FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Previsão de efeitos adversos de medicamentos

Jéssica Namora

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Professor Rui Camacho

Co-orientador: Professor Vitor Santos Costa

27 de Janeiro de 2017

Previsão de efeitos adversos de medicamentos

Jéssica Namora

Mestrado Integrado em Engenharia Informática e Computação

27 de Janeiro de 2017

Resumo

Hoje em dia há uma grande quantidade de pessoas a consumir medicamentos diariamente, principalmente pessoas na 3ª idade. Todos esses medicamentos foram obrigatoriamente sujeitos a um conjunto de ensaios clínicos que permitem avaliar a sua eficácia e determinar efeitos adversos associados. No entanto os ensaios clínicos são realizados numa amostra extremamente pequena quando comparada com a população alvo.

Os efeitos adversos associados a um medicamento podem proporcionar o aparecimento de novas doenças ou, em casos extremos, podem levar à morte do paciente. Os profissionais de saúde tomam em conta os efeitos adversos publicados na bula de cada medicamento aquando da prescrição de um medicamento a um doente.

O trabalho realizado é baseado em *data mining* e pretende relacionar a informação sobre os princípios ativos dos fármacos e os efeitos adversos já conhecidos, com o objectivo de criar um modelo capaz de prever efeitos adversos de medicamentos. Esta abordagem ao problema tira partido de sistemas de recomendação para fazer a previsão de efeitos adversos de medicamentos.

A segunda abordagem ao problema, tira partido de algoritmos preditivos com recurso aos descritores moleculares do princípio ativo do fármaco com o objetivo de encontrar justificações para o aparecimento de um dado efeito adverso.

Assim, poderá ser possível descobrir novos efeitos adversos, sem recorrer a ensaios clínicos. Esta última prática para além de pôr a vida da população em risco, é mais dispendiosa e demorada do que a alternativa proposta.

Abstract

Nowadays there are lots of people consuming medicines daily, mainly elderly people. All of these medicines were subjected to a set of clinical trials to assess their efficacy, safety and to determine associated adverse reactions. However, clinical trials are performed on an extremely small sample when compared to the target population.

The adverse reactions associated with a medicine may lead to the emergence of new diseases or, in extreme cases, may lead to the death of the patient. Health professionals take into account the adverse reactions publicly available when they want to prescribe a medicine.

This led to the idea of a data mining project that intends to manage the information on the active principles of the drugs and the adverse reactions already known, in order to create a model capable of predicting adverse drug reactions.

The first approach to the problem makes use of recommended systems, and verifies the similarities between drugs and adverse effects of already known drug-adverse pairs in order to predict adverse effects not yet known by the model. This approach to the problem is made up of recommendation systems to forecast adverse drug reactions.

The second approach to the problem takes advantage of predictive algorithms using the molecular descriptors of the drug's active principle in order to find justifications for the appearance of a given adverse drug reaction.

Thus, it may be possible to discover new adverse effects without resorting to clinical trials. The latter practice, in addition to putting the lives of the population at risk, is more costly and time-consuming than the proposed alternative.

Agradecimentos

Quero deixar o meu agradecimento ao professor Rui Camacho que foi o meu orientador durante este projeto. Quero agradecer a pertilha de conhecimento, a disponibilidade prestada e todo o tempo que disponibilizou para que este trabalho fosse possível.

Quero também agradecer ao meu namorado e amigos por todo o apoio e pelas opiniões que partilharam comigo para melhorar este projeto.

Por último, quero agradecer ao projeto "NanoSTIMA: Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics/NORTE-01-0145-FEDER-000016"financiado pelo Programa Operacional Regional do Norte (NORTE 2020), sob o Acordo de Parceria PORTUGAL 2020, e através do Fundo Europeu de Desenvolvimento Regional (European Regional Development Fund - ERDF) pela disponibilização de dados utilizados para a realização deste projeto.

Jéssica Namora

Conteúdo

CONTEÚDO

Lista de Figuras

LISTA DE FIGURAS

Lista de Tabelas

Abreviaturas

AD	Árvore de Decisão
ADR/EAM	Adverse Drug Reaction - Efeitos adversos de medicamentos
ADReCS	Adverse Drug Reaction Classification System – Sistema de Classificação de reacções adversas a medicamentos
AERS	Adverse Event Reporting System – Sistema de Relatórios de Reacções Adversas
ATC	Anatomical Therapeutic Chemical classication system – Sistema de Classificação anatómica de produtos químicos terapêuticos
CAS	Chemical Abstracts Service registry number – Número de Registo do sistema de resumos químicos
ChEBI	Chemical Entities of Biological Interest - Entidades Químicas de Interesse Biológico
CTD	Comparative Toxicogenomics Database - Base de dados comparativos de toxico genomas
DM	Data Mining - Extração de informação de dados
FDA	Food and Drug Administration – Administração de Alimentos e Medicamentos (US)
ICD	International Classification of Diseases – Classificação Internacional de Doenças
KEGG	Kyoto Encyclopedia of Genes and Genomes – Enciclopédia de Kyoto dos Genes e Genomas
K-NN	K- Nearest Neighbors – k- Vizinhos mais próximos
MedDRA	Medical Dictionary for Regulatory Activities - Dicionário médico para actividades reguladoras
MEDLINE	Medical Literature Analysis and Retrieval System Online - Sistema de Análise e Recuperação de Literatura Médica Online
MeSH	Medical Subject Headings – Cabeçalhos de assuntos médicos
NCBI	National Center for Biotechnology Information – Centro Nacional de Informação Biotecnológica (US)
OMIM	Online Mendelian Inheritance in Man – Herança Mendeliana do Homem Online
RF	Random Forest - Florestas aleatórias
RNA	Redes Neuronais Artificiais
SIDER	Side Effect Resource – Recurso de Efeitos Secundários
SVM	Support Vector Machines – Maquinas de Vectores de Suporte
THIN	The Health Improvement Network - A Rede de Melhoria da Saúde
Weka	Waikato Environment for Knowledge Analysis – Ambiente de análise de Conhecimento Waikato
WHO/OMS	World Health Organization - Organização Mundial de Saúde

Capítulo 1

Introdução

Neste trabalho, pretende-se tirar partido de técnicas de *data mining* para prever Efeitos Adversos de Medicamentos (EAM) que ainda não tenham sido identificados. Para isso, serão utilizados algoritmos de recomendação e classificação aplicados às moleculas de cada medicamento. A informação sobre os EAM provém do repositório web ADReCS [1] mantida por investigadores da universidade de Xiamen.

As experiências realizadas incluem uma avaliação quantitativa dos métodos utilizados e têm como objetivo a previsão de efeitos adversos de medicamentos.

1.1 Motivação e Objetivos

O termo medicamento vem associado a uma conotação positiva. Os medicamentos são destinados a curar, prevenir ou controlar um dado estado de saúde indesejável. No entanto, cada medicamento tem associado, muitas vezes, um número variável de efeitos adversos. Esses efeitos adversos são normalmente previstos através de estudos clínicos. Infelizmente, esses estudos clínicos são feitos a um número reduzido de pessoas quando comparado com o público-alvo. Por este motivo, muito dos efeitos adversos de um medicamento não são relatados aquando dos seus ensaios clínicos, o que se poderá refletir no agravamento ou aparecimento de uma nova doença ou até mesmo levar à morte do paciente. Posto isto, existe uma necessidade continuada de procura de novos efeitos adversos, após a comercialização de cada medicamento.

Os ensaios clínicos feitos a um medicamento são um processo demorado, dispendioso e que em si também podem levar à morte de várias pessoas e animais. Essas são as razões por que são feitos em número reduzido, e a motivação para encontrar outros meios de previsão de efeitos adversos de medicamentos, sem riscos para a população.

O objetivo principal deste trabalho é então prever efeitos adversos de medicamentos não na fase experimental do mesmo, mas na fase da comercialização, para assim reduzir o número de relatos de consequências negativas após a toma de um medicamento.

[1] http://bioinf.xmu.edu.cn/ADReCS/

1.2 Estrutura da Dissertação

Este trabalho foi iniciado por uma análise do estado da arte em técnicas de *data mining*.

No Capítulo 2 são descritas algumas técnicas para o pré-processamento de dados, as principais tarefas de *data mining* e as bases de dados e sítios web relevantes para a realização deste projeto.

Ainda no Capítulo 2, são referenciadas as principais ferramentas utilizadas para o processo de *data mining* e são detalhados alguns métodos de avaliação utilizados para medir o grau de confiança do mesmo.

O Capítulo 3 refere toda a implementação do projeto desde o pré-processamento dos dados, aos processos e algoritmos utilizados.

No Capítulo 4, são detalhadas as experiências concretizadas para avaliar o trabalho descrito no Capítulo 3.

Por fim, no Capítulo 5, são descritas as principais conclusões que apareceram ao longo do projeto.

Capítulo 2

Efeitos Adversos de Medicamentos e *Data Mining*

A Organização Mundial de Saúde (OMS) definiu Efeito Adverso de um Medicamento (EAM) como sendo uma resposta a um medicamento que é nociva e não intencional, e que ocorre em doses geralmente utilizadas no homem [Wor02].

O principal método utilizado para identificação de efeitos adversos de medicamentos são os ensaios clínicos. Infelizmente, devido aos elevados custos que estes acarretam, são normalmente elaborados numa amostra reduzida de pessoas, e durante um curto espaço de tempo. Estas limitações fazem com que só seja possível identificar com certeza os efeitos adversos mais frequentes, e apenas aqueles que se evidenciam durante o tempo em que o ensaio clínico está a ser feito.

Por outro lado, existem sistemas que agrupam informação de relatos espontâneos de efeitos adversos de medicamentos [HGMMO02]. Estes sistemas contêm suspeitas de efeitos adversos reportados por profissionais de saúde, farmacêuticos ou consumidores de medicamentos. Nestes sistemas, a informação relatada é muito mais vasta, a amostra de consumidores muito mais diversificados e o tempo de estudo mais alargado. Contudo, estes sistemas podem conter informações erradas, duplicadas ou em falta.

O trabalho aqui descrito pretende tirar partido de técnicas de *Data Mining* (DM) para transformar a enorme quantidade de dados presentes nos sistemas de relato espontâneo de efeitos adversos de medicamento em alguns efeitos adversos de medicamento com elevado grau de confiança.

2.1 Extração de Conhecimento de Dados

O aparecimento da *World Wide Web* como um sistema de informação global inundou-nos com uma enorme quantidade de dados. Esses dados, são muitas vezes incoerentes ou contraditórios e muitas vezes pouco fiáveis. Para além disso, poderão ser inconclusivos e pouco úteis. Posto isto, surgiu a necessidade de criar ferramentas automatizadas que permitam o tratamento e agrupamento desses dados com o objetivo de os transformar em informação útil. Este processo, tem o nome

de *data mining* ou *knowledge discovery in databases* (KDD)[FPSS96] - exploração de dados e descoberta de conhecimento em bases de dados, respectivamente.

No intuito de extrair conhecimento de uma vasta gama de dados, é aconselhado que se sigam os seguintes sete passos, referidos em [Han05], que irão ser explicados na Secção 2.1:

Limpeza dos dados- extração de dados incorretos e/ou irrelevantes

Integração dos dados- onde são combinados dados de diferentes fontes

Seleção de dados- extração dos dados relevantes da base de dados

Transformação dos dados- realização de operações de síntese ou agregação de dados

Data Mining- processo em que diferentes métodos podem ser aplicados com o objetivo de extrair padrões de dados construíndo assim modelos para os dados

Avaliação dos modelos- identificação dos padrões que realmente representam conhecimento

Utilização do conhecimento- Onde o conhecimento extraído é apresentado ao utilizador para "uso corrente"

Limpeza de Dados

É comum, em bases de dados de grande dimensão, existirem dados **inconsistentes, incompletos ou errados**. Por este motivo, é necessário tratar esses dados, antes de iniciar qualquer outro passo do processo de *data mining*.

Designam-se por **dados incompletos** os dados que têm em falta um ou mais dos seus atributos. Este problema pode ser resolvido por um dos seguintes métodos:

Ignorar o atributo- Este método é normalmente usado quando um atributo tem vários valores em falta. No entanto, caso a base de dados contenha muitos dados incompletos, este método levará a uma perda considerável de dados, o que não é normalmente o desejado.

Preenchimento manual- Esta abordagem pode ser demorada, caso o número de dados incompletos seja elevado.

Uso de uma constante global- Esta técnica substitui todos os atributos em falta por uma variável global como por exemplo: *Unknown*. Embora esta técnica seja muito simples, pode levar a erros de *clustering*, uma vez que vários exemplos podem ser considerados semelhantes resultado dos atributos *Unknown* que irão passar a ter em comum.

Uso da média- Caso o atributo em falta se trate de um atributo numérico, pode substituir-se esse valor pela média de todos os valores encontrados nesse mesmo atributo.

Uso do valor mais provável- Este método é normalmente auxiliado por um algoritmo *bayesiano* ou por uma árvore de decisão. O objetivo é inferir qual o valor mais provável para os atributos em falta.

Os dados errados podem ser fruto de um erro humano ou de um erro computacional, muitas vezes derivados das aproximações feitas aos valores de vírgula flutuante. Existem vários métodos para reduzir o efeito destes erros, são eles:

Método da caixa- Este método promove a redução do erro através da observação dos exemplos vizinhos. Estes vizinhos são equiparados e a partir deles é calculada a média. Por fim, a média calculada é substituída pelos valores iniciais, conseguindo-se assim suavizar o erro localmente.

Clustering- Todos os valores são organizados em *clusters* segundo o grau de similaridade entre si. Os valores que não tenham similaridade suficiente com nenhum dos *clusters* existentes são considerados *outliers*.

Inspeção humana e computacional- Inicialmente é feita uma inspeção computacional que encontra os *outliers* comparando os exemplos entre si. Posteriormente, os *outliers* encontrados são reavaliados e corrigidos manualmente.

Regressão- Os dados são avaliados por uma regressão linear. Uma regressão linear tenta encontrar uma linha entre dois pontos de modo que a partir de um ponto se consiga prever o outro.

Por último, podem ainda encontrar-se erros de inconsistência. Alguns desses erros podem ser corrigidos manualmente utilizando referências externas. Um método mais automático de encontrar inconsistências é recorrendo ao uso de ontologias. Erros de inconsistência ocorrem principalmente quando os dados são provenientes de mais que uma fonte de dados, sendo necessário fazer a integração dos diferentes tipos de vocabulário utilizado.

Esta fase de limpeza dos dados é geralmente uma tarefa muito demorada, uma vez que muitos dos passos têm de ser feitos de uma forma não automatizada.

Integração dos dados

Para fazer um estudo de *data mining*, normalmente é incentivado o uso de uma grande quantidade de dados. Esses dados são muitas vezes provenientes de diferentes fontes, o que leva a que seja necessário fazer uma integração cuidada dos mesmos. Esta integração foca-se principalmente na tentativa de evitar/corrigir três tipos de problemas:

Problemas de identificação- O mesmo conceito, pode ser identificado de forma diferente, dependendo da fonte proveniente. Neste caso, é necessário adotar-se por uma das identificações e uniformizar toda a base de dados.

Redundância- Um atributo é considerado redundante quando pode ser derivado a partir de um ou mais atributos. Neste caso, os atributos redundantes devem ser eliminados da base de dados. Um exemplo é considerado redundante quando diferentes fontes se referem a um mesmo elemento o que após a integração se reproduz num elemento duplicado. Para uma boa integração, um dos elementos deverá ser eliminado.

Conflitos- Este tipo de erros provem principalmente da existência de diferentes unidades de medida como o m/cm. Para uma boa integração, estas medidas têm de ser uniformizadas para que posteriormente a sua comparação seja possível.

Seleção de Dados

A fase de seleção de dados tem como objetivo reduzir significativamente o volume de dados, enquanto se mantem toda a informação relevante encontrada nos dados iniciais. Podem ser utilizadas várias técnicas para reduzir o volume de dados em estudo, são elas:

Agregação- Alguns atributos são unidos num só. Só se poderá recorrer a esta técnica se a mesma não adulterar as conclusões que possam vir a ser retiradas.

Redução- Esta técnica pretende eliminar todos os atributos presentes na base de dados, que não têm interesse para o estudo que está a ser realizado. Caso os atributos irrelevantes não sejam eliminados, poderá levar a que sejam encontrados padrões de baixa qualidade.

Redução Numérica- os dados são substituídos por representações de dados menores tais como modelos paramétricos (que armazenam apenas os parâmetros do modelo, em vez dos dados reais), ou não paramétrico, tais como o agrupamento, a amostragem e a utilização de histogramas.

A fase de seleção de dados, também chamada de fase de redução de dados, permite um processamento muito mais rápido dos dados nas fases seguintes.

Transformação de dados

A fase de transformação de dados tem quatro objetivos fundamentais, são eles:

Normalização- Os atributos são dimensionados de modo a pertencerem a um dado intervalo de valores.

Smoothing- Tentativa de reduzir os erros que podem surgir na fase da integração. Aqui são utilizados métodos como o método da caixa, *clustering* e regressão.

Agregação- Atributos como as vendas diárias de um dado objeto podem ser agregadas com o objetivo de saber as vendas mensais. Ou seja, atributos de diferentes tuplos são agregados com o objetivo de gerar novos dados.

Generalização- dados "primitivos" como por exemplo a idade de um cliente podem ser substituídos por valores como jovem adulto ou idoso, de modo a tornar mais simples e intuitiva a sua classificação.

2.2 Tarefas de *Data Mining*

Nesta secção são apresentadas as cinco tarefas principais de *data mining*, são elas: Regressão, Classificação, *Clustering*, Regras de Associação e Sistemas de Recomendação.

Classificação

Classificação é o nome dado ao processo de identificar a qual das categorias existentes pertence uma nova observação. Este processo é feito tendo em conta um conjunto de dados de treino onde a relação observação-categoria já é conhecida.

A classificação é, por isso, um método de aprendizagem supervisionada, uma vez que recorre a um conjunto de exemplos corretamente classificados, para conseguir prever a classificação de uma nova observação.

Um algoritmo que implementa classificação é chamado de um classificador. São exemplos de classificadores as Redes Neuronais Artificiais (RNA) e as Árvores de Decisão (AD), que serão descritas mais à frente.

Clustering

Clustering é uma técnica de *data mining* que tem como objetivo agrupar um conjunto de dados em *clusters*, que se podem considerar como classes. Esse agrupamento é feito segundo o grau de similaridade que os dados têm entre si.

A técnica de *clustering*, ao contrário da técnica descrita anteriormente, não depende da existência de dados pré-classificados.

Sendo assim, esta é uma técnica de aprendizagem não supervisionada que aprende com a observação ao invés de aprender através de exemplos.

Os algoritmos de *clustering* são muitas vezes utilizados para o pré-processamento de dados. Posteriormente, algoritmos alternativos de *data mining* são aplicados aos *clusters* obtidos.

