

**Aplicação de Técnicas de Data Mining em
Extracção de Elementos de Documentos Comerciais**

por

Ana Cristina da Silva Anacleto

**Tese de Mestrado em
Análise de Dados e Sistemas de Apoio à Decisão**

Orientada por

Professor Doutor Alípio Jorge

Professor Doutor Carlos Soares

Faculdade de Economia

Universidade do Porto

2009

“Nunca sabemos do que somos capazes até o tentarmos fazê-lo.”

Charles Dickens

Nota Biográfica

Nascida em Agosto de 1971, Ana Cristina da Silva Anacleto é a última de três irmãos. Começou desde cedo o seu interesse pela área das ciências, em particular pela matemática, incentivada pelo seu irmão mais velho. Foi esse interesse que a levou a optar pelo curso de Matemática Aplicada – Ramo de Ciências de Computadores na Faculdade de Ciência da Universidade do Porto no ano de 1989. Após estágio no INESC - Instituto de Engenharia de Sistemas e Computadores, iniciou a sua carreira em 1993, como programadora no departamento de informática da Bolsa de Valores do Porto (actual Euronext – Lisboa). Foi no exercício da sua actividade profissional, na área dos sistemas de informação, que encontrou a motivação pela análise de dados, motivo pelo qual a levou a optar por este mestrado.

Actualmente e desde 2006 assume a função de directora do departamento de Sistemas de Informação do Hospital da Arrábida – Gaia SA, uma unidade do Grupo Espírito Santo Saúde, SGPS, SA, onde espera, entre outras tarefas, ter a oportunidade de aplicar na prática os conhecimentos obtidos ao longo deste mestrado.

Agradecimentos

Quero agradecer aos meus orientadores, Professor Doutor Alípio Jorge e Professor Doutor Carlos Soares, por todo o apoio, disponibilidade e sábias orientações. Agradeço, também, o constante incentivo e confiança que demonstraram ao longo deste trabalho.

Agradeço a todos os amigos que, de alguma forma, me ajudaram e me incentivaram a não desistir deste projecto.

Ao meu marido um agradecimento especial pelo apoio e paciência que sempre demonstrou.

Aos meus filhos um carinho muito especial. Desculpem o tempo perdido, agora já posso brincar mais convosco.

Resumo

Manter a contabilidade organizada e actualizada nas empresas envolve meios humanos e materiais reflectidos em elevados custos administrativos para as empresas. Sabemos que, ao nível da gestão, grande parte das decisões são tomadas com base nos dados da actividade das empresas e que é reflectida na troca de documentos comerciais. Idealmente, estes dados devem ser tratados e analisados rapidamente para cumprimento de exigências legais, tais como pagamento a fornecedores, entrega de declarações às finanças, por questões de logística e, também, para permitir uma rápida e correcta tomada de decisões e definição ou ajuste de estratégias. Muitos documentos são ainda enviados em papel, pelo que se torna necessário inseri-los manualmente no Sistema de Informação.

O recurso a técnicas de extracção de informação para simplificar estes processos administrativos é ser uma opção a considerar. Os métodos de Data Mining permitem a descoberta de informação não acessível através dos tradicionais métodos estatísticos de análise de dados. Neste trabalho propomos aplicar técnicas de Data Mining para extracção de informação de documentos comerciais. Em particular, abordamos o problema de identificar o número de documento em facturas.

Índice

1.	Introdução	1
2.	Data Mining	3
2.1.	Definição de Data Mining	3
2.2.	Tarefas de Data Mining	8
2.2.1.	Métodos de Classificação	9
2.3.	Medidas de Avaliação	14
2.4.	Oracle Data Mining e Oracle Data Miner	19
2.4.1.	Métodos de Classificação no Oracle Data Miner	24
3.	Extracção de Informação	28
3.1.	Extracção de Informação	28
3.2.	Extracção de Informação de Documentos	30
3.2.1.	Extracção via Dupla Classificação	31
3.2.2.	Extracção de Títulos	33
3.2.3.	Extracção de Citações	35
3.2.4.	Extracção de Artigos de Facturas	38
3.3.	Extracção de informação – World Wide Web	39
3.4.	Adaptabilidade da Extracção de informação	44
3.5.	Caso Particular de Documentos em Papel	45
3.5.1.	OCR – Optical Character Recognition	46
3.5.2.	XML – Extensible Markup Language	50
4.	Extracção de Elementos de Facturas: Caso de estudo	51
4.1.	Definição do Problema	54
4.2.	Digitalização dos Documentos	55
4.3.	Pré-processamento de Dados	57
4.3.1.	Extracção do Número do Documento	62
4.3.2.	Construção do Conjunto de Dados	63
4.4.	Uniformização de Dados	67
5.	Modelos para Extracção do “Número de Factura”	69
5.1.	Conjuntos de Dados	69
5.1.1.	Metodologia Experimental	70
5.1.2.	Definição dos custos	72
5.1.3.	<i>Build Model by Oracle Data Miner</i>	73
5.2.	Resultados dos testes aos modelos MDO	83
5.2.1.	Comparação dos Modelos	89
5.2.2.	Comparação com os Modelos Dados Uniformizados	92
5.3.	Aplicação a novos dados	94
6.	Conclusão	96
6.1.	Contribuições deste trabalho	97
6.2.	Limitações	97
6.3.	Trabalho Futuro	98
	Bibliografia	100
	Anexos	103
	Scripts	103

