunto é dividido em 50 subconjuntos com 50.000 amostras aleatórias do conjunto inicial, aproximadamente metade das amostras em cada tentativa (listagem A.3, linhas 9-13). Todos os subconjuntos são submetidos ao mesmo algoritmo, guardando-se o peso de cada atributo nas iterações intermédias. No final os pesos são somados para produzir o peso total, que, depois de ordenado de forma crescente, permite identificar os atributos mais relevantes. Na primeira função, *calculateWeightsSampling*, obtém-se os pesos dos atributos em cada subconjunto (listagem A.3, linhas 16-25). A segunda função, *calculateRankSampling*, soma o peso obtido por cada atributo em cada uma das iterações ordenando os atributos de forma decrescente (listagem A.3, linhas 27-40). A ordenação final é influenciada pela posição ocupada pelo atributo em cada uma das iterações. Dado que é necessário atribuir um peso a cada atributo, e como se verifica que

	Information Gain	Gain Ratio	Chi Squared	Symmetrical Uncertainty
1	F.9	F.10	F.9	F.9
2	F.10	F.9	F.10	F.10
3	F.20	F.20	F.20	F.20
4	Left.Breast	Left.Breast	Left.Breast	Left.Breast
5	MLO	MLO	MLO	MLO
6	X.location	F.21	F.21	F.21
7	F.21	X.location	X.location	X.location
8	Y.location	Y.location	Y.location	Y.location
9	X.nipple.location	X.nipple.location	X.nipple.location	X.nipple.location
10	F.3	F.3	F.3	F.3
11	F.4	F.4	F.4	F.4
12	Y.nipple.location	Y.nipple.location	Y.nipple.location	Y.nipple.location
13	F.1	F.12	F.1	F.1
14	F.12	F.1	F.12	F.12
15	F.2	F.2	F.2	F.2

Tabela 3.4: Selecção de atributos - Comparação de resultados da aplicação dos diferentes algoritmos, *ranking* dos primeiros 15 atributos

os últimos 50 atributos não variam de posição, optou-se por atribuir igual valor a estes. A cada atributo que se encontre nas primeiras 73 posições é atribuído um peso entre $[123; 50[$, sendo que o primeiro atributo tem o peso de 123. Desta forma, os atributos que se encontram mais vezes nas primeiras posições têm um maior peso relativo aos outros. A fórmula de cálculo e ordenação pode ser consultada na listagem A.3, linhas 48-50.

São usados quatro algoritmos diferentes para a obtenção dos pesos dos atributos. Três são baseados nos princípios da teoria da informação, e usam a entropia como medida base para o cálculo da correlação.

O primeiro, *Information Gain* (IG), é uma medida assimétrica entre duas distribuições de probabilidade

$$H(Class) + H(Attribute) - H(Class, Attribute),$$

com $H(S)$ a entropia de **S** e **S** representa o conjunto de treino [21, 109]. Se o valor de IG de um atributo para uma classe for elevado, este é considerado importante e informativo para essa classe. Por outro lado, se o valor obtido for baixo, este não traz informação da classe e consequentemente pode ser removido. Contudo, quando o número de atributos chave ou identificadores é elevado, numa determinada classe, apresenta um IG alto mas não é relevante para a determinação de casos cancerígenos uma vez que o resultado está demasiadamente especializado para os casos conhecidos e sendo por isso difícil de generalizar para os novos

casos [36, 91]. Este facto tem menor relevância no caso estudado uma vez que os atributos identificadores foram retirados deste estudo.

O segundo algoritmo, *Gain Ratio* (GR) [93, 109, 126], é dado por

$$\frac{H(Class) + H(Attribute) - H(Class, Attribute)}{H(Attribute)}.$$

Este algoritmo é uma variante normalizada do anterior. O GR tenta resolver uma deficiência apresentada pelo IG, que apresenta valores elevados quando há um aumento da dependência entre classe e atributos, e da entropia. Como solução, o ganho de informação é dividido pela entropia do nó o que suaviza o favorecimento de atributos com maior entropia [22].

O terceiro, *Symmetrical Uncertainty* (SU) [101, 109, 126], mede a mútua dependência de duas variáveis aleatórias e é dado por

$$2 \times \frac{H(Class) + H(Attribute) - H(Class, Attribute)}{(H(Attribute) + H(Class))}.$$

O último algoritmo baseia-se na distribuição chi-quadrado. A medida caracteriza-se por ser um teste estatístico que mede o desvio entre uma distribuição esperada ao assumir que o atributo é independente da classe. Denomina-se por *Chi Squared* (CS), e calcula os coeficientes *Cramer's V* entre dois atributos [82, 109].

Como é verificado na tabela 3.4, a ordenação dos resultados em todos os algoritmos são muito semelhantes, sofrendo alterações mínimas. O atributo `F.9` aparece por três vezes no primeiro lugar e uma vez em segundo por troca com o atributo `F.10` que surge em segundo lugar em todos os outros algoritmos. `F.20`, `Left.Breast` e `MLO`, surgem sempre no terceiro, quarto e quinto lugar, respectivamente. Verifica-se assim que estes atributos têm bastante peso e como tal não podem ser retirados do conjunto. Os restantes atributos não são removidos, uma vez que ainda apresentam um valor considerável quando comparados com os restantes atributos presentes no conjunto de dados.

3.2.3 Determinação das componentes principais

O PCA é outro método que pode ser usado na redução da dimensão de um conjunto de dados [102]. O PCA é uma técnica matemática que utiliza uma transformação linear ortogonal dos dados, projectando-os num novo plano. Cada iteração é uma nova componente que capta a máxima variância dos dados em relação

	PC1	PC2	...	PC60	PC61	PC62	...	PC122	PC123
Standard deviation	5,555363242	3,698825378	-	0,345997778	0,325557157	0,315535163	-	0,000544351	0,000370978
Proportion of Variance	0,25091	0,11123	-	0,00097	0,00086	0,00081	-	0	0
Cumulative Proportion	0,25091	0,36214	-	0,9892	0,99006	0,99087	-	1	1

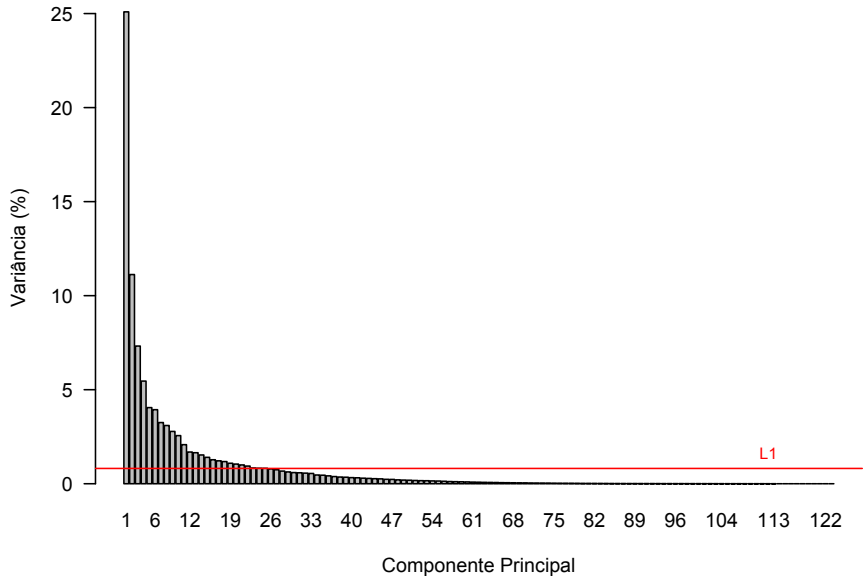
Tabela 3.5: Sumário análise PCA

à componente anterior. Este processo é repetido até se obter toda a variância do conjunto, com um número máximo de iterações igual ou inferior ao número de atributos [126]. Recorreu-se à função *prcomp* disponível no *R*. A figura 3.4(a) mostra a variância em percentagem de cada uma das componentes principais, a linha *L1* define a percentagem que cada atributo deveria obter se todos contribuíssem de igual forma. Analisando o gráfico juntamente com o sumário dos resultados obtidos, tabela 3.5, verifica-se que 61 componentes permitem obter 99% de toda a variância. Esta informação também pode ser retirada da figura 3.4(b) onde se observa a variância acumulada. O ponto *P1* é a intersecção entre o valor acumulado até à componente 61 e a linha *L1*, correspondente a 99%. Deste modo, em vez de 123 atributos, é possível reduzir a dimensão do conjunto para metade. No entanto, esta questão não é assim tão linear. Recorde-se que o conjunto de dados é desequilibrado e, pode ser muito mais importante identificar casos muito específicos, discrepantes da maioria. Além disso, este método, como mencionado acima, é definido como uma transformação linear, então se a relação entre atributos é linear, produz melhores resultados [102]. Voltaremos a abordar este assunto no capítulo 4.

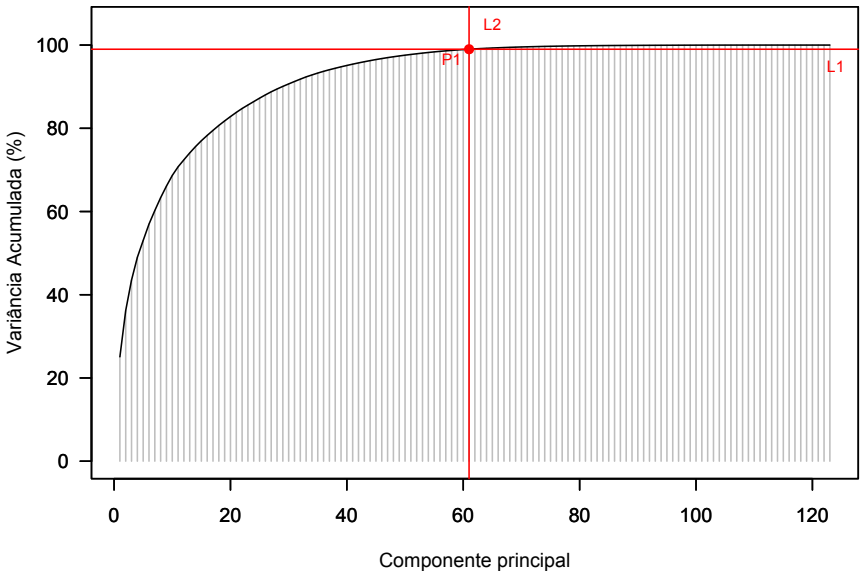
3.3 Conjunto de dados desequilibrado: técnicas de *Undersampling* e *Oversampling*

Normalmente, em bases de dados médicas verifica-se um desequilíbrio entre amostras de casos positivos e negativos no conjunto de dados. Esse desequilíbrio pode não permitir uma correcta classificação dos casos minoritários, em que a identificação do tumor é positiva. Se tal acontece-se, o modelo pode não ser utilizado, mesmo que apresente, no total, valores de detecção elevados. Uma alternativa a este problema é adoptar técnicas que permitam balancear o conjunto de dados. Técnicas que foram aplicadas na selecção dos atributos, por forma a verificar como se comportam na ordenação dos atributos.

As técnicas abordadas são, *undersampling* e *oversampling* [27, 43, 60]. No primeiro caso, a estratégia é manter o conjunto minoritário e retirar amostras aleatórias



(a)



(b)

Figura 3.4: Análise da componente principal

do conjunto maioritário até que o conjunto final tenha um número equivalente de amostras (ver subcapítulo 2.2.2). Neste caso, o conjunto final é reduzido para 1.246 amostras, nas quais se encontram todos os casos positivos de cancro e 623

	IG	Under IG	Over IG	GR	Under GR	Over GR	CS	Under CS	Over CS	SU	Under SU	Over SU
1	F.9	F.5	F.9	F.10	F.4	F.9	F.9	F.4	F.9	F.9	F.6	F.9
2	F.10	F.6	F.10	F.9	F.3	F.10	F.10	F.3	F.65	F.10	F.5	F.10
3	F.20	F.4	F.20	F.20	F.6	F.65	F.20	F.11	F.20	F.20	F.4	F.20
4	Left.Breast	F.3	F.4	Left.Breast	F.5	F.35	Left.Breast	F.109	F.35	Left.Breast	F.3	F.21
5	MLO	F.14	F.21	MLO	F.68	F.64	MLO	F.24	F.10	MLO	F.14	F.4
6	X.loc	F.20	F.3	F.21	F.65	F.63	F.21	F.55	F.64	F.21	F.20	F.3
7	F.21	F.58	F.12	X.loc	F.9	F.20	X.loc	F.26	F.6	X.loc	F.58	F.12
8	Y.loc	F.21	F.35	Y.loc	F.12	F.62	Y.loc	F.57	F.4	Y.loc	F.21	F.35
9	X.nipple.loc	F.68	F.6	X.nipple.loc	F.55	F.21	X.nipple.loc	F.68	F.3	X.nipple.loc	F.68	F.65
10	F.3	F.9	F.65	F.3	F.26	F.60	F.3	F.17	F.5	F.3	F.9	F.64
11	F.4	F.35	F.5	F.4	F.58	F.12	F.4	F.56	F.21	F.4	F.12	F.6
12	Y.nipple.loc	F.12	F.64	Y.nipple.loc	F.17	F.3	Y.nipple.loc	F.105	F.63	Y.nipple.loc	F.17	F.5
13	F.1	F.17	F.58	F.12	F.64	F.4	F.1	F.54	F.12	F.1	F.10	F.63
14	F.12	F.10	F.63	F.1	F.105	F.58	F.12	F.5	F.58	F.12	F.35	F.58
15	F.2	F.105	F.68	F.2	F.54	F.6	F.2	F.6	F.62	F.2	F.105	F.62

Tabela 3.6: Selecção de atributos - Comparação de diferentes técnicas de amostragem, *ranking* dos primeiros 15 atributos

amostras escolhidas aleatoriamente dos casos negativos. Esta redução de domínio tem impacto na determinação do peso de atributos e no modelo de classificação, recorre-se ao algoritmo de Monte Carlo para testar diferentes conjuntos de amostras construídos aleatoriamente. A técnica de *oversampling* consiste na repetição de casos do conjunto minoritário até igualar o conjunto maioritário. O conjunto de dados final é reduzido para 49.840 amostras, das quais 24.920 amostras são do conjunto maioritário e as restantes são obtidas pela replicação do conjunto minoritário. Os conjuntos resultantes passam então por todo o processo descrito na secção 3.2.2 e são posteriormente usados nas etapas seguintes.

A selecção de atributos com os novos conjuntos revela que a adopção das diferentes técnicas de amostragem produz resultados distintos na determinação dos pesos dos atributos, confirmando o impacto esperado nos resultados, ver tabela 3.6. A tabela 3.6 é uma versão alargada da tabela 3.4, cujos campos da tabela são os resultados, sem, com *undersampling* e *oversampling* dos algoritmos para obtenção de pesos dos atributos, precedidos por *Under* e *Over* respectivamente. A aplicação da técnica de *undersampling* faz emergir, a relevância dos atributos `F.3`, `F.4`, `F.5` e `F.6`. Por exemplo, `F.5` e `F.6` não aparecem nos primeiros 15 atributos com mais peso no conjunto de dados original. Os atributos `F.3` e `F.4` destacam-se na posição cimeira, com maior peso face aos outros atributos, quando comparado com o resultado obtido sem balanceamento. O mesmo não acontece com `F.9` e `F.10` cujo peso reduziu significativamente. Os resultados da aplicação da técnica de *oversampling* também produzem impacto nos pesos de cada atributo, não tanto nos lugares do topo da tabela, mas nos lugares intermédios com a introdução de novos atributos nestas posições.

A confirmação da melhor técnica a seguir só pode ser feita em fases posteriores, com o teste e determinação do erro dos modelos de classificação, ver capítulo 4.

4

Modelo de detecção e previsão

Para classificar as áreas potencialmente cancerígenas recorre-se a um exame de rastreio do cancro da mama, vulgarmente conhecido por mamografia. Deste, resulta um conjunto de informação necessária a aferir através de um modelo de DM. No entanto, este objectivo contém outro aspecto importante tal como a redução do erro na classificação de pacientes, com especial ênfase para os pacientes doentes erroneamente classificados como saudáveis. Neste caso, o objectivo é que, após a selecção e optimização, um destes modelos tenha uma sensibilidade de 100% [94].

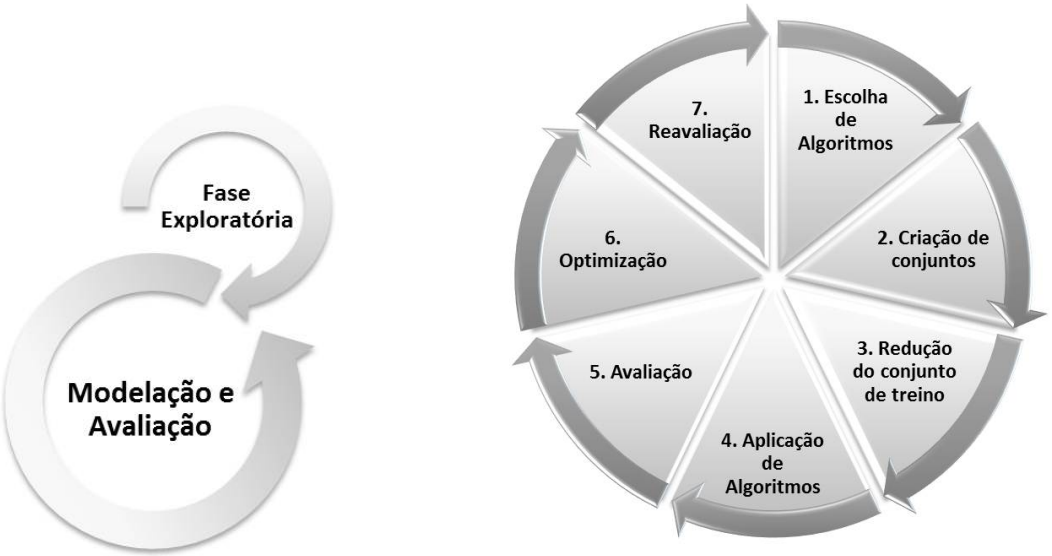
Entende-se por sensibilidade a capacidade dos modelos preverem correctamente todos os casos positivos, designadamente os TP [88, 130, 131]. Os pacientes doentes devem ser correctamente classificados. Outro parâmetro importante é a especificidade. Os modelos devem apresentar a maior especificidade possível, a especificidade é a capacidade do modelo classificar correctamente os pacientes saudáveis, Verdadeiros Negativos (TN) [88, 130, 131]. Se a relevância do primeiro parâmetro parece indiscutível dado que a finalidade do modelo é detectar a presença de cancro e todos reconhecemos a importância de não classificar como saudável uma pessoa doente. O segundo parâmetro também é bastante importante pois implica a realização de menos exames complementares de diagnóstico. Desta forma serão reavaliados apenas os casos mais relevantes, libertando assim os recursos, e possibilitando a realização de mais exames em tempo útil para a

detecção precoce de cancro e assim aumentar probabilidade de um doente superar a doença. Além disso, evita-se a realização de exames invasivos em pacientes saudáveis [63, 117].