Descoberta de Regras de Associação

A descoberta de regras de associação é geralmente dividida em dois passos. Inicialmente os itens presentes nos exemplos da base de dados são combinados de todas as maneiras possíveis de forma a serem subdivididos em combinações raras e em combinações frequentemente encontradas. Posto isto, segue-se o segundo passo, onde os dados raros são descartados e se utilizam os dados frequentes para identificar as regras de associação.

O objetivo das regras de associação é encontrar padrões frequentes nos dados, ou seja, encontrar elementos (A) que impliquem necessariamente a presença de outros elementos (B) no mesmo tuplo. Gerando assim a regra de associação A -> B.

Regressão

Regressão é uma tarefa de DM de aprendizagem supervisionada, muitas vezes equiparada à classificação. Enquanto a classificação agrupa conjuntos finitos de valores discretos como nomes, datas ou locais, a regressão prevê valores contínuos.

A técnica de regressão é muitas vezes utilizada durante o processo de pré-processamento de dados, para preencher atributos numéricos que possam estar em falta.

Sistemas de Recomendação

Os sistemas de recomendação podem ser divididos em dois grupos:

-Item Recommendation

-Item Rating Prediction

O primeiro grupo utiliza uma matriz que contém informação dos utilizadores e dos itens do sistema. O segundo grupo contém ainda informações adicionais: classificações/ratings que descrevem a relação existente entre cada par utilizador-item [MBŠ12].

Cada um destes grupos pode ainda dividir-se em dois subgrupos: recomendação baseada em conteúdo e recomendação com filtragem colaborativa. Estas tarefas de *data mining* são muito utilizadas para comércio online, com o objetivo de recomendar ao utilizador produtos do seu interesse. Exemplos de grandes empresas que têm vindo a utilizar os sistemas de recomendação são a *Google, a Amazon e a Netflix*.

A primeira abordagem (recomendação baseada em conteúdo) utiliza uma série de atributos de um dado item, a fim de recomendar itens adicionais que contenham propriedades semelhantes entre si.

A segunda abordagem (recomendação com filtragem colaborativa) constrói um modelo que agrupa utilizadores com gostos semelhantes. Assim, os produtos sugeridos a um dado utilizador, serão os produtos preferidos por outros utilizadores com gostos semelhantes.

2.3 Algoritmos de Classificação

Indução de Regras e Árvores de Decisão

Árvore de Decisão (AD) é o nome dado a um conjunto de algoritmos de classificação que constroem modelos em forma de árvore.

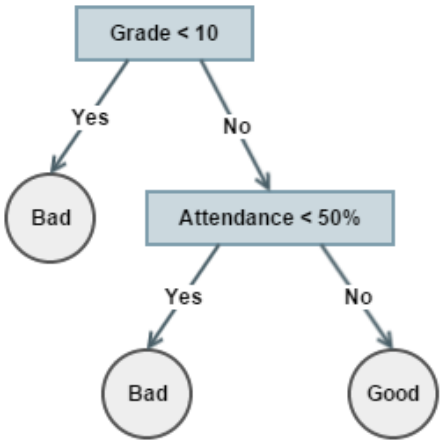

Figura 2.1: Árvore de Decisão utilizada para a classificação de alunos

Cada nó interior da AD representa um teste de um atributo e cada ramo representa um possível resultado para esse teste. O caminho percorrido desde a raiz até atingir um nó final chama-se uma regra de classificação. Cada nó final representa uma classe.

A elaboração de uma árvore de decisão inicia-se pela escolha do atributo raiz. De seguida, esse nó é expandido e cada ramo representa um valor possível desse mesmo atributo. Deste modo, a árvore de decisão pode ser subdividida num conjunto de problemas menos complexos. A partir deste momento, o processo é repetido recursivamente para cada ramo. Se, num dado momento, todas as instâncias de um nó obtiverem a mesma classificação, encontramos um nó final (uma folha).

Existem vários algoritmos desenhados para a construção de árvores de decisão. Entre eles: ID3, C4.5, CART e CHAID. Os algoritmos diferenciam-se principalmente pelo critério que decide qual o próximo nó a ser explorado e pelo tipo de teste realizado a cada nó interior da árvore.

Este tipo de algoritmo é considerado muito fácil de aprender e de interpretar e, por isso, é muitas vezes escolhido em detrimento de outros algoritmos que possam alcançar melhores resultados.

K-Nearest Neighbors

K-Nearest Neighbors (K-NN) dispõe os seus elementos de treino no espaço, cada um associado à classe a que pertence.

Para se classificar um elemento teste, verificam-se os votos dos seus k vizinhos mais próximos, e o novo elemento é classificado pela maioria das respostas recolhidas. Se k=1, o elemento de teste é classificado com a classe correspondente ao seu vizinho mais próximo.

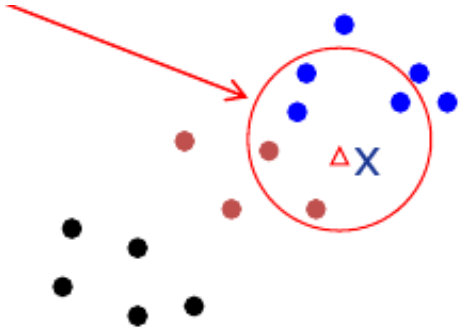

Figura 2.2: Exemplo de classificação usando o Algoritmo K-NN

Na Figura acima vemos um exemplo onde está a ser aplicado o algoritmo K-NN com k = 5. Uma vez que obtivemos 3 elementos correspondentes à classe azul e apenas 2 elementos correspondentes à cor vermelha, o novo elemento será classificado como pertencente à classe azul.

Random Forest

Random Forest é um método de *ensemble learning* utilizado para classificação e regressão. Métodos de *ensemble learning* combinam vários classificadores "básicos", e consideram como certa a resposta mais frequente ou a média das respostas fornecidas por esses métodos (no caso da regressão) ou consideram como certa a classe mais votada (no caso da classificação).

No caso do método *Random Forest* são usadas árvores de decisão (algoritmo CART), descritas anteriormente, construídas a partir da amostragem com reposição. O resultado de cada predição é o voto de cada método utilizado. O resultado que obtiver mais votos é o resultado selecionado.

Em suma, o algoritmo *Random Forest* tira partido de todos os classificadores construídos resultando geralmente num aumento do desempenho global.

Redes Neuronais Artificiais

As Redes Neuronais Artificiais (RNA) como um método de *data mining* foram inspiradas pelas redes neurais presentes no sistema nervoso central dos animais. É geralmente dividida em camadas com uma camada de entrada, uma camada de saída e "n"camadas intermédias. Entre cada camada há uma conexão que faz a ligação entre dois neurónios, ver Figura 2.3. Essa conexão tem associado um peso para a ligação.

Cada neurónio recebe vários valores reais como entrada, que provêm dos neurônios das camadas precedentes. Cada neurónio terá de combinar todos os seus valores de entrada num único valor de saída. Essa combinação é feita através de uma fórmula que combina os valores de entrada com o valor dos pesos (presentes nas conexões).

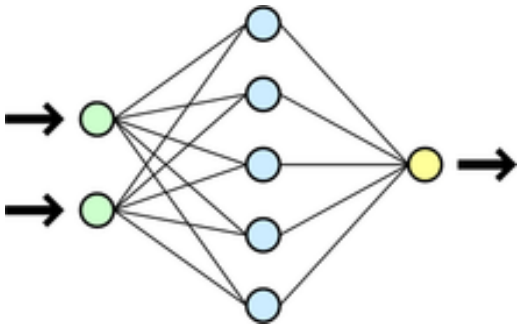

Figura 2.3: Exemplo de uma RNA com 3 camadas

Support Vector Machines

Support Vector Machines (SVM) é um algoritmo de aprendizagem supervisionada utilizado para classificação ou análise de regressão.

Dado um conjunto de dados de treino, cada exemplo é representado como um ponto no espaço, mapeados de modo a que os exemplos das duas classes disponíveis possam ser linearmente separados. A linha que os separa deverá estar o mais afastada possível dos exemplos das duas classes que se encontrem mais próximos da linha.

Assim que o modelo esteja construído, é possível inferir a qual das classes pertencerá um novo exemplo, dependendo de que lado da linha é que esse exemplo é mapeado.

Para além de realizar a classificação linear, SVMs podem também realizar uma classificação não-linear recorrendo a funções *kernel* que transformam o espaço dos exemplos de treino de forma que possam ser representados num espaço n-dimensional e de forma que as suas classes possam ser separadas por um hiperplano. Um exemplo desta transformação pode ser vista na figura abaixo:

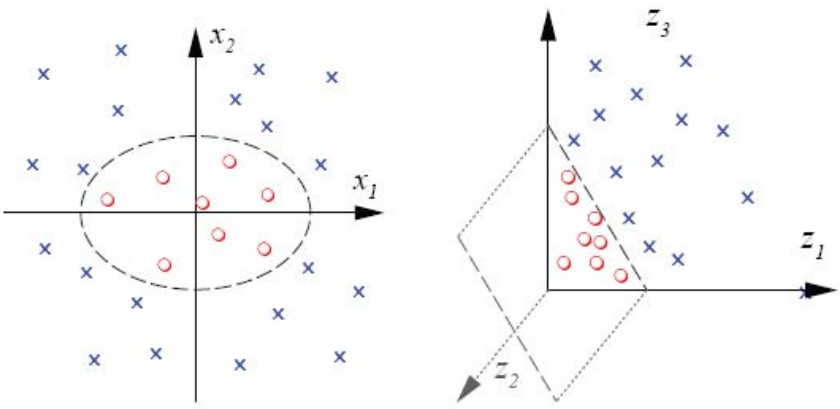

Figura 2.4: Separação dos dados num plano tridimensional

2.4 *Algoritmos de Clustering*

k-means/k-medoids

k-means é um método de *clustering* que tem como objetivo dividir a amostra de dados em k grupos diferentes. Cada *cluster* agrupa um conjunto de dados com semelhanças entre si.

De cada vez que um novo elemento se junta a um dos *clusters*, o centro desse mesmo *cluster* será recalculado, ver Figura 2.5. O centro de cada *cluster* é calculado através da média de todos os elementos presentes nesse *cluster*.

O algoritmo *k-medoids* diferencia do algoritmo *k-means* no facto de que o centro de um cluster tem que ser um dos seus objetos.

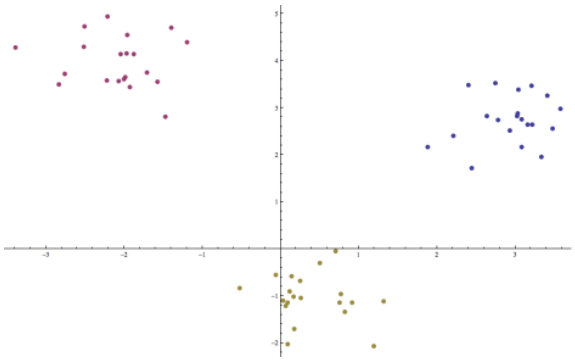

Figura 2.5: Exemplo de 3 clusters encontrados com o algoritmo k-means com k=3

2.5 Ferramentas

No domínio de DM foram desenvolvidas ferramentas que facilitam bastante a realização de tarefas de análise de dados. Nesta secção serão descritas as feramentas mais relevantes.

Orange

Orange [1] é uma ferramenta de *data mining* de software livre, escrita em *Python*.

O programa é mantido e desenvolvido pelo Laboratório de Bioinformática da Faculdade de Computação e Ciência da Informação na Universidade de Ljubljana.

Esta ferramenta disponibiliza uma interface intuitiva o que permite a utilização da mesma por parte daqueles que não têm qualquer tipo de experiência em *data mining*. Para os mais experientes, esta ferramenta pode ser utilizada como uma *Python library*.

Ao contrário das restantes ferramentas revistas nesta secção, *Orange* é a única que só suporta um tipo de base de dados - *MySQL*.

Weka

Waikato Environment for Knowledge Analysis (Weka) [FHW16] é uma ferramenta *open sorce*, escrita em *Java*, que fornece um vasto conjunto de algoritmos de *data mining* e ainda algumas ferramentas que possibilitam o pré-processamento de dados. Weka inclui algoritmos para regressão, classificação, *clustering*, regras de associação entre outros [HNF+].

Os dados podem ser carregados a partir de diferentes tipos de fontes como: ficheiros, URLs e bases de dados. Os formatos de ficheiros suportados incluem: ARFF, CSV, LIBSVM e C4.5.

A ferramenta Weka tem a possibilidade de acrescentar um vasto número de *plugins*, que possuem funcionalidades mais especificas a uma dada área, é o caso do *plugin* BioWEKA específico para biologia, bioinformática e bioquímica [GSZ07].

Porém, esta ferramenta não tem muito apoio quanto a visualização dos modelos finais obtidos. O que acontece noutras ferramentas que foram exploradas.

Por último, esta ferramenta inclui também algoritmos de avaliação, que permitem a comparação entre diferentes algoritmos ou entre diferentes data sets. Esta funcionalidade traz ainda a possibilidade de exportar ou revisitar os modelos mais tarde.

Rapid Miner

RapidMiner[2] é também uma ferramenta *open sorce* escrita em Java que suporta todas as etapas do processo de *data mining*. Esta ferramenta utiliza XML internamente para a uniformização dos dados.

A maioria das fontes de dados são suportadas incluindo Excel, Access, Oracle, IBM DB2, Microsoft SQL Server, ficheiros de texto, entre outras.

[1] http://orange.biolab.si/
[2] https://rapidminer.com/

Os utilizadores de RapidMiner dificilmente têm de escrever código, o que o torna numa ferramenta muito intuitiva. Para além disso, a visualização do resultado final também é feita automaticamente pela feramenta, em vários formatos como por exemplo: gráficos de barras, bolhas, densidade, 3D, entre outros.

A ferramenta *Rapid Miner* pode executar todos os algoritmos presentes na ferramenta Weka e ainda executar os seus algoritmos próprios.

Knime

KNIME[3] é uma ferramenta baseada na plataforma Eclipse IDE o que o torna tanto numa plataforma de desenvolvimento como numa plataforma de *data mining*. Knime está escrito em Java e, tal como o Eclipse, faz uso de *Plugins* para integrar funcionalidades adicionais. A versão da ferramenta KNIME, sem *plugins*, inclui algoritmos de integração, transformação e visualização de dados.

R

RStudio[4] é um IDE *(integrated development environment)* escrito em C++.

Entre as ferramenta estudadas nesta secção, o *RStudio* é a única que não disponibiliza uma forma de programação visual, o que poderá ser uma desvantagem.

[3] https://www.knime.org/
[4] https://www.rstudio.com/

2.6 Repositórios Web Relevantes

2.6.1 Medicamentos e Efeitos Adversos

ADReCS

Adverse Drug Reaction Classification System ADReCS)[5] é uma base de dados em XML, mantida por investigadores da universidade de Xiamen e oferece uma classificação hierárquica de Efeitos Adversos de Medicamentos.

Esta base de dados integra dados de vários repositórios médicos públicos como DailyMed, MedDRA 2.6.2, SIDER 2.6.1, DrugBank 2.6.2, PubChem, UM LS, entre outros.

Estes efeitos adversos provêm de fontes muito distintas como registos por parte de consumidores, resultados laboratoriais, registos médicos e farmacêuticos, entre outros. Posto isto, é frequente encontrar variadíssimos nomes para um mesmo medicamento, ou para um mesmo efeito adverso encontrado. Para colmatar esta situação, foi necessário padronizar todos os dados encontrados e, para isso, a ADReCS adotou as bases de dados MedDRA e UM LS como principais referências.

Nesta base de dados é utilizado um ID único com quatro campos separados por '.' (exemplo: xx.xx.xx.xxx) atribuído a cada efeito adverso. Esta combinação facilita a pesquisa de efeitos adversos uma vez que estes estão dispostos numa hierarquia com quatro níveis.

Database Statistics

ADRs	System Organ Classes	26
	High Level Group Terms	323
	High Level Terms	1,306
	Preferred Terms	5,094
	All unique ADR terms	6,749
	All ADR terms (include synonyms)	40,943
Drugs		1,698
Drug-ADR pairs		154,355

Figura 2.6: Estatísticas da Base de Dados ADReCS

Atualmente, a ADReCS disponibiliza informação de 1698 medicamentos e 6749 efeitos adversos, num total de 154 355 ocorrências medicamento - efeito adverso.

Para cada efeito adverso está registado o nome, o id, uma lista de sinónimos, uma breve descrição do efeito adverso sentido, o código proveniente da MedDRA 2.6.2 correspondente a esse ADR e a lista de medicamentos onde esse efeito adverso foi encontrado. A lista de medicamentos contém o nome e o id do medicamento em causa.

[5] http://bioinf.xmu.edu.cn/ADReCS/

Para cada medicamento é registado: ID, nome, uma breve descrição do medicamento, ATC *(Anatomical Therapeutic Chemical classication system)*, lista de sinónimos, indicações, CAS (*Chemical Abstracts Service registry number*) e uma lista de efeitos adversos que esse medicamento pode provocar. A lista de efeitos adversos contém o nome do ADR, o id e ainda a frequência com que esse ADR é encontrado aquando da toma do medicamento em causa.

O código ATC classifica os medicamentos em 5 níveis. O primeiro nível faz uma divisão entre 14 tipos de medicamentos. No segundo nível, os medicamentos são agrupados segundo a sua terapêutica. No terceiro nível é feita uma divisão farmacológica. No quarto nível, os medicamentos são divididos em diferentes grupos químicos e, por fim, no 5º nível, são divididos pelas suas substâncias químicas.

A	Alimentary tract and metabolism (1st level, anatomical main group)
A10	Drugs used in diabetes (2nd level, therapeutic subgroup)
A10B	Blood glucose lowering drugs, excl. insulins (3rd level, pharmacological subgroup)
A10BA	Biguanides (4th level, chemical subgroup)
A10BA02	metformin (5th level, chemical substance)

Figura 2.7: Classificação ATC do medicamento Metformin

Metformin é um medicamento com código ATC = A10BA02 e é utilizado para o tratamento da diabetes.

AERS

Adverse Event Reporting System (AERS) [6] é a maior base de dados de efeitos adversos de medicamentos do mundo [RSG01] e é gerida pela FDA.*Food and Drug Administration* (FDA) - Administração de alimentos e medicamentos - é o órgão governamental dos Estados Unidos da América responsável pelo controlo de alimentos e medicamentos.

Os registos presentes no AERS são provenientes de relatos por parte de profissionais de saúde, ou por parte de consumidores finais de medicamentos. Os termos inicialmente relatados para a FDA são muito distintos, e muitas vezes sinónimos. Para personalizar todos esses termos numa linguagem homogeneizada, os termos são traduzidos para os termos existentes na MedDRA 2.6.2.