Índice de Figuras

Fig. 1 – Fases do Processo KDD (Knowledge Discovery in Data)	4
Fig. 2 – Fases do Processo Data Mining descritas em CRISP-DM	5
Fig. 3 – Exemplo duma árvore de decisão	10
Fig. 4 – Matriz de Confusão	15
Fig. 5 – Transformação de dados no Oracle Data Miner	22
Fig. 6 – <i>Wizard</i> para construção de modelos	22
Fig. 7 – Várias apresentações dos resultados no Oracle Data Miner	23
Fig. 8 – <i>Viewer</i> para análise Curvas ROC no Oracle Data Miner	27
Fig. 9 – Representação da transformação da citação numa sequência de proteínas	36
Fig. 10 – Fases da aprendizagem activa.....	40
Fig. 11 – Arquitectura do sistema proposto	43
Fig. 12 – Etapas da tecnologia base do OCR.....	47
Fig. 13 – Caracter digitalizado vs candidatos a integrar o texto	47
Fig. 14 – Contrastes entre cores de fundo e texto	48
Fig. 15 – Exemplo de uma factura e sua correspondente segmentação em regiões	53
Fig. 16 – Exemplo de uma factura e sua conversão em formato ASCII.....	56
Fig. 17 – Outro exemplo de uma factura e sua conversão em formato ASCII	57
Fig. 18 – Ficheiro marcado com os <i>tags</i> do XML	58
Fig. 19 – Estrutura em árvore das <i>tags</i> XML.....	59
Fig. 20 – Ficheiro com marcação intercalada dos <i>tags</i> do XML	60
Fig. 21 – Cabeçalho de facturas	62
Fig. 22 – Algoritmo para recolha das palavras e <i>strings</i> vizinhas	64
Fig. 23 – Exemplos dos dados em análise e da separação em classes	70
Fig. 24 – Dados em análise recolhidos dos documentos	71
Fig. 25 – Opções construção do modelo de Árvores de Decisão.....	74
Fig. 26 – Árvore de decisão gerada no Oracle Data Miner	75
Fig. 27 – Regra principal gerada pela modelo MDO de Árvores de Decisão	75
Fig. 28 – Regra alternativa gerada pelas Árvores de Decisão	76
Fig. 29 – Regras geradas pelas Árvores de Decisão aplicado aos modelos MDO	77
Fig. 30 – Probabilidades geradas pelo modelo de <i>Naive Bayes</i>	78
Fig. 31 – Coeficientes gerados pelo modelo MDO com SVMs	82
Fig. 32 – Resultados dos testes ao modelo de Árvores de Decisão	83

Índice de Tabelas

Tabela 1 – Modelos disponíveis no Oracle Data Mining	20
Tabela 2 – Resultados do método aplicado às entidades individualmente	32
Tabela 3 – Resultados dos diferentes métodos à extracção de títulos	34
Tabela 4 – Resultados experimentais da extracção de citações	36
Tabela 5 – Comparação dos resultados obtidos pelo BibPro e o Infomap	37
Tabela 6 – Resultados experimentais do método	38
Tabela 7 – Resumo da análise comparativa	41
Tabela 8 – Resultados dos métodos aplicados a anúncios de seminários	42
Tabela 9 – Exemplos de conversão do OCR	65
Tabela 10 – Descrição e tipo dos atributos do conjunto de dados	66
Tabela 11 – Exemplos de palavras e diferentes representações	67
Tabela 12 – Variação dos resultados em função do <i>Threshold</i> nas Árvores de Decisão	84
Tabela 13 – Variação dos resultados em função do <i>Threshold</i> no <i>Naive Bayes</i>	86
Tabela 14 – Variação dos resultados em função do <i>Threshold</i> nas ABN	87
Tabela 15 – Variação dos resultados em função do <i>Threshold</i> nas SVMs	88
Tabela 16 – Comparação dos indicadores nos testes aos 4 modelos MDO	89
Tabela 17 – Comparação dos Modelos MDO e MDU	92
Tabela 18 – Matriz de confusão da aplicação do modelo MDO aos novos dados	94
Tabela 19 – Comparação dos resultados de teste e da aplicação aos novos dados	95

Índice de Gráficos

Gráfico 1 – Separação das classes através da maximização das margens	13
Gráfico 2 – Comparação de Curvas ROC	17
Gráfico 3 – Distribuição de formatos de ficheiros na Internet e na intranet	33
Gráfico 4 – Resultados experimentais da extracção de citações	36
Gráfico 5 – Perspectiva gráfica da análise comparativa	41
Gráfico 6 – Comparação <i>tags</i> por ficheiro e total de <i>tags</i>	60
Gráfico 7 – Percentagem de <i>Tags</i> no total dos ficheiros	61
Gráfico 8 – Comparação análise Curvas ROC dos Modelos MDO	90
Gráfico 9 – Comparação de custos e ganhos dos Modelos MDO e MDU	93

1. Introdução

Nos dias de hoje, as empresas armazenam e tratam grandes quantidades de informação, quer em documentos em papel quer em formatos digitais sob a forma de ficheiros ou em base de dados. A facturação é parte fundamental desta informação global. A actividade das empresas, reflectida em produtos e serviços, é oficializada pela troca de documentos, designados por documentos comerciais. A maioria destes documentos tem suporte físico em papel. Extrair informação relevante destes documentos é um trabalho minucioso, penoso e, em muitos casos, falível. Por um lado, a informação contida em alguns desses documentos pode envolver vários graus de subjectividade e por outro esta tarefa requer muito tempo e meios humanos para alimentar as bases de dados com a informação considerada importante.

Neste trabalho, estudamos a extracção de elementos de documentos comerciais, cujo suporte físico é papel, através da aplicação de técnicas de Data Mining. Existem muitos elementos importantes em documentos comerciais. Estes elementos são de natureza muito diversificada (p. ex. números, datas, nomes, moradas, valores monetários). Seria preciso, em princípio, desenvolver abordagens diferentes para cada um deles. Assim, optámos por focar o trabalho num único elemento, o número do documento.

Um dos objectivos do trabalho é analisar a viabilidade de ferramentas comerciais de Data Mining e processamento de dados para este problema em particular. Assim, optámos por usar o Oracle Data Miner, uma interface gráfica sobre o Oracle Data Mining, a ferramenta de Data Mining do Sistema de Gestão de Bases de Dados, o Oracle Database 11g Enterprise Edition. Para complementar as funcionalidades do Oracle Data Miner recorreremos aos *queries* em SQL, aos procedimentos em PL/SQL e funções do MS Excel.

Os documentos comerciais usados neste estudo são facturas de uma empresa com actividade no ramo da indústria de calçado. Através da digitalização dos documentos e aplicação de um software de OCR, foi possível a recolha dos termos que os compõem. Com a etiquetação destes termos, através de técnicas de XML, foi construída uma base de dados com palavras, candidatas a número de documento, e com os termos que as antecedem e sucedem.