Sendo que é difícil obter sensibilidade e especificidade máximas, a relação de compromisso que se pretende é atingir os 100% de sensibilidade e redução máxima dos falsos positivos, isto é, máxima especificidade [94].

Um dos objectivos proposto no desafio do KDD Cup 2008 foi o de obter uma percentagem entre 0,2 e 0,3 de falsos alarmes por imagem [94]. Significa isto que no total do conjunto de dados apenas poderiam ser encontrados entre 1.370 e 2.054 falsos alarmes, dado que o conjunto contém informação sobre 1.712 pacientes, o que corresponde a 6.848 imagens, e dado que cada paciente possui 4 imagens referentes às duas perspectivas de cada uma das mamas do paciente (MLO e CC).

De acordo com a metodologia CRISP-DM, descrita no capítulo 3, os passos que se seguem são a modelação e a avaliação. O objectivo desta fase é desenvolver um modelo que se adeque ao problema. Para isso, são testados vários algoritmos, de diferentes famílias, sendo escolhidos os mais promissores, com melhor desempenho e que melhor se adequam ao cumprimento dos objectivos que se pretendem atingir.



(a) Ciclos do processo (b) Etapas do processo em cada ciclo

Figura 4.1: Processo de modelação e avaliação

Também dentro desta fase, existem algumas etapas que têm de ser alcançadas tais como a definição de uma metodologia de treino e teste, balanceamento do conjunto de treino, aplicação de algoritmos e avaliação. Nomeadamente, é criado

um processo de modelação e avaliação dos algoritmos. Este é composto pelas seguintes etapas, ver figura 4.1:

1. Escolha dos algoritmos
2. Criação dos conjuntos de treino e teste
3. Redução do conjunto de treino
4. Aplicação dos algoritmos
5. Avaliação
6. Optimização
7. Reavaliação

Numa primeira fase é feita uma modelação exploratória para obter uns resultados preliminares dos algoritmos e do seu desempenho. Pode-se considerar esta fase como apenas mais uma iteração do processo, mas pretende-se reforçar a sua importância na modelação.

Os primeiros resultados são importantes para avaliar quais os modelos que mais se destacam e assim definir o restante processo de modelação. Nesta fase pretende-se uma visão rápida, pelo que o processo é simplificado sem a exaustão necessária das iterações seguintes. Por exemplo, o uso de técnicas de balanceamento de dados implica uma redução do número de casos considerados. Para garantir a representatividade destes, é necessário adoptar técnicas que permitam validar os resultados obtidos.

Uma das técnicas é a de *Monte Carlo* que utiliza o mesmo algoritmo com distribuições diferentes para determinar a tendência dos resultados. Com esta técnica, tenta-se reduzir a influência de uma distribuição específica e não representativa, da aplicação do método a diferentes distribuições do conjunto de dados. As iterações seguintes, representadas pelo ciclo de maior dimensão (figura 4.1(a)) servem sobretudo para validar os resultados obtidos com testes mais exaustivos e mais representativos e realizar, pequenos ajustes nos algoritmos ou explorar novas opções. A figura 4.1(b) ilustra as etapas do processo de modelação em cada ciclo.

4.1 Etapa 1: Escolha de algoritmos

É necessário criar um modelo capaz de classificar uma amostra com base no conhecimento adquirido na observação, e encontrar padrões em dados existentes.

Os esquemas de aprendizagem podem ser variados pelo que é necessário escolher os algoritmos e técnicas mais adequadas. Esta é uma das tarefas essenciais na modelação. Para esta tarefa é ainda necessário, compreender os diversos algoritmos e técnicas para aplicá-los e parametrizá-los correctamente, ver subcapítulo 2.4. Não existe um processo instituído para a escolha dos algoritmos.

A escolha dos algoritmos teve por base três aspectos: (i) algoritmos usados em problemas semelhantes na área da medicina; (ii) algoritmos usualmente utilizados em comparações nos problemas de DM; e (iii) escolha de algoritmos representativos de diferentes famílias de algoritmos. Foi dada ênfase ao primeiro aspecto dado que o uso de um algoritmo em casos semelhantes dá uma boa noção da adequação de um determinado algoritmo para este problema.

Com base nesta premissa foram consultados alguns trabalhos, já referidos no subcapítulo 2.7. Serviu de base os trabalhos desenvolvidos por Perlich *et al.* [98], Rosset *et al.* [111], Bellaachia e Guven [6], Delen *et al.* [35], Ayer *et al.* [4], Bellazi e Zupan [7], Chen *et al.* [28], Malley *et al.* [86], cujo âmbito está ligado à aplicação de técnicas de DM à área médica. Também foram tidos em conta os trabalhos de Meyer *et al.* [89] e Wu *et al.*[127], onde foram escolhidos alguns algoritmos usualmente usados em comparações de desempenho e alguns representativos de diferentes famílias.

Dos métodos apresentados no capítulo 2.7, apenas alguns foram seleccionados de acordo com a aplicação desses algoritmos a casos semelhantes ou devido à curiosidade em algoritmos cujo o conhecimento sobre a sua aplicação a casos semelhantes é incipiente. Pretende-se num próximo trabalho regressar a este assunto e validar os seus resultados.

Destacam-se sobretudo os algoritmos SVM e o ANN. Vários autores referem o uso do SVM em casos semelhantes entre os quais, os vencedores do desafio do KDD 2008 [98, 111] e outros [4, 7, 86]. O ANN também é bastante usado nesta temática [3, 4, 6, 7, 35, 86]. Outros dois algoritmos destacam-se também pelos vários estudos já realizados sobre a sua aplicação a casos semelhantes, é o caso do LDA [4] e LR [4, 35].

A tabela 4.1 apresenta a lista dos algoritmos usados, família de algoritmos e os pacotes do *R* cuja implementação do algoritmo foi usada. A escolha do pacote do R é importante dado que diferentes pacotes contêm diferentes implementações de um mesmo algoritmo.

Algoritmo	Familia	Meta-Algoritmo	Algoritmo base	R Package
Adaboost J48	DT	Boosting	C4.5	Rweka
Bagging using RPART	DT	Bagging	RPART	ipred
CFOREST	DT	Bagging	conditional inference trees	party
CTREE	DT		conditional inference trees	party
Double - Bagging	DT e LM	Bagging	RPART e LDA	ipred
KNN	IBL		knn	class
Linear Discriminant Analysis	LM		LDA	MASS
Logistic Regression	LM		Logistic Regression	stats
Multinomial Log-linear Models	IBL		ANN	nnet
Multivariate Adaptive Regression Splines	LM		MARS	mda
Naive Bayes	SL		Naive Bayes	e1071
Neural Networks	IBL		ANN	nnet
OneR	RBL		1-R	Rweka
Quadratic Discriminant Analysis	LM		QDA	MASS
Random Forest	DT	Bagging	-	randomForest
RPART	DT		RPART	RPART
SOM Supervised - bdk	IBL		SOM	kohonen
SOM Supervised - XYF	IBL		SOM	kohonen
SOM Unsupervised	IBL		SOM	kohonen
SVM C-classification linear	IBL		SVM	e1071
SVM C-classification polynomial	IBL		SVM	e1071
SVM C-classification radial	IBL		SVM	e1071
SVM eps-regression linear	IBL		SVM	e1071
SVM eps-regression polynomial	IBL		SVM	e1071
SVM eps-regression radial	IBL		SVM	e1071
SVM nu-classification linear	IBL		SVM	e1071
SVM nu-classification polynomial	IBL		SVM	e1071
SVM nu-classification radial	IBL		SVM	e1071
SVM nu-regression linear	IBL		SVM	e1071
SVM nu-regression polynomial	IBL		SVM	e1071
SVM nu-regression radial	IBL		SVM	e1071
SVM one-classification linear	IBL		SVM	e1071
SVM one-classification polynomial	IBL		SVM	e1071
SVM one-classification radial	IBL		SVM	e1071

Tabela 4.1: Algoritmos utilizados. (Legenda: DT - Árvores de Decisão, LM - Modelos Lineares, IBL - *Instance-based Learning*, RBL - *Rule-based Learning*, SL - *Statistical Learning*)

4.2 Etapa 2: Criação dos conjuntos de treino e teste

A avaliação de um modelo implica a validação deste através de um conjunto aleatório de novas amostras. O KDD Cup 2008 disponibilizou dois conjuntos de dados, treino e teste. Estes conjuntos contêm informação de 1.712 e 1.000 pacientes, respectivamente. No entanto, o conjunto de teste não inclui informação do valor da classe e `Lesion.ID`. Não é assim possível usar este conjunto de teste para avaliar os resultados.

Segundo Witten *et al.* [126] e Kohavi *et al.* [75], a separação em conjuntos de treino e teste é uma parte importante da avaliação de modelos de DM, desde que o primeiro seja representativo do segundo e permita a sua validação.

A inexistência do conjunto de testes implica assim a adopção de uma estratégia para efectivar essa validação, como por exemplo, o *hold-out*, o *k-fold cross-validation* e o *bootstrap*, descritos no subcapítulo 2.3. O *hold-out* é uma boa opção dada a sua simplicidade, aceitação generalizada e pela sua aplicabilidade ao problema em causa. Mas como foi referido no subcapítulo 2.3.1, é necessário parametrizar esta técnica, como por exemplo, o rácio do conjunto de dados para treino em relação ao conjunto de dados original. Uma vez que a classe minoritária tem poucos casos e o conjunto é bastante desequilibrado, optou-se pela divisão genericamente usada nos processos de DM, em que $2/3$ dos dados são usados no conjunto de treino e $1/3$ no conjunto de teste [75, 126]. A representatividade dos dados é preservada através da escolha aleatória dos casos que fazem parte de cada subconjunto e manutenção da proporcionalidade entre casos considerados malignos e benignos, tanto no conjunto de treino como no de testes. Foram usadas duas técnicas distintas, divisão tendo em conta as amostras e divisão tendo em conta os pacientes, ambas variantes da técnica *hold-out*.

A primeira técnica consiste na identificação das amostras que representam os casos malignos e benignos presentes no conjunto inicial. São criados dois subconjuntos, um com as amostras correspondentes aos casos benignos e outro subconjunto com as amostras dos casos malignos. $2/3$ de cada um destes novos subconjuntos é colocado no conjunto de treino, independentemente do paciente a que pertencem e o restante é colocado no conjunto de teste.

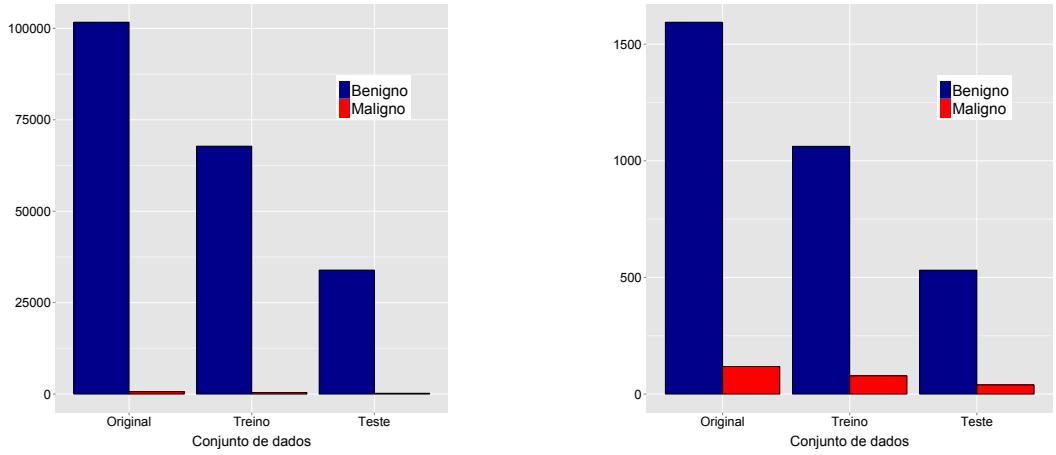
A segunda técnica é um pouco mais complexa. Consiste primeiramente na agregação dos casos por paciente e classificação desse paciente. A identificação do paciente é dada pelo atributo *Study.ID*, e é este atributo que serve para agregar as amostras dos exames efectuados por cada paciente, ver tabela 4.2. Esta tabela contém apenas uma pequena fracção de todos os casos do paciente 14280 que no total contém 133 amostras das quais apenas 3 amostras são consideradas malignas. No entanto basta apenas uma amostra positiva para que o paciente seja considerado como doente de cancro, é o caso deste paciente.

A divisão do conjunto de dados por paciente consiste na colocação de todos os casos de um determinado paciente em apenas um dos conjuntos. Por exemplo, se este paciente for escolhido aleatoriamente para fazer parte do conjunto de treino, todas as suas amostras farão parte desse mesmo conjunto. Tem-se assim que todas as amostras de $2/3$ dos pacientes identificados como saudáveis e $2/3$ dos pacientes dados como portadores da doença são colocados no conjunto de treino. De referir que também nesta técnica, a escolha dos pacientes que constituem o conjunto de treino é feito de forma aleatória mas mantendo a proporcionalidade

M.M	I.F.ID	S.F.ID	Image.ID	Study.ID	L.B	MLO	X.l	Y.l	X.n.l	Y.n.l	F.1	F.2
Benign	0	0	100499	14280	Left	MLO	1812	1818	2543	2483	0,084363752	0,22992367
Benign	0	0	100499	14280	Left	MLO	1099	1384	2543	2483	0,17940398	0,20656769
Benign	0	0	100508	14280	Right	MLO	1518	1916	937	2425	0,092573207	0,1076179
Malign	112567	110419	100508	14280	Right	MLO	2685	1867	937	2425	0,086889738	0,22115597
Benign	0	0	100508	14280	Right	MLO	1689	1758	937	2425	0,14846065	-0,059189482
Malign	112567	110419	100508	14280	Right	MLO	2684	1869	937	2425	0,11878031	0,22019815
Benign	0	0	100517	14280	Left	CC	1108	1779	2387	2354	0,14656616	0,12765836
Malign	112561	110419	100526	14280	Right	CC	2959	1312	1054	2080	0,017740865	0,20479941

Tabela 4.2: Exemplo de agregação dos dados de um paciente (Parte das 133 amostras deste paciente). (Legenda: M.M - Malignant.Mass, I.F.ID - Image.Finding.ID, S.F.ID - Study.Finding.ID, L.B - Left.Breast, X.l - X.location, Y.l - Y.location, X.nipple.location - X.n.l, Y.n.l - Y.nipple.location)

entre pacientes com cancro e sem cancro por forma a manter a representatividade dos dados. Esta particularidade pode levar a que a proporcionalidade entre casos malignos e positivos difira um pouco entre os subconjuntos de treino e teste. A figura 4.2 ilustra a divisão que foi efectuada. Observa-se a grande desproporcionalidade entre os casos das duas classes.



(a) Por amostra (b) Por paciente

Figura 4.2: Divisão do conjunto de dados

4.3 Etapa 3: Redução do conjunto de dados de treino

Como foi descrito anteriormente, no capítulo 3, o conjunto de dados é extenso e por isso a manipulação de todos os dados é inviabilizada em alguns contextos, dada a falta de recursos e lentidão dos processos. Nestes casos é essencial reduzir a dimensão dos dados.

A primeira redução do conjunto prende-se com a eliminação dos atributos identificadores das imagens e dos pacientes. Estes atributos não são usados na modelação porque se pretende uma independência face à proveniência das amostras. As amostras provêm de diferentes locais, pelo que, o identificador pode ser facilmente associado a determinada origem. A proporcionalidade entre casos, nos conjuntos de origem, é diferente, o que pode condicionar a análise dos casos com cancro [98]. A eliminação dos identificadores permite a redução de 4 atributos. O uso destes identificadores pode deturpar os resultados na classificação de uma amostra. Dada a dimensão do conjunto de dados, esta redução é insignificante. Assim, segue-se uma outra técnica.

Como referido no subcapítulo 2.2.2, existem três possibilidades, redução do número de atributos, redução do número de casos ou uma combinação destas duas.

A primeira abordagem centra-se na redução do número de atributos, redução de colunas. Nos subcapítulos 3.2.1 e 3.2.2 foram introduzidas duas técnicas que são usadas neste contexto, eliminação de redundâncias e selecção de atributos. A primeira técnica consiste no cálculo da correlação entre atributos, utilizando somente o conjunto de treino nesse cálculo, mantendo a independência face ao conjunto de teste. Os atributos com maior correlação são retirados do conjunto de treino dado que uma elevada correlação indicia uma redundância na informação que se pode retirar dos dois atributos. Como referido em 3.2.1, considera-se que correlações superiores a $|0,90|$ revelam uma elevada dependência entre atributos. Deste modo, pode-se escolher qualquer intervalo entre $[0,90;1]$, dependendo da quantidade de informação que se quer prescindir e do número de variáveis que se pretende retirar do conjunto de treino. Do conjunto verifica-se que existem 37 pares de atributos com uma correlação de 0,99 a 1. Destes, foi possível retirar 26 atributos do conjunto de treino. Em alguns casos pretende-se ir um pouco mais longe e retirar mais alguns atributos. Para o efeito, aplica-se uma outra técnica designada por o *Feature Ranking* (FR), que consiste na identificação dos atributos que mais contribuem para a classificação da amostra ou melhor, que mais peso têm nessa determinação. Neste caso opta-se por considerar apenas $2/3$ dos atributos do conjunto. Assim e juntamente à eliminação de redundâncias, o conjunto fica reduzido a 66 atributos, menos de metade dos atributos do conjunto inicial que contém 128 atributos. A aplicação destas técnicas, por si só, não produziu os resultados desejados dado que o treino dos modelos provocou a falta de recursos no seu processamento com o consequente erro e fecho inesperado da aplicação. Algo já esperado dados os problemas mencionados no capítulo 3.

