

*Aprendizagem e Mineração de Dados*

Projeto Final – A

Semestre de inverno 2023/2024 - MI1D

Docente

Eng.º Paulo Trigo

Grupo 04   
Gonçalo Silva – 47255   
João Rocha – 47196

Luís Morgado - 51358

Data: 1/10/2023

Índice

Conteúdo

[Introdução 2](#_Toc150202380)

[Enquadramento 2](#_Toc150202381)

[Análise dos dados 3](#_Toc150202382)

[Modelo de dados 3](#_Toc150202383)

[Modelo Entidade-Relação 4](#_Toc150202385)

[Modelo Lógico 5](#_Toc150202387)

[Detalhes de Implementação Base de dados 6](#_Toc150202388)

[Discretização 6](#_Toc150202389)

[1R – One Rule 7](#_Toc150202390)

[Implementação 1R – One Rule 8](#_Toc150202391)

[Arvore de decisão – ID3 11](#_Toc150202392)

[Naive bayes - NB 11](#_Toc150202393)

[Implementação ID3 e NB 11](#_Toc150202394)

## Introdução

Com este trabalho, pretende-se explorar e compreender inter-relações entre dados, informação e conhecimento, aplicando esses conceitos em cenários práticos e explorando diversas técnicas e métodos que permitam converter dados em conhecimento, de modo a desenvolver a competência de analisar, modelar e validar um projeto de mineração de dados.

## Enquadramento

#### O centro médico especializado em oftalmologia, MedKnow, utiliza um sistema de gestão de bases de dados (SGBD) que armazena todos os dados reunidos ao longo do tempo referentes a cada consulta. A MedKnow forneceu o arquivo "d01\_lenses.xls", contendo um conjunto de dados específico relacionado à atividade de prescrição de lentes nas últimas duas semanas (conforme apresentado na Tabela 1), algumas premissas e padrões conseguem ser identificados, fornecendo informação valiosa para a identificação de tendências na atividade diária da clínica oftalmológica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **age** | **prescription** | **astigmatic** | **tear\_rate** | **lenses** |
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| young | hypermetrope | yes | reduced | none |
| young | hypermetrope | no | normal | soft |
| young | hypermetrope | no | reduced | none |
| presbyopic | myope | yes | reduced | none |
| presbyopic | myope | yes | normal | hard |
| presbyopic | hypermetrope | yes | reduced | none |
| presbyopic | hypermetrope | yes | normal | none |
| presbyopic | hypermetrope | no | normal | soft |
| presbyopic | hypermetrope | no | reduced | none |
| pre-presbyopic | myope | yes | reduced | none |
| pre-presbyopic | myope | yes | normal | hard |
| pre-presbyopic | myope | no | normal | soft |
| pre-presbyopic | hypermetrope | yes | normal | none |
| pre-presbyopic | hypermetrope | no | normal | soft |

*Tabela 1 - snippet de dados relacionado com a atividade de prescrição de lentes*

Com base nos dados da Tabela 1, podemos afirmar que a *Medknow* categoriza os seus clientes com base na sua idade (jovem, pré-presbiópico, presbiópico), tipos de prescrição (miopia e hipermetropia), presença de astigmatismo (sim/não), taxa de lágrima (normal/reduzida) e para cada um deles, podem ser prescritos 3 tipos diferentes de lentes (nenhuma, moles, rígidas).

## Análise dos dados

#### Depois de uma análise completa deveremos ser capazes de saber que tipo de lentes devem ser escolhidas para cada paciente dependendo da sua idade e dos seus aspetos de saúde ocular.

#### Podemos observar que os dados da Tabela 1, são estruturados num formato tabular, com valores nominais em domínio discreto, quanto à sua semântica, o dataset apresenta 5 valores, que podem ser divididos em 4 atributos, sendo estes a sua idade (jovem, pré-presbiópico, presbiópico), tipos de prescrição (miopia e hipermetropia), presença de astigmatismo (sim/não), taxa de lágrima (normal/reduzida) e 1 classe/alvo que são as lentes, que podem ser representadas por 3 tipos diferentes de lentes (nenhuma, moles, rígidas).