Esta base de dados contém mais de nove milhões de notificações de eventos adversos e reflete os dados desde 1969 até ao presente.

Para cada relato de efeitos adversos são pedidas informações como a idade, o peso e o sexo do paciente. Quanto ao medicamento em jogo são guardadas informações como a sua substância ativa, o intervalo de tempo entre doses, a dose e a via por onde o medicamento foi administrado, entre outros.

[6] http://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ucm082193.htm

Nesta base de dados são também recolhidos dados sobre a seriedade dos efeitos adversos encontrados como por exemplo se levou à morte, se provocou internamento prolongado ou se provocou incapacidade ao doente.

Uma outra mais-valia desta vasta base de dados é a classificação da pessoa responsável por inserir os dados no sistema: físicos, farmacêuticos, médicos, advogados ou o consumidor final do medicamento.

Esta base de dados, para além de relacionar vários medicamentos com os efeitos adversos associados, também organiza informação sobre o recetor do medicamento, o consumidor final. Estas características são muitas vezes relacionadas com os efeitos adversos sentidos.

THIN

The Health Improvement Network (THIN)[7] – Rede de melhoria da saúde – é uma base de dados que coleciona os registos clínicos de 11.1 milhões de pacientes do reino unido. Estes registos são processados por profissionais de saúde e seguem um conjunto de regras para que se torne mais fácil a sua interpretação.

Esta base de dados armazena dados sobre os pacientes como por exemplo a idade, a localização, o sexo, a altura, o peso e a etnia.

Em relação aos medicamentos tomados por cada paciente, são guardados dados como a dosagem e a quantidade de medicamentos administrados.

Para além disso, esta base de dados ainda possui dados adicionais como por exemplo se o paciente é fumador ou não, se a paciente se encontra grávida ou não, e quais as vacinas administradas ao paciente.

Estas informações adicionais, não foram encontradas nas outras bases de dados estudadas. No entanto, todos estes fatores podem desencadear alterações nos efeitos adversos associados à toma de um medicamento.

Uma outra vantagem desta base de dados em relação às demais é possuir informação sobre a quantidade de vezes que o paciente se dirigiu a uma consulta médica. Contudo, esta base de dados não está disponível gratuitamente.

SIDER

SIDER[8] é uma base de dados que agrega informação no que diz respeito a efeitos adversos de medicamentos. Esta base de dados contém atualmente informação de 1430 medicamentos, 5868 efeitos adversos, e 139756 pares medicamento-efeito adverso.

As informações armazenadas em SIDER provêm essencialmente de documentos publicados que posteriormente são validados com recurso à MedDRA 2.6.2, STITCH 2.6.2 e PubChem.

[7] https://www.ucl.ac.uk/pcph/research-groups-themes/thin-pub/database
[8] http://sideeffects.embl.de/

2.6.2 Biologia e Interações moleculares

ICD

International Classification of Diseases (ICD) [9] – Classificação Internacional de Doenças é um sistema projetado para traduzir condições de saúde de palavras para um código alfanumérico até 6 dígitos. Este código facilita assim o armazenamento, interpretação e comparação dos dados.

Uma família de doenças (categoria) contém doenças que partilham características semelhantes entre si. Uma doença que ocorra com elevada frequência deve ter uma categoria própria, para assim facilitar o estudo da doença.

A estrutura geral da classificação de doenças é a seguinte:

- Doenças epidémicas
- Doenças constitucionais ou gerais
- Doenças locais organizadas por local
- Doenças do desenvolvimento
- Lesões

O ICD é revisto periodicamente pela Organização Mundial de Saúde (WHO) e está atualmente na sua décima revisão – ICD10.

OMIM

Online Mendelian Inheritance in Man (OMIM) [10] é uma base de dados que contêm informações sobre todos os distúrbios mendelianos conhecidos e mais de 15.000 genes. Distúrbios mendelianos são doenças hereditárias produzidas pela mutação ou alteração de um gene. OMIM foca então a relação entre fenótipo e genótipo.

Esta base de dados foi iniciada em 1960 e foi colocada online pela NCBI (*National Center for Biotechnology Information*) em 1985. Hoje em dia, esta base de dados é atualizada diariamente.

MedDRA

MedDRA[11] (*Medical Dictionary for Regulatory Activities*) - Dicionário médico para atividades reguladoras – é uma terminologia médica padronizada para facilitar a partilha de informações regulatórias sobre produtos médicos usados por seres humanos. Os produtos médicos abrangidos pelo âmbito da MedDRA incluem produtos farmacêuticos, biológicos e vacinas.

O principal objetivo deste dicionário é uniformizar e agrupar por grau de similaridade toda a linguagem médica, para assim facilitar a análise e recolha de dados.

[9] http://www.who.int/classifications/icd/en/
[10] http://www.omim.org/
[11] http://www.meddra.org/

DrugBank

A base de dados DrugBank[12] é um recurso único de biologia e química que combina dados detalhados de fármacos como a sequência de aminoácidos, estrutura e via de administração. DugBank contém atualmente 8246 entradas de medicamentos (6000 experimentais).

DrugBank permite pesquisas por medicamento, categoria, gene, reação, *pathway*, classe, proteína alvo ou indicação.

DrugBank coleciona e revê informação em mais de 50 bases de dados/aplicações web.

Uniprot

O UniProt[13] é uma base de dados com informações funcionais sobre proteínas retiradas da literatura médica. Para cada entrada UniProt, são armazenadas informações como: sequência de aminoácidos, nome, descrição da proteína, dados taxonómicos e informações de citação.

Para além disso, UniProt contém uma secção de informações analisadas computacionalmente (anotações possíveis, que aguardam uma verificação manual). Estas informações constam de ontologias biológicas, classificações e referências cruzadas.

STITCH

STITCH[14] é uma base de dados / servidor web de interações proteína-proteína conhecidas e previstas. Cada interação prevista é associada a uma pontuação do nível de confiança.

STITCH importa informações de bases de dados como: MINT, HPRD, BIND, DIP, BioGRID 2.6.2, KEGG 2.6.2, Reactome 2.6.2, IntAct, EcoCyc, GO 2.6.2. Para além disso, STITCH também utiliza ferramentas de *text mining* para importar interações proteína-proteína vindas de textos científicos.

Atualmente, STITCH contém interações de 300 mil pequenas moléculas e 2,6 milhões de proteínas de 1.133 organismos.

BioGRID

BioGRID[15] é um repositório de interações que pesquisa em 57 513 publicações presentes na PubMed2.6.3. Neste repositório constam 1.079.789 interações proteicas e genéticas e 27.745 associações.

BioGRID é atualizado mensalmente e encontra-se na versão 3.4.142.

Na base de dados, para cada interação detetada, é guardado o ID PubMed da publicação onde a interação está documentada.

[12] https://www.drugbank.ca/
[13] http://www.uniprot.org/
[14] http://stitch.embl.de/
[15] https://thebiogrid.org/

REACTOME

O Reactome[16] é uma base de dados que fornece ferramentas para a visualização, interpretação e análise de reações moleculares. Reação molecular é um evento biológico que desencadeia uma mudança no estado de uma molécula tais como uma ligação, uma ativação, ou a degradação da mesma.

ChEBI

ChEBI[17] (*Chemical Entities of Biological Interest*) - Entidades Químicas de Interesse Biológico - é uma base de dados de entidades moleculares. Para além disso, O ChEBI incorpora uma classificação ontológica, segundo a qual são especificadas as relações entre entidades moleculares. ChEBI baseia-se principalmente em quatro bases de dados, são elas: KEGG 2.6.2, IntEnz, PDBeChem e ChEMBL.

Ontologia dos Genes - GO

A Ontologia dos Genes[18] fornece vocabulário referente a propriedades dos genes abrangendo três domínios:

Componente Celular- referente à estrutura anatómica de uma célula ou a um grupo de genes;

Função Molecular- descreve atividades moleculares tal como ligação, transporte ou catálise;

Processos Biológicos- descrevem operações ou conjuntos de eventos moleculares especificamente pertinente para o funcionamento de unidades de vida integradas.

A ontologia GO é estruturada como um grafo acíclico dirigido e contém 745264 componentes celulares, 816681 processos biológicos e 778973 funções moleculares.

KEGG PATHWAY

KEGG[19] (*Kyoto Encyclopedia of Genes and Genomes*) é uma base de dados que representa conhecimento sobre interações moleculares e redes de reação para:

- Metabolismo
- Processamento de Informação Genética
- Processamento da Informação Ambiental
- Processos Celulares
- Sistemas Orgânicos
- Doenças
- Desenvolvimento de fármacos

[16] http://www.reactome.org/
[17] https://www.ebi.ac.uk/chebi/
[18] http://www.geneontology.org/
[19] http://www.genome.jp/kegg/

2.6.3 Ferramentas Química informática

Open Babel

Open Babel[20] é uma ferramenta que permite pesquisar, converter e analisar dados químicos.

Atualmente, o Open Babel suporta 111 formatos de arquivos químicos no total, sendo essa uma das principais vantagens quando comparado com outras ferramentas de análise de químicos.

Para além disso, fornece bibliotecas de programação que permitem o desenvolvimento de *software* de química em C++, Phyton, Perl, Ruby e Java.

O Open Babel permite o armazenamento de uma molécula em diferentes formatos, alguns deles descritos de seguida:

***Fingerprints* (impressões digitais)** - Open Babel identifica todas as subestruturas lineares e de anel de uma molécula e mapeia-as numa sequência de bits de comprimento 1024 usando uma função hash. Essa sequência de bits é chamada da impressão digital da molécula. As impressões digitais, para além de reduzirem o tempo de pesquisa de uma molécula, também facilitam a identificação de moléculas semelhantes através da comparação das suas impressões digitais.

SMILES – linguagem utilizada para identificar uma molécula. Esta linguagem segue um conjunto de regras, o que permite que uma molécula seja identificada por um e só um SMILES, o que não acontece, por exemplo, quando uma molécula é traduzida num ficheiro MOL. Este tipo de identificação, é benéfico para a eliminação de duplicados.

Coordenadas 2D e 3D – Open Babel tem a capacidade de produzir as coordenadas 2D e 3D de uma molécula através do seu identificador SMILES

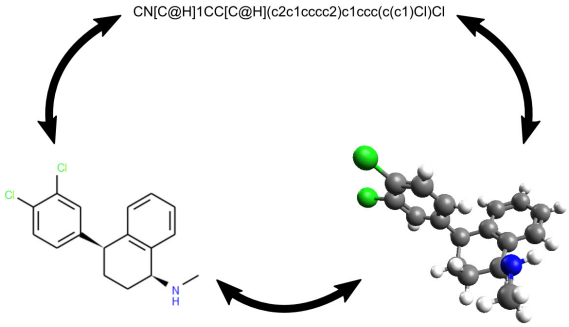

Figura 2.8: Identificadores da molécula setralina

A imagem acima ilustra três formas diferentes de identificar uma mesma molécula. Em cima, vemos o identificador SMILES – considerado uma estrutura 0D. Do lado esquerdo, encontra-se a molécula sertraline expressa em 2D e do lado direito a mesma molécula expressa em 3D. O Open Babel tem a capacidade de identificar as três estruturas como identificadores da molécula sertraline. Para além disso, todas elas podem ser convertidas entre si com o uso do Open Babel.

[20] http://openbabel.org

PaDEL-Descriptor

PaDEL-Descriptor[21] foi desenvolvido em Java e é um *software* gratuito e open source.

Este software é utilizado para o cálculo de descritores moleculares e impressões digitais. Através da fórmula química de uma dada substância, este *software* é capaz de a traduzir em valores matemáticos, que fornecem informação detalhada sobre a substância a ser explorada. Dados esses que não são perecíveis através da análise da fórmula química.

Atualmente PaDEL é capaz de calcular 1875 descritores (1444 descritores 1D e 2D e 431 escritores 3D) e ainda 12 tipos de impressões digitais.

IntSide

A base de dados IntSide[22] integra informações químicas e biológicas com o objetivo de conseguir explicar o porquê da existência dos efeitos adversos dos medicamentos.

Database	Version
UniprotKB	2015_05
SIDER	2015-08-06
MedDRA	17.1
CTD	2015_08

Database	Version
DrugBank	4.3
STITCH	v4.0
Gene Ontology	1.2_20150910
KEGG	75.1 2015-09-01
ChEBI	131

Figura 2.9: Base de dados utilizadas para a construção de IntSide

O conteúdo reunido na base de dados IntSide baseia-se nas bases de dados da Fig 2.9: Uniprot 2.6.2, SIDER 2.6.1, MedDRA 2.6.2, CTD 2.6.3, DrugBank 2.6.2, STITCH 2.6.2, Gene Ontology 2.6.2, KEGG 2.6.2 e ChEBI 2.6.2.

Atualmente reúne informação acerca de 996 drogas e 1175 efeitos adversos [JBDFA15]. Esses efeitos adversos estão subdivididos em 8 categorias pertencentes à biologia ou à química. Na imagem imediatamente abaixo podemos verificar quais as 8 categorias ou causas dos 1175 efeitos adversos catalogados.

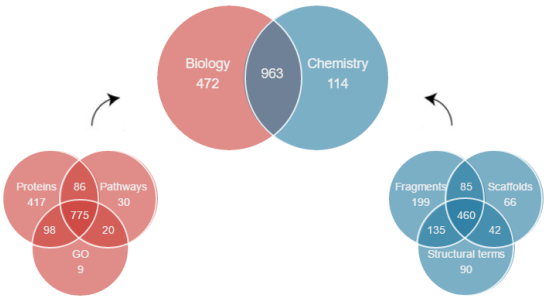

Figura 2.10: Estatísticas da base de dados IntSide

Como podemos verificar pela imagem acima, existem mais ADR provenientes da associação de diferentes características biológicas (775), do que ADR provenientes de interações químicas (460).

[21] http://www.scbdd.com
[22] http://intside.irbbarcelona.org/

No lado esquerdo, a vermelho, podemos verificar as características biológicas: Proteínas, *Pathways* e *Go*. Do lado direito, a azul, podemos ver as características químicas estudadas: *Scaffolds*, Fragmentos e Termos estruturais.

Proteínas – informação recolhida através da base de dados STITCH 2.6.2 – nome da proteína e id.

Pathways – informação que provem da base de dados KEGG. São registados o nome e o id de cada medicamento. O id proveniente da base de dados KEGG representa conhecimento sobre a interação molecular do medicamento.

Funções e Processos – informação retirada da Ontologia dos Genes. Diz respeito aos processos biológicos e a funções moleculares.

Scaffolds - são *strings* que correspondem a uma forma de representação de uma molécula – SMILES. Esta nomenclatura permite-nos visualizar os elementos químicos existentes na molécula, assim como as ligações químicas existentes entre eles.

Fragmentos - têm informação na linguagem SMARTS, que tal como o SMILES descrito acima, especifica a substrutura das moléculas. A sintaxe utilizada para representar um SMART id é a mesma utilizada para uma representação SMILES, mas com algumas regras e informações adicionais. Esta informação é retirada do Open Babel.

Termos estruturais - informação retirada da base de dados CHEBI 2.6.2, que subdivide as moléculas em categorias que contêm um dado grupo orgânico como por exemplo: *ethanols, organic salt, lipid* etc.

CTD

CTD[23] fornece informações sobre medicamentos, genes e doenças. As vertentes principais da base de dados CTD são as interações medicamento-gene, medicamento-doença e gene-doença.

- Interações Medicamento – Gene/Proteína - Estas interações são tanto diretas (ligações químicas à proteína) como indiretas (resultados químicos no aumento da fosforilação de uma proteína através de eventos intermediários);

- Associações Gene – Doença – contem associações comprovadas e associações inferidas;

- Associação medicamento – Doença – contem associações comprovadas e associações inferidas;

Para além disto, a base de dados CTD ainda integra informação da ontologia genética Go 2.6.2 e informação acerca das interações moleculares provenientes da base de dados KEGG 2.6.2.

[23] http://ctdbase.org/

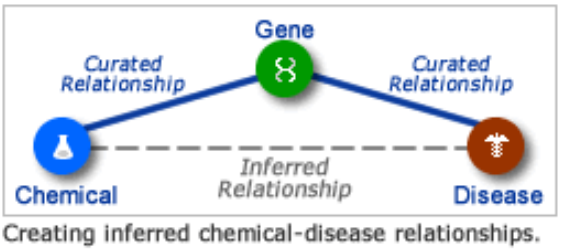

Figura 2.11: Inferencia de uma relação medicamento-doença

Na imagem acima[24] podemos comprovar como são inferidas as relações Medicamento (Chemical) - Doença (Disease). Se existe uma relação comprovada entre o medicamento A e o Gene C e existe uma relação comprovada entre o Gene C e a Doença B, então é provável que exista uma relação entre o medicamento A e a Doença B.

Atualmente, CTD inclui mais de 30,5 milhões de conexões tóxico genómicas [DGJ+16].

MEDLINE-PubMed-MESH

MEDLINE (*Medical Literature Analysis and Retrieval System Online*) - Sistema Online de Análise e Obtenção de Literatura Médica - é a base de dados bibliográficos da Biblioteca Nacional de Medicina dos Estados Unidos da América. Atualmente, o MEDLINE é atualizado mensalmente e contém mais de 18 milhões de referências a artigos de jornais científicos. Para facilitar a pesquisa e utilização de dados, todos os dados gravados no sistema são indexados com palavras-chave específicas de um sistema chamado MeSH.

MeSH[25] (*Medical Subject Headings*) é um dicionário de termos (descritores) relacionados com a medicina. Este dicionário organiza os conteúdos de uma forma hierárquica que permite pesquisar em vários níveis de especificidade.

MeSH contém 27.883 descritores. É possível a pesquisa por sinónimos ou nomes semelhantes, perfazendo assim 87.000 termos de entrada, divididos em 14 categorias principais.

O PubMed[26] é um serviço que permite o acesso gratuito às citações, resumos e artigos fornecidos pelo MEDLINE. Atualmente, PubMed contem mais de 26 milhões de citações de literatura biomédica publicada no MEDLINE.

[24] http://ctdbase.org/
[25] https://www.ncbi.nlm.nih.gov/mesh
[26] https://www.ncbi.nlm.nih.gov/pubmed/

2.7 Metodologias e Métricas de Avaliação

O objetivo de um sistema de classificação é ser capaz de prever acertadamente a que classe pertence um objeto. Caso o classificador não consiga prever acertadamente a classe do objeto, então deparamo-nos com um erro. A medida do desempenho de um classificador é calculada pela sua taxa de erros, ou seja, a proporção de erros encontrados face a um dado número de instâncias testado.

Para prever o desempenho de um classificador, é preciso calcular a sua taxa de erro num conjunto de dados que não os dados que criaram o classificador. Os dados usados para a formação do classificador chamam-se dados de treino. Os dados utilizados para avaliar o desempenho do classificador chamam-se dados de teste.