A segunda abordagem consiste na redução do número de casos para treino, técnica de *Undersampling*. Esta consiste na redução do número de casos da classe maioritária e manutenção do número de casos da classe minoritária. A redução da classe maioritária é obtida com a escolha aleatória sem repetição destas amostras. Assim obteve-se um conjunto de dados reduzido a 830 amostras, com igual distribuição de casos negativos e positivos. Esta abordagem permitiu a realização dos testes dos vários algoritmos, dado que a redução da dimensão dos dados é significativa. E, que vai de encontro com outros trabalhos já referidos. Optou-se pela não repetição de casos da classe maioritária dado o grande desequilíbrio entre as duas classes. Foi ainda considerada uma outra técnica de *Undersampling* que consiste na agregação das amostras por paciente. Caso tenha sido usada a técnica de divisão do conjunto de treino por paciente, implica que este conjunto é composto por cerca de 79 pacientes com cancro. De salientar que, feita uma agregação das amostras por paciente, todas as amostras dos pacientes diagnosticados com cancro, mesmo as que não indiciam a presença de cancro, são consideradas como fazendo parte do conjunto minoritário. São considerados para o conjunto de treino todos os pacientes com cancro e igual número de pacientes classificados como não tendo cancro. O conjunto de treino final contém 158 pacientes. Uma vez que, em média, cada paciente tem 59 amostras, o conjunto de dados é cerca de 9.300 amostras. Apesar da grande dimensão para a maioria dos modelos, já permite a obtenção de resultados. Esta abordagem só pode ser feita se for usada a divisão do conjunto de treino por paciente. Caso contrário, o conjunto pode apresentar pelo menos uma amostra de todos pacientes com cancro e, consequentemente tem-se um conjunto de treino com 236 pacientes, dos quais 118 com cancro e, 13.000 amostras.

Uma terceira abordagem é a combinação das técnicas anteriores. O uso das várias técnicas em conjunto podem melhorar o desempenho dos algoritmos.

Uma das técnicas que também pode ser usada neste caso é a PCA. O objectivo é, por transformação ortogonal, converter um conjunto de observações de variáveis possivelmente correlacionadas a um conjunto de valores de variáveis linearmente descorrelacionadas. O n. de componentes principais é muito menor ao n. de variáveis originais. Pode explicar a elevada proporção da variação total associada ao conjunto original [108]. Se assim for, a aplicação desta técnica permite escolher apenas as componentes que captam um determinado valor de variância do conjunto, as restantes são retiradas pois apresentam redundância entre dados. Nos testes realizados, esta técnica é usada no conjunto de treino para obter as n componentes principais que captam 99% da variância total. O primeiro passo é

efectuar a PCA, onde se obtém as componentes e a variância que cada uma capta. De seguida a variância de cada uma das componentes é somada até ser igual ou superior ao valor que se pretende captar. As componentes necessárias para perfazer esse valor são mantidas e as restantes são retiradas do conjunto. Verificou-se que são necessárias apenas 61 componentes para captar pelo menos 99% da variância do conjunto de treino, ou seja, trata-se de uma redução significativa da cardinalidade do conjunto tendo em conta que o conjunto original contém 124 atributos. Se mesmo assim for necessário reduzir mais a dimensão do conjunto ou analisar o desempenho do modelo, esta técnica pode ser usada em conjunto com as técnicas descritas anteriormente.

As técnicas usadas estão identificadas no quadro resumo, tabela 4.3.

		Tudo	FS			PCA	PCA + FS		
			ER	FR	ER + FR		ER	FR	ER + FR
Tudo		-	-	-	-	-	-	-	-
Undersampling	Amostra	X	X	X	X	X	X	X	X
	Paciente	-	-	-	-	-	-	-	X
Oversampling	Amostra	-	-	-	-	-	-	-	-
	Paciente	-	-	-	-	-	-	-	-
SMOTE		X	X	X	X	X	X	X	X

Tabela 4.3: Resumo das técnicas de redução da dimensão do conjunto de dados. (Legenda: X Técnicas implementadas, - Técnicas não implementadas, ER - Eliminação de Redundâncias, FR - Feature Ranking, FS - Selecção de atributos)

4.4 Etapa 4: Aplicação dos algoritmos

Como foi referido inicialmente, a modelação tem dois ciclos ou fases. Um ciclo exploratório dos dados que se pretende rápido para obtenção dos primeiros resultados com uma visão abrangente, mas não detalhada, da aplicação dos algoritmos e que servem de guia para o restante processo. Este ciclo foi repetido várias vezes introduzindo progressivamente novos algoritmos e afinando alguns pormenores do processo. No segundo ciclo um dos objectivos é a consolidação dos resultados obtidos na fase exploratória. Os algoritmos são executados repetidas vezes, em vários conjuntos de dados com distribuições diferentes, visando uma maior representatividade dos resultados obtidos. A diversidade de conjuntos e distribuições permitem validar se determinado algoritmo teve uma boa ou má classificação devido ao conjunto de dados ou à distribuição dos dados.

4.4.1 Fase exploratória

Tendo os objectivos definidos e escolhidos os modelos, esta é a fase em que são feitos alguns testes preliminares para adquirir alguma sensibilidade e analisar o desempenho dos algoritmos. Desde logo se observou que a cardinalidade e o balanceamento do conjunto de dados introduzem algumas dificuldades na aplicação dos algoritmos. O grande número de casos e atributos impediu a realização desta fase. Foram detectados dois problemas, elevado tempo de processamento e falta de memória aquando a aplicação[1]. Este problema foi ultrapassado através da redução do conjunto de dados de treino. Por um lado, através da redução de atributos (Colunas) e por outro, através da redução do número de amostras (Linhas). A redução do conjunto de dados é feita respeitando a proporcionalidade entre as duas classes. Para balancear o conjunto são utilizadas técnicas como o *Undersampling* e o SMOTE. A técnica de *Oversampling* é adequada para conjuntos de reduzida dimensão [69],o que não se verifica neste problema. Para conjuntos de dados com esta dimensão é recomendado o uso da técnica de *Undersampling* [69]. Existe ainda a possibilidade de utilização de uma técnica mista com *Undersampling* da classe maioritária e *Oversampling* da classe minoritária mas esta opção não foi explorada. Opta-se por utilizar um método semelhante de *Oversampling*, que consiste na geração de amostras sintéticas da classe minoritária, SMOTE, referido no subcapítulo 2.2.2. Assim esta fase consiste na execução dos diversos algoritmos sob o mesmo conjunto e distribuição algumas vezes mas, sem a preocupação que os resultados sejam verdadeiramente representativos. Os resultados aqui obtidos servem apenas como preparação e orientação para os passos seguintes. A tabela 4.4 apresenta uma medida de desempenho dos algoritmos, o AUC, em diferentes distribuições dos dados. Observa-se que o NB é um dos algoritmos com melhor desempenho na maioria das distribuições.

Analisando os resultados observa-se que na maioria dos casos, os resultados com a utilização de *undersampling* e FR numa divisão por amostra é superior aos das restantes técnicas. Dos 33 algoritmos, 20 obtiveram melhores resultados usando esta técnica. Se excluirmos as diferentes variantes do SOM e SVM, ficamos com apenas 15 algoritmos e 8 dos quais apresentaram resultados idênticos. Esta característica foi tida em conta na próxima fase, com uma exploração de técnicas associadas à redução dos atributos.

[1]Foi utilizado um computador equipado com processador Intel(R) Core(TM) i7 M620 a 2,67GHz, 4GB de memória e sistema operativo Windows 7 a 32 bits

Algoritmo	Amostra				Paciente	
	Undersampling			SMOTED	Undersampling	
	FR	ER	Original	Original	FR	ER
Adaboost J48	0,90134	0,87606	0,87081	0,88457		
Bagging using RPART	0,91995	0,90303	0,90773	0,90058		
CFOREST	0,92286	0,91842	0,91889	0,91776		
CTREE	0,86282	0,84776	0,84776	0,88208	0,87171	0,86709
Double - Bagging	0,92642	0,89766	0,90312	0,90809		0,85393
KNN	0,50481	0,45245	0,50787	0,49021		0,48042
Linear Discriminant Analysis	0,91315	0,91997	0,91330	0,89586		0,87613
Multinomial Log-linear Models	0,92408	0,91672	0,89771	0,90231		0,92020
Multivariate Adaptive Regression Splines	0,81773	0,74174	0,77711	0,69891		0,88896
Naive Bayes	0,95522	0,90852	0,89953	0,90727	0,93771	0,92763
Neural Networks	0,81519	0,81189	0,79218	0,85843	0,87020	0,73832
OneR	0,79096	0,78264	0,78264	0,77584		
Quadratic Discriminant Analysis	0,85850	0,82929	0,81954	0,81439		0,81518
Random Forest	0,93120	0,92425	0,92507	0,91997	0,80284	0,80082
RPART	0,81340	0,85216	0,84249	0,83688	0,60765	0,60825
SOM Supervised - bdk	0,85201	0,83134	0,80198	0,81060		0,76963
SOM Supervised - XYF	0,84525	0,80920	0,81513	0,79860		0,70771
SOM Unsupervised	0,79368	0,76798	0,73536			0,75868
SVM C-classification linear	0,92684	0,92665	0,91965	0,91394	0,91939	0,91333
SVM C-classification polynomial	0,92429	0,86282	0,89158	0,88319	0,85280	0,82608
SVM C-classification radial	0,93434	0,86601	0,88308	0,86950	0,85342	0,83534
SVM eps-regression linear	0,90038	0,90665	0,91024	0,90581		
SVM eps-regression polynomial	0,89119	0,76464	0,81577	0,90947		
SVM eps-regression radial	0,93204	0,86583	0,87662	0,86980		
SVM nu-classification linear	0,93231	0,92497	0,92197	0,92208		
SVM nu-classification polynomial	0,93067	0,91500	0,91209	0,90578		
SVM nu-classification radial	0,92967	0,86480	0,90753	0,87410		
SVM nu-regression linear	0,91506	0,91749	0,91734	0,90828		
SVM nu-regression polynomial	0,89530	0,76144	0,81543	0,91790		
SVM nu-regression radial	0,93173	0,86472	0,87740	0,86754		
SVM one-classification linear	0,51779	0,51723	0,51310	0,51769	0,51484	0,52614
SVM one-classification polynomial	0,44973	0,48506	0,50067	0,44994	0,65507	0,61987
SVM one-classification radial	0,53958	0,51877	0,53856	0,52518	0,36898	0,39536

Tabela 4.4: Resultados preliminares. (Legenda: FR - Selecção de atributos, ER - Eliminação de Redundâncias)

Por outro lado, a divisão do conjunto de treino por paciente apenas obteve melhores resultados em 4 algoritmos, sendo que dois deles não têm muita expressão dado o resultado da AUC ser inferior a 0,65 e como tal, ligeiramente superior a uma escolha aleatória da classificação de um atributo. Dados os problemas referidos acima, não foi possível ir mais para além do descrito. Num trabalho futuro pensa-se regressar a este assunto. Comparando os resultados que consideram exclusivamente a eliminação de redundâncias, verifica-se que dos 20 algoritmos analisados, a divisão por paciente foi superior em 7 algoritmos. Este facto, conjugado com as dificuldades de testar os diversos algoritmos neste conjunto de

dados, também foi tido em conta na fase posterior, dando-se por isso preponderância ao estudo dos algoritmos utilizando a divisão dos conjuntos por amostra. Isto não implica que este tipo de divisão não deva ser estudado, indica apenas a prioridade no seu estudo por forma a optimizar os recursos.

Por outro lado, esta fase permitiu ainda identificar os algoritmos mais promissores. O NB, surgiu destacado com o melhor resultado, com uma AUC de 0,95. Nos lugares imediatamente seguintes destaca-se um conjunto de variantes do SVM (*C-classification radial*, *nu-classification linear*, *SVM eps-regression radial* e *nu-regression radial*), com valores de AUC acima dos 0,93. Também acima dos 0,93 surge o RF. Referência ainda para os algoritmos *double-bagging*, MLM e *CFOREST*, acima dos 0,92 e, o *bagging* e LDA acima dos 0,91. Não quer isto dizer que os restantes algoritmos não sejam adequados para este problema mas dadas as poucas parametrizações que foram feitas e o conjunto de dados em questão, tiveram neste caso um desempenho inferior. Relembra-se que nesta fase, os algoritmos foram testados com poucas ou nenhumas alterações face aos parâmetros por omissão. No capítulo seguinte veremos se estes resultados são confirmados ou existe uma alteração significativa no desempenho destes algoritmos.

4.4.2 Fase Modelação

Como foi dito no subcapítulo anterior, é necesssário garantir que os resultados são representativos do conjunto de dados. Uma das formas de o fazer, é repetindo o mesmo algoritmo com diferentes distribuições. Pretende-se confirmar, ou validar, que os resultados são independentes da distribuição e convergem para um mesmo valor, excluindo ainda a hipótese de escolher um conjunto ou balanceamento mais favorável, cujo resultado não seja reproduzível em situações semelhantes. Esta questão pode surgir se por acaso, um conjunto estiver demasiadamente especializado na classificação de um determinado conjunto de treino. Deve-se então perceber, até que ponto, determinado conjunto de dados influencia ou não o resultado final da aplicação de um algoritmo. Para despistar este problema e tal como em situações anteriores, ver capítulo 3, foi usada uma técnica inspirada no método de *Monte Carlo*. A técnica consiste na criação de cinco conjuntos de treino e teste, aleatórios e distintos, cada um deles com cinco distribuições, perfazendo um total de vinte e cinco distribuições de dados diferentes, ver figura 4.3. Dada a dimensão dos dados e para se obter um maior grau de certeza, seria necessário executar os algoritmos num maior conjunto de distribuições. Esta técnica torna-se pouco prática neste tipo de conjuntos, não obstante a

sua representatividade.

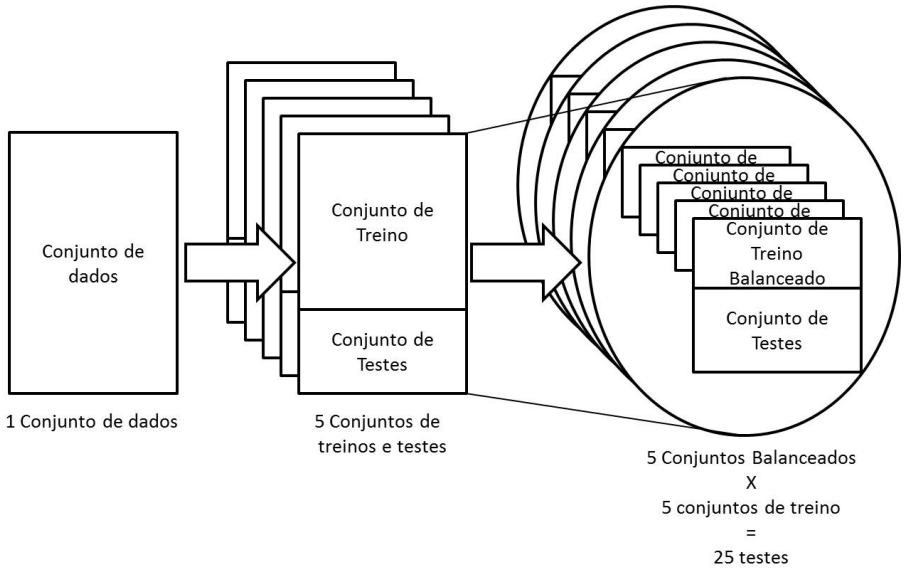


Figura 4.3: Ilustração do método utilizado na execução dos algoritmos

Os testes foram repetidos para cada uma das técnicas descritas na tabela 4.3 e que foram assinaladas como utilizadas nesta tese. Nesta fase, nenhum dos algoritmos propostos na fase anterior foi excluído. Pretende-se avaliar o impacto do conjunto no seu desempenho.

O desenvolvimento desta etapa envolveu a estruturação do código usado nas etapas anteriores, para que fosse possível realizar os testes de uma forma iterativa, aplicando um conjunto de algoritmos sobre diversos conjuntos de dados com diferentes distribuições.

Os algoritmos só são passíveis de ser comparados se forem realizados sob as mesmas condições, neste caso, usando os mesmos conjuntos de dados com as mesmas distribuições. Para garantir esta condição é necessário guardar os diferentes conjuntos ou realizar exactamente os mesmos passos na sua criação. Como os conjuntos são gerados de forma aleatória, ou melhor, de forma pseudo-aleatória é necessário guardar a semente usada na geração dos conjuntos e no seu balanceamento. Das 100 criadas, são usadas as primeiras cinco sementes na divisão do conjunto de dados e posteriormente no balanceamento do mesmo, o que perfaz os 25 conjuntos testados. Este conjunto de sementes foi guardado num ficheiro para poder ser utilizado nos diferentes testes. Trata-se de uma pré-condição necessária para salvaguardar a possibilidade de repetição dos testes.

O programa desenvolvido consiste, numa primeira fase, na configuração das técnicas utilizadas (listagem A.4) e em dois ciclos onde se itera pelos conjuntos de

treino, segundo a divisão e balanceamento. Os conjuntos são criados em cada iteração (listagem A.5, linha 17 e listagem A.8). Segue-se a remoção de atributos e balanceamento dos dados (listagem A.5, linha 35-45). Neste passo, existe um procedimento relevante no caso de se usar a PCA. Como foi dito nos subcapítulos 3.2.3 e 4.3, o uso da PCA implica a projecção dos atributos num outro plano. A PCA é aplicada unicamente ao conjunto de treino, pois é importante que se mantenha a independência em relação ao conjunto de teste. No conjunto de treino passam a figurar as componentes em vez dos atributos, pelo que é necessário transformar o conjunto de teste, aplicando a rotação obtida no processo de determinação das componentes principais. Este processo está descrito na listagem A.12.

O passo seguinte é a aplicação dos algoritmos. Para cada algoritmo, foi criada uma função que recebe como parâmetros, os conjuntos de treino e teste, além do nome do ficheiro onde é guardada a compilação dos resultados obtidos. O corpo principal do programa, desenvolvido para o efeito, chama as funções de cada algoritmo (listagem A.6).

Cada função está estruturada da seguinte forma: (i) preparação dos dados ou pré-processamento, (ii) escolha dos melhores parâmetros, (iii) aprendizagem do algoritmo, (iv) pós-processamento, (v) cálculo do tempo de execução, (vi) classificação das amostras do conjunto de treino, (vii) cálculo de probabilidade de cada amostra, e (viii) guardar os dados obtidos. Existem alguns passos que não foram implementados por alguns dos algoritmos, como por exemplo, o primeiro passo. A maioria dos algoritmos não necessitam de pré-processamento dos dados de treino. As listagens A.10 e A.11 ilustram duas funções, uma que envolve a aplicação do algoritmo NB e a segunda que envolve a aplicação do algoritmo SOM. Estas são diferentes, o SOM necessita de uma preparação do conjunto de dados para ser aplicado o algoritmo, porque trabalha com dados numéricos. Daí a necessidade de transformação dos dados antes da sua aplicação. O segundo passo foi utilizado na parametrização do SVM. Dada a complexidade de parametrização, foi usada uma função (`tune`) que encontra os melhores parâmetros para este algoritmo. O quarto passo só é usado pelo algoritmo *Recursive Partitioning and Regression Trees* (RPART) para "podar" a árvore criada.

No final os resultados são compilados num ficheiro para poderem ser avaliados na etapa posterior (listagem-A.7).

4.5 Etapa 5: Avaliação dos modelos

Nesta etapa são analisados e avaliados os resultados obtidos, aplicando as métricas de avaliação descritas no subcapítulo 2.6. Esta primeira avaliação serve de base na orientação do próximo passo de optimização dos algoritmos.