A aplicação de técnicas de Data Mining, para classificação destas palavras, permitiu concluir que é possível reduzir, significativamente, o esforço necessário em identificar os números de documento.

Esta dissertação está organizada da seguinte forma: no capítulo 2 definimos Data Mining e apresentamos a ferramenta usada neste estudo. No capítulo 3 fazemos referência a artigos sobre trabalhos na área de extracção de informação e que serviram de apoio a este trabalho. No capítulo 4 é abordada a preparação de dados. No capítulo 5 é explicada a construção dos modelos sobre os dados e apresentados os resultados. As conclusões deste trabalho são apresentadas no capítulo 6.

2. Data Mining

2.1. *Definição de Data Mining*

Um problema com que se deparam muitas empresa nos dias de hoje: demasiados dados e pouca informação.

Com a utilização generalizada de bases de dados e o crescimento substancial da sua dimensão, em número de casos e em número de campos, as organizações deparam-se com o problema em usar, convenientemente, esses dados (Bradley, et al., 1999). Tradicionalmente, o “uso” destes dados está limitado às usuais técnicas de pesquisa e à criação de relatórios que resumem a informação histórica armazenada e na análise e interpretação manual dos dados (Fayyad, et al., 1996). Este modo de interacção é satisfatório para certos processos bem definidos, no entanto não está projectado para suportar exploração de dados e aplicações de suporte à decisão.

Resultado da necessidade urgente de uma nova geração de ferramentas para extracção de informação útil (conhecimento) das suas bases de dados, surgem os processos de descoberta de conhecimento nos dados (Knowledge Discovery in Data – KDD). A descoberta de conhecimento nos dados é definida (Fayyad, et al., 1996a) como o processo de identificar nos dados uma estrutura válida, nova, compreensível e potencialmente útil. Aqui entende-se “dados” como um conjunto de factos (os casos na base de dados) e estrutura refere-se a padrões ou modelos.

O processo de descoberta de conhecimento nos dados é iterativo e interactivo e é uma sequência de passos, desde da preparação dos dados, procura de padrões, avaliação de conhecimento e refinamento de todo o processo. Na Fig. 1 temos, esquematicamente, as fase do processo KDD:

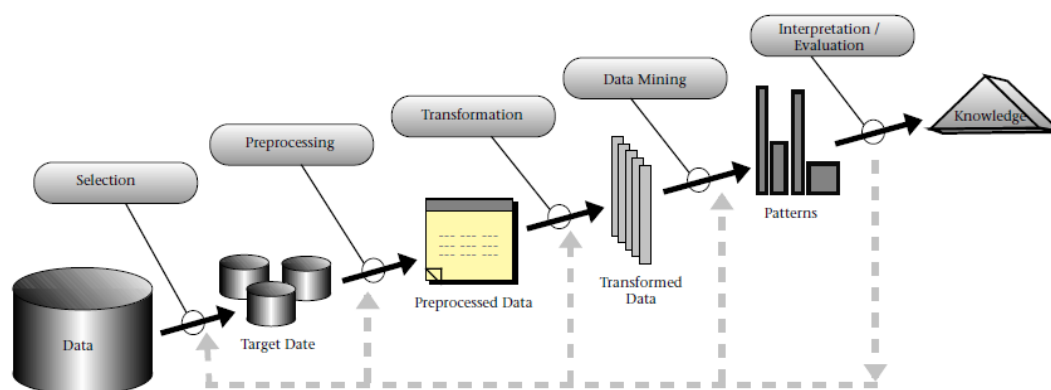


Fig. 1 – Fases do Processo KDD (Knowledge Discovery in Data)

O termo “Data Mining” é usado, frequentemente, como sinónimo para o processo de extracção de informação útil de bases de dados (Fayyad, 1996). No entanto, a descoberta de conhecimento nos dados refere-se a todo o processo de descoberta de conhecimento útil nos dados enquanto Data Mining refere-se a um passo particular neste processo. Data Mining é definido (Fayyad, et al., 1996a) como um passo no processo KDD que, sob aceitáveis limitações de eficiência computacional, identifica estruturas (padrões ou modelos) nos dados que foram preparados em passos anteriores. As estruturas obtidas serão avaliadas em passos subsequentes.

O Data Mining não elimina a necessidade de conhecimento dos dados, dos métodos analíticos e da área de negócio a que se aplicam. O Data Mining descobre nova informação nos dados, mas não revela qual o valor dessa informação e de que forma se pode aplicar para rentabilização de um negócio. Sem uma orientação, as soluções adequadas para resolver o problema de negócio não são encontradas automaticamente, e estas podem ser muito distintas dependendo da forma como o problema é formulado. A correcta definição do problema é fundamental nesta abordagem. Por exemplo, mais importante do que perceber “como evoluíram as vendas de um produto” é, antes de tudo, tentar encontrar características comuns nas pessoas que, no passado, compraram esse produto.

Percebemos que a definição do problema é uma parte muito importante do processo de Data Mining. É a primeira fase deste processo. O processo Data Mining é dinâmico e composto por seis fases (CRISP-DM, 2000). Na Fig. 2 temos, esquematicamente, o

ciclo de vida do projecto de Data Mining. Este ciclo contém as fases do projecto de Data Mining, as tarefas respectivas e as relações entre as tarefas. O fluxo descrito mostra que as diversas fases não têm uma sequência rígida e os processos não param obrigatoriamente quando uma primeira solução é encontrada. O processo de Data Mining é interactivo. Dos resultados podem surgir novas questões que fazem despoletar todo o processo, através da redefinição do problema inicial.