Geralmente, quanto maior for o conjunto de dados de treino, melhor o classificador. Embora os retornos comecem a diminuir depois de ultrapassado um certo volume de dados de treino [WFH11].

2.7.1 Métricas de Avaliação

De seguida são apresentadas algumas métricas de avaliação, sendo que:

VP: valores verdadeiros positivos

VN: valores verdadeiros negativos

FP: valores falsos positivos

FN: valores falsos negativos

Accuracy

Proporção entre os dados previstos e o seu verdadeiro valor.

$$Accuracy = \frac{VP+VN}{VP+VN+FP+FN}$$

Precision ou Positive Predictive Value

Proporção dos casos positivos previstos que estão corretamente identificados.

$$Precision = \frac{VP}{VP+FP}$$

Recall ou Sensitivity

Proporção de casos positivos reais que estão corretamente identificados.

$$Recall = \frac{VP}{VP+FN}$$

Negative Predictive Value

Proporção dos casos negativos previstos que estão corretamente identificados.

$$NPV = \frac{VN}{VN + FN}$$

Specificity

Proporção de casos negativos reais que estão corretamente identificados.

$$Specificity = \frac{VN}{VN + FP}$$

Medida F - F Measure

Média ponderada entre as métricas *recall* e *precision* mencionadas acima

$$FMeasure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

2.7.2 Metodologias de Avaliação

Hold-Out

Consiste em dividir o conjunto de dados em dois subconjuntos: de treino e de teste. As proporções de cada conjunto são variáveis. Normalmente, o conjunto de treino é bastante maior que o conjunto de teste.

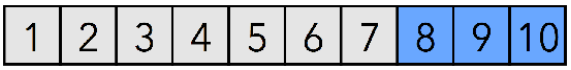

Figura 2.12: Exemplo do método Hold-Out utilizando 70% dos exemplos para treino e 30% para teste

Na figura 2.12 vê-se um exemplo do uso da técnica *Hold-Out* para a separação dos dados de treino (representados pela cor cinzenta) e de teste (representados pela cor azul). Neste caso, as proporções utilizadas foram de 0.7 para os dados de treino, e 0.3 para os dados de teste.

k-fold Cross-Validation

Os dados são divididos em K subconjuntos de igual tamanho. Os modelos são construídos K vezes. Cada modelo é construído utilizando um subconjunto a representar os dados de teste e os restantes K-1 subconjuntos a representar os dados de treino.

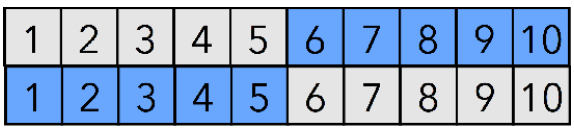

Figura 2.13: Exemplo 2-fold Cross-Validation

Na Figura 2.13 vemos um exemplo da técnica *2-fold Cross-Validation*. Os dados são divididos em 2. São portanto construídos 2 modelos. Para cada modelo metade dos exemplos são usados para treino e a outra metade dos exemplos são usados para teste.

Leave-one-out

Este método utiliza apenas um dos seus exemplos para o teste, e todos os outros exemplos como treino. Deste modo, são criados tantos modelos quantos exemplos existirem (dados). Este método é geralmente usado quando os dados disponíveis são limitados.

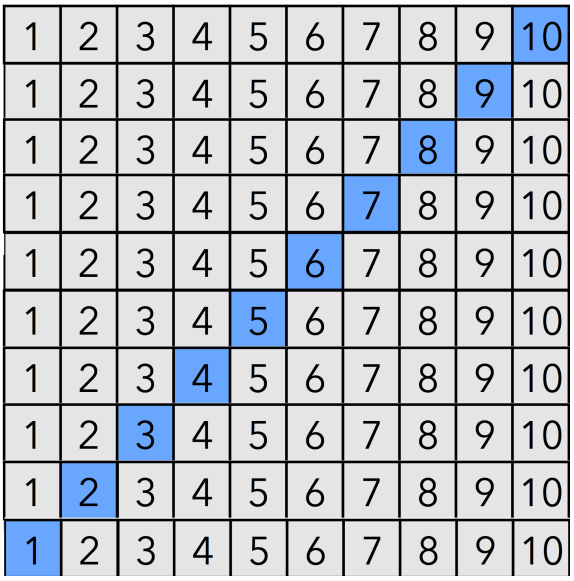

Figura 2.14: Exemplo Leave-one-out

Na Figura 2.14 vemos um exemplo do método *Leave-one-out*. Assim como nas subsecções anteriores, a cor cinzenta representa os dados de treino e a cor azul representa os dados de teste.

2.8 Resumo

Neste capítulo estão sumarizados os principais tópicos que tornam este trabalho possível. Na Secção 2.1 foi feito um levantamento das tarefas de pré-processamento de dados transversais a todas as tarefas de *data mining - DM*.

Na Secção 2.2 foram resumidas as principais tarefas de DM. Consecutivamente, na Secção 3.2 e 2.4 foram sumarizados alguns algoritmos de DM que pareciam interessantes para a resolução do problema.

Na Secção 2.5 foi feito um levantamento de algumas ferramentas de DM disponibilizadas gratuitamente, assim como algumas das suas vantagens e desvantagens.

Na Secção 2.6 foram descritos alguns sítios Web com informação relevante para este projeto. Esta secção subdivide-se em informação à cerca de medicamentos e efeitos adversos, informação detalhada à cerca de genes, proteínas e interações moleculares e ainda uma terceira secção que envolve sítios Web que albergam as duas primeiras categorias.

Por fim, na Secção 2.7, foi feito um levantamento de algumas métricas e metodologias de avaliação que permitirão avaliar e melhorar os modelos de *data mining* desenvolvidos.

Capítulo 3

Implementação

Neste capítulo são detalhados os algoritmos de pré-processamento e de *data mining* utilizados no decorrer deste projeto.

A base de dados selecionada para a avaliação experimental do projeto foi a ADReCS descrita na Secção 2.6.1.

A ferramenta escolhida para a elaboração deste projeto foi a ferramenta RapidMiner descrita na Secção 2.5.

Esta ferramenta está atualmente disponível em três versões. Uma versão gratuita que restringe o número de linhas utilizadas, uma versão comercial não gratuita e, ainda, uma versão educacional que permite o acesso gratuito e sem restrição do número de linhas a estudantes e professores. Para este processo foi utilizada a versão educacional.

Este capítulo está dividido em duas grandes secções: Sistemas de Recomendação (como já tinha sido descrito em [PCC]) e Algoritmos de Classificação.

Na primeira secção é utilizada unicamente informação da base de dados ADReCS e são utilizados 3 algoritmos de sistemas de recomendação para prever os pares medicamento-efeito adverso existentes.

Na segunda secção, para além dos dados utilizados para a primeira experiência, são ainda adicionados dados relativos aos descritores moleculares de cada fármaco. O objetivo desta experiência, para além de encontrar relações medicamento-efeito adverso, é também encontrar potenciais justificações para a existência de tal relação.

3.1 Sistemas de Recomendação

Experiência 1

Pré-processamento dos Dados

Uma vez que a base de dados ADReCS só está disponível em formato XML, foi feita uma conversão para o formato CSV - um dos formatos aceites pela ferramenta RapidMiner.

Este processo de transformação foi implementado recorrendo ao sistema de desenvolvimento Eclipse Java Neon, onde foram selecionados apenas os atributos relevantes (ADR e Drug). Seria interessante utilizar o atributo frequência disponibilizado em ADReCS mas, infelizmente, a maioria dos valores encontrava-se vazia. Na Figura 3.1 pode ver-se uma amostra do ficheiro CSV elaborado.

BADD_D00001	08.01.03.025
BADD_D00001	10.01.06.001
BADD_D00001	23.04.02.001
BADD_D00001	07.01.07.003
BADD_D00002	10.01.03.003
BADD_D00003	01.03.02.001
BADD_D00003	19.06.02.002

Figura 3.1: Amostra dos atributos selecionados de ADReCS

De seguida, o RapidMiner constroi um ficheiro de entrada compatível com a realização de um sistema de recomendação. (Faz a transformação do ficheiro de entrada para a forma de matriz com os medicamentos nas linhas e os efeitos adversos nas colunas).

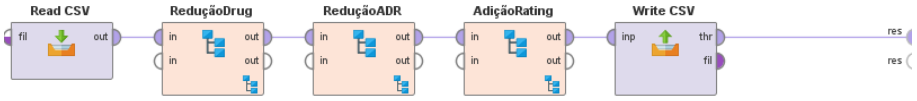

Figura 3.2: Processo de pré-processamento dos dados

O processo inicia-se com o operador 'Read CSV' que recebe como entrada um ficheiro do tipo CSV como o representado na Figura 3.1. De seguida, estão representados os subprocessos responsáveis pela redução dos medicamentos e efeitos adversos menos frequentes. Este passo reduz o tempo de execução das experiências que serão descritas neste capítulo.

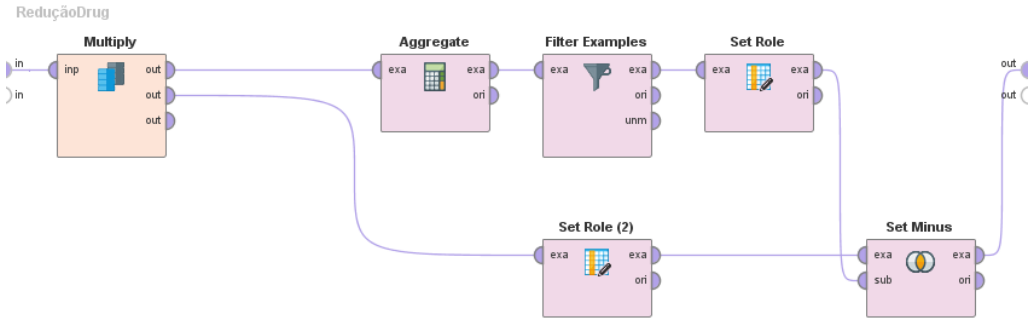

Figura 3.3: Processo referente ao subprocesso 'Redução Drug'

Inicialmente é aplicado o operador 'Multiply'. Este operador recebe um objeto na porta de entrada e multiplica-o para todas as portas de saídas conectadas a ele.

A um dos objetos de saída é aplicado o operador ‘Aggregate’. Este operador é capaz de contabilizar para cada medicamento existente, quantos ADR esse medicamento tem associados. Para isso, os parâmetros de entrada foram os seguintes:

-Aggregation atribute: adr

-Aggregation function: count

-Selected Attributes: drug

Uma amostra do ficheiro de saída do operador ‘Aggregate’ está representada na Figura 3.4.

Row No.	drug	count(adr)
1	BADD_D00001	110
2	BADD_D00002	1
3	BADD_D00003	99
4	BADD_D00004	30
5	BADD_D00005	307

Figura 3.4: Amostra do ficheiro de saída do operador ’Aggregate’

Após a agregação de efeitos adversos, optou-se por filtrar todos os medicamentos com menos de 100 efeitos adversos associados. Para isso recorreu-se ao operador ‘FilterExamples’, com o filtro count (adr) <= 100.

Os dois operadores ‘Set Role’ mostrados na Figura 3.3 são necessários para realizar a operação seguinte. Em ambos os operadores ‘Set Role’ o atributo drug fica com Role = id.

O operador ‘Set Minus’ é responsável por subtrair dois ficheiros de dados. Neste caso, o operador é utilizado para subtrair todos os medicamentos com menos do que 100 ADR ao ficheiro de dados iniciais (que contém todos os medicamentos).

Após a eliminação dos medicamentos e efeitos adversos com menor frequência, surge a necessidade de adicionar um subprocesso ‘AdiçãoRating’, o quarto operador presente na Figura 3.2.

Até aqui, trabalhou-se com medicamentos e efeitos adversos com relação entre si. A esses pares medicamento-ADR vai ser adicionado um atributo rating = ’true’. Ainda neste subprocesso, vão ser adicionados os pares medicamento-ADR não verificados (rating = ’false’) (Figura 3.5).

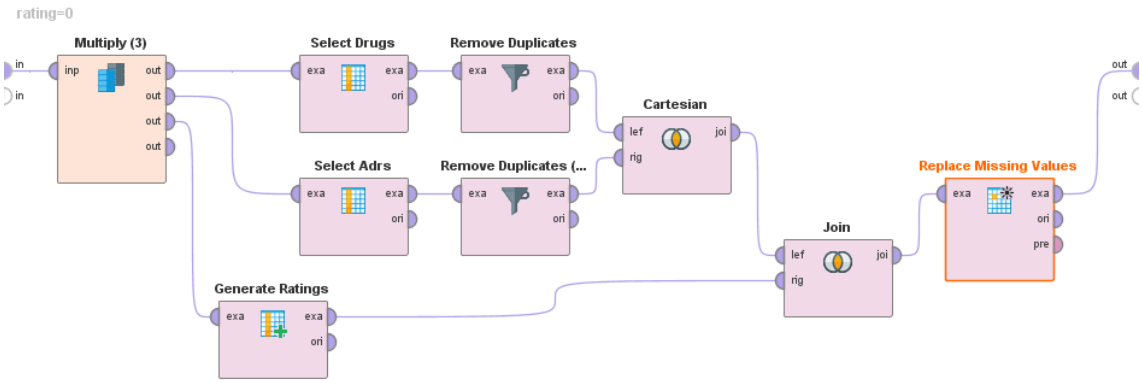

Figura 3.5: Subprocesso ’rating = 0’

O primeiro operador deste subprocesso é, mais uma vez, o operador ‘Multiply’.

A primeira e segunda portas de saída deste operador vão ser utilizadas para gerar os pares medicamento-ADR ainda não identificados (rating = 0). A terceira porta representa os pares medicamento-ADR já verificados, ou seja, os pares provenientes da base de dados ADReCS.

Os operadores ‘Select Attribute’ representados na Figura 3.5 pelos operadores ’Select Drugs’ e ’Select Adrs’ são utilizados para selecionar o atributo drug e o atributo adr. De seguida, é utilizado o atributo ‘Remove Duplicates’. Neste momento, existe um conjunto de dados com todos os medicamentos e outro com todos os efeitos adversos presentes nos dados iniciais. Posteriormente recorre-se ao operador ‘Cartesian’.

Á saída deste operador encontra-se um conjunto de dados correspondentes a todas as combinações possíveis entre medicamentos e efeitos adversos.

Voltando ao operador ‘Multiply’, existe ainda um operador de saída que mantém intactos os dados de entrada deste subprocesso. A esta porta é conectado o operador ‘Generate Attributes’, representado na Figura 3.5 pelo operador ’Generate Ratings’. Com este operador foi possível gerar um novo atributo ‘rating’. Todos os exemplos foram inicializados com rating = 1.

De seguida, é utilizado o operador ‘Join’ do tipo Left. Na porta Right é conectado o exemplo de dados agora com três atributos, correspondente aos dados da base de dados ADReCS. Na porta Left é conectado o exemplo de dados com apenas dois atributos correspondente a todas as relações possíveis entre os medicamentos e ADR existentes.

Do operador ‘Join’ sai um exemplo de dados com todas as relações possíveis entre medicamentos e efeitos adversos, sendo que as relações que realmente se verificam possuem rating = 1 e as restantes têm o atributo Rating=null.

O passo seguinte, permite substituir todos os exemplos com o atributo Rating=null pelo valor 0 através do atributo ‘Replace Missing Values’.

No final do pré-processamento de dados obtemos um ficheiro de três atributos (drug, adr e rating). Se rating = 1 significa que o ADR se verificou aquando da toma do medicamento se rating = 0 significa que este par medicamento-ADR não tem relação estabelecida.

No capítulo seguinte será relatado outro processo de pré-processamento de dados, onde serão separados os dados de teste e os dados de treino utilizados para as experiências também relatadas no capítulo seguinte.

Processo de Recomendação

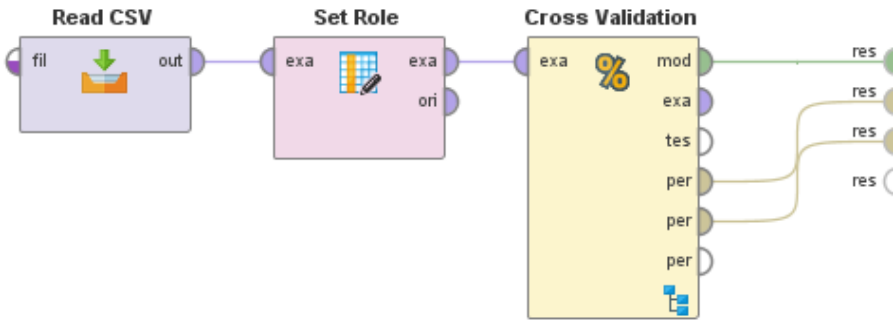

Figura 3.6: Processo de Recomendação

Na Figura 3.6 vemos o processo geral de recomendação utilizado nesta experiência. Este processo recorre a uma extensão de nome: *Recommender Extension*[1] [MBŠ12]. Esta extensão oferece operadores que aplicam os sistemas de recomendação descritos na Secção 2.2. Os algoritmos explorados foram os de previsão de *rating* com filtragem colaborativa. Tendo em conta a origem do problema, os medicamentos irão ser considerados utilizadores e os efeitos adversos produtos/conteúdo/itens.

Na Figura 3.6 vê-se que o primeiro operador a ser utilizado é o operador 'Read CSV'. Este operador é responsável por ler o ficheiro CSV elaborado na fase de pré-processamento de dados, descrita na secção anterior.

Se seguida, apresenta-se o operador 'Set Role'. Com este operador relacionou-se cada um dos atributos (drug, adr, rating) aos Role predefinidos pela extensão *Recommender Extension* são eles: 'user identification', 'item identification' e 'label'. Consoante se vê na Figura 3.7.

attribute name	target role
drug	user identification
adr	item identification
rating	label

Figura 3.7: Parâmetros utilizados no operador 'Set Role'

De seguida, temos o operador 'Cross Validation'. Este operador é responsável por fazer a divisão dos ficheiros de treino e de teste, como abordado na Secção 2.7.2. A Figura 3.8 mostra os operadores incluídos dentro do operador 'Cross Validation':

[1] http://www.e-lico.eu/recommender-extension.html

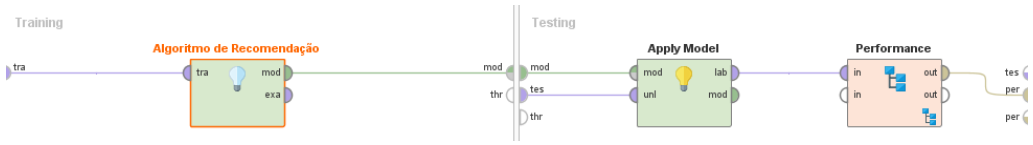

Figura 3.8: Operadores incluídos no operador 'Cross Validation'

O painel do processo foi automaticamente dividido. O lado esquerdo corresponde ao treino e o lado direito corresponde ao teste.