A estratégia desta fase passa por fazer uma apresentação geral dos resultados e por detalhar a análise segundo as técnicas utilizadas. Partindo dos resultados genéricos para os resultados mais específicos, com o objectivo de avaliar quais as técnicas que melhor contribuem para a obtenção dos melhores resultados.

Dos 33 algoritmos usados neste estudo, são seleccionados alguns para ilustrar pormenorizadamente o problema.

Os principais indicadores usados são os AUC, TP, TN, FP e FN. Para facilitar a compreensão e visualização dos resultados, é apresentada, quando possível, a percentagem de TP, FN, TN e FP. Estes são usados na avaliação de cada um dos casos e na classificação de cada um dos pacientes, tendo por base a agregação dos casos de cada um destes. Para isso é necessário agregar os resultados das classificações das amostras por paciente, obtendo as n amostras de um paciente, usando como função agregadora por omissão, a função *max*. Considera-se que, uma amostra positiva tem o valor 1 e a amostra negativa o valor -1. Através desta função obtém-se o valor máximo do conjunto, ou seja, basta uma amostra positiva para o paciente ser considerado positivo, logo com cancro (ver listagem A.9). Esta agregação pode ser parametrizada para obter uma classificação média das amostras ou usar uma outra métrica. Neste caso, foi usada a métrica definida por omissão, isto é, o paciente é considerado como tendo cancro se uma das amostras for classificada como cancerígena (valor 1).

4.5.1 Análise geral

A primeira análise consiste na agregação dos resultados obtidos por cada algoritmo independentemente das técnicas usadas, isto é, não diferenciando as técnicas de selecção de atributos e balanceamento, que foram aplicadas. A tabela 4.5 apresenta a média dos resultados obtidos por cada algoritmo.

Sem surpresas, o NB é o algoritmo com melhor desempenho em termos de AUC com 0,956. Este lugar no topo da tabela era previsível dado os resultados nos testes preliminares. Além do mais, o resultado agora obtido está um pouco acima do resultado observado na fase anterior. Porém, essa diferença não é significativa. O

conjunto utilizado nos testes preliminares encontra-se próximo da média destes testes.

São confirmadas as boas prestações dos algoritmos RF e *double-bagging*, dois algoritmos da família das árvores de decisão, se bem que o *double-bagging* resulta da combinação de uma árvore com um algoritmo linear, o LDA.

A classificação obtida pelo *double-bagging* é bastante interessante se tivermos em conta que este meta-algoritmo combina um conjunto de algoritmos com uma prestação bastante inferior. É o caso dos dois algoritmos de base, LDA e *RPART*, e do meta-algoritmo *bagging*. Este resultado confirma a teoria de que modelos eventualmente mais fracos podem-se tornar fortes quando usados em conjunto [59].

O SVM também tem boas prestações em várias das suas parametrizações, com excepção do SVM *one-classification*. Não se trata de algo inesperado, dado que nos testes preliminares já se registava este facto. O SVM *one-classification* é usado sobretudo para detecção de novos dados (termo designado em inglês por *novelty detection*), isto é, dado um conjunto de dados com uma distribuição normal, este algoritmo indica se a amostra em causa está ou não dentro da distribuição normal [116].

Menção ainda para os algoritmos baseados em modelos lineares, MLM e LR, que se encontram na primeira metade da tabela. De realçar, um desempenho pouco conseguido do algoritmo ANN, um dos mais usados em estudos comparativos do género e em que muitos destes, apontam-no como um dos algoritmos com melhor desempenho com dados médicos (ver subcapítulos 2.4.4.3 e 4.1). Este facto tem de ser desvalorizado, dada a utilização deste algoritmo com a maioria dos parâmetros preenchidos com os valores por omissão. Além do mais, a sua parametrização é complexa, o que exige um estudo aprofundado deste algoritmo, tendo como objectivo a sua optimização para este conjunto de treino.

Num olhar ingénuo para a tabela 4.5, e observando apenas a coluna dos TP, comparando NB com MARS, pode-se depreender que o algoritmo MARS se aproxima mais do objectivo uma vez que tem um maior número de casos TP. Mas observando mais atentamente as restantes colunas, verifica-se que este resultado não tem a mesma expressão nos restantes indicadores, com excepção do FN, que está intimamente relacionado com o número de TP. A diferença entre NB e MARS reside essencialmente no número de TN. O MARS tem uma percentagem de casos TP mais elevada porque por defeito classifica todos os casos como positivos, daí a percentagem de 98% de casos negativos erradamente classificados como

Algoritmo	AUC	TP	TN	FP	FN	% TP	% TN	% FP	% FN
Naive Bayes	0,95639	180	31306	2585	28	0,86678	0,92373	0,07627	0,13322
Random Forest	0,93263	167	30654	3237	41	0,80499	0,90449	0,09551	0,19501
Double - Bagging	0,93072	171	30240	3651	37	0,82154	0,89229	0,10771	0,17846
SVM C-classification linear	0,92549	174	29292	4599	34	0,83499	0,8643	0,1357	0,16501
Multinomial Log-linear Models	0,92443	174	29035	4856	34	0,83748	0,85673	0,14327	0,16252
SVM nu-classification linear	0,92417	169	29832	4059	39	0,81248	0,88023	0,11977	0,18752
SVM nu-regression linear	0,924	173	29164	4727	35	0,83383	0,86053	0,13947	0,16617
SVM nu-regression radial	0,92376	154	31092	2799	54	0,73845	0,9174	0,0826	0,26155
SVM eps-regression radial	0,9234	153	31025	2866	55	0,73781	0,91543	0,08457	0,26219
SVM C-classification radial	0,92306	153	31068	2823	55	0,73743	0,91671	0,08329	0,26257
Linear Discriminant Analysis	0,92258	173	28951	4940	35	0,83239	0,85425	0,14575	0,16761
Logistic Regression	0,92218	173	29081	4810	35	0,83058	0,85808	0,14192	0,16942
SVM eps-regression linear	0,92152	176	28314	5577	32	0,84681	0,83545	0,16455	0,15319
SVM nu-classification polynomial	0,92107	163	30579	3312	45	0,78578	0,90227	0,09773	0,21422
CFOREST	0,92048	168	29426	4465	40	0,80955	0,86826	0,13174	0,19045
Bagging using RPART	0,9187	167	29807	4084	41	0,80365	0,87949	0,12051	0,19635
SVM nu-classification radial	0,91821	149	31056	2835	59	0,71415	0,91634	0,08366	0,28585
Adaboost J48	0,90503	165	29169	4722	43	0,79327	0,86066	0,13934	0,20673
SVM C-classification polynomial	0,90353	141	30554	3337	67	0,67825	0,90152	0,09848	0,32175
SVM nu-regression polynomial	0,90099	168	28643	5248	40	0,80644	0,84516	0,15484	0,19356
SVM eps-regression polynomial	0,89381	153	29049	4842	55	0,73549	0,85713	0,14287	0,26451
Neural Networks	0,87559	165	28731	5160	43	0,79566	0,84774	0,15226	0,20434
CTREE	0,8737	160	28398	5493	48	0,77108	0,83792	0,16208	0,22892
Quadratic Discriminant Analysis	0,86382	161	27257	6634	47	0,77626	0,80426	0,19574	0,22374
RPART	0,84715	164	28365	5526	44	0,78808	0,83694	0,16306	0,21192
Multivariate Adaptive Regression Splines	0,84274	205	456	33435	3	0,98764	0,01344	0,98656	0,01236
SOM Supervised - XYF	0,82754	188	20583	13308	20	0,90229	0,60734	0,39266	0,09771
SOM Supervised - bdk	0,82664	188	20513	13378	20	0,90381	0,60526	0,39474	0,09619
SOM Unsupervised	0,78884	171	22648	11227	37	0,82167	0,66858	0,33142	0,17833
OneR	0,77244	155	27026	6865	53	0,74745	0,79743	0,20257	0,25255
KNN	0,52069	163	23438	10453	45	0,78332	0,69156	0,30844	0,21668
SVM one-classification polynomial	0,51872	77	22646	11245	131	0,36924	0,6682	0,3318	0,63076
SVM one-classification linear	0,50709	89	19805	14086	119	0,42981	0,58437	0,41563	0,57019
SVM one-classification radial	0,49668	92	18709	15182	116	0,44133	0,55202	0,44798	0,55867

Tabela 4.5: Média dos resultados por algoritmo

positivos.

Comparando a matriz de confusão, tabelas 4.6 e 4.7, de NB e MARS respectivamente, verifica-se que o NB tem uma sensibilidade relativamente menor mas uma especificidade muito maior que o MARS. Isso reflete-se na exactidão dos algoritmos, em que o NB tem uma exactidão de 0,923 contra 0,019 do MARS. Logo o algoritmo NB é mais adequado aos objectivos propostos.

A próxima análise consiste na avaliação dos resultados quanto à classificação dos pacientes. A tabela 4.8 apresenta a média dos resultados obtidos por paciente, ordenada pelo valor da AUC por amostra. Verifica-se que o NB não teve um resultado tão bom nesta comparação. Comparando exclusivamente a AUC, o melhor resultado foi obtido pelo SVM *nu-regression radial*, com 0,89148. Este desempenho fica aquém do desejável visto que classifica erradamente 6 pacientes doentes como saudáveis, relembra-se que um dos objectivos é classificar correctamente os pacientes doentes (ver capítulo 1). Existem outros algoritmos com uma maior precisão nestes casos, como por exemplo, o próprio NB. O problema do NB, nesta comparação, reside no número de TN. O NB classifica incorrectamente

		Prediçao		
		Positivo	Negativo	
Actual	Positivo	180	28	Sensibilidade $= 0,865$
	Negativo	2585	31306	Especificidade $= 0,923$
		Positivos Correctamente Identificados $= 0,065$	Negativos Correctamente Identificados $= 0,999$	

Tabela 4.6: Matriz Confusão NB

		Prediçao		
		Positivo	Negativo	
Actual	Positivo	205	3	Sensibilidade $= 0,986$
	Negativo	33435	456	Especificidade $= 0,013$
		Positivos Correctamente Identificados $= 0,006$	Negativos Correctamente Identificados $= 0,993$	

Tabela 4.7: Matriz Confusão MARS

818 pacientes, mais 15 pacientes que o SVM *nu-regression radial*. Relembra-se que este é o segundo objectivo, como tal, pretende-se um algoritmo com um bom desempenho nestes dois indicadores.

Os resultados apresentados nas tabelas 4.5 e 4.8 dizem respeito à média da classificação dos algoritmos, independentemente das técnicas utilizadas, trata-se de uma visão de conjunto. Mas é necessária uma visão mais detalhada, para apurar o impacto destas na classificação de um algoritmo.

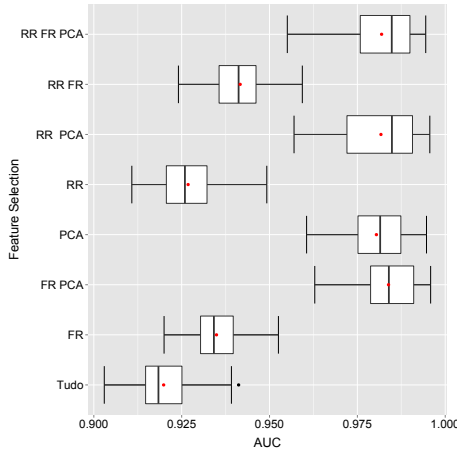
Algoritmo	AUC	TP	TN	FP	FN	% TP	% TN	% FP	% FN
Naive Bayes	0,85472	92	799	818	3	0,97142	0,4943	0,5057	0,02858
Random Forest	0,88828	91	760	858	4	0,95951	0,46975	0,53025	0,04049
Double - Bagging	0,88207	91	690	928	3	0,96565	0,42635	0,57365	0,03435
SVM C-classification linear	0,87424	92	579	1038	3	0,96808	0,35793	0,64207	0,03192
Multinomial Log-linear Models	0,86912	92	547	1071	3	0,97079	0,33804	0,66196	0,02921
SVM nu-classification linear	0,88488	91	627	990	4	0,96101	0,38754	0,61246	0,03899
SVM nu-regression linear	0,87525	92	550	1067	2	0,97368	0,34027	0,65973	0,02632
SVM nu-regression radial	0,89148	88	814	803	6	0,93423	0,50322	0,49678	0,06577
SVM eps-regression radial	0,88888	89	800	817	6	0,93616	0,49467	0,50533	0,06384
SVM C-classification radial	0,88569	88	815	803	6	0,9348	0,5037	0,4963	0,0652
Linear Discriminant Analysis	0,87097	92	527	1090	2	0,97613	0,32588	0,67412	0,02387
Logistic Regression	0,86618	92	548	1069	3	0,97022	0,33896	0,66104	0,02978
SVM eps-regression linear	0,86819	93	463	1154	2	0,98105	0,28643	0,71357	0,01895
SVM nu-classification polynomial	0,86567	91	685	932	4	0,962	0,42341	0,57659	0,038
CFOREST	0,8678	92	594	1023	3	0,96806	0,36745	0,63255	0,03194
Bagging using RPART	0,85834	92	603	1015	3	0,97049	0,37265	0,62735	0,02951
SVM nu-classification radial	0,86879	87	800	818	7	0,92517	0,49441	0,50559	0,07483
Adaboost J48	0,84902	92	485	1133	2	0,97745	0,29962	0,70038	0,02255
SVM C-classification polynomial	0,82539	81	727	890	13	0,86079	0,44938	0,55062	0,13921
SVM nu-regression polynomial	0,81748	93	436	1181	2	0,97925	0,26951	0,73049	0,02075
SVM eps-regression polynomial	0,81183	88	517	1101	7	0,9304	0,31946	0,68054	0,0696
Neural Networks	0,76971	69	487	1130	26	0,72996	0,30141	0,69859	0,27004
CTREE	0,78041	92	433	1184	2	0,97504	0,26759	0,73241	0,02496
Quadratic Discriminant Analysis	0,77045	93	320	1297	2	0,9834	0,19811	0,80189	0,0166
RPART	0,69527	92	456	1161	3	0,97216	0,28222	0,71778	0,02784
Multivariate Adaptive Regression Splines	0,76446	95	0	1617	0	0,99995	0,00015	0,99985	5e-05
SOM Supervised - XYF	0,70171	90	56	1561	4	0,95418	0,03466	0,96534	0,04582
SOM Supervised - bdk	0,69879	90	55	1562	4	0,95335	0,0343	0,9657	0,04665
SOM Unsupervised	0,55524	88	73	1530	5	0,94153	0,0452	0,9548	0,05847
OneR	0,59738	91	370	1247	3	0,96589	0,22888	0,77112	0,03411
KNN	0,5009	94	118	1499	0	0,9971	0,07327	0,92673	0,0029
SVM one-classification polynomial	0,51039	93	60	1557	2	0,98339	0,03739	0,96261	0,01661
SVM one-classification linear	0,50458	94	22	1596	0	0,99581	0,01336	0,98664	0,00419
SVM one-classification radial	0,50659	93	48	1570	2	0,98378	0,0294	0,9706	0,01622

Tabela 4.8: Média dos resultados por algoritmo (Paciente)

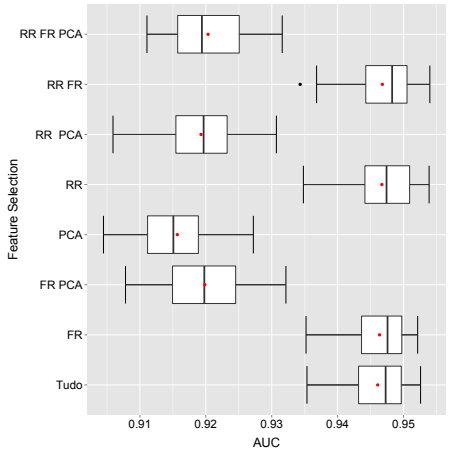
4.5.2 Análise detalhada por técnica de selecção de atributos

A próxima análise incide na comparação e avaliação dos resultados obtidos por cada algoritmo tendo em conta a utilização de diferentes técnicas de selecção de atributos. Dos 33 algoritmos, escolhemos quatro exemplos distintos: (i) NB, (ii) RF, (iii) *Double-bagging* e, (iv) SVM *C-classification linear*. Note-se que se trata dos quatro algoritmos que estão no topo da lista de resultados, (ver tabela 4.5), mas independentemente disso, apresentam resultados tão distintos.

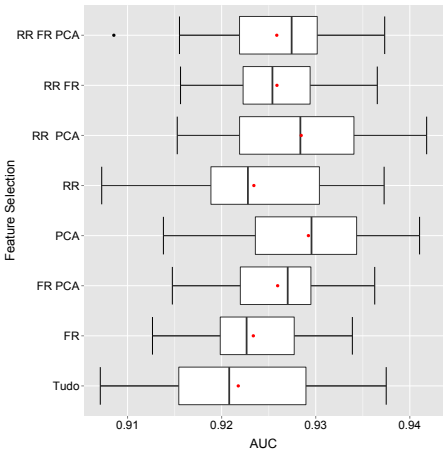
A figura 4.4 mostra a distribuição dos resultados dos quatro algoritmos, segundo a técnica usada na redução de atributos. Verifica-se que o uso da PCA tem impacto no resultado dos algoritmos, mas distinto conforme o algoritmo em causa. As duas primeiras figuras 4.4(a) e 4.4(b) mostram dois casos curiosos. No caso do RF, os resultados são penalizados ficando um pouco abaixo da média, 0,932, se contabilizarmos todos os testes independentemente das técnicas usadas. Contrariamente, o uso da PCA incrementa os resultados do NB com valores acima



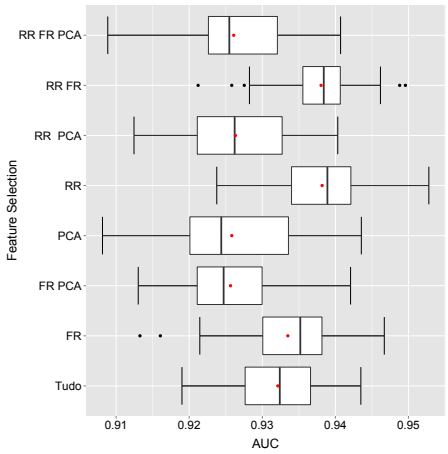
(a) Naive Bayes



(b) Random Forest



(c) SVM C-classification linear



(d) Double-bagging

Figura 4.4: Impacto da selecção de atributos. Com indicação da média (ponto), mediana (segmento de recta dentro da *boxplot*), *outliers* (pontos fora da *boxplot*), primeiro e terceiro quartis (extremidade da *boxplot*), margem de erro (segmentos de recta na extremidade da *boxplot*)

da média de 0,956. E isso acontece para todos os casos em que foi usada a PCA conjugada com outras técnicas de selecção de atributos, como a eliminação de redundâncias, FR ou as duas em conjunto. Além disso, observa-se uma separação clara entre os casos com e sem PCA. Situação que não se verifica na figura 4.4(c). Os resultados são uniformes e a fronteira entre casos com e sem PCA é difusa. Exemplo disso é o uso de eliminação de redundâncias em conjunto com o FR, onde PCA tem um impacto pouco significativo, sendo a média muito semelhante. No quarto caso, figura 4.4(d), a fronteira também é difusa mas a PCA penaliza um pouco o desempenho do algoritmo.

