Aprendizagem e Mineração de Dados

Projeto A/A1

Grupo 17

Tiago Conceição, nº 47611

Mestrado em Engenharia Informática e de Computadores

Semestre de Inverno 2021/2022

19/12/2021

Índice

Índice de figuras

1. Introdução

A realização deste trabalho prático tem como objetivo a aplicação dos conceitos de *data*, *information* e *knowledge* em dois cenários específicos. A aquisição de informação e conhecimento é feita através de algoritmos de *machine learning* e *data-mining* como o *One Rule*, *Naive Bayes* e *Iterative Dichotomiser 3* dos quais os três serão implementados porém também poderão ser utilizados através da ferramenta gráfica do Orange.

2. Projeto A

2.1. Enquadramento

A empresa MedKnow presta serviços de consulta de oftamologia onde são realizados exames médicos para a avaliação da necessidade de utilização de lentes de contacto, no final dos exames em questão é chegada à conclusão se um paciente terá que usar lentes de contacto rígidas, lentes de contacto moles ou se não precisará de utilizar lentes de contacto. Os pacientes pode ser diagnosticados com hipermetropia, miopia ou astigmatismo sendo impossível o paciente ter hipermetropia e miopia em simultâneo. Ao longo de uma consulta poderão ser realizados diversos testes, sendo um deles obrigatório que consiste em perceber a taxa de lubrificação ocular. Os dados dos respetivos testes são inseridos num sistema de gestão de base de dados onde serão correlacionados com o paciente possibilitanto uma posterior consulta e análise dos mesmos.

2.2. Modelo Entidade Relacionamento

Dado o contexto do problema foi elaborado um modelo Entidade Relacionamento com o objetivo de criar uma modelação dos dados para que posteriormente se possa implementar um sistema de gestão de base de dados com a capacidade de armazenar todos os dados recolhidos da clínica.

O modelo ER desenvolvido está representado na seguinte figura:

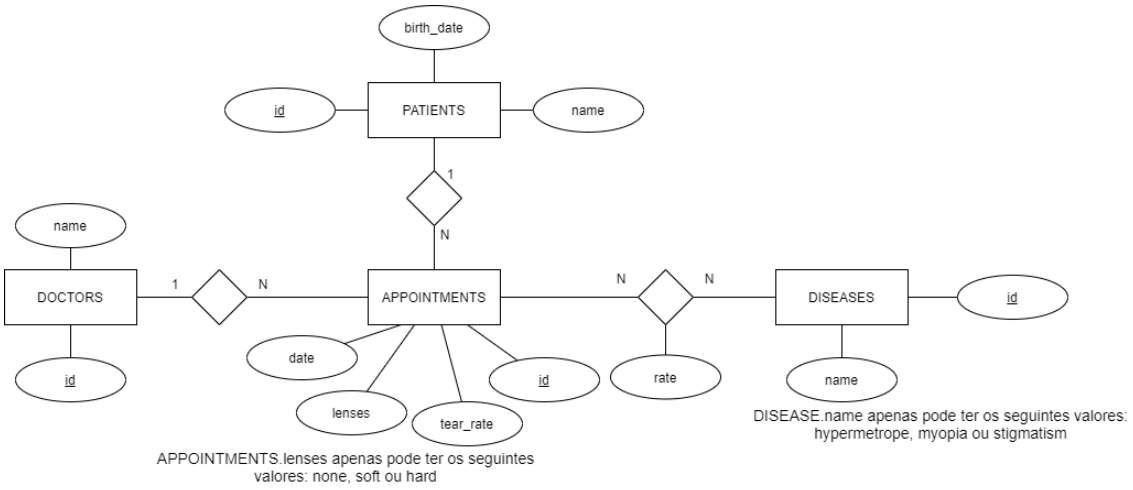

Figura 1 - Modelo Entidade Relacionamento

2.3. Modelo Relacional

A partir do modelo ER desenvolvido anteriormente foi efetuado a transição desse modelo para o modelo relacional de forma a que seja implementado um sistema de gestão de base de dados.

Desta forma a base de dados consiste na relação entre 5 tabelas:

- Patients: Representa os pacientes, nela é armazenado um identificador único, o nome e a data de aniversário.