## Modelo de dados

## Foi construído um modelo concetual de dados utilizando a notação Entidade-Relação (Entity-Relationship notation) para o apoio operacional (trabalho diário) e um modelo lógico derivado do mesmo. Tais modelos são cruciais para a organização dos dados e futura análise.

## Modelo Entidade-Relação

## 

*Figura 1 – Modelo Entidade-Relação*

## Modelo Lógico

#### Uma imagem com texto, captura de ecrã, Tipo de letra, design Descrição gerada automaticamente

*Figura 2 – Modelo Lógico*

## Detalhes de Implementação Base de dados

**Modelo de Dados e Associações entre Tabelas**

O modelo de dados é centrado na tabela "Appointment", que serve como o núcleo das informações clínicas. Essa tabela está relacionada com todas outras tabelas do sistema. Para associar as informações sobre as doenças a uma consulta, foi criada a tabela "DIAGNOSIS." Nessa associação, reconhecemos que uma única consulta pode estar relacionada a mais de uma doença. No entanto, no contexto clínico, esperamos que haja exatamente uma doença do tipo "myope" ou "hypermetrope" associada a cada consulta. Além disso, opcionalmente, pode estar associada a uma doença do tipo "astigmatic."

**Atributos Chave e Associações Diretas:**

O valor de "TearRate," assume-se que é atribuído um único valor à consulta, refletindo uma medição específica, estando este valor diretamente descrito na tabela “APPOINTMENT”. Da mesma forma, "LensHardness" também é atribuído como um único resultado, no entanto, esses resultados foram armazenados numa tabela separada, uma vez que representam os resultados específicos obtidos durante cada consulta.

## Discretização

Com base no conjunto de dadosapresentado na *tabela 1,* podemos concluir que todos os valores dos atributos são valores nominais, com um intervalo de valores bem definido.

Para chegar a estes valores, é necessário um processo de discretização, pois normalmente dados como a idade e a taxa de lágrima costumam ser valores numéricos.

Como mencionado anteriormente, precisamos converter todos os dados numéricos em valores nominais e podemos fazer isso atribuindo nomes a determinados intervalos de valores.

**Intervalos de Valores e Mapeamento:**

Para caracterizar os intervalos de valores possíveis para as medidas de idade, TearRate e LensHardness, foram criadas três tabelas de mapeamento. Estas tabelas servem como referências para definir faixas de valores aceitáveis. Cada intervalo é mapeado por meio das tabelas "OcularAge," "TearRate," e "LensHardness," proporcionando uma maneira padronizada de representar e consultar os resultados clínicos dentro dessas faixas.

Essas associações e representações foram projetadas para garantir uma estrutura de dados coerente e eficaz. O uso de tabelas de mapeamento ajuda a manter a consistência e a integridade dos dados, fornecendo uma estrutura clara para as informações clínicas em questão. Os valores foram organizados de acordo com a seguinte tabela:

|  |  |  |
| --- | --- | --- |
| **mínimo** | **máximo** | **valor** |
| Ocular Age | | |
| 0 | 30 | “young” |
| 30 | 45 | “pre-presbyopic” |
| 45 | 99 | “presbyopic” |
| Lenses Hardness | | |
| 0 | 0.1 | “none” |
| 0.1 | 0.3 | “soft” |
| 0.3 | 0.99 | “hard” |
| Tear Rate | | |
| 0 | 0.5 | “reduced” |
| 0.5 | 1.0 | “normal” |

*Tabela 2 - Discretização dos dados*

Este processo é especialmente importante neste projeto, pois o Classificador One Rule, só pode trabalhar com valores discretos.

## 1R – One Rule

O classificador "One Rule" tem como ideia principal encontrar a regra que melhor discrimina as classes alvo com base em um único atributo.