Pode-se desde já concluir que caso se opte por optimizar estes algoritmos, têm

de ser usadas estratégias diferentes de selecção de atributos, de acordo com o algoritmo utilizado.

4.5.3 Análise detalhada por técnica de balanceamento

O próximo passo é avaliar o impacto do balanceamento dos dados. Foram usadas duas técnicas, *undersampling* e SMOTE.

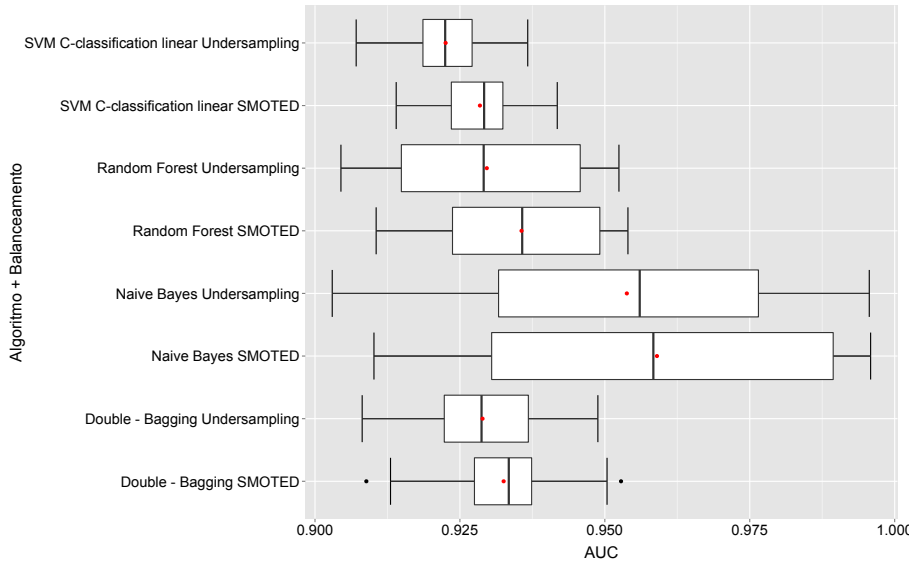


Figura 4.5: Impacto do balanceamento

A figura 4.5 apresenta os resultados dos quatro algoritmos com melhor desempenho de acordo com o balanceamento. Observa-se que os resultados obtidos através da aplicação do SMOTE são um pouco melhores comparativamente ao uso do *undersampling*. Verifica-se que apesar da média e mediana do algoritmo NB ser consideravelmente maior que qualquer outro algoritmo, os resultados têm uma maior variação comparativamente com o SVM *C-classification linear*. Aliás, este último pode não ter os melhores resultados mas tem os resultado mais consistentes, sempre muito perto da sua média. Considerando apenas estes resultados, seria aconselhável realizar mais testes com mais conjuntos de dados para avaliar se a distribuição do NB é mesmo assim tão dispersa e por isso, menos representativa do conjunto de dados. Mas considerando os resultados observados no subcapítulo 4.5.2, verifica-se que a diferença de desempenho está relacionada com o uso de PCA. Como se verifica na figura 4.4, o uso de PCA melhora significativamente o desempenho deste algoritmo e existe uma fronteira clara entre o uso ou não da PCA. Mas também é verdade, que até este momento o NB não

apresentou *outliers*, e comparativamente, o *double-bagging* apresenta aqui dois *outliers* no balanceamento por SMOTE.

Com base nos resultados observados neste subcapítulo, pode-se concluir, que o uso de diferentes técnicas de balanceamento tem impacto nos resultados da maioria dos algoritmos, sendo que, o SMOTE apresenta uns resultados ligeiramente superiores ao uso de *undersampling*.

4.5.4 Análise detalhada por conjunto de dados

Outra observação pertinente é a do desempenho dos algoritmos tendo em conta o conjunto de dados, isto é, conjuntos com diferentes distribuições de dados.

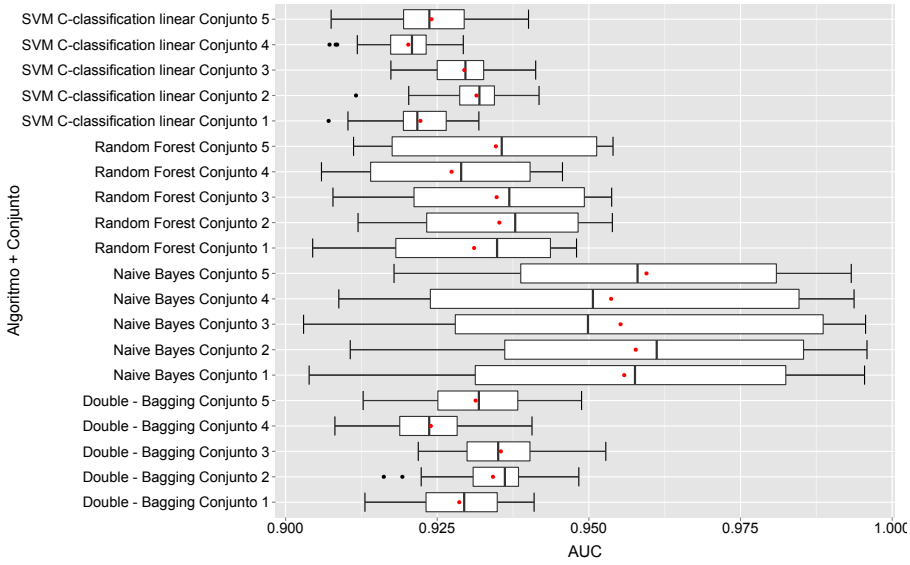


Figura 4.6: Representatividade dos conjuntos

A figura 4.6 mostra as distribuições do desempenho dos algoritmos tendo em conta o conjunto de dados. Observando a figura nota-se que existe alguma variação e que alguns conjuntos potenciam um pouco os resultados dos algoritmos, como por exemplo o conjunto 2 para os algoritmos SVM *C-classification linear* e RF, contrariamente ao conjunto 4 que penaliza os resultados de todos os algoritmos. Ora, com base nestes resultados torna-se evidente a necessidade de executar os algoritmos várias vezes, com diferentes conjuntos, para que se encontre um conjunto representativo do conjunto original ou mais provavelmente, se encontre uma média representativa da aplicação de um determinado algoritmo a um conjunto de dados. Relembra-se que de acordo com a simulação do método de

Monte Carlo, a aplicação sucessiva de um mesmo algoritmo a subconjuntos aleatórios do conjunto original, é equivalente à aplicação de um algoritmo no conjunto de dados original.

4.5.5 Análise detalhada por técnica de selecção de atributos e balanceamento

Como vimos na figura 4.5, o NB apresenta uma distribuição aproximadamente uniforme da AUC entre 0,92 e 0,99 face ao balanceamento, tendo o SMOTE alcançado um desempenho ligeiramente melhor. Contudo, a figura 4.4 mostra algo absolutamente diferente, existe uma fronteira clara entre os resultados obtidos com ou sem PCA. É com base nestas premissas que se explora uma outra possibilidade, a conjugação das técnicas de selecção de atributos com as técnicas de balanceamento.

Algoritmo	**AUC**	**TP**	**TN**	**FP**	**FN**	**% TP**	**% TN**	**% FP**	**% FN**
Naive Bayes FR PCA SMOTED	0,98919	194	33304	587	14	0,93077	0,98268	0,01732	0,06923
Naive Bayes RR PCA SMOTED	0,98868	186	33424	467	22	0,89462	0,98622	0,01378	0,10538
Naive Bayes RR FR PCA SMOTED	0,98834	188	33242	649	20	0,90423	0,98084	0,01916	0,09577
Naive Bayes PCA SMOTED	0,98472	187	33375	516	21	0,89846	0,98476	0,01524	0,10154
Naive Bayes FR PCA Undersampling	0,97855	180	32635	1256	28	0,86308	0,96295	0,03705	0,13692
Naive Bayes PCA Undersampling	0,97612	173	32578	1313	35	0,8325	0,96125	0,03875	0,1675
Naive Bayes RR FR PCA Undersampling	0,97545	173	32580	1311	35	0,83135	0,96132	0,03868	0,16865
Naive Bayes RR PCA Undersampling	0,97474	169	32581	1310	39	0,81385	0,96135	0,03865	0,18615
Random Forest RR SMOTED	0,949	167	31507	2384	41	0,80404	0,92965	0,07035	0,19596
Random Forest RR FR SMOTED	0,94856	167	31385	2506	41	0,80481	0,92606	0,07394	0,19519
Random Forest FR SMOTED	0,94822	167	31389	2502	41	0,80462	0,92616	0,07384	0,19538
Random Forest Tudo SMOTED	0,9478	167	31486	2405	41	0,80212	0,92905	0,07095	0,19788
Random Forest RR FR Undersampling	0,945	181	29582	4309	27	0,87096	0,87287	0,12713	0,12904
Random Forest FR Undersampling	0,9445	181	29612	4279	27	0,86962	0,87374	0,12626	0,13038
Random Forest RR Undersampling	0,94441	181	29636	4255	27	0,87038	0,87444	0,12556	0,12962
Random Forest Tudo Undersampling	0,94435	181	29664	4227	27	0,87192	0,87528	0,12472	0,12808
Naive Bayes RR FR Undersampling	0,94287	183	30067	3824	25	0,87846	0,88718	0,11282	0,12154
Naive Bayes RR FR SMOTED	0,94042	181	29984	3907	27	0,87135	0,88471	0,11529	0,12865
Double - Bagging RR SMOTED	0,94012	167	31326	2565	41	0,80077	0,92431	0,07569	0,19923
Double - Bagging RR FR SMOTED	0,93929	165	31305	2586	43	0,79269	0,9237	0,0763	0,20731

Tabela 4.9: Média dos resultados por algoritmo, selecção de atributos e balanceamento

A tabela 4.9 lista a média dos algoritmos segundo o balanceamento e a selecção de atributos. São listados apenas os 20 melhores resultados, dada a extensão da lista completa (538 resultados).

O NB é o algoritmo com melhor desempenho, ocupando os 8 primeiros lugares com 8 variantes distintas (diferentes conjugações de técnicas), mas cujo denominador comum é o uso de PCA. Trata-se de um resultado expectável face ao exposto no parágrafo anterior. A variante com melhor desempenho surge da combinação de várias técnicas como o FR, PCA e SMOTE. A técnica SMOTE com

a PCA representa, neste conjunto de resultados, uma melhoria média de 0,0115 na AUC face ao uso de *undersampling* com PCA. O melhor desempenho deste algoritmo sem PCA, em termos de AUC, foi de 0,942. O uso de PCA representa por isso um acréscimo da AUC, em média, superior a 3%.

Olhando para outros indicadores, verificamos que o uso de NB com PCA e FR, tem o melhor resultado de TP, com 194 amostras positivas bem classificadas e 587 FP, só ultrapassado neste último indicador por duas variantes do NB com PCA e SMOTE. É o caso da variante que usa exclusivamente a PCA (516 FP) e da variante com eliminação de redundância (467 FP).

Observando o FP e o FN, verifica-se que as oito primeiras variantes do NB estão dentro dos limites definidos para o segundo objectivo deste trabalho, menos de 2.054 amostras mal classificadas, ver início do capítulo 4. No entanto o primeiro objectivo não é atingido, o NB com FR, PCA e SMOTE, é o que mais se aproxima falhando apenas na classificação de um paciente com cancro (ver tabela 4.10).

Algoritmo	**AUC**	**TP**	**TN**	**FP**	**FN**	**% TP**	**% TN**	**% FP**	**% FN**
Naive Bayes FR PCA SMOTED	0,92551	93	1215	402	1	0,98453	0,75157	0,24843	0,01547
Naive Bayes RR PCA SMOTED	0,93502	91	1286	332	3	0,96686	0,79493	0,20507	0,03314
Naive Bayes RR FR PCA SMOTED	0,91022	93	1179	438	2	0,97803	0,72899	0,27101	0,02197
Naive Bayes PCA SMOTED	0,93458	91	1259	358	3	0,96708	0,77873	0,22127	0,03292
Naive Bayes FR PCA Undersampling	0,84471	92	929	688	3	0,97068	0,5748	0,4252	0,02932
Naive Bayes PCA Undersampling	0,83473	91	907	710	4	0,95747	0,56111	0,43889	0,04253
Naive Bayes RR FR PCA Undersampling	0,83064	92	900	718	3	0,96859	0,55629	0,44371	0,03141
Naive Bayes RR PCA Undersampling	0,82806	90	906	711	5	0,95142	0,56007	0,43993	0,04858
Random Forest RR SMOTED	0,90853	90	972	645	5	0,94769	0,60126	0,39874	0,05231
Random Forest RR FR SMOTED	0,90751	90	960	657	5	0,94813	0,59356	0,40644	0,05187
Random Forest FR SMOTED	0,90702	90	954	663	5	0,95024	0,58998	0,41002	0,04976
Random Forest Tudo SMOTED	0,90793	90	963	654	5	0,95199	0,59554	0,40446	0,04801
Random Forest RR FR Undersampling	0,90298	91	753	864	3	0,9633	0,46547	0,53453	0,0367
Random Forest FR Undersampling	0,90441	92	745	872	3	0,96881	0,46072	0,53928	0,03119
Random Forest RR Undersampling	0,903	91	755	862	3	0,96633	0,46716	0,53284	0,03367
Random Forest Tudo Undersampling	0,90419	92	747	870	3	0,96964	0,46204	0,53796	0,03036
Naive Bayes RR FR Undersampling	0,84054	92	561	1056	2	0,97607	0,34705	0,65295	0,02393
Naive Bayes RR FR SMOTED	0,86387	92	588	1030	3	0,9732	0,36341	0,63659	0,0268
Double - Bagging RR SMOTED	0,89673	90	872	745	4	0,9559	0,53938	0,46062	0,0441
Double - Bagging RR FR SMOTED	0,89238	90	879	738	4	0,95312	0,54372	0,45628	0,04688

Tabela 4.10: Média dos resultados por algoritmo, selecção de atributos e balanceamento (Paciente)

4.5.6 Resumo

Em termos absolutos, verificaram-se 14 testes a cumprir todos os objectivos, resultantes da aplicação do NB com PCA, em conjunto com outras técnicas. Considerando a média, os objectivos pretendidos não foram atingidos, mas os principais indicadores estão bem próximos. Assim, o melhor candidato resulta da

combinação do algoritmo NB com as técnicas FR e PCA para selecção de atributos e SMOTE para balanceamento do conjunto. No entanto, para que em média, este candidato possa atingir os objectivos propostos, falta apenas classificar correctamente um dos pacientes, pelo que a optimização pode fazer a diferença.

Tendo em conta que 14 algoritmos cumpriram todos os objectivos, é possível implementar um meta-algoritmo que conjuge estes modelos para classificação de uma nova amostra ou paciente.

4.6 Etapa 6: Optimização

Como foi referido no subcapítulo 2.5 existem várias possibilidades de optimização dos algoritmos: (i) procurar obter os conjuntos de dados mais representativos, (ii) alterar parâmetros do algoritmo de aprendizagem, (iii) maximizar a AUC, (iv) utilizar diferentes limiares de probabilidades na distinção de classes na fase de classificação ou (v) combinar vários algoritmos.

A estratégia passou por combinar algoritmos dentro do mesmo conjunto de dados. É essencial referir ainda que, por uma questão de simplificação do processo, numa primeira fase, o *ensemble* resulta da combinação dos algoritmos envolvidos no mesmo conjunto de teste. Isto é, com o mesmo conjunto, balanceamento e atributos. Esta análise visa perceber até que ponto os restantes algoritmos complementam a classificação atribuída pelo NB, optimizando as suas previsões.

Foi testada a combinação de vários algoritmos, escolhendo os n algoritmos com maior AUC. Foram usadas três variantes de voto na combinação dos algoritmos, (i) voto por maioria, (ii) média ponderada, e (iii) *rank* dos algoritmos (ver subcapítulo 2.5). No primeiro caso, todos os algoritmos têm o mesmo peso e uma instância é classificada tendo em conta a maioria dos votos. No segundo caso, é atribuído um peso a cada algoritmo e o seu voto é ponderado por esse peso. No terceiro caso, o peso de cada algoritmo é dado pelo seu *ranking*, por exemplo, se considerarmos apenas três algoritmos, o algoritmo que tiver maior AUC tem o maior peso. Neste caso, o primeiro algoritmo teria peso 3, os restantes algoritmos teriam um peso de 2 e 1, de acordo com a sua posição na ordenação.

Um segunda estratégia, é combinar algoritmos provenientes do mesmo conjunto de dados mas com distribuições e atributos diferentes. Isto é possível, porque todos partilham o mesmo conjunto de treino. No subcapítulo 4.5 observámos que os melhores resultados foram obtidos pelo NB combinado com a PCA. Foi criado um *ensemble* que reúne os melhores resultados do NB com a PCA, por

conjunto de dados. Este *ensemble* também pode ser designado de *bagging* uma vez que combina vários testes do mesmo algoritmo.

4.7 Etapa 7: Reavaliação

Esta etapa é muito semelhante à etapa descrita no subcapítulo 4.5 mas com a adição dos resultados obtidos na etapa de optimização (subcapítulo 4.6). É necessário avaliar se as alterações efectuadas nessa etapa realmente tiveram os resultados pretendidos, e incrementaram a qualidade dos algoritmos base.

Analisando a tabela 4.11, verifica-se que quanto menor o número de algoritmos envolvidos na criação do *ensemble*, melhor o resultado. E que apesar de se ter usado três métricas distintas na votação dos algoritmos, estas acabam por ter um resultado muito semelhante entre si. Mas o importante é comparar estes resultados com os apresentados na tabela 4.5, mais concretamente com o NB, o algoritmo com melhor desempenho nessa etapa. A combinação de 3 algoritmos significou um ligeiro acréscimo da AUC, tanto na classificação das amostras como na classificação do paciente. Contrariamente, os restantes indicadores foram piores. O TP decresceu de 180 para 175 e o número de FP aumentou para 3.608. A classificação de pacientes também sofreu um decréscimo, aumentando o número de FP para 883. Portanto, pode-se pensar que estamos perante indicadores contraditórios, aumento de AUC e decréscimo dos outros indicadores. Mas, a única coisa que estes valores querem dizer é que as classes tornaram-se mais separáveis entre si, se bem que a diferença não é significativa, e o ponto de corte da curva ROC mudou. Devemos ter em conta que os indicadores que temos em consideração derivam da curva ROC, mas num ponto de corte por omissão.