- Doctors: Representa os doutores da empresa que presta serviço clínico, estes são representados por um identificador único e o nome.

- Diseases: Representa todas as doenças possíveis de diagnosticar, para cada doença existe um identificador único e um nome.

- Appointments: Representa todas as consultas efetuadas na clínica, em cada consulta é guardado o identificador único da consulta, o identificador do doutor, o identificador do paciente, a data da realização da mesma, a taxa de lubrificação ocular do paciente no momento da consulta e a avaliação final do doutor sobre a utilização de lentes de contacto.

- Evaluations: Representa todos os exames feitos numa consulta, à exceção do exame da taxa de lubrificação ocular do paciente, para cada entrada desta tabela são guardados o identificador da doença em questão, o identificador da consulta onde foi realizado o exame, e a taxa de avaliação da doença. Uma certa doença apenas pode ser diagnostica uma vez por consulta e caso numa consulta seja diagnosticada miopia num paciente, nessa mesma consulta não poderá ser diagnosticado hipermetropia.

2.4. Base de dados

De modo a implementar o modelo relacional conceptualizado anteriormente e possibilitar toda a operacionalidade do sistema de gestão de base de dados foram desenvolvidos scripts com o objetivo de automatizar de criação da base de dados, criação do modelo relacional e da respetiva população das tabelas geradas.

2.5. Exportação de dados

Tendo a base de dados populada e presente num estado consistente é agora necessário elaborar *scripts* de modo a automatizar o processo de criação de views para criar o dataset e exportar o mesmo para um ficheiro num determinado formato, para possibilitar que o dataset seja usado. No entanto existe ainda um passo entre a modelação de dados e a utilização dos mesmos pelos algoritmos classificadores, passo este que consiste em pré-processar os dados de modo facilitar a interpretação dos dados. Foi determinado que o pré-processamento dos dados seriam feitos na base de dados.

Desta forma os atributos “age” e “tear_rate”, cujos representam a idade de um paciente e a taxa de lubrificação ocular, respetivamente, ser-lhes-á aplicada uma discretização, ou seja, serão convertidos os seus valores numéricos para valores nominais. O atributo “age” agora tem os seguintes valores possíveis: “young”, “pre-presbyopic” e “presbyopic”, enquanto o atributo “tear_rate” a partir da discretização terá os seguintes valores possíveis: “normal” e “reduced”. Além destes dois atributos foi aplicada uma binarização sobre o atributo “astigmatic”.

2.6. Arquitetura geral da solução

A arquitetura geral da solução foi projetada de modo a desenvolver componentes de software genéricos possibilitando o reaproveitamento de funcionalidades em cada um dos classificadores, aquando necessário. Desta forma foram criados três diretorias na diretoria raiz da solução:

- Classifier: Diretoria onde estarão presentes as classes que respresentam cada um dos classificadores, juntamente com uma classe abstrata responsável pela abstração das classes dos classificadores na avaliação do desempenho destes.
- Data: Diretoria de dados no qual estão presentes módulos e classes responsáveis pela aquisição e preparação dos dados, assim como as funções para adquirir dados estatísticos sobre os dados e ainda as representações do modelo de dados da solução.

- Evaluation: Diretoria onde está presente o módulo que define a classe responsável por efetuar a avaliação de um classificador através do algoritmo *stratified 10-Fold cross validation.*
- Solution: Diretoria onde está presente o módulo no qual define a classe responsável por construir soluções.

É relevante enunciar que na raiz da solução constam quatro scripts, onde três destes são destinados a executar a solução utilizando, cada um, um classificador diferente e o último script que é responsável por realizar a avaliação dos três classificadores a ter consideração neste projeto e demonstra a classificação de cada um ao utilizador, deixando ao mesmo a responsabilidade de analisar as avaliações elaboradas de modo a tirar as suas próprias conclusões sobre os classificadores.

2.7. Classificador One-Rule

O classificador One-Rule baseia-se na classificação de um padrão unicamente tendo em consideração um atributo desse padrão. Para tal é necessário determinar qual o atributo mais apto para construir as regras de classificação. A formulação do atributo mais apto consiste em escolher o atributo que apresente um erro total menor em relação aos restantes atributos.