Do lado esquerdo, temos o operador 'Algoritmo de Recomendação'. É com este operador que se elabora o modelo de treino. A extensão utilizada - *Recommender extension* - disponibiliza 10 opções de algoritmos de treino para previsão de *rating*. Na secção seguinte serão explorados os 3 algoritmos utilizados no decorrer deste projeto.

Ainda na Figura 3.8, agora do lado direito, vemos o operador 'Apply Model'. Este operador recebe na primeira porta o modelo de treino, que contém todas as informações sobre os dados com que foi treinado. Na segunda porta, recebe o ficheiro de teste (elaborado pelo operador 'Cross Validation'), que contém os pares medicamento-adr cujo rating vai ser previsto. À saída deste operador obtemos o ficheiro CSV de teste com mais um atributo de nome 'prediction' (Figura 3.9).

drug	adr	rating	prediction
BADD_D00001	13.03.01.019	0	0.280
BADD_D00001	23.04.02.001	1	0.575
BADD_D00003	23.03.10.003	0	0.867
BADD_D00003	07.01.04.001	1	0.864
BADD_D00004	23.03.01.002	0	0.916
BADD_D00004	24.08.02.001	1	0.834
BADD_D00005	14.05.03.001	0	0.175
BADD_D00005	20.02.02.007	1	0.647
BADD_D00006	01.02.01.003	0	0.894
BADD_D00006	08.01.07.007	1	0.958

Figura 3.9: Amostra do ficheiro de saída do operador 'Apply Model'

Por último, temos o subprocesso 'Performance' que tem como objetivo final obter as métricas *Accuracy*, *Recall*, *Precision* e *F Measure*. Estas medidas são apresentadas em forma de percentagem. Quanto mais próximas de 100%, melhor é a performance do modelo.

O subprocesso 'Performance' pode ser visto na Figura 3.10.

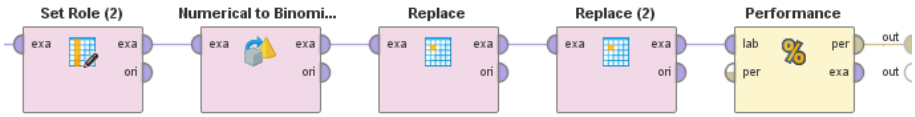

Figura 3.10: Subprocesso 'Performance'

O operador 'Set Role' presente na Figura 3.10 é responsável por transformar o Role do atributo prediction de regular para prediction. Esta transformação é necessária para que o operador 'Performance Binomial Classification' possa ser aplicado.

De seguida, temos o operador 'Numerical to Binomial' que é utilizado para transformar o atributo prediction para o tipo binomial (mais uma vez, uma condição necessária para utilizar o operador 'Performance Binomial Classification'). Após a utilização deste operador, o atributo prediction transforma todos os exemplos < 0.5 em false e todos os exemplos >= 0.5 em true.

De seguida, vemos dois operadores replace que transformam os exemplos do atributo prediction de true para 1 e de false para 0. Para que possam ser comparador ao atributo 'rating'. Uma amostra do ficheiro de saída pode ser visualizado na Figura 3.11.

drug	adr	rating	prediction
BADD_D00001	13.03.01.019	0	0
BADD_D00001	23.04.02.001	1	1
BADD_D00003	23.03.10.003	0	1
BADD_D00003	07.01.04.001	1	1
BADD_D00004	23.03.01.002	0	1
BADD_D00004	24.08.02.001	1	1
BADD_D00005	14.05.03.001	0	0
BADD_D00005	20.02.02.007	1	1
BADD_D00006	01.02.01.003	0	1
BADD_D00006	08.01.07.007	1	1

Figura 3.11: Amostra do ficheiro final

O resultado do operador 'Performance Binomial Classification' pode ser visualizado na Figura 3.12.

accuracy: 49.23%

	true 0	true 1	class precision
pred. 0	100	120	45.45%
pred. 1	1194	1174	49.58%
class recall	7.73%	90.73%	

Figura 3.12: Saída do operador 'Performance Binomial Classification' da métrica Accuracy

O mesmo operador foi também aplicado para obter as métricas *Recall*, *Precision* e *F Measure* relativas à classe 1.

Experiência 2

Esta segunda experiência relativa aos algoritmos de recomendação partilha do mesmo processo descrito na Secção 3.1. A diferença entre as duas experiências está no pré-processamento de dados.

Tirando partido da estrutura hierárquica dos efeitos adversos presentes na base de dados ADReCS, resolveu-se subir na hierarquia e tentar prever apenas a existência ou ausência dos grupos de efeitos adversos presentes nos nós imediatamente abaixo da raiz. Na Figura 3.13 podem-se ver os 26 nós explorados nesta experiência:

01 Blood and lymphatic system disorders (1091 drugs)
02 Cardiac disorders (1405 drugs)
03 Congenital, familial and genetic disorders (280 drugs)
04 Ear and labyrinth disorders (715 drugs)
05 Endocrine disorders (763 drugs)
06 Eye disorders (1030 drugs)
07 Gastrointestinal disorders (1538 drugs)
08 General disorders and administration site conditions (1586 drugs)
09 Hepatobiliary disorders (829 drugs)
10 Immune system disorders (1368 drugs)
11 Infections and infestations (1200 drugs)
12 Injury, poisoning and procedural complications (1151 drugs)
13 Investigations (1301 drugs)
14 Metabolism and nutrition disorders (1301 drugs)
15 Musculoskeletal and connective tissue disorders (1265 drugs)
16 Neoplasms benign, malignant and unspecified (incl cysts and polyps) (489 drugs)
17 Nervous system disorders (1555 drugs)
18 Pregnancy, puerperium and perinatal conditions (182 drugs)
19 Psychiatric disorders (1251 drugs)
20 Renal and urinary disorders (1101 drugs)
21 Reproductive system and breast disorders (813 drugs)
22 Respiratory, thoracic and mediastinal disorders (1309 drugs)
23 Skin and subcutaneous tissue disorders (1566 drugs)
24 Vascular disorders (1528 drugs)
25 Surgical and medical procedures (329 drugs)
26 Social circumstances (241 drugs)

Figura 3.13: Grupos de efeitos adversos no nível 1 da hierarquia.

Enquanto na experiência anterior a matriz de resultados era bastante dispersa, foi necessário eliminar os medicamentos e ARs com pouca frequência, nesta experiência, a matriz de resultados é muito densa, e foi por isso necessário eliminar os medicamentos que possuem efeitos adversos de muitos dos 26 grupos estudados.

O facto da matriz de resultados se ter tornado bastante densa deve-se ao facto dos medicamentos terem efeitos adversos muito distintos e, por isso, abrangerem muitos dos grupos de efeitos adversos.

Na Figura 3.14 ilustra-se uma amostra do ficheiro utilizado para esta experiência em forma de matriz.

drug	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
BADD_D00002	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BADD_D00003	1	1	0	0	1	1	1	1	0	1	1	1	0	1	1	0	1	0	1	1	1	1	1	1	0	0
BADD_D00004	0	1	0	0	1	0	1	1	0	0	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0	0
BADD_D00006	1	0	0	0	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
BADD_D00008	0	1	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	1	1	0	0	1	1	0	0
BADD_D00009	1	1	0	1	0	1	1	1	1	1	0	1	0	1	1	0	1	0	1	1	0	1	1	1	0	0
BADD_D00011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
BADD_D00013	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
BADD_D00014	0	1	0	1	0	1	1	1	0	1	1	0	1	1	0	0	1	0	1	0	0	1	1	1	1	1
BADD_D00015	1	0	0	1	0	0	1	1	0	1	0	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0

Figura 3.14: Matriz que relaciona medicamentos com grupos de efeitos adversos

Algoritmos de Recomendação

Matrix Factorization

Considerando um conjunto M de medicamentos, e um conjunto A de ADR's e uma matriz R de tamanho |M| * |A| que contém todas os rating's representativos das relações medicamento-ADR.

O algoritmo em questão tenta caracterizar os medicamentos e adr existentes através de vetores de fatores/características (k) inferidas através de padrões encontrados na matriz de rating's R. O objetivo do algoritmo é então encontrar as matrizes P (|M| * k) e Q (|A|*k) tal que o seu produto se aproxime da matriz R:

$$R \approx P * Q^T$$

Desta forma, para obter a previsão de um rating $(m_i - a_j)$ calcula-se o produto escalar dos dois vetores correspondentes ao medicamento m_i e o efeito adverso a_j:

$$r_{u,i} = m_i^T a_j = \sum_{k=1}^{k} m_{ik} a_{ik}$$

Para obter as matrizes P e Q , o sistema minimiza o erro quadrático regularizado para cada par medicamento-adr entre os valores previstos e o conjunto de ratings já conhecidos:

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^{k} p_{ik} q_{jk})^2 \frac{\beta}{2} \sum_{k=1}^{k} (||P||^2 + ||Q||^2)$$

O parâmetro β é utilizado para controlar as magnitudes dos vetores caraterísticas-medicamento e características-adr tal que P e Q sejam uma boa aproximação de R sem conterem números grandes. Por outras palavras, a segunda parte da fórmula acima foi acrescentada para evitar que o modelo sofra *overfitting* [KBV09].

Slope-One

Slope-one é um algoritmo utilizado para filtragem colaborativa introduzido em 2005 por Daniel Lemire e Anna Maclachlan. Este algoritmo calcula a diferença média entre os ratings de dois itens sendo que só são considerados os utilizadores que tenham classificado ambos os itens [LM05].

Dado um conjunto de treino X, e quaisquer dois itens j e i com ratings u_j e u_i, respetivamente, consideramos o desvio médio entre o item i e o item j como:

$$dev(j,i) = \sum_{u \in S_{j,i}(X)} \frac{u_j - u_i}{card(S_{j,i}(X))}$$

Tal que apenas utilizadores que contenham avaliações para os itens i e j estão incluídos no somatório. E tal que card representa a cardinalidade (número de elementos no conjunto).

Estando calculados todos os desvios médios, o cálculo da previsão é feito tal que:

$$P(u)_j = \frac{1}{card(R_j)} \sum_{i \in R_j} (dev_{j,i} + u_i)$$

onde $R_j = i | i \in S(u), i \neq j, card(S_{j,i}(X)) > 0$ é o conjunto de todos os itens relevantes.

User k-NN

A filtragem colaborativa baseada em utilizadores agrupa utilizadores segundo o seu grau de semelhança. Neste projeto, os medicamentos serão tratados como utilizadores e os efeitos adversos tratados como itens.

Sendo assim, para prever um dado rating do utilizador 'a' ao item 'b', o algoritmo user k-nn irá calcular o grau de semelhança entre o utilizador 'a' e todos os outros utilizadores do sistema.

O grau de semelhança entre os utilizadores 'a' e 'b' é medido recorrendo a uma das seguintes métricas:

Semalhança de Cosine

$$\frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Correlação de Pearson

$$\frac{\sum_{i=1}^{n} (A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n} (A_i - \bar{A})^2} \sqrt{\sum_{i=1}^{n} (B_i - \bar{B})^2}}$$

Sendo que:

-$A_i B_i$ correspondem aos ratings do item i do utilizador 'a' e 'b', respetivamente.

-n corresponde ao número total de itens classificados pelos utilizadores 'a' e 'b'

-$\bar{A}\bar{B}$ corresponde à média de todos os rating do utilizador 'a' e 'b', respetivamente.

Na tentativa de evitar o *overfitting* do modelo, o RapidMiner oferece a possibilidade de juntar as métodos acima um parâmetro de regularização. Este parâmetro é escolhido pelo programador.

Assim que o utilizador 'a' tenha sido comparado com todos os utilizadores do sistema, são guardados apenas os KNN (K-Nearest Neighbors – K vizinhos mais próximos, mais semelhantes). K corresponde a um dos parâmetros de entrada do algoritmo, assim como a métrica de semelhança (cosine ou pearson).

Por último, calcula-se o rating do utilizador 'a' ao item 'i' através da média ponderada dos rating's dos K vizinhos mais semelhantes ao utilizador 'a' multiplicado pela medida de semelhança entre o utilizador 'a' e o vizinho em jogo.

A extensão *Recommender extension* utilizada para a implementação deste projeto também fornece o algoritmo *Item k-NN*. Os dois algoritmos são muito semelhantes, com a exceção de que o Item k-NN mede a semelhança entre diferentes itens e o *User k-NN* mede a semelhança entre diferentes utilizadores.

Uma vez que o conjunto de dados utilizados tem mais medicamentos do que efeitos adversos, os dados dos efeitos adversos são, em média, mais esparsos. Posto isto, o algoritmo abordado neste projeto será o algoritmo *User K-NN*.

3.2 Algoritmos de Classificação

Experiência 3

Pré-processamento dos Dados

Como já foi referido anteriormente, o objetivo desta segunda experiência é tentar explicar (identificar uma justificação bioquímica para) a existência dos efeitos adversos nos medicamentos. Para este efeito, optou-se por abordar métodos de classificação descritos na Secção 3.2.

Para a utilização destes métodos, foi necessário enriquecer a base de dados com um conjunto de atributos que caracteriza cada medicamento quanto à sua constituição molecular, tipológica, propriedades entre outros. Este conjunto de atributos é chamado *descritores moleculares*.

Para obter o conjunto de descritores moleculares foi necessário transformar cada medicamento na sua fórmula SMILE (exemplo: Aspirin - CC(=O)Oc1ccccc1C(=O)O). Essa transformação foi obtida através de um programa realizado em Java que acede ao servidor web ChemSpider[2].

Depois de obtidas todas as fórmulas SMILE, de todos os medicamentos, foi então possível obter os descritores moleculares recorrendo ao servidor web PaDEL Descriptors.

Como já foi descrito na Secção 2.6.3, esta última plataforma agrega um total de 1875 descritores moleculares, sendo que 1444 são descritores 1D e 2D e 431 são descritores 3D.

Uma vez que os descritores 3D disponibilizados na plataforma só estão disponíveis para 333 medicamentos da base de dados ADReCS, optou-se por ignorar os descritores 3D para não reduzir excessivamente o número de medicamentos a serem utilizados.

Neste momento, é necessário criar o atributo label, ou seja, o atributo que os algoritmos vão tentar prever. Mais uma vez, esse atributo será retirado da base de dados principal ADReCS, recorrendo à hierarquia de efeitos adversos, tal como já vinha sido feito na Experiência 2 (Secção 3.1).

Cada algoritmo de classificação vai ser executado 26 vezes (número de grupos de efeitos adversos em estudo). Cada execução tenta prever se os medicamentos têm ou não efeitos adversos do grupo de efeitos adversos em estudo.

O atributo 'grupoADR' de role=label tomará então valores de true ou false. Se grupoADR = true, significa que o medicamento em jogo tem efeitos adversos do grupo de efeitos adversos estudado. Pelo contrário, se grupoADR = false, significa que o medicamento em estudo não provoca nenhum dos efeitos adversos do grupo de efeitos adversos em estudo.

Na Figura 3.15 pode ver-se uma amostra do ficheiro corresponde ao grupo de efeitos adversos 2 que diz respeito a efeitos adversos com consequências cardíacas.

[2] http://www.chemspider.com/

Name	grupoADR	nAcid	ALogP	ALogp2	AMR	apol	naAromAt	nAromBon	nAtom	nHeavyAtc	nH
BADD_D00005	true	1.0	-1.102199	1.2148448	36.9694	23.342722	0.0	0.0	22.0	11.0	11.0
BADD_D00009	true	0.0	-0.231400	0.0535459	44.754900	22.785136	6.0	6.0	20.0	11.0	9.0
BADD_D00013	true	0.0	-1.194999	1.4280249	38.2306	25.692687	0.0	0.0	26.0	10.0	16.0
BADD_D00014	true	1.0	-0.499199	0.2492006	36.811800	21.207136	0.0	0.0	19.0	10.0	9.0
BADD_D00015	false	1.0	0.4399999	0.1935999	47.9988	24.382343	6.0	6.0	21.0	13.0	8.0
BADD_D00026	true	0.0	8.9349999	79.834224	189.2906	107.74040	0.0	0.0	96.0	40.0	56.0
BADD_D00029	false	0.0	-0.734799	0.5399310	106.25440	64.246996	0.0	0.0	57.0	28.0	29.0
BADD_D00038	true	0.0	-2.0427	4.1726232	33.4662	16.669171	9.0	10.0	14.0	10.0	4.0

Figura 3.15: Amostra do ficheiro correspondente ao grupoADR=2

Processo de Classificação

O processo de classificação utilizado no decorrer desta experiência é muito semelhante ao processo apresentado nas secções anteriores é ilustrado na Figura 3.16

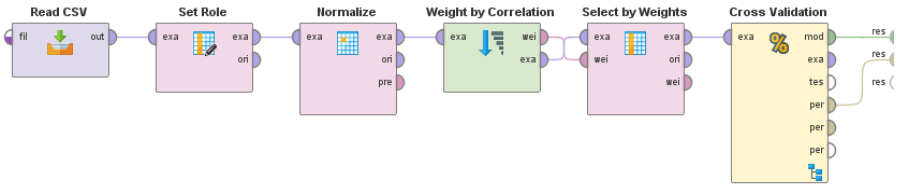

Figura 3.16: Processo de Classificação

O primeiro operador utilizado é o operador 'Read CSV'. Este operador é responsável por tratar o ficheiro descrito na Secção 3.2. Este ficheiro de entrada contém um atributo drug, um atributo grupoADR, e um conjunto de atributos representativos dos descritores moleculares de cada medicamento.

De seguida, é utilizado o operador 'Set Role'. Este operador é encarregue de alterar o role do atributo 'grupoADR' de 'regular' para 'label'. Esta mudança é necessária para utilizar qualquer algoritmo de classificação. O algoritmo escolhido, irá fazer a classificação/previsão segundo esse atributo.

De seguida, foi aplicado o operador 'normalize' a todos os atributos da base de dados. Este passo faz com que todos os atributos sejam transformados de forma a que todos eles variem entre os mesmos valores. Desta forma, impede-se que atributos com valores grandes tenham um maior peso na previsão de cada exemplo de dados.

De seguida, é feita uma pré-selecção de atributos (*feature selection*) onde são identificados os atributos mais relevantes para o estudo e são descartados os atributos menos relevantes. Esta pré-seleção é feita através dos operadores 'Weight by Correlation' e 'Select by Weight'. O primeiro operador gera um peso (weight) para cada atributo da base de dados. O peso corresponde à correlação entre cada atributo de entrada e a label. Quando maior for o peso de um atributo, mais relevante esse atributo é considerado para a resolução do problema. Posteriormente, o operador 'Select by Weight' selecciona apenas os 'k' atributos de maior peso. O parâmetro 'k' é obtido através de optimização recorrendo ao operador 'Optimize Grid'.