De forma a descobrir qual o atributo que melhor classifica os dados, é construída uma regra que relaciona cada um dos valores desse atributo com um dos valores do conceito (classe).

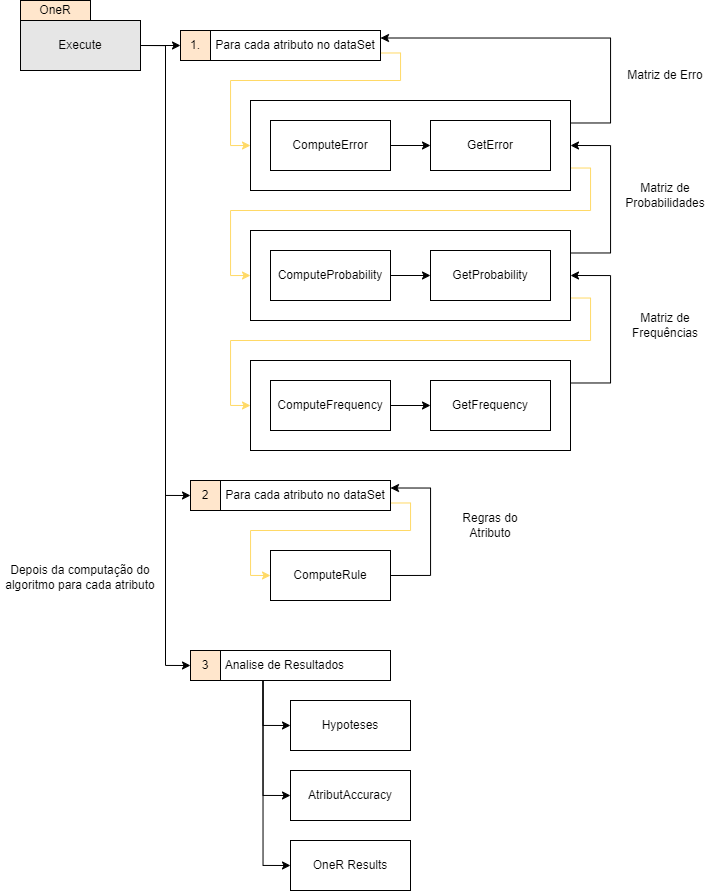
Processo para descobrir a “One Rule”:

1. Escolher um atributo para análise (o processo deve ser repetido para cada atributo).
2. Calcular a frequência de cada valor do atributo em relação às classes alvo.
3. Calcular o erro para cada valor do atributo.
4. Escolher os pares (valor-atributo, valor-classe) com menor erro, aleatório se igual.
5. Calcular o erro do atributo somando os erros dos pares escolhidos.
6. Depois de obter as regras e os erros de cada atributo (itens acima), é escolhido o atributo com o menor erro e considera-se as regras associadas a esse atributo como representativas do conjunto de dados.

## Implementação 1R – One Rule

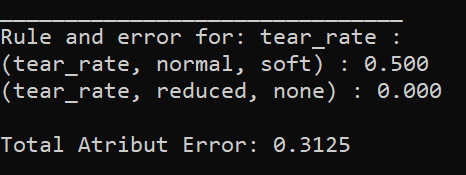
Para a implementação do algoritmo 1R foram desenvolvidas 2 implementações, uma primeira, denominada OneR, baseada no código fornecido em aulas laboratoriais, tendo sido utilizadas e adaptadas às respectivas funções necessárias nomeadamente as funções de geração de matrizes de frequência, probabilidades e erros, entre outras funções. e uma outra implementação, denominada NativeOneR, desenvolvida sem utilização prévia de código fornecido

O seguinte diagrama de fluxo ilustra a execução e processamento interno do modelo OneR tendo em conta a implementação da classe OneR:



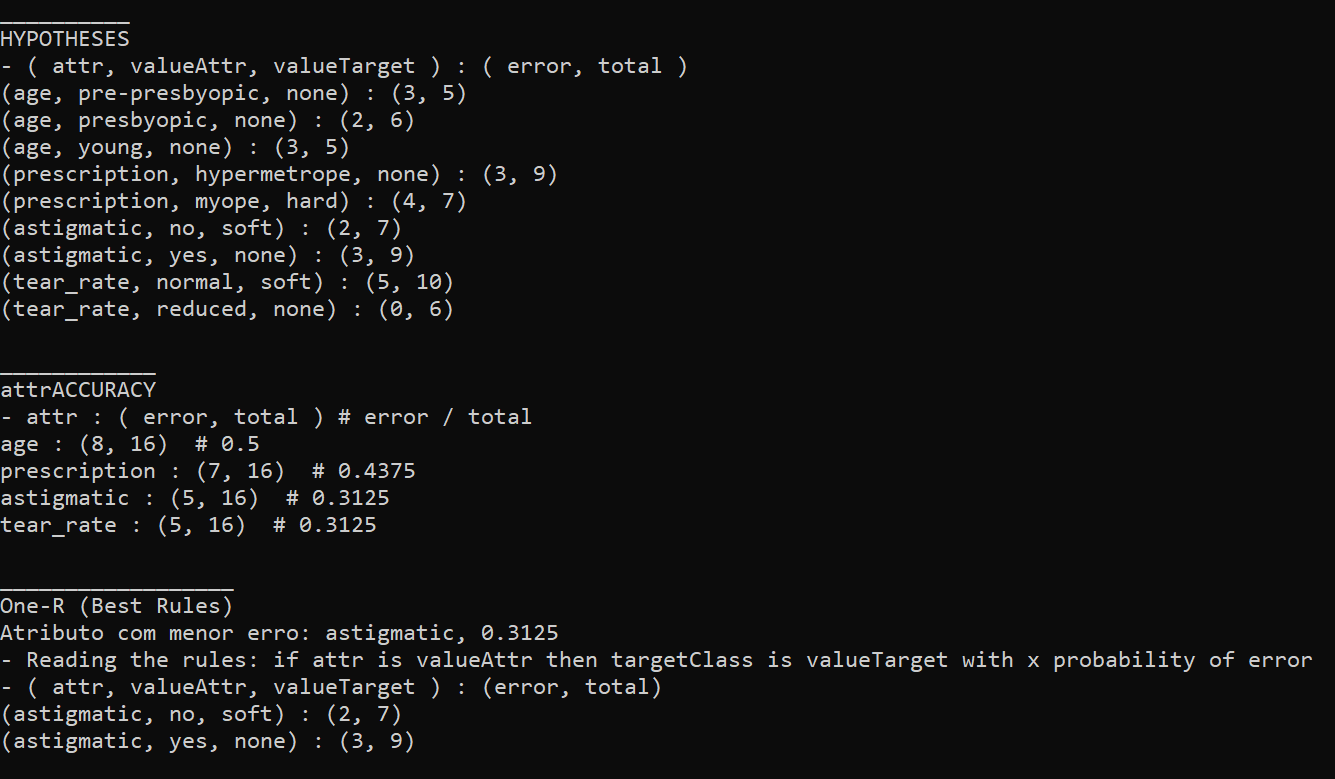
*Figura 3 - Discretização dos dados*

O algoritmo começa por carregar um ficheiro de input para o dataset sobre o qual vai ser executado o algoritmo. Inicialmente são feitas as computações necessárias para obter as regras/erro associadas a cada atributo, gerando por exemplo o seguinte resultado para o atributo tear\_rate:



*Figura 4 - Regras do atributo tear\_rate*

À medida que são geradas regras para os atributos, estas são armazenadas num dicionário. No fim da execução da geração de regras para cada atributo é amostrado um conjunto de resultados “Hypoteses” que representam todas as regras e respectivas % de erro geradas para cada atributo, posteriormente é mostrado o conjunto de resultados AtrrAccuracy que representa a soma de %erro para cada atributo e por fim é escolhida a regra de menor erro que é a que melhor representa o dataSet

*Figura 5 - Modelo 1R: Resultados finais*

As funções denominadas como “get\_” são responsáveis por efetivamente gerar a matriz respetiva pretendida enquanto que as funções denominadas como “compute\_” são responsáveis por efetivamente chamar as funções get e realizar a amostragem da matriz resultante a chamada da função “get”.