Algoritmo	**AUC**	**TP**	**TN**	**FP**	**FN**	**AUC(P)**	**TP(P)**	**TN(P)**	**FP(P)**	**FN(P)**
Ensemble Peso 3	0,9589	175	30283	3608	33	0,88569	91	735	883	3
Ensemble Voto 3	0,95856	175	30283	3608	33	0,88569	91	735	883	3
Ensemble Rank 3	0,95841	181	29752	4139	27	0,88569	92	661	956	3
Ensemble Peso 5	0,95518	175	30168	3723	33	0,86567	91	717	900	4
Ensemble Voto 5	0,95478	175	30168	3723	33	0,86567	91	717	900	4
Ensemble Rank 5	0,9537	175	30123	3768	33	0,86567	91	705	912	4
Ensemble Peso 7	0,95267	175	30213	3678	33	0,50458	91	716	901	4
Ensemble Voto 7	0,95228	175	30213	3678	33	0,50458	91	716	901	4
Ensemble Peso 9	0,95054	175	30161	3730	33	0,50659	91	704	914	4
Ensemble Rank 7	0,95023	175	30078	3813	33	0,50458	91	684	934	3

Tabela 4.11: Média dos resultados por tipo de *ensemble*
Nota: *Indicador*(P) indica os resultados por paciente

A tabela 4.12 lista os 5 primeiros resultados, agrupados por técnica de selecção

de atributos e balanceamento. Comparando com a tabela 4.9 observa-se que o melhor resultado foi obtido recorrendo às três técnicas, FR, PCA e SMOTE. No entanto a AUC é inferior, assim como todos os outros indicadores. De facto a utilização da PCA faz com que o NB fique num patamar superior aos restantes algoritmos, e neste caso, a composição com outros algoritmos não melhorou os resultados.

Algoritmo	**AUC**	**TP**	**TN**	**FP**	**FN**	**AUC(P)**	**TP(P)**	**TN(P)**	**FP(P)**	**FN(P)**
Ensemble Peso 3 FR PCA SMOTED	0,97785	173	31191	2700	35	0,87026	90	824	794	4
Ensemble Rank 3 RR FR PCA SMOTED	0,97774	189	29563	4328	19	0,86343	93	589	1029	1
Ensemble Rank 3 FR PCA SMOTED	0,97741	182	30462	3429	26	0,87026	92	694	924	3
Ensemble Rank 3 PCA SMOTED	0,97735	188	29783	4108	20	0,87564	93	607	1010	2
Ensemble Voto 3 FR PCA SMOTED	0,97698	173	31191	2700	35	0,87026	90	824	794	4

Tabela 4.12: Média dos resultados por tipo de *ensemble* e técnicas de redução do conjunto
Nota: *Indicador*(P) indica os resultados por paciente

Algoritmo	**AUC**	**TP**	**TN**	**FP**	**FN**	**AUC(P)**	**TP(P)**	**TN(P)**	**FP(P)**	**FN(P)**
Ensemble Rank 5	0,99622	194	33433	458	14	0,86428	93	1289	329	1
Ensemble Voto 5	0,99619	194	33449	442	14	0,86428	93	1297	321	1
Ensemble Peso 5	0,99619	194	33449	442	14	0,86428	93	1297	321	1
Ensemble Rank 7	0,99616	193	33415	476	15	0,8749	94	1278	340	1
Ensemble Peso 7	0,99612	194	33447	444	14	0,8749	94	1296	321	1
Ensemble Voto 7	0,99612	194	33447	444	14	0,8749	94	1296	321	1
Ensemble Peso 9	0,99599	194	33442	449	14	0,85749	93	1295	323	1
Ensemble Voto 9	0,99599	194	33442	449	14	0,85749	93	1295	323	1
Ensemble Rank 9	0,99597	193	33428	463	15	0,85749	93	1285	332	1
Ensemble Voto 3	0,99571	192	33460	431	16	0,79337	93	1302	315	2
Ensemble Peso 3	0,99571	192	33460	431	16	0,79337	93	1302	315	2
Ensemble Rank 3	0,99563	196	33346	545	12	0,79337	93	1235	382	1

Tabela 4.13: Média dos resultados por tipo de *ensemble*, *bagging* de NB
Nota: *Indicador*(P) indica os resultados por paciente

Uma última análise consiste na observação dos resultados da combinação de várias variantes do algoritmo NB. Comparando as tabelas 4.13 e 4.9, os algoritmos que estão no topo têm algumas diferenças. Verifica-se um ligeiro acréscimo da AUC, de 0,989 para 0,996 e um decréscimo de FP, de 587 para 458. Comparando a classificação de pacientes, verifica-se um decréscimo da AUC, 0,926 para 0,864, um decréscimo do número de FP, de 402 para 329, mas mantém-se a classificação errónea de um paciente. O decréscimo de AUC indica que as classes estão menos separadas entre si, e que por isso é mais difícil classificar alguns casos na fronteira das classes, no entanto foi possível reduzir o número de FP. Avaliando globalmente todos os indicadores, este algoritmo permitiu melhorar os classificadores obtidos no subcapítulo 4.4.2.

Como conclusão, verifica-se que o melhor algoritmo desta etapa é o *ensemble* dos cinco melhores algoritmos do NB com PCA em conjugação com as outras diferentes técnicas de selecção de atributos e balanceamento.

5

Conclusões

Neste capítulo é feita uma síntese do trabalho realizado, discussão de resultados e apontados alguns caminhos para trabalho futuro.

5.1 Síntese e discussão de resultados

Esta tese incidiu essencialmente na análise dos passos necessários para construir um modelo de DM na detecção de tumores em exames de rastreio, tendo em conta a especificidade deste domínio. A tolerância ao erro na área da saúde é mais reduzida, dado que envolve a vida humana e isso reflete-se nos principais indicadores usados na classificação de um método. Com base nestas especificidades foram definidos dois objectivos principais. Por um lado, obter a máxima sensibilidade na classificação dos pacientes e por outro, reduzir o número de incidências mal classificadas. Dado que nem sempre é possível optimizar estes dois indicadores, é necessário encontrar uma relação de compromisso entre estes.

Partindo de um conjunto de dados pré-estabelecido foram dados passos sucessivos na construção do modelo. Este processo é complexo pelo que foi necessário adoptar uma metodologia. Existem duas metodologias geralmente aceites e utilizadas neste processo, CRISP-DM e SEMMA. Foi adoptada a primeira, CRISP-DM considerar completa e adequada ao problema. O que se veio a confirmar durante todo o processo.

As primeiras duas etapas do CRISP-DM ocuparam cerca de 30% do tempo destinado a este trabalho. É nesta fase que se tem o primeiro contacto com o domínio do problema e com as ferramentas disponíveis. Existe um tempo de assimilação do domínio do problema e de adaptação às ferramentas que não é desprezável, além do tempo gasto em pesquisa e constituição da bibliografia base para os passos seguintes.

Da análise inicial aos dados surgiram logo os primeiros problemas, (i) a dimensão e o (ii) balanceamento dos dados, (iii) conjunto de teste e (iv) representatividade do conjunto. O primeiro condiciona o uso das ferramentas, dado os recursos necessários para o armazenamento e manipulação dos dados, e a segunda tem implicações na aprendizagem. O terceiro problema identificado nesta fase, foi a falta do atributo classificador da amostra do conjunto de teste, inviabilizando o seu uso neste trabalho. O último problema surgiu em consequência dos problemas anteriores, devido à utilização de técnicas que só usam parte dos dados e que por isso podem não representar o conjunto original.

O primeiro problema foi resolvido com o uso de técnicas de selecção de atributos, tais como a remoção dos atributos redundantes, a determinação dos atributos relevantes, a análise da componente principal e a conjugação destas três. Foram adoptadas duas soluções no balanceamento dos dados, SMOTE e *undersampling*, as soluções mais adequadas dada a dimensão do conjunto de dados. A resolução do terceiro problema passou por dividir o conjunto de dados em dois, um para treino e outro para teste. Por fim, e para garantir a representatividade dos resultados foi necessário executar os algoritmos diversas vezes, uma técnica inspirada no método de *Monte Carlo*.

A preparação dos dados foi uma etapa que exigiu pouco esforço dado que o conjunto de dados estava completo, e por isso foi limitada à importação dos dados para o *R*, análise e transformação de alguns atributos.

As etapas seguintes, modelação e avaliação, foram as mais representativas neste trabalho ocupando cerca de 70% do tempo, usado em pesquisa e desenvolvimento das ferramentas de análise. Sendo que, o tema desta tese não é novo e existem diversos trabalhos nesta área optou-se por fazer um estudo mais abrangente, comparando diversos algoritmos cuja aplicação nesta área não é muito conhecida e evitando assim a usual comparação entre árvores, ANN e SVM. Obviamente que esta opção trouxe algumas desvantagens, entre as quais, o pouco tempo para optimização dos algoritmos.

Verifica-se que as técnicas usadas no balanceamento e redução de atributos têm

impacto na execução dos algoritmos e, consequentemente, nos resultados. Pelo que foram feitos alguns testes mais exaustivos explorando esse impacto. Dos algoritmos escolhidos para este trabalho e tendo em conta os testes realizados, o NB foi o que teve melhor desempenho. Verificou-se que a aplicação de técnicas de selecção dos atributos tem um forte impacto nos algoritmos e em especial no NB. O uso de PCA incrementa substancialmente a prestação deste algoritmo contrariamente ao RF onde o desempenho é significativamente penalizado. Este é diferente conforme o algoritmo em causa. Menos significativo é a utilização de balanceamento, o SMOTE apresenta melhores resultados na maioria dos algoritmos.

O uso de meta-algoritmos, como o *ensemble* melhora o desempenho dos classificadores. Estes podem ser decompostos em duas partes essenciais, escolha e método de combinação dos algoritmos. O que os distingue, são as diferentes abordagens destas duas componentes. Por exemplo, o *bagging* combina o mesmo algoritmo com diferentes distribuições de dados, o RF é uma variante que altera os parâmetros das árvores que cria, o *boosting* altera as distribuições e o peso de cada variante de um mesmo algoritmo. Neste trabalho foi usada uma variante mais genérica que permite a combinação de algoritmos de diferentes famílias. Os resultados melhoraram mas não o suficiente para atingir os dois objectivos deste trabalho.

Os resultados obtidos não são directamente comparáveis com os resultados apresentados no subcapítulo 2.7, dado que os resultados da competição são obtidos sobre o conjunto de teste que não pode ser usado neste trabalho dada a inexistência de um atributo classificador. No entanto, se considerarmos que os conjuntos de treino e teste disponibilizados pela competição são representativos, pode-se efectuar uma comparação relativa, entre os resultados da competição e os resultados aqui apresentados. Porém, só é possível comparar com os resultados da segunda tarefa da competição. A AUC apresentada como resultado da primeira tarefa é obtido com base numa função disponibilizada pela entidade organizadora, numa outra linguagem, que não foi possível converter em tempo útil para validar os resultados.

Embora existam alguns algoritmos, que em condições únicas, cumprem os objectivos propostos, a média de resultados dos mesmos falha na classificação de alguns pacientes doentes. Além disso, os estudos efectuados não permitiram concluir de forma clara, qual o algoritmo mais adequado à detecção de cancro da mama, usando este conjunto de dados, uma vez que não foi possível fazer um estudo mais exaustivo sobre a parametrização de todos os algoritmos, facto

que terá penalizado os resultados de alguns destes, como por exemplo, o SVM e ANN. No entanto, foi usado um conjunto de técnicas que aplicadas ao conjunto de dados, demonstraram que o processo produz resultados significativamente melhores. Especialmente o uso de PCA e *ensemble*, conjuntamente com NB.

Mas se considerarmos apenas os conjuntos de algoritmos e técnicas que cumprem os objectivos propostos (sensitividade a 100% e falsos positivos por imagem entre 0,2 e 0,3), verifica-se que em média, estes têm um desempenho muito próximo do vencedor do segundo desafio. Comparativamente, estes conseguem uma especificidade média de 67,94% contra os 68,15% obtidos pelo vencedor. Este resultado possibilitaria a obtenção do segundo lugar uma vez que o segundo lugar da competição consegue uma especificidade de 64,68%. Relembra-se que os três primeiros lugares são ocupados por submissões de uma mesma equipa. Se for tida em conta a segunda equipa melhor classificada, a diferença de resultados é considerável, uma vez que essa equipa apenas conseguiu uma especificidade 17,42%. De ressalvar que os resultados obtidos pela equipa vencedora do desafio tem em conta os identificadores dos pacientes, e como tal, informação que não foi usada neste trabalho mas que lhes permitiu tirar partido para maximizar os seus resultados na competição.

5.2 Principais conclusões

Esta dissertação assumiu como principal objectivo a elaboração de um modelo de DM para detecção de tumores em exames de rastreio tendo como principais requisitos a detecção de todos os pacientes doentes, redução do número de pacientes incorrectamente classificados e redução do número de regiões de interesse mal classificadas.

Face aos objectivos propostos, conjunto de dados usado, conjunto de testes realizados e face ao exposto no subcapítulo 5.1, é possível apresentar as seguintes conclusões.

O NB revelou ser o algoritmo mais adequado para a resolução deste problema.

A combinação de várias variantes de um mesmo algoritmo, neste caso, *ensemble* de NB, demonstrou ser uma boa solução pois permitiu aumentar o desempenho deste mesmo algoritmo. Tendo-se revelado uma boa técnica de optimização deste algoritmo.

A introdução de técnicas de selecção de atributos, com especial ênfase para a PCA introduz uma melhoria significativa dos resultados.

Adicionalmente, o uso de SMOTE provoca um ligeiro acréscimo nos resultados sendo a técnica de balanceamento com melhor desempenho nos testes.

A metodologia adoptada revelou-se consistente e adequada, pois permitiu uma melhoria significativa dos resultados. Além do mais, verificou-se uma evolução dos resultados ao longo do desenvolvimento deste trabalho, através da conjugação das diferentes técnicas, demonstrando a importância da adopção desta metodologia neste trabalho.

A comparação de um vasto número de algoritmos é uma das contribuições mais importantes deste trabalho, sendo relevante a comparação de vários algoritmos sob as mesmas condições.

Verificou-se que existe um número considerável de algoritmos que atingiu um bom desempenho neste conjunto de dados. Este trabalho permitiu aumentar o conhecimento sobre o potencial destes algoritmos e a sua utilização em problemas semelhantes.

A combinação de diferentes técnicas no processo de DM e seu estudo, é outro contributo importante.

A optimização na parametrização, realizada apenas em alguns algoritmos, é um dos pontos fracos e que merece um trabalho a realizar mais tarde.

5.3 Trabalho futuro

No decorrer do trabalho foram surgindo diversos problemas mas também algumas soluções. Nem sempre foi possível seguir esses novos caminhos ou fazer um estudo mais exaustivo que permitisse tirar mais partido de determinada técnica. Algumas opções que foram tomadas, acabaram também por limitar as abordagens que se poderiam ter seguido. Tendo em conta estes aspectos, são apresentadas algumas sugestões de trabalho futuro.

A primeira sugestão prende-se com a divisão do conjunto de dados, em treino e teste. Usando a técnica de *hold-out*, seria interessante avaliar o impacto na aprendizagem, usando diferentes percentagens na divisão dos conjuntos, como por exemplo 50-50 ou 60-40. Outra hipótese é dividir o conjunto em 3 conjuntos, treino, validação e teste. Esta divisão permite utilizar um dos conjuntos para validação e ajuste dos algoritmos. Também permite ajustar o ponto de corte na classificação de um algoritmo e depois testar o impacto desses ajustes no conjunto

de teste. Ainda nesta temática, existe a possibilidade de aplicar o algoritmo *10-fold Cross-Validation* e aplicá-lo 10 vezes, perfazendo um total de 100 testes. Esta quantidade de testes permitiria dar mais garantias de representatividade, mas também representa um acréscimo de tempo de aprendizagem dos algoritmos. O seu desenvolvimento é mais complexo mas, se forem considerados somente os primeiros 10 testes, poderia dar uma primeira visão, com maior representatividade e rapidez. Poder-se-ia até concluir que 10 testes são suficientes e como tal reduzir o tempo de aprendizagem dos algoritmos.

Como se verifica neste trabalho, as técnicas usadas na escolha dos atributos teve impacto nos resultados, pelo que uma das hipóteses de trabalho futuro é variar alguns parâmetros dessas técnicas e avaliar o impacto no desempenho dos algoritmos. Sugere-se que se varie o limite acima do qual se considera que dois atributos são correlacionados, removê-los do conjunto de dados e assim verificar se a exclusão de mais atributos tem uma influência negativa ou positiva nos resultados. O mesmo se pode fazer em relação ao FR, considerando menos atributos relevantes e na PCA, variando o número de componentes consideradas ou alterando o valor mínimo de variância acumulada que se pretende captar.

Ainda no âmbito deste trabalho, sugere-se o uso das técnicas de *undersampling* e SMOTE com diferentes proporcionalidades entre casos minoritários e maioritários. Variar o número de casos, por exemplo menos casos da classe minoritária ou escolher os casos da classe maioritária de forma aleatória com reposição. Ou ainda, usar as amostras não usadas no treino, para teste. Dada dimensão do conjunto de dados não foi possível testar o *oversampling*, pelo que a sua aplicação a este conjunto de dados e consequente análise, pode ser relevante.

Outra sugestão para trabalho futuro prende-se com a utilização de diferentes algoritmos de aprendizagem. Sugere-se o uso de outros algoritmos, ou os mesmos, mas provenientes de outros pacotes do *R*, ou com implementações diferentes, como por exemplo, ANN com *backpropagation*. Fazer o estudo exaustivo dos parâmetros de cada algoritmo e registar o seu impacto na classificação. Criar vários tipos de *ensemble* com diferentes critérios de escolha e votação, por exemplo, escolha de algoritmos com base no número de TP em vez da AUC. Isso implicaria a escolha de algoritmos como o SOM ou o MARS onde existe um elevado numero de TP mas muitos FP. Experimentar a junção destes algoritmos com o NB e avaliar o impacto na classificação. Analisar o uso do SVM *one-classification* conjuntamente com outros algoritmos e verificar se este ajuda na descoberta de casos difíceis de classificar, utilizando então vários algoritmos para classificar esses casos.

Usar outras técnicas de pré e pós-processamento que permitam agregar efectivamente os casos de cada paciente, como por exemplo, calculando a média dos atributos por paciente ou por tipo de imagem. Usar métricas para correlacionar dados provenientes de diferentes imagens, como por exemplo, usar a distância da incidência ao mamilo.