Por último, temos o operador 'Cross Validation'. Tal como aconteceu nas experiências anteriores, o operador 'Cross Validation' subdivide-se automaticamente nos processo de treino e teste.

O processo de treino contém o operador de classificação escolhido. O processo de teste contém o operador 'Apply Model' e o operador performance.

Na secção seguinte são descritos os algoritmos de classificação escolhidos para a realização desta experiência.

Algoritmos de Classificação

Árvores de Decisão

Árvore de Decisão é um operador que organiza os seus exemplos no formato de uma árvore invertida, ou seja, uma árvore que tem a sua raiz na parte superior e que vai sucessivamente sendo dividida em vários ramos (de cima para baixo). Os nós mais inferiores são chamados de nós folha. Os nós folha representam as classes que o algoritmo está a tentar prever. No *RapidMiner*, o algoritmo irá tentar prever o atributo que estiver definido com role = label. O algoritmo de árvores de decisão utilizado neste projeto foi o CART.

Em cada recursão de qualquer algoritmo árvore de decisão são usados os seguintes passos, descritos em [Man]:

- Um atributo A é selecionado para se dividir.

- O atributo A é subdividido em subconjuntos disjuntos.

- Uma árvore é retornada com o atributo A como raiz e com um conjunto de ramos no nível inferior. Cada ramo terá um descendente subárvore ou nó folha.

Para o algoritmo CART, o critério de seleção utilizado para escolher o nó A a ser explorado é a métrica *Gini Index* ou *Gini impurity*. O nó escolhido é aquele que obtiver uma medida de impureza menor. A impureza é calculada através da seguinte fórmula:

$$GiniIndex(f) = \sum_{i=1}^{J} fi(1 - fi)$$

Sendo que:

- J corresponde ao número de classes

- fi corresponde à fração de itens rotulados com classe i

Em geral, a recursão pára quando todos os exemplos de dados, ou a maioria dos exemplos de dados têm o mesmo valor de label. Contudo, existem outras condições que fazem com que a árvore deixe de se ramificar, são elas:

- Há menos de um certo número de instâncias ou exemplos na subárvore atual. Esta condição é ajustada através do parâmetro *minimal size for split* (tamanho mínimo para haver uma divisão).

- Nenhum atributo que possa ser dividido atinge um dado ganho mínimo em relação à árvore existente antes dessa divisão. Isto pode ser ajustado usando o parâmetro *minimal gain* (ganho mínimo).

- A profundidade máxima é atingida. Esta condição de paragem é alterada segundo o parâmetro *maximal depth* (profundidade máxima).

Random Forest

O operador *random forest* gera um conjunto de árvores de decisão (tipo CART) usando amostragem com reposição no conjunto de dados original. O número de árvores de decisão geradas é especificado pelo programador. O modelo resultante é um modelo de votação de entre todas as árvores de decisão geradas. Um atributo x é classificado como pertencente à classe 1 se o atributo x foi previsto como pertencente à classe 1 pela maioria das árvores geradas.

Cada árvore de decisão é gerada exactamente da mesma forma que as árvores de decisão especificadas na secção anterior (através do cálculo da impureza), com a exceção de que para cada divisão, o nó selecionado é escolhido de entre um conjunto aleatório de nós, ao invés de ser seleccionado de entre todos os nós disponíveis para serem ramificados.

Naive Bayes

Naive Bayes é um algoritmo baseado no Teorema de *Bayes*. Este algoritmo assume que a presença (ou ausência) de um dado atributo não está relacionada com a presença (ou ausência) de qualquer outro atributo [Mur06].

Este classificador considera então que cada descritor molecular contribui independentemente para a probabilidade do medicamento provocar ou não os efeitos adversos em estudo.

Dado um atributo de role = label 'C' de classes c1, c2, ..., cn e um vetor de atributos 'a' correspondente a todos os outros atributos, a probabilidade conditional de uma classe ci pode ser expressa como:

$$P(C = ci|a) = \frac{P(C = ci) * P(a|C = ci)}{P(a)}$$

A ferramenta utilizada para a elaboração deste trabalho disponibiliza ainda um operador 'Naive Bayes (kernel)'. Com este operador é possível utilizar atributos numéricos contínuos, o que não acontece com o algoritmo básico 'Naive Bayes'.

Tendo em conta a natureza dos descritores moleculares utilizados para esta experiência, o operador utilizado será o operador 'Naive Bayes (kernel)'. Este operador é baseado na estimativa de densidade do kernel. Posto isto, a probabilidade condicional P(a|C = ci) será calculada como a estimativa da densidade do kernel para a classe ci, tal que:

$$P(a|C = ci) = f_i(a)$$

com

$$f_i(a) = \frac{1}{Nh} \sum_{n=1}^{N} K_i \frac{(a - a_n)}{h}$$

Sendo que:

-a_n representa um ponto de treino -$K_i(a, a_n)$ representa a função kernel -h representa um parâmetro de suavização -N representa o número total de classes

'h' é normalmente chamado de *bandwidth* (largura de banda) e é um parâmetro escolhido pelo programador.

SVM

Dado um conjunto de exemplos de treino, cada um marcado como pertencente a uma de duas classes, o algoritmo SVM constrói um modelo que representa cada um dos exemplos como pontos no espaço. Esses pontos são mapeados de forma que os exemplos de cada categoria sejam divididos por uma linha recta. Exemplos pertencentes a classes diferentes devem ser representados no espaço o mais distante quanto possível.

Os novos exemplos, presentes no ficheiro de teste, são então mapeados no mesmo espaço e previstos como pertencentes a uma das duas classes, conforme o lado da recta em que sejam colocados. Svm é considerado por isso um classificador linear binário.

Tal como foi referido na Secção 2.3, através da função de *kernel*, o algoritmo SVM também é capaz de classificar dados não separáveis linearmente e atributos não binários.

Para este projeto será utilizado o operador 'Support Vector Machine (LibSVM)' que se baseia na libraria Java LibSVM [CL13].

Operadores de pré-seleção de atributos

De maneira a tentar optimizar os resultados obtidos com a realização da experiência 3, nesta Secção vão ser explorados alguns dos operadores de pré-seleção de algoritmos disponibilizados pela ferramenta RapidMiner, são eles:

- Weight by Correlation
- Weight by Gini Index
- Weight by Information Gain Ratio

Weight by Correlation

O operador 'Weight by Correlation' foi o algoritmo utilizado na realização da 3ª experiência. Este operador calcula o peso (weight) de um dado atributo em relação à label através da medida de correlação entre ambos.

A medida de correlação é um valor que pode variar entre -1 e 1 e que mede o grau de associação entre um atributo e a label. Quanto maior o peso de um atributo, mais relevante esse atributo é considerado para a resolução de um problema.

Se um atributo 'a' obtém uma correlação positiva significa que o atributo 'a' é diretamente proporcional à label. Se um atributo 'a' obtém uma correlação negativa, significa que esse atributo é inversamente proporcional à label. A correlação é calculada através da correlação de Pearson abordada em 3.1.

Weight by Gini Index

O operador Weight by Gini Index calcula o peso de um dado atributo em relação à label através do calculo da medida de impureza de Gini Index da distribuição da label. O cálculo da impureza de gini index é calculada através da fórmula:

$$GiniIndex = 1 - \sum_{i=1}^{J} p_i^2$$

Sendo que J representa o número de classes (labels) e pi representa a fracção de exemplos rotulados com a classe i.

Weight by Information Gain Ratio

O operador Weight by Information Gain Ratio calcula o peso de um atributo em relação à label através da proporção entre o cálculo da Information Gain (IG) e do valor intrínseco (IV) tal que:

$$InformationGainRatio(IGR) = \frac{InformationGain(IG)}{IntrinsicValue(IV)}$$

$$IG(E_x, a) = H(E_x) - \sum_{v \in values(a)} (\frac{|x \in E_x | value(x,a) = v|}{|E_x|} H(x \in E_x | value(x,a) = v)))$$

$$IV(E_x, a) = - \sum_{v \in values(a)} \frac{|x \in E_x | value(x,a) = v|}{|E_x|} \log_2 (\frac{|x \in E_x | value(x,a) = v|}{|E_x|})$$

Sendo que:

- Ex é o conjunto de todos os dados de treino
- values(x,a) com x$\in E_x$ corresponde ao valor do exemplo 'x' para o atributo 'a'
- H corresponde ao cálculo da Entropia
- value(a) corresponde a todos os valores possíveis para o aributo 'a'

A entropia é calculada segundo a seguinte fórmula:

$$Entropia = - \sum_{i=1}^{J} P_i \log_2 P_i$$

Sendo que J representa o número de classes(labels) e P_i representa a fracção de exemplos rotulados com a classe i.

Capítulo 4

Experiências e Resultados

Este capítulo visa avaliar o trabalho realizado ao longo deste projeto. Descrevemos os dados utilizados nas experiências bem como as experiências e os resultados obtidos.

4.1 Descrição dos Dados

O conjunto de dados utilizados provém da base de dados ADReCS. Como já foi referido na Secção 2.6.1 esta base de dados disponibiliza informações acerca de medicamentos e dos efeitos adversos associados a esses medicamentos.

Nas secções 3.1, 3.1 e 3.2 são referidos os processos de limpeza e redução de dados aplicados à base de dados para cada uma das experiências. As principais razões para a redução de atributos é a diminuição do tempo de treino de cada algoritmo e a melhoria geral do modelo, uma vez que atributos pouco frequentes poderão levar o modelo a tomar decisões baseadas no ruído e, por isso, levar ao overfitting do sistema. Alguns atributos representando descritores moleculares foram removidos por haver um número elevado de valores em falta *missing values*.

Resumidamente, foram selecionados os dados relevantes para o estudo, sendo eles Drug e Adr. Na experiência 1 foram explorados os nós folha da hierarquia de ADR na forma 03.01.04.012. Nas experiências 2 e 3, foram explorados apenas os nós imediatamente a seguir à raiz, ou seja, o ADR anterior tornou-se num ADR do grupo 3 (identificado pelos 2 primeiros dígitos do identificador).

Após este processo, para a experiência 1 obteve-se um conjunto de dados de 3 atributos (Drug, Adr e Rating) com 618 medicamentos e 437 efeitos adversos.

Para as experiências 2 obteve-se um conjunto de dados com 3 atributos (Drug, GrupoADR e Rating) 618 medicamentos e 26 grupos de efeitos adversos.

Para a experiência 3 obteve-se um conjunto de dados com atributos Drug, rating e 1524 descritores moleculares, posteriormente reduzidos durante a execução do processo. Para esta experiência foram utilizados um total de 340 exemplos (medicamentos).

4.2 Experiência 1

Nesta experiência foram utilizados os algoritmos *user k-nn, matrix factorization e slope one* recorrendo ao operador 'Optimize Parameters' disponibilizado através da extensão *Recommender Extension*. Este operador permite-nos escolher quais os parâmetros que pretendemos otimizar, a margem de valores possíveis e o número de iterações.

Os três algoritmos foram testados recorrendo à metodologia *K-fold Cross-Validation* referida na Figura 2.7.2 com K=10.

Os resultados obtidos estão sumarizados na Tabela 4.1.

	Accuracy	Precision	Recall	F-measure
Matrix Factorization	47.71(±0.29)	35.60(±0.25)	58.07(±0.61)	44.14(±0.35)
User K-NN	38.90(±1.67)	29.56(±0.64)	52.07(±5.19)	37.65(±1.88)
Slope One	23.89(±0.16)	22.82(±0.18)	47.82(±0.53)	30.89(±0.28)

Tabela 4.1: Resultados Experiência 1

Como se pode ver na Tabela 4.1, o algoritmo *Matrix Factorization* obteve os melhores resultados quando comparado aos algoritmos *User K-NN e Slope One*. Sendo que o algoritmo *Slope One* foi o que obteve pior desempenho entre os três.

O melhor desempenho obtido pelo algoritmo *Matrix Factorization* deve-se ao facto de entre os três, este ser o único algoritmo que considera os efeitos adversos e os medicamentos para a tomada de decisão. O algoritmo *User k-NN* prevê as classificações de um medicamento com base nas semelhanças encontradas com um outro grupo de medicamentos e o algoritmo *Slope One* prevê as classificações de um efeito adverso com base nas classificações de um outro efeito adverso.

A métrica *F-measure* relaciona as métricas de *Precision* e *Recall*, como já foi mencionado na Secção 2.7.1. Um modelo terá melhor desempenho quanto maior a sua *F-Measure*. A métrica *Accuracy* resume-se à percentagem de previsões corretas feitas pelo modelo.

A métrica *Recall* é responsável por medir os efeitos adversos reais que o sistema consegue prever como sendo, efetivamente, efeitos adversos. Para o algoritmo *Matrix Factorization* o sistema consegue encontrar 58,07% dos efeitos adversos reais.

A métrica *Precision*, obtém o seu melhor resultado mais uma vez utilizando o algoritmo *Matriz Factorization*. Esta medida indica-nos que apenas 35.60% dos efeitos adversos previstos pelo sistema são, de facto, efeitos adversos no mundo real, ou seja, o principal erro está em 0's reais serem previstos como 1's.

Foi também possível verificar que o parâmetro que mais influenciou a otimização do operador *Matrix Factorization* foi o número de fatores (NF). Quando maior o número de NF, maior a *accuracy* do modelo. Isto explica-se porque o modelo é capaz de capturar diferentes aspetos ou fatores sobre os dados. No entanto, após atingir um certo número de fatores, o algoritmo deixa de conseguir melhorar a sua performance devido ao *overfitting*.

Já no operador *User KNN*, verificou-se que o valor de k escolhido influenciava em muito o desempenho do modelo. Em geral, quanto menor o valor de k, melhor a performance do sistema. Note-se que o número de k diz respeito ao número de utilizadores (drug) com que o utilizador em estudo vai ser comparado. O valor de k escolhido pelo modelo através do operador 'Optimize Parameters' foi k=1. Isto só é possível devido ao grande número de medicamentos disponibilizados nos dados de treino.

4.3 Experiência 2

Os algoritmos utilizados no âmbito desta experiência dizem respeito à previsão de rating. São eles: *Matrix Factorization, User K-NN e Slope One*. São, portanto, os algoritmos utilizados na experiência anterior.

A diferença entre estas duas experiências, tal como foi referido em 3.1, está no atributo ADR.

Com esta experiência pretende-se prever quais os grupos de efeitos adversos sentidos aquando da toma de um medicamento em vez de prever um ADR individual. Os grupos de efeitos de adversos em estudo pode ser revisto na Figura 3.13.

Os resultados obtidos estão representados na Tabela 4.2.

	Accuracy	Precision	Recall	F-measure
Matrix Factorization	79.52(±0.87)	77.46(±0.88)	79.84(±1.44)	78.63(±0.97)
User K-NN	78.97(±0.82)	76.32(±0.78)	80.40(±1.32)	78.30(±0.91)
Slope One	78.71(±0.76)	76.24(±1.02)	79.75(±1.08)	77.95(±0.76)

Tabela 4.2: Resultados Experiência 2

Tal como aconteceu na experiência 1, o algoritmo que atingiu melhor desempenho foi o algoritmo *Matrix Factorization*, seguindo-se do algoritmo *User K-NN* e, por último, o algoritmo *Slope One*.

Também se verificou uma redução do parâmetro ótimo *NF (Number of factors)* proveniente do algoritmo *Matrix Factorization*. Esta redução tem origem na redução efetuada ao parâmetro adr que na experiência 1 contava com 437 efeitos adversos e na experiência 2 foi reduzido para 26.

Já o parâmetro k referente ao algoritmo *User K-NN* sofreu um aumento quando comparado com o da primeira experiência.

Os resultados gerais da experiência foram significativamente melhores aos resultados alcançados na Experiência 1. Isto deve-se ao facto de ter sido feita uma agregação aos efeitos adversos em estudo. Com esta redução, tornou-se mais fácil encontrar medicamentos mais semelhantes entre si.

4.4 Experiência 3

Como já foi referido na Secçãp 3.2, esta experiência visa tirar partido de algoritmos de classificação, com particular foco em algoritmos para os quais se possa conhecer os atributos mais relevantes usados na classificação. O conhecimento dos atributos mais relevantes usados pelo classificador podem ser pistas importantes para os especialistas perceberem a causa do efeito adverso. Os algoritmos de classificação selecionados para este projeto foram:

- Árvores de decisão com CART
- Random Forest com CART
- Naive Bayes
- Support Vector Machines

Na Tabela 4.3 podem ver-se os resultados obtidos para os 26 grupos de efeitos adversos utilizando o algoritmo árvores de decisão com CART. Esta experiência foi executada com *10-fold Cross Validation*.

GrupoADR	% true	Accuracy	Precision(T)	Recall(T)	Precision(M)	Recall(M)
G1	60.88	67.94(±6.76)	71.93(±4.71)	77.81(±11.59)	67.45(±7.56)	65.22(±6.27)
G2	78.53	80.59(±4.40)	82.82(±3.14)	95.17(±3.72)	73.66(±14.56)	61.34(±7.68)
G3	19.41	81.18(±1.95)	66.67(±0.00)	6.19(±12.43)	48.26(±16.88)	52.74(±5.48)
G4	45.59	63.53(±5.76)	59.75(±7.97)	68.46(±12.55)	65.04(±5.97)	64.01(±5.42)
G5	45.30	70.00(±6.28)	68.42(±7.11)	62.33(±12.51)	69.99(±6.16)	69.38(±6.55)
G6	62.35	67.65(±7.56)	70.10(±4.93)	83.90(±9.63)	65.64(±11.14)	62.30(±7.50)
G7	88.24	89.41(±1.44)	89.57(±1.79)	99.67(±1.00)	63.12(±23.12)	56.08(±8.02)
G8	94.41	94.41(±0.88)	94.41(±0.88)	100.00(±0.00)	47.21(±0.44)	50.00(±0.00)
G9	47.35	66.47(±9.59)	66.02(±12.11)	60.15(±15.26)	66.85(±10.17)	66.17(±9.88)
G10	81.76	84.41(±3.49)	86.99(±3.27)	95.30(±2.35)	75.49(±9.33)	65.39(±8.59)
G11	62.94	71.47(±5.89)	76.01(±5.13)	79.83(±5.45)	69.42(±6.19)	68.57(±6.09)
G12	60.29	65.29(±7.30)	70.72(±7.54)	74.19(±6.92)	63.46(±7.99)	63.00(±8.45)
G13	73.53	74.71(±3.00)	77.59(±3.19)	92.80(±5.60)	61.79(±13.59)	58.62(±6.65)
G14	75.59	79.12(±5.33)	81.90(±3.42)	93.02(±4.84)	73.18(±11.76)	64.56(±7.66)
G15	69.12	70.00(±7.42)	75.73(±4.76)	83.33(±8.64)	65.00(±10.07)	61.53(±8.03)
G16	69.12	70.59(±6.71)	68.80(±19.21)	44.00(±16.94)	73.63(±10.97)	66.93(±8.92)
G17	90.59	91.18(±1.86)	91.94(±2.18)	99.02(±1.49)	57.22(±18.97)	56.59(±11.65)
G18	10.00	91.76(±2.88)	71.43(±0.00)	28.33(±24.78)	74.68(±21.50)	63.51(±12.39)
G19	77.06	80.29(±4.17)	83.15(±4.28)	93.90(±4.91)	75.41(±13.14)	64.01(±9.77)
G20	64.70	70.00(±3.90)	73.28(±2.83)	84.55(±4.17)	67.02(±5.30)	63.94(±4.36)
G21	48.24	67.65(±6.44)	67.33(±7.93)	66.51(±10.50)	68.22(±6.32)	67.65(±6.38)
G22	75.88	77.65(±3.77)	78.41(±3.25)	97.69(±3.53)	62.70(±22.59)	56.14(±7.61)
G23	92.06	92.65(±1.97)	92.63(±1.93)	100.00(±0.00)	51.31(±15.72)	53.33(±10.00)

G24	88.53	90.00(±1.44)	89.87(±1.33)	100.00(±0.00)	69.93(±25.52)	56.25(±6.25)
G25	22.06	78.24(±5.29)	48.65(±19.99)	43.75(±21.42)	66.74(±12.02)	66.03(±9.93)
G26	17.35	83.53(±4.59)	54.95(±28.70)	33.333(±21.08)	71.10(±15.24)	63.64(±10.59)

Tabela 4.3: Resultados Experiência 3 - árvores de decisão com CART

A primeira coluna representa o grupo ADR a ser testado. A segunda coluna representa a percentagem de exemplos verdadeiros de cada ficheiro de dados. De seguida, temos a métrica Accuracy, seguindo-se das métricas Precision e Recall da classe = true. Por último, vemos as métricas Precision e Recall médias (classe true e classe false).