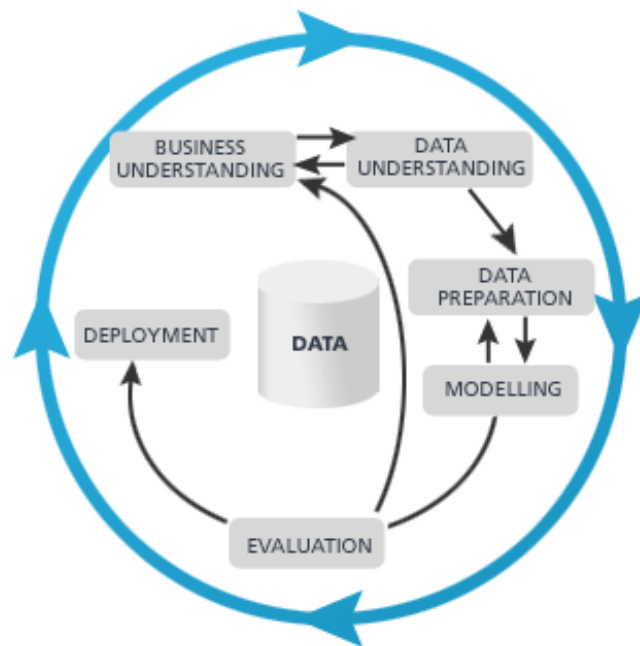


Fig. 2 – Fases do Processo Data Mining descritas em CRISP-DM

As fases do processo de Data Mining, tal como estão descritas em CRISP-DM, estão resumidas a seguir:

- **Bussiness Understanding**

Esta fase inicial do projecto de Data Mining baseia-se em entender os objectivos, requisitos e exigências do projecto. É fundamental formular o problema numa perspectiva de negócio e perceber qual o enquadramento e necessidades. Uma vez definido o problema numa perspectiva de negócio, este deve ser reformulado como um problema de Data Mining e desenhado um plano preliminar de implementação. Devem, também, ser definidos os critérios de sucesso e as medidas de avaliação dos resultados.

- **Data Understanding**

A fase de compreensão dos dados inicia com a identificação das fontes disponíveis, recolha de dados e sua exploração. O objectivo é aumentar a familiaridade com os dados e descobrir as suas características principais. A compreensão dos dados e dos atributos é necessária para o bom desenvolvimento deste processo. Isto passa, por exemplo, por identificar subgrupos nos dados, identificar o atributo alvo para as tarefas de previsão e analisar estatísticas simples.

- **Data Preparation**

A preparação dos dados inclui todas as tarefas para a construção do conjunto final dos dados. Inicia com a escolha dos dados a usar para a análise. Os critérios para a escolha devem ter em conta os objectivos do problema de Data Mining e condições técnicas tais como o volume e tipo de dados. Nesta fase, os dados são limpos, tratados e construído um conjunto de dados de acordo com as necessidades que resultam da definição do problema.

- **Modeling**

Nesta fase, os algoritmos a usar são seleccionados e os parâmetros são ajustados para otimizar os resultados. O ajuste dos parâmetros implica a reconstrução do modelo até encontrar resultados óptimos. É frequente, nesta fase, a necessidade de voltar à fase preparação para reajustar o conjunto de dados usada. São criados procedimentos para validar e testar a qualidade dos modelos.

- **Evaluation**

A fase de avaliação dos métodos é essencial e permite comparar e avaliar de que forma o(s) modelo(s) responde(m) ao problema formulado na fase 1 (*Bussiness Understanding*). O objectivo chave é determinar se existem aspectos importantes de negócio que não ainda foram considerados. A avaliação dos métodos pode determinar uma nova estratégia para abordar o problema. No final desta fase, deve ser tomada a decisão quanto à aplicação dos resultados obtidos.

- **Deployment**

Da análise e interpretação dos resultados pode resultar a aplicação dos modelos a novos conjuntos de dados, à recolha de detalhes ou regras, à integração destes com aplicações externas ou de construção de relatórios. Pode também implicar alterações mais profundas, como introduzir soluções e estratégias de desenvolvimento na área de negócio em estudo, assentes nos resultados obtidos.

Os métodos de Data Mining podem ser divididos em duas categorias (Han, et al., 2006): *supervised* e *unsupervised*. Os *supervised methods* são usados para prever um valor e requerem a especificação de um atributo alvo (*outcome*). Ao contrário, os *unsupervised methods* são aplicados para descobrir a estrutura intrínseca, padrões ou afinidades entre os dados. Estes não necessitam da especificação de um atributo alvo.

Podemos, também, classificar os métodos de Data Mining em *predictive* e *descriptive*, ou seja, métodos de previsão e de descrição (Han, et al., 2006). Os métodos de previsão constroem um ou mais modelos que são usados para prever o valor do atributo alvo em novos conjuntos de dados. Os métodos de descrição descrevem os conjuntos de dados duma forma concisa e apresentam as características interessantes dos dados. Os *supervised methods* são normalmente associados aos métodos de previsão, tais como os que incluem os métodos de classificação e de regressão e as *unsupervised methods* aos métodos de descrição incluindo os de *clustering* e as regras de associação.

Diferentes métodos servem propósitos distintos e apresentam vantagens e desvantagens. Diferem no tempo de processamento e de construção dos modelos, na interpretação dos dados, na interpretação e leitura dos resultados obtidos e na aplicação a diversos domínios. A natureza dos dados pode determinar qual o método que promove uma melhor solução para o problema em causa. O mesmo problema pode ser estudado por vários métodos de Data Mining.

2.2. *Tarefas de Data Mining*

Há vários tipos de tarefas em que se pode usar os métodos de Data Mining (Han, et al., 2006). Os métodos de classificação e previsão têm por objectivo a identificação de padrões em dados históricos que relacionem os valores das variáveis independentes e o de uma variável objectivo. Esses padrões são depois usados para prever o valor do variável objectivo para novos exemplos. No caso dos métodos de classificação, a variável objectivo assume valores discretos, sem ordem de grandeza, referidos como as classes. Para dados cujo valor da variável objectivo seja numérico ou em que exista ordem de grandeza, os métodos usados são de regressão. As classes numéricas podem ser convertidas em classes discretas através da discretização de valores. Os métodos de classificação são aplicados a problemas onde é importante prever comportamentos ou resultados.

Para problemas onde é necessário encontrar sub-conjuntos de dados semelhantes entre si, são usados métodos de *clustering*. Os métodos de *clustering* são úteis para agrupar os dados onde estes grupos (*clusters*) não são conhecidos previamente. Um bom método de *clustering* cria *clusters* cuja semelhança entre elementos do mesmo grupo é elevada e a semelhança entre elementos de grupos diferente é reduzida. Os modelos de associação são especialmente usados para encontrar relações e co-ocorrências em conjunto de dados. Tradicionalmente estes métodos são usados em análises “*market basket*”, mas são poderosos aplicados a áreas de marketing, personalização de páginas da Web, construção de catálogos, definição de promoções, etc.