Até este ponto foram apresentadas sugestões para evoluir este trabalho, mas mais importante é apontar sugestões para outros trabalhos com base neste e por isso seria interessante aplicar as técnicas aqui descritas a conjuntos de dados reais. Por exemplo, dados provenientes do Instituto Português de Oncologia, ou outras bases de dados provenientes de outros países, e avaliar os resultados. Será que estas técnicas têm bom desempenho quando aplicadas a conjuntos reais? Seria também interessante avaliar o desempenho dos algoritmos nesses conjuntos de dados e verificar se se mantém o desempenho dos 4 primeiros algoritmos deste estudo. Outra opção é aplicar estas técnicas a outro tipo de dados biomédicos.

Por último sugere-se a aplicação deste conjunto de técnicas a outras realidades distintas como categorização de texto, classificação de imagens, marketing, entre outras.

Bibliografia

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, volume 1215, pages 487–499, 1994.

[2] F.J. Anscombe. Graphs in statistical analysis. *The American Statistician*, 27 (1):17–21, 1973.

[3] Maria-Luiza Antonie, Osmar R Zaiane, and Alexandru Coman. Application of data mining techniques for medical image classification. In *MDM/KDD*, pages 94–101, 2001.

[4] Turgay Ayer, Mehmet US Ayvaci, Ze Xiu Liu, Oguzhan Alagoz, and Elizabeth S Burnside. Computer-aided diagnostic models in breast cancer screening. *Imaging*, 2(3):313–323, 2010.

[5] A. Azevedo and M. F. Santos. KDD, SEMMA, and CRISP-DM: A parallel overview. In *Proceedings of the IADIS European Conference on Data Mining*, pages 182–185, Amsterdam, 2008. IADIS.

[6] Abdelghani Bellaachia and Erhan Guven. Predicting breast cancer survivability using data mining techniques. *Age*, 58(13):10–110, 2006.

[7] Riccardo Bellazzi and Blaz Zupan. Predictive data mining in clinical medicine: current issues and guidelines. *International Journal of Medical Informatics*, 77(2):81–97, 2008.

[8] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305, 2012.

[9] L. Bernstein, B.E. Henderson, R. Hanisch, J. Sullivan-Halley, and R.K. Ross. Physical exercise and reduced risk of breast cancer in young women. *Journal of the National Cancer Institute*, 86(18):1403–1408, 1994.

[10] Christian Borgelt. Efficient implementations of apriori and eclat. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90*, page 90, 2003.

[11] N.F. Boyd, G.A. Lockwood, J.W. Byng, D.L. Tritchler, and M.J. Yaffe. Mammographic densities and breast cancer risk. *Cancer Epidemiology Biomarkers & Prevention*, 7(12):1133–1144, 1998.

[12] Andrew P Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[13] AC Braga. Curvas ROC: aspectos funcionais e aplicações. 2000.

[14] Ulf Brefeld and Tobias Scheffer. AUC maximizing support vector learning. In *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*. Citeseer, 2005.

[15] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[16] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[17] Leo Breiman, Jerome H Freidman, Richard A Olshen, and Charles J Stone. Classification and regression trees. 1984.

[18] P. Brown and A.R. Allen. Obesity linked to some forms of cancer. *WV Med J*, 98(6):271–272, 2002.

[19] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[20] Harry B. Burke, David B. Rosen, and Philip H. Goodman. Comparing the prediction accuracy of artifical neural networks and other statistical models for breast cancer survival. In *Neural Information Processing Systems*, pages 1063–1067, 1994.

[21] Kenneth P Burnham and David R Anderson. *Model selection and multi-model inference: a practical information-theoretic approach*. Springer, 2002.

[22] L.G. Castanheira. Aplicação de técnicas de mineração de dados em problemas de classificação de padrões. *Belo Horizonte: UFMG*, 2008.

[23] R.A. Castellino. Computer aided detection (CAD): an overview. *Cancer Imaging*, 5(1):17, 2005.

[24] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. CRISP-DM 1.0 step-by-step data mining guide. 2000.

[25] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[26] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[27] N.V. Chawla. Data mining for imbalanced datasets: An overview. *Data Mining and Knowledge Discovery Handbook*, pages 875–886, 2010.

[28] Dar-Ren Chen, Ruey-Feng Chang, and Yu-Len Huang. Breast cancer diagnosis using self-organizing map for sonography. *Ultrasound in Medicine & Biology*, 26(3):405–411, 2000.

[29] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 101–108. Morgan Kaufmann Publishers Inc., 1999.

[30] E.B. Claus, M. Stowe, and D. Carter. Family history of breast and ovarian cancer and the risk of breast carcinoma in situ. *Breast cancer research and treatment*, 78(1):7–15, 2003.

[31] D. Cook and D.F. Swayne. *Interactive and Dynamic Graphics for Data Analysis: with R and GGobi*. Springer, 2007.

[32] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[33] Portal da Oncologia Português. A mama, 2013. [Online; acedido 10-Janeiro-2013], disponível em http://www.pop.eu.com/.

[34] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[35] Dursun Delen, Glenn Walker, and Amit Kadam. Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial intelligence in medicine*, 34(2):113–127, 2005.

[36] H. Deng, G. Runger, and E. Tuv. Bias of importance measures for multi-valued attributes and solutions. *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 293–300, 2011.

[37] DD Dershaw, J. Yahalom, and JA Petrek. Breast carcinoma in women previously treated for hodgkin disease: mammographic evaluation. *Radiology*, 184(2):421–423, 1992.

[38] G.T. Diamandopoulos. Cancer: an historical perspective. *Anticancer research*, 16(4A):1595–1602, 1996.

[39] G.S. Dite, M.A. Jenkins, M.C. Southey, J.S. Hocking, G.G. Giles, M.R.E. McCredie, D.J. Venter, and J.L. Hopper. Familial risks, early-onset breast cancer, and BRAC1 and BRAC2 germline mutations. *Journal of the National Cancer Institute*, 95(6):448–457, 2003.

[40] M. Dramiński, A. Rada-Iglesias, S. Enroth, C. Wadelius, J. Koronacki, and J. Komorowski. Monte Carlo feature selection for supervised classification. *Bioinformatics*, 24(1):110–117, 2008.

[41] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[42] B.K. Edwards, H.L. Howe, L.A.G. Ries, M.J. Thun, H.M. Rosenberg, R. Yancik, P.A. Wingo, A. Jemal, and E.G. Feigal. Annual report to the nation on the status of cancer, 1973–1999, featuring implications of age and aging on us cancer burden. *Cancer*, 94(10):2766–2792, 2002.

[43] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36, 2004.

[44] Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[45] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[46] J.J. Fenton, S.H. Taplin, P.A. Carney, L. Abraham, E.A. Sickles, C. D'Orsi, E.A. Berns, G. Cutter, R.E. Hendrick, W.E. Barlow, et al. Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356(14):1399–1409, 2007.

[47] César Ferri, Peter Flach, and José Hernández-Orallo. Learning decision trees using the area under the ROC curve. In *ICML*, volume 2, pages 139–146, 2002.

[48] D. Ford and DF Easton. The genetics of breast and ovarian cancer. *British Journal of Cancer*, 72(4):805, 1995.

[49] Timothy W. Freer and Michael J. Ulissey. Screening mammography with computer-aided detection: Prospective study of 12,860 patients in a community breast center. *Radiology*, 220(3):781–786, 2001. doi: 10.1148/radiol.2203001282. URL http://pubs.rsna.org/doi/abs/10.1148/radiol.2203001282. PMID: 11526282.

[50] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[51] CM Friedenreich, HE Bryant, and KS Courneya. Case-control study of lifetime physical activity and breast cancer risk. *American Journal of Epidemiology*, 154(4):336–347, 2001.

[52] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, pages 1–67, 1991.

[53] B.B. Gallucci et al. Selected concepts of cancer as a disease: from the greeks to 1900. In *Oncology nursing forum*, volume 12, page 67, 1985.

[54] M.A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[55] Jiawei Han and Micheline Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann, 2006.

[56] David J Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123, 2009.

[57] D.J. Hand and C. Anagnostopoulos. When is the area under the receiver operating characteristic curve an appropriate measure of classifier performance? *Pattern Recognition Letters*, 34(5):492 – 495, 2013. ISSN 0167-8655.

[58] James A Hanley. Characteristic (ROC) curvel. *Radiology*, 743:29–36, 1982.

[59] Trevor Hastie, Robert Tibshirani, and J Jerome H Friedman. *The elements of statistical learning: data mining, inference and prediction*, volume 1. New York: Springer, 2009.

[60] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1263–1284, 2009.

[61] S.P. Helmrich, S. Shapiro, L. Rosenberg, D.W. Kaufman, D. Slone, C. Bain, O.S. Miettinen, P.D. Stolley, N.B. Rosenshein, R.C. Knapp, et al. Risk factors for breast cancer. *American journal of epidemiology*, 117(1):35–45, 1983.

[62] BE Henderson, RK Ross, and L. Bernstein. Estrogens as a cause of human cancer: the Richard and Hinda rosenthal foundation award lecture. *Cancer Research*, 48(2):246–253, 1988.

[63] Sylvia H Heywang-Köbrunner, Ingrid Schreer, Walter Heindel, and Alexander Katalinic. Imaging studies for the early detection of breast cancer. *Deutsches Arzteblatt International*, 105(31-32):541, 2008.

[64] K. Hirose, K. Tajima, N. Hamajima, T. Takezaki, M. Inoue, T. Kuroishi, S. Miura, and S. Tokudome. Association of family history and other risk factors with breast cancer risk among japanese premenopausal and postmenopausal women. *Cancer Causes and Control*, 12(4):349–358, 2001.

[65] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90, 1993.

[66] Torsten Hothorn and Berthold Lausen. Double-bagging: Combining classifiers by bootstrap aggregation. *Pattern Recognition*, 36(6):1303–1309, 2003.

[67] Cheng-Lung Huang, Hung-Chang Liao, and Mu-Chen Chen. Prediction model building and feature selection with support vector machines in breast cancer diagnosis. *Expert Systems with Applications*, 34(1):578–587, 2008.

[68] IARC. Globocan 2008, 2013. [Online; acedido 10-Janeiro-2013], disponível em http://globocan.iarc.fr/.

[69] Nathalie Japkowicz et al. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68, 2000.

[70] C.G. Kardinal and J.W. Yarbro. A conceptual history of cancer. *USA*, 6(4): 396–408, 1979.

[71] J.L. Kelsey and M.D. Gammon. The epidemiology of breast cancer. *CA: A Cancer Journal for Clinicians*, 41(3):146–165, 2008.

[72] J.L. Kelsey, M.D. Gammon, and E.M. John. Reproductive factors and breast cancer. *Epidemiologic reviews*, 15(1):36, 1993.

[73] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.

[74] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Proceedings of the ninth International Workshop on Machine Learning*, pages 249–256. Morgan Kaufmann Publishers Inc., 1992.

[75] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

[76] Teuvo Kohonen. *Self-organizing maps*, volume 30. Springer, 2001.

[77] L.A. Kurgan and P. Musilek. A survey of knowledge discovery and data mining process models. *Knowledge Engineering Review*, 21(1):1–24, 2006.

[78] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. 1992.

[79] Jan Larsen and Cyril Goutte. On optimal data split for generalization estimation and model selection. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 225–234. IEEE, 1999.

[80] Nada Lavrač. Selected techniques for data mining in medicine. *Artificial intelligence in medicine*, 16(1):3–23, 1999.

[81] SA Lee, RK Ross, and MC Pike. An overview of menopausal oestrogen–progestin hormone therapy and breast cancer risk. *British journal of cancer*, 92(11):2049–2058, 2005.

[82] Albert M Liebetrau. *Measures of association*, volume 32. Sage Publications, Incorporated, 1983.

[83] Hung-Yi Lo, Chun-Min Chang, Tsung-Hsien Chiang, Cho-Yi Hsiao, Anta Huang, Tsung-Ting Kuo, Wei-Chi Lai, Ming-Han Yang, Jung-Jung Yeh, Chun-Chao Yen, et al. Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. *ACM SIGKDD Explorations Newsletter*, 10(2):43–46, 2008.

[84] Wei-Yin Loh and Yu-Shan Shih. Split selection methods for classification trees. *Statistica sinica*, 7(4):815–840, 1997.

[85] Mariëtte Lokate, Michiel GJ Kallenberg, Nico Karssemeijer, Maurice AAJ Van den Bosch, Petra HM Peeters, and Carla H Van Gils. Volumetric breast density from full-field digital mammograms and its association with breast cancer risk factors: a comparison with a threshold method. *Cancer Epidemiology Biomarkers & Prevention*, 19(12):3096–3105, 2010.

[86] James D Malley, Karen G Malley, and Sinisa Pajevic. *Statistical learning for biomedical data*. Cambridge University Press, 2011.

[87] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for Naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, volume 752, pages 41–48. Citeseer, 1998.

[88] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[89] David Meyer, Friedrich Leisch, and Kurt Hornik. Benchmarking support vector machines. 2002.

[90] B.A. Miller, L.N. Kolonel, L. Bernstein, JL Young Jr, G.M. Swanson, D. West, C.R. Key, J.M. Liff, C.S. Glover, G.A. Alexander, et al. Racial/ethnic patterns of cancer in the United States, 1988-1992. pages A–8+, 1996.

[91] T.M. Mitchell et al. Machine learning, 1997.

[92] C.Z. Mooney. *Monte Carlo simulation*, volume 116. Sage Publications, Incorporated, 1997.

[93] Tatsunori Mori. Information gain ratio as term weight: the case of summarization of ir results. In *Proceedings of the 19th International Conference on*

Computational linguistics-Volume 1, pages 1–7. Association for Computational Linguistics, 2002.

[94] ACM Special Interest Group on Knowledge and Data Mining. KDD Cup 2008: Tasks, 2013. [Online; acedido 10-Janeiro-2013], disponível em http://www.sigkdd.org/kddcup/index.php.

[95] World Health Organization. Cancer mortality database, 2013. [Online; acedido 10-Janeiro-2013], disponível em http://www-dep.iarc.fr/WHOdb/WHOdb.htm.

[96] World Health Organization. WHO - World Health Organization, 2013. [Online; acedido 05-Fevereiro-2013], disponível em http://www.who.int.

[97] Edgar Osuna, Robert Freund, and Federico Girosit. Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 130–136. IEEE, 1997.

[98] Claudia Perlich, Prem Melville, Yan Liu, Grzegorz Świrszcz, Richard Lawrence, and Saharon Rosset. Breast cancer identification: KDD Cup winner's report. *ACM SIGKDD Explorations Newsletter*, 10(2):39–42, 2008.

[99] JD Picard. History of mammography]. *Bulletin de l'Académie National de Médecine*, 182(8):1613, 1998.

[100] Selwyn Piramuthu. Evaluating feature selection methods for learning in data mining applications. *European Journal of Operational Research*, 156(2): 483–494, 2004.

[101] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, 2007.

[102] Dorian Pyle. *Data preparation for data mining*, volume 1. Morgan Kaufmann, 1999.

[103] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[104] John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[105] Irina Rish. An empirical study of the Naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[106] ROCHE. Info cancro, tudo sobre o cancro: O cancro da mama, 2013. [Online; acedido 10-Janeiro-2013], disponível em http://www.roche.pt/.

[107] Joseph Lee Rodgers and W Alan Nicewander. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66, 1988.

[108] Paulo Canas Rodrigues and João A Branco. A análise de componentes principais sobre dados dependentes. In *XIV conference of the Portuguese Statistical Society*, page 27, 2006.

[109] P. Romanski and M.P. Romanski. Package fselector. 2009.

[110] C.M. Ronckers, C.A. Erdmann, C.E. Land, et al. Radiation and breast cancer: a review of current evidence. *Breast Cancer Res*, 7(1):21–32, 2005.

[111] Saharon Rosset, Claudia Perlich, Grzergorz Świrszcz, Prem Melville, and Yan Liu. Medical data mining: insights from winning two competitions. *Data Mining and Knowledge Discovery*, 20(3):439–468, 2010.

[112] Stuart Jonathan Russell and Peter Norvig. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.

[113] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[114] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels*. The MIT press, 2002.

[115] Bernhard Schölkopf, Christopher JC Burges, and Alexander J Smola. *Advances in kernel methods: support vector learning*. The MIT press, 1999.

[116] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588, 1999.

[117] Ingrid Schreer and Jutta Lüttges. Breast cancer: early detection. In *Radiologic-Pathologic Correlations from Head to Toe*, pages 767–784. Springer, 2005.

[118] P. Taylor and RM Given-Wilson. Evaluation of computer-aided detection (CAD) devices. *British Journal of Radiology*, 78(suppl 1):S26–S30, 2005.

[119] A. Thomas, A.K. Banerjee, and U. Busch. *Classic papers in modern diagnostic radiology*. Springer Verlag, 2005.

[120] Luís Torgo. *Data Mining With R Learning With Case Studies*. Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, Boca Raton, Florida, 2011.

[121] Cancer Research UK, 2013. [Online; acedido 27-Janeiro-2013], disponível em http://www.cancerresearchuk.org/home/.

[122] M. Verleysen and D. François. The curse of dimensionality in data mining and time series prediction. *Computational Intelligence and Bioinspired Systems*, pages 85–125, 2005.

[123] Juanjuan Wang, Mantao Xu, Hui Wang, and Jiwu Zhang. Classification of imbalanced data by using the smote algorithm and locally linear embedding. In *Signal Processing, 2006 8th International Conference on*, volume 3. IEEE, 2006.

[124] WHO. *World Health Organization Histological Typing of Breast Tumors*. World Health Organization, Geneva, 2nd edition, 1981.

[125] Graham Williams. *Use R: Data Mining with Rattle and R: the Art of Excavating Data for Knowledge Discovery*. Springer, 2011.

[126] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[127] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[128] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

[129] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[130] Kelly H Zou, A James O'Malley, and Laura Mauri. Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models. *Circulation*, 115(5):654–657, 2007.

[131] Mark H Zweig and Gregory Campbell. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clinical chemistry*, 39(4):561–577, 1993.