2.7.1. Implementação

A implementação deste algoritmo resume-se numa classe *OneRule* que extende de Classifier, esta segunda classe representa abstratamente um classificador. A classe OneRule é constituida por dois campos, um que representa uma instância de um objeto de auxilio de dados, isto é, é um objeto que disponibiliza métodos relacionados com dados. O segundo campo representa o atributo mais apto para exercer a classificação deste algoritmo. Esta classe disponibiliza dois métodos privados e dois métodos públicos, sendo que dos privados um tem o objetivo de calcular o erro de um atributo e o outro de selecionar o atributo mais apto, ou seja, de escolher o atributo com menor erro total. Os dois métodos públicos mencionados são os métodos de treino do classificador e de previsão. O método de treino irá determinar o atributo mais apto, determinando também o valor da classe aos valores do atributo. Enquanto o método de previsão, recebendo um padrão como parâmetro irá prever o valor da classe tendo em consideração o valor do atributo que representa a regra de classificação.

2.8. Classificador Naive-Bayes

O classificador Naive-Bayes baseia-se na classificação de um padrão através de probabilidades a-posteriori, isto é, através de uma probabilidade condicionada da hipótese, sabendo as evidências, sendo a hipótese o valor da classe e as evidências os valores dos atributos do padrão em questão. Dado um determinado conjunto de dados, para cada atributo é calculado um conjunto de probabilidades condicionais dos valores do atributo sabendo a classe. Sabendo estes dados estatísticos em antemão (probabilidades a-priori), é possível de elaborar estatísticamente previsões de classes, resultanto na multiplicação de todas as probabilidades a-priori no qual cada respetivo valor de um atributo se inclue, dado um padrão, consequentemente a previsão da classe será a hipótese da respetiva probabilidade com maior valor.

2.8.1. Implementação

Para implementar este algoritmo foi desenvolvida uma classe NaiveBayes que é caracterizada por ser constituida por dois campos, um que representa a probabilidade associado a cada valor de cada atributo do dominio do dataset e o outro representa o domínio da classe. Esta classe é também constituida por dois métodos públicos, sendo o primeiro responsável pelo treino do classificador, isto é, pelo calculo das probabilidades dos valores de cada atributo tendo em conta o valor da classe associado. O segundo método é responsável pela previsão da respetiva classe de um padrão recebido por argumento.

2.9. Classificador Iterative Dichotomiser 3

O classificador Iterative Dichotomiser 3 (ID3) é um algoritmo de classificação que representa uma árvore de decisão, cujos nós representam condições sobre os atributos de um dado padrão e as folhas da mesma representam os valores das classes. O processo de contrução da árvore passa por averiguar qual o atributo que minimiza a incerteza sobre a classe, isto é, o atributo que cuja entropia é menor.

A entropia do valor de um atributo representa a informação esperada, enquanto a média ponderada da entropia do conjunto de valores de um atributo representa a quantidade de informação que se espera ser necessária, logo o atributo que apresente uma média de entropia menor é o que minimiza a incerteza sobre a classe.

Partindo de um nó na raiz é recursivamente determinado os próximos nós da árvore, até que esta esteja completa.

2.9.1. Implementação

Para implementar este algoritmo foi desenvolvida uma classe denominada por ID3, esta é caracterizada por ser constituida por um único campo, este representa a raiz da árvore de decisão a ser contruída no treinamento do algortimo. Esta classe é também constituída por três métodos privados e dois públicos.

Dos métodos privados, o primeiro tem como objetivo averiguar se o conjunto de dados passados por argumento é constituido por um único valor da classe, o segundo método tem como objetivo remover um atributo do dataset, pois após ser formulado qual o atributo mais apto para um determinado ponto da árvore, este tem que ser retirado do dataset para ser possível construir a seguinte camada da árvore. O terceiro e último método privado tem como objetivo construir a árvore de decisão em si através de iterações recursivas deste método, o método em si pode retornar ou uma instância que representa um nó da árvore, cujo nó poderá ter um conjunto de descendesntes, podendos os descendentes ser outros nós ou folhas da árvore, ou pode retornar uma instância que representa uma folha da árvore, isto é, um valor da classe.