Existem ainda outro tipo de tarefas de Data Mining, como por exemplo detecção de *outliers* (podemos ver mais detalhes em (Han, et al., 2006)) mas vamos focar-nos na que é mais relevante para este trabalho, a classificação.

2.2.1. Métodos de Classificação

A classificação de um conjunto de dados rotulados com uma classe, consiste na identificação de padrões que discriminem os exemplos de classes diferentes. As classes podem ser binárias (o atributo assume um de dois valores, por exemplo, 0 ou 1, sim ou não, preto ou branco) ou multi-valor (o atributo tem neste caso mais do que dois valores e não há uma relação de ordem ou grandeza entre eles). No contexto de Data Mining, a classificação é feita usando um modelo que é construído com dados históricos ou dados de referência. O objectivo dos métodos de classificação é gerar modelos que prevêem com exactidão a classe alvo, para cada registo de um novo conjunto de dados. Os métodos de classificação constroem um modelo a partir de um conjunto de dados, também denominados como dados de treino. Diferentes métodos de classificação usam técnicas diferentes para criação dos modelos. Os modelos construídos podem, então, ser aplicados a novos casos para previsão da classe ou categoria até aqui desconhecida. Os métodos de classificação têm aplicação em diversas áreas tais como análise de crédito, acções de marketing, detecção de fraudes, telecomunicações, segmentação de clientes e em diagnósticos médicos (Fayyad, et al., 1996).

Árvores de Decisão

As árvores de decisão são modelos baseados em probabilidades condicionais e geram modelos que podem ser facilmente interpretados e aplicados a um conjunto de dados para identificar a classe a que os registos pertencem. A designação deste modelo prende-se com a forma como o modelo é gerado e aplicado. Um valor alvo é previsto a partir duma sequência de perguntas. Num determinado estado a pergunta depende da resposta que foi obtida no estado anterior. Desta forma, um problema complexo é decomposto em problemas mais simples. Recursivamente, a mesma estratégia é aplicada a cada sub-problema. O objectivo é que este processo identifique unicamente um valor alvo. Graficamente este processo forma uma estrutura em árvore, como no exemplo da Fig. 3.

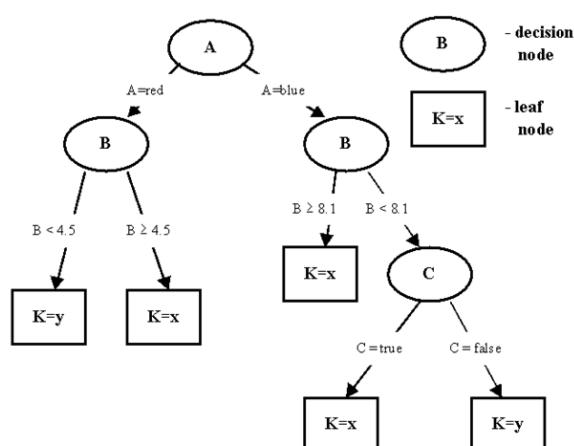


Fig. 3 – Exemplo duma árvore de decisão

Durante o processo de treino, as árvores de decisão devem, repetidamente, encontrar a forma mais eficiente de separar os registos em dois nós. O objectivo é encontrar o atributo mais útil para a classificação dos exemplos. As medidas que quantificam a utilidade de um atributo avaliam como é feita a separação dos exemplos de treino de acordo com o valor da classe alvo. Existem duas métricas de homogeneidade, *Gini* e *Entropy*, para avaliação da separação dos registos. O algoritmo usa estas métricas para avaliar a qualidade das várias alternativas das condições de separação e selecciona a mais adequada, consoante a métrica escolhida. A métrica de Gini atribui a um dos ramos a percentagem mais elevada de uma classe e a entropia tenta balancear e separar as classes tanto quanto possível (Han, et al., 2006).

As Árvores de Decisão são muito sensíveis a pequenas variações nos dados e à existência de ruído. Estas condições podem originar grandes alterações nos modelos. Em modelos muito complexos ou construídos a partir de dados que apresentem ruído ou com poucos exemplos no conjunto de treino, pode ocorrer sobre-ajustamento (*Overfitting*) dos dados. O *Overfitting* impede a capacidade de generalização do modelo a novos conjuntos de dados. Existem várias técnicas para eliminar o *Overfitting* nas árvores de decisão. São técnicas que determinam os critérios de paragem na construção e crescimento da árvore de decisão.

Naive Bayes

Tal como o modelo das Árvore de Decisão, o algoritmo de *Naive Bayes*, é baseado em probabilidades condicionais. Este algoritmo usa o teorema de *Bayes*, que deve o seu nome a Thomas Bayes:

$$p(\text{Decisão}_i | x) = \frac{p(x|\text{Decisão}_i) p(\text{Decisão}_i)}{p(x)}$$

A regra de *Bayes* mostra como a variação das probabilidades hipótese (ou decisão) varia, tendo em conta nova evidências. Os requisitos necessários para a aplicação do teorema de *Bayes* como classificador requerem:

- Conhecer as probabilidades à priori $p(\text{Decisão}_i)$
- As probabilidades condicionais $p(x|\text{Decisão}_i)$

Este algoritmo assume que os atributos são independentes, mas tem demonstrado um bom desempenho mesmo em situações onde encontramos claras dependências entre atributos. Tem um comportamento bastante robusto ao ruído e a atributos irrelevantes. Toda a informação necessária à construção do modelo é adquirida com uma única passagem pelos dados o que o torna um dos algoritmos de classificação mais rápidos.

Support Vector Machines

As *Support Vector Machines* – SVM – são o estado actual da arte ao nível dos algoritmos de classificação e de regressão. Têm um forte fundamento teórico e um também um bom desempenho na construção de modelos para conjunto de dados com muitos atributos e comparativamente ao número de registos.