Anexos

```
##################################
# Estudo das correlacoes lineares
##################################
library(Matrix)

getNames<-function(x){
  colnames(corre)[x]
}

prepareMatrix<-function( formula ){
  mat <- which(formula, arr.ind=TRUE, useNames=FALSE)
  # Obter nome das colunas (atraves dos indices)
  mat<-apply(mat,2,getNames)
  # Ordenar os resultados atraves da primeira coluna (Sem relevancia)
  mat<-mat[order(mat[,1]),]
  return (mat)
}

# Obter Matriz Triangular Inferior sem Matriz identidade (s/ Diagonal)
corre<-cor(allData[,c(1,6:128)])
corre<-tril(corre,-1)

# Obter os mais altos valores de correlacao
inds99<-prepareMatrix(formula(quote(corre > 0.99)))
inds98<-prepareMatrix(formula(quote(corre > 0.98 & corre <= 0.99)))
inds97<-prepareMatrix(formula(quote(corre > 0.97 & corre <= 0.98)))
inds96<-prepareMatrix(formula(quote(corre > 0.96 & corre <= 0.97)))
inds<-prepareMatrix(formula(quote(corre > 0.96)))
```

Listagem A.1: Estudo das correlações lineares

```
####################
# Feature Selection
####################
featureSelection<-function(indices){
 vindices <-unique(as.vector(indices))
 featuresNOTRemove<-vindices
 for( i in 1:nrow(indices)){
  if( indices[i,1] %in% featuresNOTRemove & indices[i,2] %in% featuresNOTRemove){
   featuresNOTRemove <- featuresNOTRemove[!featuresNOTRemove==indices[i,2]]
  }
 }
 return(setdiff(vindices,featuresNOTRemove))
}

# Remover factores correlacionados cor>0.99
featuresToRemove99<-featureSelection(inds99)
write.csv(featuresToRemove99, "featuresToRemove99.csv",row.names=F)

# Remover factores cor>0.98 & cor<=0.99
featuresToRemove98<-featureSelection(inds98)
write.csv(featuresToRemove98, "featuresToRemove98.csv",row.names=F)

# Remover factores cor>0.97 & cor<=0.98
featuresToRemove97<-featureSelection(inds97)
write.csv(featuresToRemove97, "featuresToRemove97.csv",row.names=F)

# Remover factores cor>0.96 & cor<=0.97
featuresToRemove96<-featureSelection(inds96)
write.csv(featuresToRemove96, "featuresToRemove96.csv",row.names=F)

# Remover todos os factores cor>0.96
featuresToRemove<-featureSelection(inds)
write.csv(featuresToRemove, "featuresToRemove.csv",row.names=F)
```

Listagem A.2: Selecção de atributos

```
#########################################################################
# Calculo do peso de cada atributo na classificacao da massa tumorial
#########################################################################

library(FSelector)

# Subset maximo para determinar o peso das variaveis
# Escolha de 50 subsets aleatorios
randomRowsN<-{}
nIterations = 50
for( i in 1:nIterations){
  randomRowsN<-rbind(randomRowsN,c(sample(nrow(allData),50000)))
}
write.csv(randomRowsN, "randomRowsN.csv",row.names=F)

calculateWeightsSampling<-function(RandomRows, functionToUse, features){
  weightsIG<-{}
  for( i in 1:nrow(RandomRows) ){
    subsetAllData<-allData[RandomRows[i,],features]
    # calculate weights for each atribute using some function
    weights <- functionToUse(Malignant.Mass~., subsetAllData)
    weightsIG<-rbind(weightsIG,as.data.frame(t(weights)))
  }
  return (weightsIG)
}

calculateRankSampling<-function(weights, featuresNames,weightPerGroup){
  rank<-matrix(ncol=1,nrow=ncol(allData[,featuresNames]), dimnames= list(featuresNames,c(
      "Weight")))
  rownames(rank)<-featuresNames
  rank<-apply(rank,c(1,2),function(x){x=0})
  for(i in 1:nrow(weights)){
    auxt<-as.data.frame(as.data.frame(t(weights[i,])))
    aux<-cutoff.k(auxt,nrow(auxt))
    for( j in 1:length(aux)){
      rank[aux[j],1]<-rank[aux[j],1]+weightPerGroup[j]
    }
  }
  rank<-cutoff.k(rank,nrow(rank))
  return (rank)
}

# Vector usado no calculo do peso de cada atributo
weightPerGroup<-c(c(123:50),rep(1,49))

#################################################################
# Metodo 1 - Information Gain
# Calcular peso dos atributos
WeightsIG<-calculateWeightsSampling(randomRowsN, information.gain,c(1,6:128))
# Calcular rank dos atributos
RankIG<-calculateRankSampling(WeightsIG, colnames(allData[,c(6:128)]),weightPerGroup)
```

Listagem A.3: Determinação do peso de cada atributo na classificação da massa potencialmente cancerígena

```
1  ######################################################################
   # Configuracao
3  ######################################################################
   test.parameter.Remove.Redundancy<-TRUE
5  test.parameter.Feature.Ranking<-TRUE
   test.parameter.PCA<-FALSE
7  test.parameter.PCA.varmax.perc<-99 # [0..100]
   test.parameter.split.by<-'Amostra' # 'Amostra' ou 'Paciente'
9  test.parameter.split.ratio<-2/3 #[0..1] Examplo: 2/3 -> 2/3 Trainset, 1/3 Testset
   # c('Tudo','CC','MLO','Left Breast','Right Breast',
11 # 'Left Breast CC', 'Left Breast MLO','Right Breast CC', 'Right Breast MLO')
   test.parameter.subset<-'Tudo'
13 # c('Original','Undersampling','Oversampling','SMOTED',
   # 'Undersampling Paciente','Oversampling Paciente')
15 test.parameter.balance<-'Undersampling'
```

Listagem A.4: Configuração do processo de modelação

```
for( divi in 1:test.parameter.n.div){
 for( bali in 1:test.parameter.n.bal){

  source("library.R")
  source("Funcoes.R")
  source("Modelos.R")

  # Sementes para a divisao e balanceamento do conjunto de dados
  # Semente escolhida de acordo com a iteracao corrente
  ds.division.seed<-test.seed[divi]
  ds.balance.seed<-test.seed[bali]

  # Obter conjunto de dados original
  allData<-getAllData()

  # Obter conjunto de treino e teste
  result<-getTrainAndTestData(allData, ds.division.seed,
                       test.parameter.split.by, test.parameter.split.ratio)
  trainset<-result$trainset
  testset<-result$testset

  # Manter informacao essencial sobre cada caso e permitindo a agregacao por paciente (
      Study.ID)
  # em pos processamento
  patients.ID<-allData[,c("Image.ID","Study.ID","Malignant.Mass")]

  # Agregacao por paciente
  patients.ID.testset<-patients.ID[rownames(testset),]

  # Remover atributos sem relevancia para a deteccao de tumor (Remover identificadores)
  identifiers<-c("Image.Finding.ID","Study.Finding.ID","Image.ID","Study.ID")
  trainset<-trainset[,!(colnames(trainset) %in% identifiers)]
  testset<-testset[,!(colnames(testset) %in% identifiers)]

  # Seleccao de atributos
  result<-removeFeatures(trainset, testset, test.parameter.Remove.Redundancy,
                  test.parameter.Feature.Ranking,
                  test.parameter.PCA, percent.over=0.99, percent.under=1,
                  nRows=5000, nTries=50, method=information.gain,
                  varmax.percent=test.parameter.PCA.varmax.perc)
  trainset<-result$trainset
  testset<-result$testset

  # Balanceamento dos dados
  result<-dataset.sampling( trainset, testset, patients.ID, ds.balance.seed,
                    test.parameter.balance )
  trainset<-result$trainset
  testset<-result$testset
```

Listagem A.5: Pré-processo - Preparação dos conjuntos de dados

```
#######################################################################
# Aplicacao de modelos
#######################################################################
tryCatch({
  svm.tune <- tune(svm, Malignant.Mass ~ ., data = trainset,
                   ranges = list(gamma = 2^(-8:1), cost = 2^(0:4)),
                   tunecontrol = tune.control(sampling = "fix"))
}, interrupt = function(ex) {
  cat("An interrupt was detected.\n");
  print(ex);
}, error = function(ex) {
  cat("An error was detected.\n");
  print(ex);
}, finally = {
  cat("Releasing resources... SVM TUNE");
})
svm.method<-c('C-classification','nu-classification','one-classification','eps-
    regression','nu-regression')
svm.kernel<-c('linear','polynomial','radial')
svm.model<-list()
gc()
ind<-1
for(met in 1:length(svm.method)) {
  for(ker in 1:length(svm.kernel)) {
    cont<-TRUE
    svm.model<-svm.trainer(trainset,testset,filename, svm.tune, ind, svm.model, method
        =met, kernel=ker)
    ind<-length(svm.model) + 1
  }
}
rpart.trainer(trainset,testset,filename)
ctree.trainer(trainset,testset,filename)
randomForest.trainer(trainset,testset,filename)
cforest.trainer(trainset,testset,filename)
bagging.trainer(trainset,testset,filename)
dbagg.trainer(trainset,testset,filename)
adaboost.trainer(trainset,testset,filename)
naivebayes.trainer(trainset,testset,filename)
neuralnet.trainer(trainset,testset,filename)
glm.trainer(trainset,testset,filename)
oneR.trainer(trainset,testset,filename)
som.trainer(trainset,testset,filename)
xyf.trainer(trainset,testset,filename)
bdk.trainer(trainset,testset,filename)
knn.trainer(trainset,testset,filename)
lda.trainer(trainset,testset,filename)
qda.trainer(trainset,testset,filename)
multinom.trainer(trainset,testset,filename)
mars.trainer(trainset,testset,filename)
```

Listagem A.6: Aplicação de algoritmos

```
    ######################################################################
    # Pos processamento
    ######################################################################

    # Resultados por paciente
    patients.ID.testset$Malignant.Mass<-as.numeric(patients.ID.testset$Malignant.Mass)
    testset.patient<-aggregate(patients.ID.testset, by=list(patients.ID.testset$Study.ID)
        , FUN=max)
    testset.patient[,"Malignant.Mass"]<-factor(testset.patient[,"Malignant.Mass"], labels
        =c("Benign","Malign"))

    # Juntar resultados obtidos por amostra e por paciente
    comparacao<-{}
    for( i in 1:length(models)){
      models[[i]]<-get.model.aggregation.by.patient(models[[i]],patients.ID.testset)$model
      comparacao<-rbind(comparacao, evaluate(models[[i]]))
    }
  }
}
```

Listagem A.7: Pós processamento

```
# Dividir o conjunto em treino e teste - default 2/3 - 1/3
getTrainAndTestData<-function( dataset, tseed, split.by='Amostra', ratio=2/3 ){
  set.seed(tseed)
  # Divisao por casos
  if( split.by == 'Amostra'){
    # Todas as amostras sem cancro
    negativeInc<-rownames(dataset[dataset$Malignant.Mass == "Benign",])
    # Todas as amostras com cancro
    positiveInc<-rownames(dataset[dataset$Malignant.Mass == "Malign",])
    trainsetPos <- sample(positiveInc, length(positiveInc)*(ratio),
                    replace = FALSE)
    trainsetNeg <- sample(negativeInc, length(negativeInc)*(ratio),
                    replace = FALSE)
    trainset<-dataset[c(trainsetPos,trainsetNeg),]
    # Dados para teste 1/3 dos dados totais
    testsetrow<-setdiff(rownames(dataset),c(trainsetPos,trainsetNeg))
    testset<-dataset[testsetrow,]
  }
  # Divisao por pacientes
  if(split.by == 'Paciente'){
    # Identificao de todos os pacientes
    patients<-unique(dataset$Study.ID)
    # Identificao de todos os pacientes com cancro
    patients.with.cancer<-unique(subset(dataset, Malignant.Mass == "Malign")$Study.ID)
    # Identificao de todos os pacientes sem cancro
    patients.without.cancer<-setdiff(patients,patients.with.cancer)
    train.patients.with.cancer<-patients.with.cancer[sample(length(patients.with.cancer),
                  length(patients.with.cancer)*(ratio), replace = FALSE)]
    train.patients.without.cancer<-patients.without.cancer[sample(length(patients.without
        .cancer),
                  length(patients.without.cancer)*(ratio), replace = FALSE)]
    test.patients.with.cancer<-setdiff(patients.with.cancer,train.patients.with.cancer)
    test.patients.without.cancer<-setdiff(patients.without.cancer,train.patients.without.
        cancer)
    trainset<-dataset[dataset$Study.ID %in%
                  c(train.patients.without.cancer,train.patients.with.cancer),]
    testset<-dataset[dataset$Study.ID %in%
                 c(test.patients.without.cancer,test.patients.with.cancer),]
  }
  return(list(trainset=trainset, testset=testset))
}
```

Listagem A.8: Divisão de conjuntos (Função)

```
#########################################################
# Agregacao por paciente
#########################################################
get.model.aggregation.by.patient<-function(model,patients.ID.testset,aggregation.func=
    max){
  pred<-as.data.frame(cbind(model$pred,patients.ID.testset$Study.ID))
  if(class(pred$V1) == "factor"){
    levels<-c(1,2)
  } else {
    levels<-c(min(pred$V1),max(pred$V1))
  }

  patients.ID.testset1<-patients.ID.testset
  patients.ID.testset1$Malignant.Mass<-as.numeric(patients.ID.testset1$Malignant.Mass)
  testset.patient<-aggregate(patients.ID.testset1, by=list(patients.ID.testset1$Study.ID)
      , FUN=aggregation.func)
  testset.patient[,"Malignant.Mass"]<-factor(testset.patient[,"Malignant.Mass"], labels=c
      ("Benign","Malign"))
  rm(patients.ID.testset1)

  pred$V2<-as.numeric(pred$V2)
  pred$V1<-as.numeric(pred$V1)
  pred<-aggregate(pred, by=list(pred$V2), FUN=aggregation.func)
  pred$V1<-factor(pred$V1, levels=levels, labels=c("Benign","Malign"))
  pred<-pred$V1

  rocr.predictions<-as.data.frame(cbind(models[[i]]$rocr.predictions,patients.ID.testset$
      Study.ID))
  rocr.predictions$V2<-as.numeric(rocr.predictions$V2)
  rocr.predictions$V1<-as.numeric(rocr.predictions$V1)
  rocr.predictions<-aggregate(rocr.predictions, by=list(rocr.predictions$V2), FUN=
      aggregation.func)
  rocr.predictions<-rocr.predictions$V1
  rocr <- prediction(rocr.predictions, testset.patient$Malignant.Mass)
  perf.auc <- performance(rocr,"auc")
  auc <- as.numeric(perf.auc@y.values)
  cm <- confusionMatrix(pred, testset.patient$Malignant.Mass, positive = 'Malign')
  model$patient.pred<-pred
  model$patient.rocr.predictions<-rocr.predictions
  model$patient.auc<-auc
  model$patient.cm<-cm
  return(list(model=model))
}
```

Listagem A.9: Agregação por paciente (Função)

```
#############################################################
# Naive Bayes
#############################################################
gc()
naivebayes.trainer<-function(trainset,testset,filename, ...){
 tryCatch({
   startTime <- Sys.time()
   model <- naiveBayes(trainset, trainset$Malignant.Mass)
   stopTime <- Sys.time()
   pred <- predict(model, newdata=testset)
   prob <- predict(model, type="raw", newdata=testset)
   rocr.predictions<-prob[,'Malign']
   name<-paste('Naive Bayes', collapse = NULL)
   nb.model<-getModel(name, model, pred, prob, rocr.predictions,
                startTime, stopTime, testset, threshold = NULL)
   if(is.error(nb.model)){
     return()
   }
   environ<-environment()
   append.Rda.env(environ, nb.model, file=paste(filename,collapse=NULL))
   rm(nb.model)
   rm(startTime,model,stopTime,pred,prob,rocr.predictions,name)
 }, interrupt = function(ex) {
   cat("An interrupt was detected.\n");
   print(ex);
 }, error = function(ex) {
   cat("An error was detected.\n");
   print(ex);
 }, finally = {
   cat("Releasing resources... NB");
 })
}
```

Listagem A.10: Exemplo da função de treino NB (Função)

```
#############################################################
#SOM
#############################################################
gc()
som.trainer<-function(trainset,testset,filename, ...){
 tryCatch({
   startTime <- Sys.time()
   som.trainset <- data.frame(lapply(trainset[,-1], as.numeric), stringsAsFactors=FALSE)
   som.trainset <- scale(som.trainset)
   som.testset <- data.frame(lapply(testset[,-1], as.numeric), stringsAsFactors=FALSE)
   som.testset <- scale(som.testset)
   som.grid <- ifelse(sqrt(nrow(som.trainset))>25,25,sqrt(nrow(som.trainset))/2)
   model <- som(som.trainset, grid = somgrid(som.grid, som.grid, "hexagonal")) #25,25
   stopTime <- Sys.time()
   totalTime <- stopTime - startTime
   save(testset,
       file=paste('Temp',filename, collapse = NULL))
   rm(testset)
   pred <- predict(model, newdata = som.testset,
               trainX = som.trainset,
               trainY = trainset$Malignant.Mass, type='Class')

   load(paste('Temp',filename, collapse = NULL))
   prob<-pred$unit.prediction[pred$unit.classif,'Malign']
   pred<-pred$prediction
   rocr.predictions<-prob
   name<-paste('SOM Unsupervised', collapse = NULL)
   som.model<-getModel(name, model, pred, prob, rocr.predictions,
                  startTime, stopTime, testset, threshold = NULL)
   if(is.error(som.model)){
     return()
   }
   environ<-environment()
   append.Rda.env(environ, som.model, file=paste(filename,collapse=NULL))
 }, interrupt = function(ex) {
   cat("An interrupt was detected.\n");
   print(ex);
 }, error = function(ex) {
   cat("An error was detected.\n");
   print(ex);
 }, finally = {
   load(paste('Temp',filename, collapse = NULL))
   unlink(paste('Temp',filename, collapse = NULL))
   cat("Releasing resources... SOM");
 })
}
```

Listagem A.11: Exemplo da função de treino SOM (Função)

```
# PCA
usePCA<-function(trainset, testset, varmax.percent ){
 #Kernel PCA
 trainset <- droplevels(trainset)
 testset <- droplevels(testset)
 trainset <- data.frame(Malignant.Mass = trainset$Malignant.Mass,lapply(trainset[,-1],
     as.numeric), stringsAsFactors=FALSE)
 testset <- data.frame(Malignant.Mass = testset$Malignant.Mass,lapply(testset[,-1], as.
     numeric), stringsAsFactors=FALSE)
 pc <- prcomp(trainset[,-1], scale.=TRUE)
 sd <- pc$sdev
 # Variancia
 var <- sd^2
 # Variancia %
 var.percent <- var/sum(var) * 100
 # Variancia Acumulada
 cumvar<-cumsum(var.percent)
 cumvar<-which.min(abs(cumvar-varmax.percent))
 # Usar as componentes como conjunto de treino
 trainset <- data.frame(Malignant.Mass = trainset[,"Malignant.Mass"], pc$x)
 # Rotacao do conjunto de treino para estar de acordo com as componentes principais
 pc <- predict(pc, newdata = subset(testset,select=-Malignant.Mass))
 testset <- data.frame(Malignant.Mass = testset[,"Malignant.Mass"], pc )
 trainset <- trainset[,c(1:(cumvar+1))]
 testset <- testset[,c(1:(cumvar+1))]
 return(list(trainset=trainset,testset=testset))
}
```

Listagem A.12: Aplicação da PCA na redução do conjunto de dados (Função)