Pode ver-se que os resultados de Accuracy obtidos estão a cima da classe maioritária. Em alguns casos bastante acima (ex: G9, G21).

Na Tabela 4.4, são apresentados os resultados utilizando o algoritmo Random Forest utilizando mais uma vez *10-fold Cross Validation*.

GrupoADR	**% true**	**Accuracy**	**Precision(T)**	**Recall(T)**	**Precision(M)**	**Recall(M)**
G1	60.88	69.41(±7.23)	70.10(±5.51)	87.45(±7.15)	69.27(±10.91)	64.55(±7.46)
G2	78.53	80.88(±1.47)	81.03(±1.75)	98.87(±1.72)	70.51(±21.20)	56.85(±4.99)
G3	19.41	82.06(±3.07)	100.00(±0.00)	7.86(±12.60)	55.92(±24.14)	53.93(±6.30)
G4	45.59	66.18(±6.34)	70.96(±11.70)	44.79(±13.24)	67.96(±7.61)	64.59(±6.55)
G5	45.30	71.76(±7.80)	74.14(±14.80)	60.96(±8.08)	72.74(±9.43)	70.84(±7.72)
G6	62.35	70.00(±8.40)	71.75(±6.18)	85.84(±11.86)	70.73(±10.61)	64.85(±8.40)
G7	88.24	89.41(±1.95)	89.83(±2.13)	99.33(±2.00)	62.42(±23.88)	57.17(±9.45)
G8	94.41	95.00(±2.65)	95.53(±1.99)	99.38(±1.25)	62.77(±23.72)	59.69(±16.78)
G9	47.35	69.12(±6.99)	69.48(±7.66)	60.85(±12.87)	69.32(±7.16)	68.69(±7.28)
G10	81.76	84.41(±1.88)	84.22(±1.59)	99.64(±1.07)	79.61(±20.62)	57.80(±4.95)
G11	62.94	72.35(±3.99)	71.95(±3.97)	92.51(±3.83)	74.43(±5.13)	65.36(±5.10)
G12	60.29	66.18(±8.55)	64.48(±4.76)	80.64(±13.56)	66.88(±12.13)	62.57(±7.82)
G13	73.53	76.76(±1.58)	76.53(±1.68)	98.80(±1.83)	78.68(±15.42)	57.18(±3.79)
G14	75.59	78.82(±5.39)	78.95(±4.31)	98.43(±1.92)	70.72(±23.81)	58.59(±7.92)
G15	69.12	73.82(±4.04)	75.20(±3.04)	92.74(±2.81)	70.51(±7.20)	62.10(±4.80)
G16	69.12	76.18(±4.45)	77.50(±18.58)	34.45(±15.07)	77.08(±9.69)	64.71(±7.17)
G17	90.59	92.06(±3.24)	92.51(±2.97)	99.34(±1.31)	69.59(±25.03)	61.34(±12.91)
G18	10.00	92.35(±1.95)	100.00(±0.00)	23.33(±16.58)	81.10(±23.58)	61.67(±8.29)
G19	77.06	80.29(±2.96)	80.46(±2.30)	98.48(±3.52)	75.23(±20.50)	58.70(±6.45)
G20	62.94	70.00(±5.33)	72.36(±4.45)	85.97(±4.69)	68.20(±7.62)	64.85(±6.26)
G21	48.24	71.47(±8.11)	70.23(±8.70)	73.27(±5.66)	71.64(±8.08)	71.44(±8.07)

G22	75.88	77.94(±1.97)	77.69(±1.84)	99.62(±1.15)	62.18(±24.35)	54.60(±5.00)
G23	92.06	92.94(±1.95)	92.90(±1.90)	100.00(±0.00)	61.45(±23.59)	55.83(±9.17)
G24	88.53	90.00(±2.35)	90.36(±1.69)	99.34(±1.31)	75.18(±25.13)	58.42(±8.38)
G25	22.06	80.00(±2.88)	65.00(±29.30)	9.25(±18.67)	72.95(±15.17)	58.06(±4.90)
G26	17.35	85.00(±3.82)	72.22(±0.00)	22.33(±17.32)	72.06(±22.07)	60.28(±8.93)

Tabela 4.4: Resultados Experiência 3 - random forest com CART

Verificou-se uma melhoria na medida de accuracy em 20/26 dos grupos testados com o algoritmo Random Forest quando comparado com o algoritmo CART em árvores de decisão. Sendo que ambos os algoritmos alcançaram exatamente o mesmo valor de accuracy para 6/26 grupos. Essa melhoria pode ser verificada no gráfico da Figura 4.3.

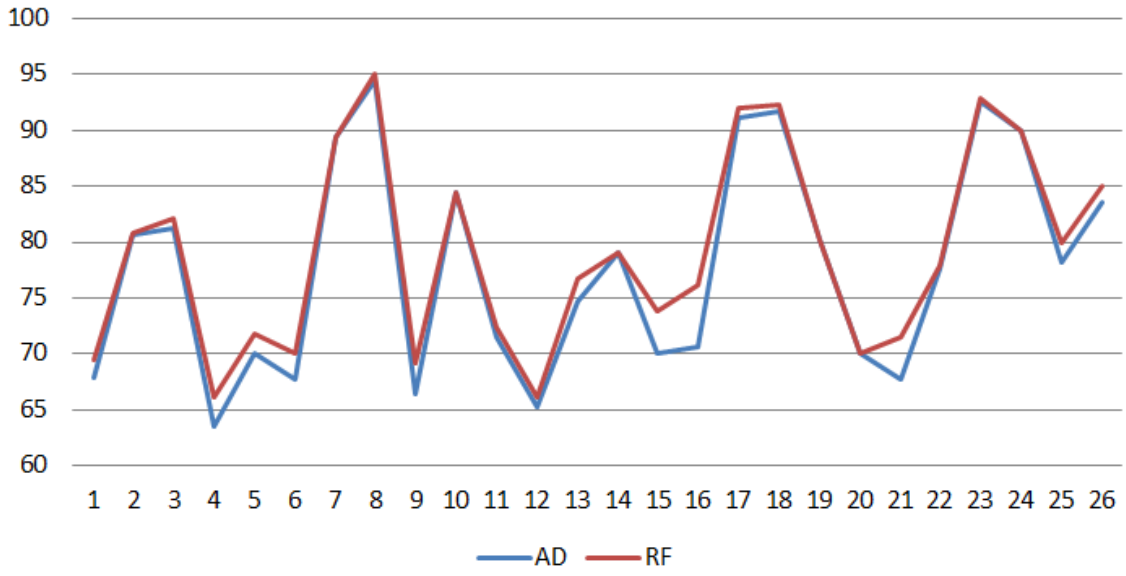

Figura 4.1: Métrica Accuracy para o CART (linha azul no gráfico) e o Random Forest (linha vermelha no gráfico).

Na Figura 4.3 é possível verificar que as maiores diferenças de valores se verificam para os grupos de efeitos adversos que obtiveram piores valores de *accuracy* no algoritmo das árvores de decisão.

De seguida, foi testado o algoritmo Naive Bayes (NB). Os resultados são apresentados na Tabela 4.5.

GrupoADR	**% true**	**Accuracy**	**Precision(T)**	**Recall(T)**	**Precision(M)**	**Recall(M)**
G1	60.88	64.84(±2.90)	63.78(±2.23)	98.52(±3.11)	71.89(±17.24)	55.20(±2.90)

G2	78.53	81.48(±2.98)	81.67(±2.77)	99.25(±1.51)	60.84(±22.67)	54.31(±4.88)
G3	19.41	81.87(±3.06)	75.00(±0.00)	5.36(±8.64)	55.98(±23.72)	52.50(±4.46)
G4	45.59	60.88(±6.45)	56.88(±6.53)	59.50(±14.73)	61.23(±6.54)	60.88(±6.60)
G5	45.30	64.71(±6.03)	76.24(±26.76)	28.12(±11.08)	68.96(±15.29)	61.64(±5.73)
G6	62.35	66.18(±4.00)	65.31(±2.61)	97.62(±3.19)	73.49(±13.03)	55.86(±3.95)
G7	88.24	88.24(±0.00)	88.24(±0.00)	100.00(±0.00)	44.12(±0.00)	50.00(±0.00)
G8	94.41	95.75(±1.44)	96.03(±1.29)	99.68(±0.97)	58.01(±20.22)	54.84(±10.09)
G9	47.35	60.88(±6.58)	58.86(±0.00)	57.76(±31.10)	60.71(±13.37)	60.77(±7.17)
G10	81.76	83.28(±2.73)	83.56(±2.28)	99.29(±1.43)	59.28(±23.18)	53.81(±5.64)
G11	62.94	64.44(±3.64)	64.17(±3.75)	99.09(±2.73)	50.66(±23.94)	52.81(±4.83)
G12	60.29	62.91(±6.26)	66.11(±4.78)	78.56(±9.39)	61.02(±7.55)	58.87(±6.39)
G13	73.53	76.47(±2.94)	76.79(±2.10)	97.60(±3.20)	73.90(±17.00)	57.69(±4.81)
G14	75.59	77.10(±3.00)	77.74(±3.30)	98.41(±2.60)	52.20(±21.81)	53.58(±7.23)
G15	69.12	70.88(±4.45)	72.27(±2.66)	94.06(±3.85)	65.72(±16.04)	56.44(±6.25)
G16	69.12	70.29(±4.04)	61.11(±0.00)	10.36(±6.60)	67.06(±20.55)	53.70(±4.75)
G17	90.59	91.93(±3.29)	92.17(±3.29)	99.67(±1.00)	53.59(±16.47)	53.17(±6.34)
G18	10.00	90.00(±1.44)	0.00(±0.00)	0.00(±0.00)	45.00(±0.72)	50.00(±0.00)
G19	77.06	79.41(±2.94)	80.04(±2.12)	97.71(±1.87)	71.27(±15.58)	57.69(±5.84)
G20	64.70	65.88(±3.28)	65.97(±1.84)	97.73(±3.66)	57.99(±23.16)	52.61(±3.79)
G21	48.24	68.24(±5.55)	69.53(±12.09)	68.49(±10.79)	69.93(±6.34)	68.49(±5.36)
G22	75.88	78.48(±3.35)	79.89(±1.84)	96.03(±4.25)	76.94(±13.77)	59.09(±4.30)
G23	92.06	92.58(±2.99)	93.37(±2.54)	99.03(±1.48)	59.19(±21.23)	57.85(±12.93)
G24	88.53	89.93(±2.71)	90.41(±2.02)	99.30(±1.41)	60.21(±23.30)	53.81(±6.72)
G25	22.06	79.42(±2.76)	100.00(±0.00)	5.77(±9.58)	54.60(±23.76)	52.89N±4.79)
G26	17.35	82.94(±1.18)	100.00(±0.00)	1.67(±5.00)	46.45(±15.33)	50.83(±2.50)

Tabela 4.5: Resultados Experiência 3 Naive Bayes

O algoritmo NB obteve uma métrica de accuracy inferior à obtida no algoritmo RF para 24/26 grupos de efeitos adversos em estudo.

De seguida, foi testado o algoritmo Support Vector Machine (SVM) com kernel Radial Basis Function (RBF). Os resultados são apresentados na Tabela 4.6.

GrupoADR	**% true**	**Accuracy**	**Precision(T)**	**Recall(T)**	**Precision(M)**	**Recall(M)**
G1	60.88	72.35(±7.91)	74.52(±7.91)	84.64(±8.37)	72.10(±8.86)	68.94(±9.24)
G2	78.53	81.47(±2.30)	81.56(±2.50)	98.89(±1.70)	75.36(±19.91)	58.37(±5.75)
G3	19.41	81.76(±2.88)	58.33(±0.00)	21.19(±13.74)	67.22(±16.13)	58.77(±6.55)

G4	45.59	63.53(±11.63)	60.55(±16.67)	54.45(±18.16)	63.03(±13.07)	62.68(±12.01)
G5	45.30	72.06(±7.47)	72.62(±9.69)	61.12(±14.50)	72.47(±8.13)	71.15(±7.99)
G6	62.35	71.18(±5.23)	76.48(±5.80)	78.74(±8.28)	69.81(±6.44)	68.51(±6.53)
G7	88.24	89.41(±3.53)	89.84(±2.92)	99.33(±1.33)	59.92(±24.22)	57.17(±12.96)
G8	94.41	95.29(±1.44)	95.27(±1.41)	100.00(±0.00)	62.63(±23.47)	57.50(±11.46)
G9	47.35	68.24(±6.94)	71.46(±12.60)	58.38(±6.85)	64.29(±8.25)	67.75(±6.71)
G10	81.76	84.12(±4.40)	85.12(±3.39)	97.57(±2.37)	67.56(±22.69)	60.00(±10.83)
G11	62.94	72.35(±7.80)	76.77(±7.13)	80.80(±9.12)	71.07(±8.77)	69.50(±8.11)
G12	60.29	65.29(±6.28)	66.68(±4.84)	86.43(±6.21)	63.23(±10.05)	59.75(±8.07)
G13	73.53	77.65(±6.86)	78.73(±4.56)	95.60(±4.88)	75.45(±15.35)	61.69(±9.71)
G14	75.59	78.53(±4.57)	79.37(±3.38)	96.89(±3.36)	75.85(±12.77)	59.42(±6.71)
G15	69.12	72.35(±5.46)	73.83(±3.53)	93.19(±5.44)	68.75(±15.52)	59.37(±7.16)
G16	69.12	75.59(±4.75)	73.88(±19.37)	31.09(±12.59)	74.83(±10.77)	63.23(±7.07)
G17	90.59	91.58(±2.28)	91.89(±1.83)	99.02(±1.49)	65.94(±22.54)	57.43(±7.93)
G18	10.00	91.18(±2.28)	83.33(±0.00)	15.00(±15.28)	70.67(±25.71)	57.34(±7.81)
G19	77.06	79.12(±4.04)	79.41(±2.59)	98.46(±2.55)	73.04(±23.23)	56.28(±6.08)
G20	64.70	69.12(±6.06)	71.95(±6.11)	87.27(±5.30)	64.99(±9.69)	61.55(±8.69)
G21	48.24	71.76(±4.20)	81.13(±5.81)	54.19(±11.09)	74.47(±3.95)	71.13(±4.64)
G22	75.88	78.82(±3.90)	79.17(±3.19)	98.06(±2.59)	66.25(±21.09)	58.27(±6.82)
G23	92.06	93.24(±1.88)	93.17(±1.84)	100.00(±0.00)	66.59(±25.13)	57.50(±9.46)
G24	88.53	90.00(±1.44)	89.87(±1.33)	100.00(±0.00)	69.93(±25.52)	56.25(±6.25)
G25	22.06	79.71(±3.07)	72.33(±25.77)	21.43(±10.83)	76.80(±12.69)	58.83(±5.01)
G26	17.35	84.12(±4.40)	59.79(±23.40)	39.00(±10.55)	73.88(±12.43)	66.30(±6.20)

Tabela 4.6: Resultados Experiência 3 – Support Vector Machines

A Figura 4.3 mostra uma comparação da accuracy para os 26 grupos considerados e para os quatro algoritmos utilizados.

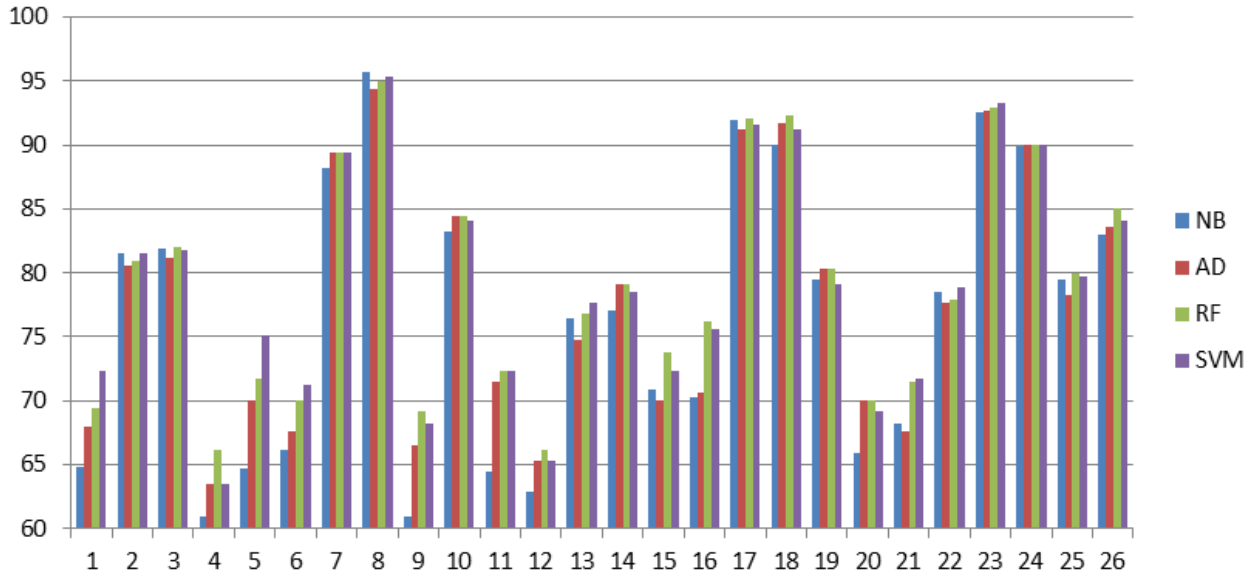

Figura 4.2: Métrica Accuracy para os quatro algoritmos em estudo

Pode ver-se na figura que para todos os algoritmos, o grupo de ADR que obteve a melhor accuracy foi o grupo G8. Grupo esse que contém a maior percentagem da classe maioritária (94.41%). O pior resultado, para todos os algoritmos, corresponde ao grupo de ADR G4 cuja classe maioritária contém 54.41% dos dados.