As SVMs usam *Kernels* para projectar os dados de entrada num espaço onde é esperado que a separação das classes seja mais fácil. Tipicamente, esse espaço é de muito maior dimensionalidade do que o espaço original. Outra característica importante é que o algoritmo das SVMs tenta separar as classes através da maximização das margens e não pela minimização do erro. Esquemáticamente, no Gráfico 1, vemos um exemplo da separação das classes através de modelos com SVMs.

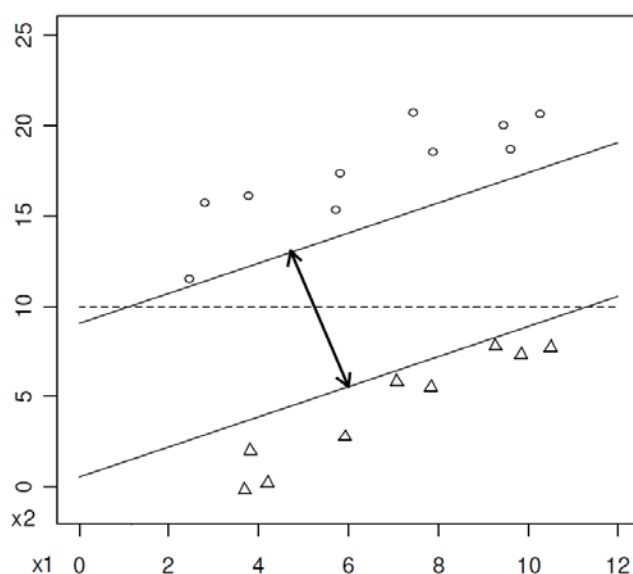


Gráfico 1 – Separação das classes através da maximização das margens

Os dois *Kernels* mais comuns são o *Kernel* linear e *Kernel* *Gaussiano* (não-linear). Diferentes tipos de *Kernels* e diferentes escolhas dos seus parâmetros podem gerar diferentes propostas para os limites das margens. A utilização de *kernels* não lineares permite obter fronteiras não lineares com um algoritmo que determina uma fronteira linear.

2.3. *Medidas de Avaliação*

Dado que os modelos de classificação são desenvolvidos para serem aplicados a dados diferentes daqueles que foram usados para os construir, torna-se essencial avaliá-los para medir até que ponto eles são capazes de fazerem as previsões correctamente. Essa avaliação é tipicamente feita separando o conjunto de dados disponíveis em 2 partes: o conjunto de treino e o conjunto de teste. O modelo é construído com base no conjunto de treino. Depois é aplicado aos dados de testes e compara-se o valor da classe de cada exemplo neste conjunto com o que se obtém na previsão. Esta técnica serve para avaliar o modelo e estimar a incerteza das suas previsões. Existem variantes desta técnica, como, por exemplo, a validação cruzada (Han, et al., 2006).

Existem várias medidas de avaliação do teste aos modelos que permitem saber o interesse de cada modelo e avaliar a sua performance na classificação dos dados:

- **Matriz de Confusão**

A Matriz de Confusão é uma tabela para visualização dos resultados tipicamente usada em aprendizagem classificação. Cada linha da matriz representa as instâncias previstas de uma classe enquanto cada coluna da matriz representa as instâncias reais de uma classe. Um dos benefícios da matriz de confusão é que é fácil de analisar, principalmente se o sistema prevê duas classes. No caso de sistemas que prevêem mais do que duas classes, estes podem ser reduzidas a duas, a classe alvo, designada como classe positiva, e as restantes podem ser agrupadas para formar uma só classe, designada como a classe negativa. Esta estratégia permite comparar, mais facilmente, as previsões da classe positiva com as restantes classes, sendo possível construir uma tabela desta para cada classe. Desta forma, podemos considerar que a matriz de confusão é uma tabela com duas linhas e duas colunas que regista o número de True Negatives (TN), False Positives (FP), False Negatives (FN) e True Positives (TP).

		actual value		total
		<i>p</i>	<i>n</i>	
prediction outcome	<i>p'</i>	True Positive	False Positive	P'
	<i>n'</i>	False Negative	True Negative	N'
total		P	N	

TN – Instâncias negativas classificadas como negativas
 FP – Instâncias negativas classificadas como positivas
 FN – Instâncias positivas classificadas como negativas
 TP – Instâncias positivas classificadas como positivas

Fig. 4 – Matriz de Confusão

- **Accuracy**

Accuracy é o acerto do sistema considerando a proporção de instâncias correctamente classificadas no total dos registos. Esta medida é calculada através da fórmula:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

- **Recall, Precision e F1-Measure**

Recall e *Precision* são duas medidas de avaliação amplamente utilizadas. *Recall* pode ser considerada como uma medida de completude enquanto *Precision* pode ser considerada como uma medida de precisão ou de fidelidade. *Recall* mede a capacidade de um sistema em encontrar o que se quer e *Precision* mede a capacidade do mesmo sistema em rejeitar o que não se quer. Podemos obter estas medidas a partir da matriz de confusão a partir das fórmulas:

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

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

F1-Measure é definida como a média harmónica entre os valores de Recall e do Precision. É obtida da seguinte forma:

$$F1 - Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

A média harmónica é mais intuitiva do que a média aritmética quando envolve o cálculo da média de rácios. Esta medida é apropriada para situações onde é desejada a média das taxas ou percentagens.

- **False Positive Rate (FPR) e True Positive Rate (TPR = Recall)**

False Positive Rate é um indicador que permite saber qual a percentagem de instâncias que são erradamente classificadas como positivas, isto é, são erradamente classificadas como pertencentes à classe alvo. Pode ser obtida através da fórmula:

$$FPR = \frac{FP}{FP + TN}$$

O indicador que permite saber qual a percentagem de instâncias negativas classificadas correctamente é o TNR – *True Negative Rate*. É definido desta forma:

$$TNR = \frac{TN}{FP + TN}$$

Este indicador é conhecido como o indicador da “especificidade”. O indicador FPR pode também ser obtido através da fórmula:

$$FPR = 1 - TNR$$

True Positive Rate é um indicador que permite saber qual a percentagem de instâncias classificadas correctamente como positivas. Este indicador é o mesmo que o *Recall*. Podemos obtê-la através da seguinte fórmula:

$$TPR = \frac{TP}{FN + TP}$$

Este é o indicador que representa a “sensibilidade” do sistema.