## Arvore de decisão – ID3

As árvores de decisão são uma técnica de aprendizagem que permite a classificação de instâncias em categorias tendo como base um conjunto de atributos observados.

O algoritmo ID3 (*inductive decision tree*) é um dos algoritmos mais utilizados na construção de árvores de decisão.

O processo de construção da árvore de decisão com o algoritmo ID3 pode ser resumido nas seguintes etapas:

1. Inicialmente, o algoritmo avalia todos os atributos disponíveis para determinar qual deles resulta no maior ganho de informação, e escolhe esse atributo para ser o nó raiz da árvore de decisão.
2. É criado um nó filho para cada valor do atributo escolhido na etapa anterior
3. Para cada nó filho criado, o algoritmo ID3 repete o processo, avaliando quais atributos são mais informativos para a classificação daquele subconjunto específico.
4. A condição de paragem ocorre quando todos os exemplos em um subconjunto têm o mesmo valor para um atributo, tornando o subconjunto homogêneo e não exigindo mais divisões.

## Naive bayes - NB

O classificador Naïve Bayes, fundamentado no Teorema de Bayes, é um algoritmo de aprendizagem que trata todos os atributos como igualmente importantes durante a fase de treinamento, ao contrário do classificador 1R, que se baseia em apenas um atributo.

Este classificador presume que os atributos são independentes entre si, o que pode não ser totalmente verdade na prática. No entanto, essa simplificação permite um treinamento mais eficiente e é particularmente útil para lidar com problemas de classificação de texto e mineração de dados.

## Implementação ID3 e NB

Ao contrário do modelo 1R, a implementação dos classificadores ID3 e NB, não tiveram que ser implementados de raiz, então para tal foi utilizada a biblioteca *sklearn* para a implementação de ambos os modelos.

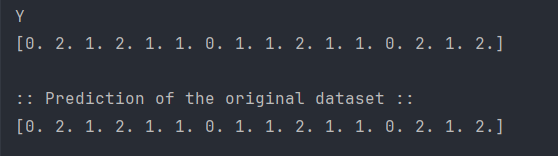
Foram criadas 2 classes, *‘ModelID3’* e *‘ModelNaiveBayes’,*  onde cada uma contém a implementação do algoritmo correspondente.

Cada uma destas classes tem como base a implementação do método **fit()**, e **predict()**.

O método **fit()** é usado para treinar o modelo de Naive Bayes com o dataset de treino.

Depois de ser realizado o treino, o modelo está pronto para fazer previsões, para tal é usado o método **predict()** que faz previsões com base no modelo treinado, recebendo um conjunto de dados *X* como entrada e usa o modelo de Naive Bayes para prever as classes correspondentes.

Nesta fase da implementação treinou-se o modelo com o dataset original, e fez-se as previsões também para o dataset original, esperando resultados iguais verificando assim o bom funcionamento do modelo.

Posteriormente serão criados datasets de treino e de teste de modo a avaliar corretamente o desempenho do modelo.

*Figura 6 – ID3 and NB results*

Implementa do ID3 e do Naïve Bayes no Orange

Uma imagem com texto, diagrama, file, captura de ecrã

Descrição gerada automaticamente

# Desempenho dos modelos:

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente

Conclusão

Em resumo, embora o conjunto de dados utilizado neste trabalho seja relativamente pequeno, o modelo One-R demonstrou ser uma abordagem simples para realizar classificações com base em um único atributo, permitindo a interpretação das regras de classificação. No entanto, ao avaliar o desempenho global dos modelos, observamos que outras técnicas mais complexas, como árvores de decisão (ID3) e o classificador Naïve Bayes, superaram o One-R em termos de precisão e capacidade de generalização.