Na Tabela 4.7 pode ver-se a média global de *Accuracy*, nos 26 grupos, obtida por cada algoritmo.

Algortimo	Accuracy Médias	Nº Grupos Optimos
NB	76.50	2
AD	77.68	6
SVM	78.95	17
RF	79.03	10

Tabela 4.7: Resultados médios da métrica de accuracy

Resumidamente, o algoritmo RF foi o algoritmo que alcançou a melhor métrica de accuracy, em média, para os 26 grupos de EAM em estudo (79.03%). Seguindo-se do algoritmo SVM com uma média de accuracy de 78.95%.

A tabela 4.8 apresenta as médias de Accuracy para todos os grupos de ADR. Na terceira coluna podemos ver número de grupos de EAM para os quais cada algoritmo obteve a melhor Accuracy. O algortimo RF obteve o resultado optimo para 17/26 grupos de EAM.

Posto isto, para o algoritmo RF foi elaborado um novo processo que diferencia o operador de selecção de atributos. Recorde-se que para as experiências iniciais foi utilizado o operador 'Weight by correlation' para todos os algoritmos. Na tabela abaixo, são apresentados os resultados obtidos

para os grupos 4 e 8, recorrendo aos métodos de pré selecção 'Weight by correlation' (WC), Weight by gini index' (WGI) e 'Weight by information gain ratio' (WIGR):

Grupo EAM	Seleção Atributos	Accuracy
G4	WC	63.24(±4.79)
G4	WGI	65.59(±8.22)
G4	WIGR	65.00(±4.25)
G8	WC	95.00(±2.65)
G8	WGI	95.00(±1.35)
G8	WIGR	95.00(±1.35)

Tabela 4.8: Resultados para diferentes métodos de seleção de atributos

Como se pode ver na tabela imediatamente acima, para o G8 (o grupo que tem a maior percentagem de classe maioritária), a mudança do critério de seleção de atributos não obteve qualquer mudança na métrica de *accuracy*. Já para o grupo G4 (um grupo que tem a classe true e false quase nas mesmas proporções), foi possível melhorar os resultados de *accuracy*, *Precision* e *Recall* através da utilização do operador 'Weight by Gini Index' para a pré-selecção de atributos.

É sabido que o algoritmo RF utiliza o cálculo da impureza de Gini Index para a selecção do próximo nó da árvore de decisão a ser explorado.No entanto, o cálculo do gini index não é calculado para todos os nós da árvore, mas apenas para uma selecção aleatoria de nós. Utilizando a pré-seleção dos nós mais relevantes, faz com que os nós menos relevantes sejam excluidos da base de dados e, por isso, não sejam selecionados para exploração. Posto isto, tendo-se verificado que o critério de selecção de atributos *Gini Index* melhora a accuracy dos resultados, a experiência foi repetida, para todos os grupos de efeitos adversos:

GrupoADR	**% true**	**Accuracy**	**Precision(T)**	**Recall(T)**	**Precision(M)**	**Recall(M)**
G1	60.88	71.76(±7.80)	72.44(±6.47)	87.45(±6.85)	71.63(±10.47)	67.38(±8.93)
G2	78.53	81.18(±3.53)	81.14(±3.28)	99.25(±1.51)	66.82(±23.80)	57.30(±7.21)
G3	19.41	82.65(±3.59)	88.89(±0.00)	12.62(±11.72)	71.26(±25.84)	56.12(±6.08)
G4	45.59	65.59(±8.22)	63.62(±11.11)	56.21(±16.11)	65.60(±8.99)	64.90(±8.67)
G5	45.30	72.06(±8.85)	71.57(±11.82)	63.50(±13.00)	72.22(±9.27)	71.33(±9.27)
G6	62.35	71.47(±7.21)	71.70(±4.40)	89.68(±11.90)	74.58(±11.13)	65.54(±6.88)
G7	88.24	89.41(±1.44)	89.30(±1.31)	100.00(±0.00)	64.65(±25.15)	55.00(±6.12)
G8	94.41	95.00(±1.35)	94.98(±1.32)	100.00(±0.00)	57.49(±20.50)	55.00(±10.00)
G9	47.35	72.35(±4.40)	72.45(±7.69)	69.63(±10.33)	73.16(±4.89)	72.27(±4.39)
G10	81.76	84.12(±2.70)	84.37(±1.96)	98.90(±1.68)	75.52(±20.12)	58.26(±5.54)
G11	62.94	70.59(±6.03)	72.34(±5.28)	86.88(±4.16)	68.71(±7.21)	64.85(±7.24)
G12	60.29	67.65(±2.63)	67.45(±3.16)	90.71(±8.03)	72.58(±8.77)	61.59(±4.02)
G13	73.53	77.06(±4.32)	77.66(±3.17)	96.80(±2.40)	74.08(±12.68)	59.51(±7.01)

G14	75.59	79.71(±4.45)	80.16(±4.11)	97.69(±2.55)	72.41(±20.12)	60.93(±9.14)
G15	69.12	75.00(±2.71)	75.35(±3.10)	95.38(±4.73)	76.82(±9.41)	62.55(±4.55)
G16	69.12	75.29(±4.78)	77.74(±14.09)	28.73(±12.86)	76.47(±8.31)	62.43(±6.60)
G17	90.59	92.35(±2.35)	92.73(±1.45)	99.34(±1.31)	78.87(±23.17)	61.76(±8.67)
G18	10.00	92.65(±3.29)	84.62(±0.00)	34.17(±23.99)	81.51(±20.96)	66.76(±12.08)
G19	77.06	81.76(±4.12)	82.11(±3.24)	97.71(±2.54)	81.47(±10.96)	63.14(±6.61)
G20	64.70	70.29(±5.65)	71.58(±3.72)	90.00(±6.98)	69.86(±11.22)	62.08(±6.71)
G21	48.24	73.24(±5.33)	71.04(±5.60)	74.89(±9.28)	73.59(±5.75)	73.29(±5.33)
G22	75.88	77.06(±2.56)	77.03(±2.68)	99.62(±1.15)	52.27(±22.24)	52.93(±5.25)
G23	92.06	93.24(±1.88)	93.17(±1.84)	100.00(±0.00)	66.59(±25.13)	57.50(±9.46)
G24	88.53	90.29(±1.88)	90.65(±1.92)	99.34(±1.31)	75.32(±22.46)	60.09(±7.59)
G25	22.06	80.59(±2.35)	84.62(±0.00)	14.29(±8.71)	75.21(±20.55)	56.76(±4.50)
G26	17.35	84.41(±3.96)	61.08(±6.52)	25.33(±10.97)	77.63(±18.16)	61.08(±6.52)

Tabela 4.9: Resultados Experiência 3 - random forest com CART e Gini Index

Comparando o algoritmo RF com pré-seleção de atributos utilizando correlação e utilizando a impureza de gini index verificou-se um aumento da métrica de accuracy média de 0.46%. A optimização do critério de pré-seleção de atributos conseguiu trazer melhorias para 18/26 grupos em estudo, alcançando uma média de accuracy de 79.49%. O gárifo da Figura 4.3 mostra a comparação da métrica de accuracy para o algoritmo RF com os dois critérios de pré-seleção estudados: correlação e Gini index.

Figura 4.3: Métrica Accuracy para os dois testes ao algoritmo RF

No gráfico da Figura 4.3 é possível verificar que tal como aconteceu nas experiências anteriores, os grupos de ADR que conseguiram uma melhoria mais acentuada foram os grupos com menor accuracy.

Tirando partido do facto do algoritmo RF não ser um algoritmo do tipo *black box*, foi feito um estudo relativamente aos descritores moleculares que o algoritmo de DM escolheu como raiz de cada árvore construída.

Os melhores descritores foram os seguintes:

- Os descritores moleculares *spmax*1_*bhe* e mats2s foram seleccionados para nó raíz para 23% dos grupos de ADR testados.
- Os descritores moleculares $spmin1_bhm$, sssch2, mwc10, tpipc, aatsc0c lipoaffinityindex, $spmin3_bhv$ e mic4 foram selecionados para 12% dos grupos de ADR.

O significado destes descritores é o seguinte: [1]

spmax1_bhe maior valor absoluto presente na matrix de Burden n1 sobre o valor de eletronegatividade de Sanderson.

mats2s autocorrelação de Moran - lag 2 sobre o estado-I

spmin1_bhm menor valor absoluto presente na matriz de Burden n1 sobre a massa relativa

sssch2 soma de todos os átomos do tipo CH2

mwc10 contagem de passos moleculares de ordem 10 (ln (1 + x)

tpipc ordem de ligação convencional total (até à ordem 10) (ln (1 + x))

aatsc0c Autocorrelação centrada média Broto-Moreau - lag 0 sobre a carga

lipoaffinityindex índice de lipoaffinity

spmin3_bhv menor valor absoluto da matriz de Burden n3 sobre os volumes relativos de van der Waals

mic4 índice de conteúdo de informação modificado (simetria de vizinhança de ordem 4)

Esta experiência sumariza os descritores moleculares presentes como atributo raiz para o maior número de grupos de efeitos adversos estudados. Estes descritores não serão por isso explicativos de um dado grupo de ADR em particular.

Foi ainda feito um segundo estudo que contabilizou grupo a grupo quais os descritores moleculares selecionados para cada uma das árvores e com que frequência. Nesta segunda experiência foram tomados em conta todos os nós das árvores e não apenas os atributos raiz.

[1] http://www.scbdd.com/padel_desc/descriptors/

De notar que para cada grupo de ADR, foram feitas 10 árvores de decisão.

Foi então possível sumarizar os grupos que selecionaram o mesmo descritor molecular em mais do que 5/10 árvores de decisão geradas:

- G2 selecionou o descritor minaach para 7/10 árvores e o descritor maxaasc para 6/10 árvores
- G16 selecionou o descritor atsc4m para 6/10 árvores
- G19 selecionou o descritor aatsc2v para 6/10 árvores
- G23 selecionou os descritores aatsc2e e vr2_d para 6/10 árvores
- G26 selecionou o descritor aatsc1s para 6/10 árvores

O significado destes descritores é o seguinte:

minaach mínimo átomos-tipo E-State: :CH:

maxaasc máximo de átomos-tipo E-State: :C:-

atsc4m Autocorrelação Broto-Moreau centrada - lag 4 sobre a massa

aatsc2v Autocorrelação centrada média de Broto-Moreau - lag 2 sobre os volumes de van der Waals

aatsc2e Autocorrelação centrada média de Broto-Moreau - lag 2 sobre as eletronegatividades de Sanderson

vr2_d Índice normalizado baseado em eigenvector tipo Randic a partir da matriz de distância topológica

aatsc1s Autocorrelação Broto-Moreau média centrada - lag 1 sobre o estado-I

Esta segunda experiência visa relacionar os descritores moleculares mais frequentemente selecionados para cada um dos grupos de efeitos adversos de medicamentos. Com esta experiência tentou-se relacionar um dado descritor molecular como a causa do aparecimento do grupo de efeitos adversos em estudo.

Até à conclusão desta dissertação não foi possível obter a opinião de um especialista sobre o interesse dos atributos para a explicação dos ADRs.

Capítulo 5

Conclusões e Trabalho Futuro

Satisfação dos Objetivos

Este trabalho foi motivado pela crescente ocorrência de efeitos adversos sentido e pela possibilidade de conseguir prever esses efeitos adversos antes de recorrer aos usuais testes clínicos.

A primeira abordagem ao problema recorreu a sistemas de recomendação com recurso à base de dados ADReCS. Os modelos foram alimentados com medicamentos, efeitos adversos e a relação existente entre cada par (medicamento-efeito adverso). Os algoritmos utilizados foram: *Slope One, User K-NN e Matrix Factorization*. O algoritmo que obteve a maior percentagem de acertos foi o algoritmo *Matrix Factorization*, com *accuracy*=47.71%.

Tendo em conta os resultados insatisfatórios, os dados de entrada dos algoritmos foram agrupados em efeitos adversos que atuam no mesmo sistema de órgãos. Recorrendo aos mesmos algoritmos referidos acima, a melhor percentagem de acertos foi mais uma vez obtida com o algoritmo *Matrix Factorization*, com uma *accuracy* de 79.52%.

Posteriormente, os dados utilizados na Experiência 2, foram alimentados por uma segunda base de dados que relaciona cada medicamento a um conjunto de descritores moleculares. Estes dados foram explorados recorrendo a algoritmos de classificação como Árvores de Decisão com CART,*Random Forest*, *Naive Bayes* e SVM.

O algoritmo que obteve, em média, a melhor métrica de accuracy na experiência 3 foi o algoritmo Random Forest com uma métrica de accuracy de 79.02%. Posto isto, o algoritmo foi optimizado recorrendo a pré-selecção dos atributos mais relevantes recorrendo ao cálculo da impureza de Gini Index. Esta nova experiência obteve uma taxa de accuracy média de 70.49%.

Os resultados obtidos são encorajantes para a prossecução desta linha de investigação. Há no entanto muito trabalho ainda a fazer tal como é referido na secção seguinte.

Trabalho Futuro

Uma das propostas para a melhoria dos modelos, é que seja levada em consideração a frequência com que cada efeito adverso foi encontrado aquando da toma de um medicamento. Em vez de modelos binários, onde se prevê se um medicamento provoca ou não um dado efeito adverso, poderia prever-se a frequência com que cada medicamento provoca cada efeito adverso.

Outro ponto a considerar para a previsão de efeitos adversos, são os dados do doente. Dados como a idade, a raça, o peso, a altura e doenças atuais ou prévias são alguns dos fatores que podem influenciar os efeitos adversos sentidos. Para além de relacionar as características de cada medicamento com o efeito adverso sentido (o que aconteceu na experiência 3), seria interessante relacionar os dados do doente com os efeitos adversos encontrados.

Por fim, seria também interessante estudar os efeitos adversos provocados pela interaccão entre medicamentos. Através das interações entre medicamentos poderá ser possível encontrar efeitos adversos que não seriam provocados se os medicamentos fossem ingeridos em exclusividade. Esta experiência poderia ser feita recorrendo a sistemas de recomendação, onde ambos os itens e os utilizadores seriam representados por medicamentos e o rating representaria a interação indesejada entre os mesmos.

Referências

[CL13] Chih-chung Chang e Chih-jen Lin. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–39, 2013. doi:10.1145/1961189.1961199.

[DGJ+16] Allan Peter Davis, Cynthia J. Grondin, Robin J. Johnson, Daniela Sciaky, Benjamin L. King, Roy McMorran, Jolene Wiegers, Thomas C. Wiegers e Carolyn J. Mattingly. The Comparative Toxicogenomics Database: update 2017. *Nucleic Acids Research*, 9880:gkw838, 2016. URL: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gkw838, doi:10.1093/nar/gkw838.

[FHW16] Eibe Frank, Mark A Hall e Ian H Witten. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques". *Morgan Kaufmann, Fourth Edition*, 2016.

[FPSS96] Usama Fayyad, G Piatetsky-Shapiro e Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, pages 37–54, 1996. URL: http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1230, doi:10.1145/240455.240463.

[GSZ07] Jan E Gewehr, Martin Szugat e Ralf Zimmer. BioWeka — extending the Weka framework for bioinformatics. 23(5):651–653, 2007. doi:10.1093/bioinformatics/btl671.

[Han05] Jiawei Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[HGMMO02] J Hasford, M Goettler, K-H Munter e B Müller-Oerlinghausen. Physicians' knowledge and attitudes regarding the spontaneous reporting system for adverse drug reactions. *Journal of clinical epidemiology*, 55:945–950, 2002. doi:10.1016/S0895-4356(02)00450-X.

[HNF+] Mark Hall, Hazeltine National, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann e Ian H Witten. The WEKA Data Mining Software : An Update. 11(1):10–18.

[JBDFA15] Teresa Juan-Blanco, Miquel Duran-Frigola e Patrick Aloy. IntSide: A web server for the chemical and biological examination of drug side effects. *Bioinformatics*, 31(4):612–613, 2015. doi:10.1093/bioinformatics/btu688.

[KBV09] Y. Koren, R. Bell e C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):42–49, 2009. doi:10.1109/MC.2009.263.

[LM05] Daniel Lemire e Anna Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. *Society for Industrial and Applied Mathematics*, pages 471–475, 2005. URL: http://cogprints.org/4031/, doi:10.1145/2891406.

[Man] Operator Reference Manual. RapidMiner 7.

[MBŠ12] M Mihelčić, M Bošnjak e T Šmuc. Extending RapidMiner with recommender systems algorithms Recommender Extension operators. *RapidMiner Community Meeting and Conference*, 2012.

[Mur06] Kevin P Murphy. Naive Bayes classifiers Generative classifiers. *Bernoulli*, 4701(October):1–8, 2006. URL: http://www.springerlink.com/index/0060368703612735.pdf, doi:10.1007/978-3-540-74958-5_35.

[PCC] Diogo Pinto, Pedro Costa e Rui Camacho. Predicting Drugs Adverse Side-Effects using a recommender-system.

[RSG01] Evelyn M. Rodriguez, Judy A. Staffa e David J. Graham. The role of databases in drug postmarketing surveillance. *Pharmacoepidemiology and Drug Safety*, 10(5):407–410, 2001. doi:10.1002/pds.615.

[WFH11] Ian H. Witten, Eibe Frank e Mark a. Hall. Data Mining: Practical Machine Learning Tools and Techniques, Third Edition. In *Annals of Physics*, volume 54, chapter 5, page 664. 2011. URL: http://www.cs.waikato.ac.nz/{~{}}ml/weka/book.html{\T1\textbackslash}nhttp://www.amazon.com/Data-Mining-Practical-Techniques-Management/dp/0123748569, doi:10.1002/1521-3773(20010316)40:6<9823::AID-ANIE9823>3.3.CO;2-C.

[Wor02] World Health Organisation. Safety of Medicines. 2002.