Para efeitos de exemplo foi criada uma árvore com o *dataset* lenses utilizados nas aulas, a árvore gerada é representada na seguinte figura:

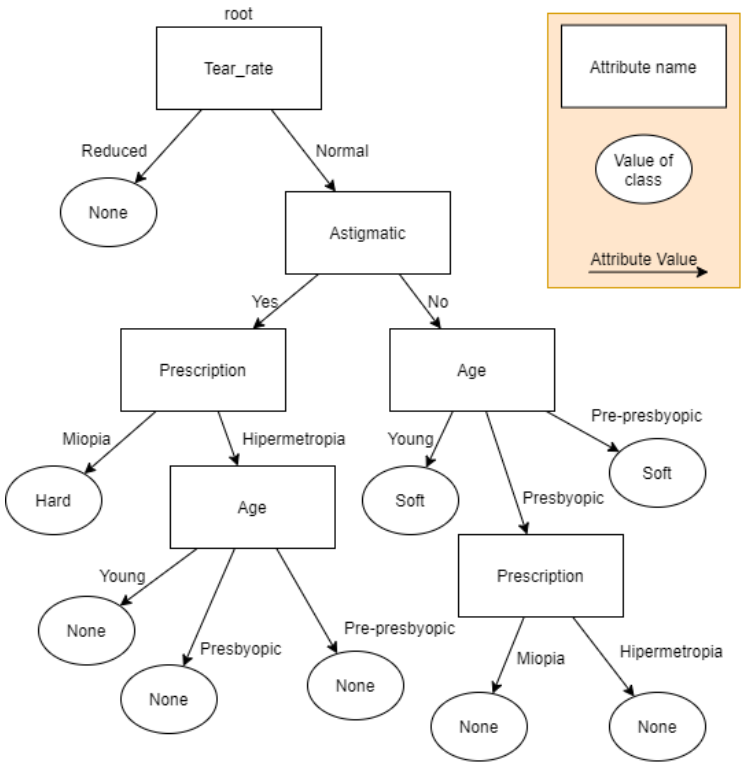

Figura 2 - Exemplo de árvore de decisão

Foi introduzido na figura de exemplo de uma árvore de decisão um figura com o objetivo de facilitar a leitura da mesma, nesta é demonstado que as figuras retângulares representam nós da árvore, sendo o seu conteúdo representado pelo nome do atributo em questão, as figuras ovais representam uma folha da árvore e o seu conteúdo o valor da

classe da classificação e por último as setas que ligam os retângulos às figuras ovais representam as ligações entre as camadas da árvore, no qual o seu valor é o valor do atributo do nó antecessor.

2.10. Avaliação dos classificadores

A avaliação dos classificadores é feita através algoritmo *stratified 10-fold cross-validation*, este algoritmo é caracterizado por a partir de um conjunto de dados, gera dez cópias do mesmo e para cada cópia irá dividir o conjunto em dez partes, sendo que uma parte, isto é, um décimo desse *dataset* será destinado para teste e as outras nove partes, ou nove décimos do dataset, será destinado para treinar o classificador. O algoritmo também garante que para cada uma das cópias do *dataset* o conjunto de teste é sempre diferentes dos conjuntos de teste das outras cópias do dataset. Desta forma para cada cópia do dataset, este será treinado, testado e por fim é avaliado os resultados dos testes e formulada a taxa de sucesso dos mesmos. O resultado final deste algoritmo consiste no cálculo da média aritmética das taxas de sucesso de cada cópia do dataset original.