- **Análise Curvas ROC e Area Under Curve (AUC)**

As curvas ROC – *Receiver Operating Characteristic* – representam graficamente a relação entre *True Positive Rate* e *False Positive Rate*. A análise das Curvas ROC é uma ferramenta que promove a comparação do desempenho dos modelos e que permite visualizar o compromisso entre os dois tipos de erros referidos. No Gráfico 2 temos um exemplo:

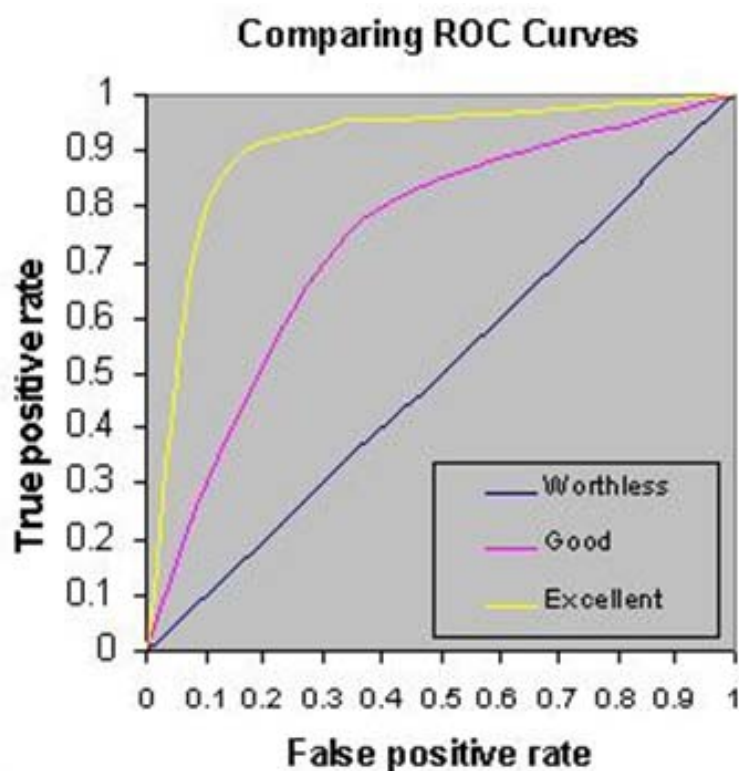


Gráfico 2 – Comparação de Curvas ROC

Para comparar classificadores pode-se reduzir a Curva ROC a um valor escalar. Este valor é representado pela área abaixo da curva do gráfico que relaciona o TPR e o FPR. Este valor é designado por AUC – *Area Under Curve*. O valor da AUC igual a 1 representa um teste perfeito. Ao contrário, um valor de AUC igual a 0,5 representa um teste fraco e sem valor.

O *Threshold* é uma medida que marca, no contexto da classificação em Data Mining, o limiar a partir do qual a nossa previsão se altera. Cada previsão tem uma confiança associada e são seleccionados como positivos, os casos cuja confiança é superior ao valor de *Threshold*. É a linha fronteira entre as classes. A variação deste valor provoca o reajuste das previsões e nova classificação dos registos em teste.

O valor do *Threshold* varia entre 0 e 1 sendo que, quando assume estes valores extremos, classifica todos os registos como negativos ou todos os registos como positivos respectivamente. A diminuição do valor de *Threshold* promove a classificação dos registos como positivos e diminui a classificação dos registos como negativos. Este cenário implica que aumentem os registos correctamente classificados como positivos (True Positives), mas também os registos erradamente classificados como positivos (False Positives). Ao contrário, diminuem os registos correctamente classificados como negativos (True Negatives) e diminuem os incorrectamente classificados como negativos (False Negatives). Em resumo, a diminuição do valor de *Threshold* faz aumentar o número dos True Positives e reduzir número dos False Negatives. Contudo, este favorável comportamento é acompanhado de um aumento dos False Positives. Idealmente era desejável ter um elevado número de registos correctamente identificados e um baixo número de registos classificados incorrectamente. Não sendo possível, a variação do *Threshold* permite avaliar qual o equilíbrio em termos de classificação de registos.

2.4. Oracle Data Mining e Oracle Data Miner

Neste trabalho optamos pela aplicação de tecnologias disponibilizadas pela Oracle – **Oracle Data Mining**.

O Oracle Data Mining é uma opção do Oracle Database 11g Enterprise Edition que permite construir e desenvolver aplicações de última geração que proporcionam novas abordagens aos dados e previsões antecipadas de comportamentos (Oracle Corporation , 2008). É um poderoso *Software* que disponibiliza técnicas para encontrar padrões, identificar atributos chave e associações escondidas ou valores anormais num conjunto de dados e que se encontra embebido no Oracle Database. O Oracle Data Mining está implementado no núcleo do Oracle Database e os modelos são a primeira classe dos objectos da base de dados. Os processos de Oracle Data Mining usam os princípios de implementação do Oracle Database para maximizar o desempenho e disponibilizar um eficiente aproveitamento dos recursos do sistema. Esta característica elimina a necessidade de extracção dos dados, da base de dados, para análise por aplicações externas, conforme descrito no início do capítulo. Este processo permite a transformação e o processamento grande quantidade de informação automaticamente, de forma assíncrona e independente da interface gráfica usada. Ao contrário do que tradicionalmente acontecia, os dados eram recolhidos das bases de dados através de pesquisas e pré-processados para posterior aplicação dos métodos de Data Mining. Isto permite manter o nível de segurança e estabilidade que estão associadas às tecnologias Oracle.