Ao avaliar todos os classificadores foi notável o facto que o classificador Iterative Dichotomiser 3 sempre obteve a maior média de taxa de sucesso, enquanto o classificador One Rule sempre obteve a pior média de taxa de sucesso entre os três classificadores, como se pode verificar na seguinte figura que demonstra a taxa de sucesso média dos classificadores após os mesmos terem sido avaliados:

```
The mean accuracy of OneRule is: 60.83 %
The mean accuracy of Naive-Bayes is: 68.33 %
The mean accuracy of ID3 is: 71.67 %
```

Figura 3 - Avaliação dos algoritmos

3. Projeto A1

3.1. Enquadramento

Após a implementação e testagem do algoritmo One-Rule,a instituição "FungiData" cordialmente pediu uma análise sobre um dataset com informações sobre a comestibilidade de cogumelos. Após uma breve análise ao dataset foi relevante a dimensão do dataset, constituído por 8416 instâncias sobre um total de vinte e três espécies de cogumelos. Cada cogumelo pode ser classificado como comestível, venenoso ou comestibilidade desconhecido e não recomendada, sendo representado no dataset a classe como uma variável nominal. Cada instância do dataset é representada por vinte e dois atributos, tendo todos valores nominais, porém verificou-se que existem 2480 valores em falta no dataset, curiosamente todos os valores em questão estão em falta no mesmo atributo “stalk-root”. No total existem 4488 instâncias classificadas como comestível e outras 3928 instâncias classificadas como venonoso.

3.2. Preparação dos dados

Tendo o dataset completo à disposição apenas é necessário converter o mesmo para um formato compatível com a biblioteca do Orange para que o dataset possa ser aplicado no algoritmo de classificação One-Rule, para tal foi criado um workflow no ambiente gráfico de desenvolvimento do Orange de modo a que carregasse o dataset, convertesse e guardasse o mesmo num ficheiro com a extensão .tab.

3.3. Aplicação do algoritmo One-Rule

Foi aplicada a implementação do algoritmo One-Rule descrita no ponto 2.7.1 no dataset em causa de modo a ter uma forte medida de desempenho deste algoritmo, para tal efeito, foi criado um *script* que treina o algoritmo com o dataset e posteriormente é testado com o mesmo, ou seja, com o dataset na sua totalidade.

Foi denotado que o atributo do dataset com menor erro associado é o atributo “odor” e unicamente se verificou falhas na previsão da classe partindo deste atributo quando o mesmo tem o valor “NONE”, concretamente foram feitas 120 previsões erradas num total de 3808 testes que abrangem o valor do atributo regra descrito.

Na seguinte figura é possível verificar o conteúdo do ficheiro oneR_OUTPUT.txt que tem como objetivo demonstrar a avaliação feita ao algoritmo implementado com o dataset em causa:

```
( odor, ALMOND, EDIBLE ) : (0, 400)
( odor, ANISE, EDIBLE ) : (0, 400)
( odor, CREOSOTE, POISONOUS ) : (0, 192)
( odor, FISHY, POISONOUS ) : (0, 576)
( odor, FOUL, POISONOUS ) : (0, 2160)
( odor, MUSTY, POISONOUS ) : (0, 48)
( odor, NONE, EDIBLE ) : (120, 3808)
( odor, PUNGENT, POISONOUS ) : (0, 256)
( odor, SPICY, POISONOUS ) : (0, 576)
```

Figura 4 - Conteúdo do ficheiro oneR_OUTPUT.txt

3.4. Árvores de decisão

Recorrendo à ferramenta gráfica do Orange foi criado um *workflow* capaz de avaliar dois tipos de árvores de decisão através do dataset em questão deste projeto, o primeiro tipo é umas simples àrvore de decisão, já o segundo tipo é uma floresta aleatória, isto é, dado um dataset, este *widget* vai criar sub-datasets e construir àrvores de decisão a partir desses sub-datasets. Tendo estes dois tipos de modelos configurados para terem no máximo 5 níveis de profundidade o primeiro tipo foi capaz de construir uma árvore capaz de mapear o dataset na sua totalidade, ou seja, obter 100% de taxa de sucesso na sua posterior testagem. Já o segundo tipo, criando 5 árvores com 5 datasets provenientes do dataset original, obteve uma taxa de sucesso de 97.9%.

4. Conclusão

A realização deste conjunto de projetos proporcionou um contacto mais prático com os temas e conceitos introduzidos, permitindo uma visão mais abrangente e clara sobre os processos e mecanismos de mineração de dados envolventes. Foi também possível avaliar e comparar o desempenho dos algoritmos classificadores implementados.