O Oracle Data Mining disponibiliza métodos de Data Mining desenvolvidos e aperfeiçoados ao longo da última década. Diferentes tipos de algoritmos aplicam-se a diferentes tipos de análises. Oracle Data Mining suporta técnicas de *supervised* e *unsupervised learning* e inclui algoritmos de Classificação e Regressão, Regras de Associação, Modelos de Clustering, Selecção e Importância de Atributos embebidos no Oracle Database. A maioria destes algoritmos podem ser aplicados a dados estruturados ou não-estruturados. Na tabela seguinte temos um resumo dos algoritmos disponíveis, para cada técnica, no Oracle Data Mining:

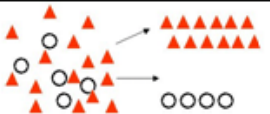
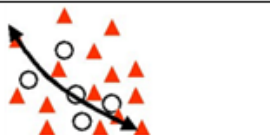
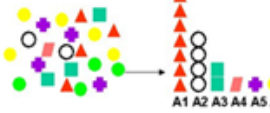
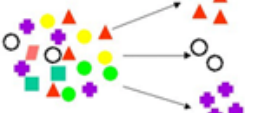
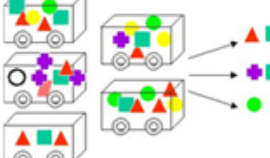
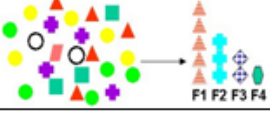
	Técnica	Algoritmo
Classificação		• Logistic Regression • Naive Bayes • Support Vector Machine • Decision Tree
Regressão		• Multiple Regression • Support Vector Machine
Attribute Importance		• Minimum Description Length
Anomaly Detection		• One-Class Support Vector Machine
Clustering		• Enhanced K-Means • Orthogonal Partitioning Clustering
Regras de Associação		• Apriori
Feature Extraction		• Non-negative Matrix Factorization

Tabela 1 – Modelos disponíveis no Oracle Data Mining

O Oracle Data Mining está acessível (Oracle Corporation., 2007) através de interfaces programáticos para SQL, PL/SQL, Java e Add-Ins para o Excel. Estão disponíveis *packages* e funções que permitem o desenvolvimento dos modelos a partir de SQL e de PL/SQL e o Java Data Mining (JSR-73) permite integração destas funções em aplicações Java. Como interfaces gráficas temos o Oracle Spreadsheet Add-In para o Microsoft Excel e o Oracle Data Miner que ajuda a construção de modelos. O Oracle Data Mining Java Code Generator é uma extensão do Oracle JDeveloper que exporta os modelos criados em Oracle Data Miner para código Java. É também possível exportação dos modelos para *packages* de PL/SQL.

Oracle Data Miner

A interface que vamos usar é o Oracle Data Miner. Com esta opção pretende-se avaliar as capacidades desta ferramenta comercial que ainda não está muito divulgada. Vamos descrever algumas das suas capacidades e limitações e perceber de que forma influencia este trabalho.

O Oracle Data Miner é uma interface gráfica a partir do qual se constroem modelos sobre o Oracle Data Mining que, para além das capacidades de pesquisa e de pré-processamento dos dados, garante a correcta ordem das operações e disponibiliza configurações inteligentes e optimizações para todos os parâmetros (Oracle Data Miner Tutorial, 2008). Para estudos mais avançados, os parâmetros pré-definidos podem ser alterados. Os analistas de dados podem usar os assistentes (*wizards*) disponíveis do Oracle Data Miner para pesquisa e preparação de dados, construção e avaliação de modelos através da aplicação das técnicas de Data Mining disponibilizadas e interpretar os resultados. O Oracle Data Miner proporciona também a geração de código PL/SQL e Java, passo a passo, dos modelos criados para futuras integrações de aplicações Data Mining ou Business Intelligence.

Nas Fig. 5 e Fig. 6 percebemos algumas das possibilidades oferecidas no Oracle Data Miner ao nível da transformação e pré-processamento dos dados e um exemplo do *Wizard* de construção de modelo de classificação respectivamente:

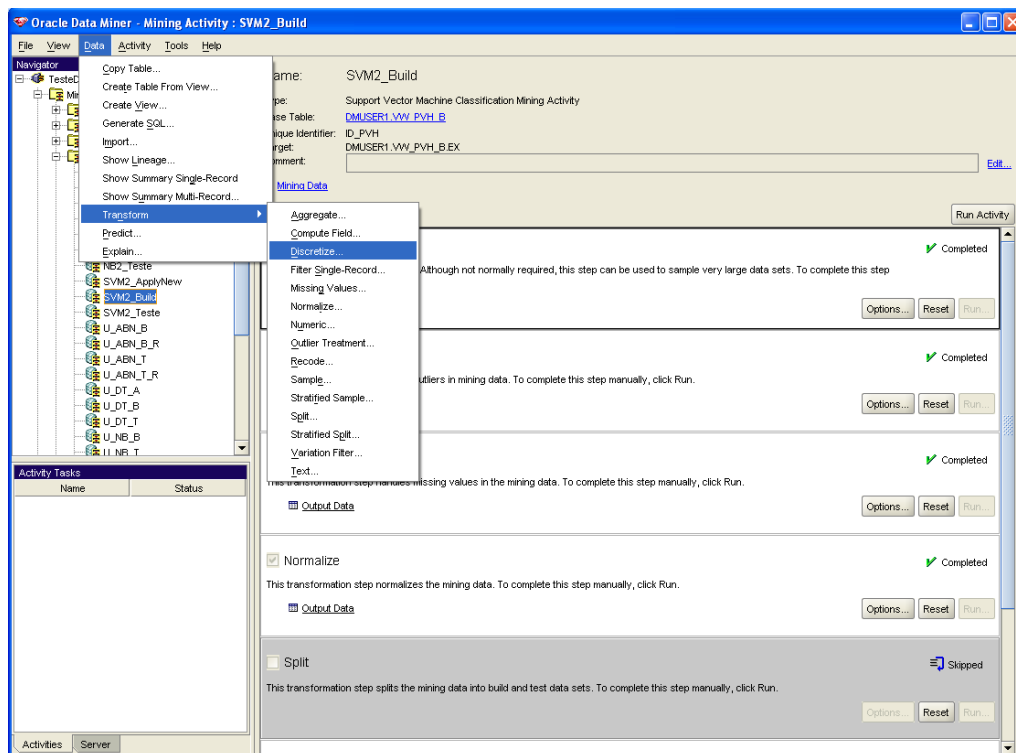


Fig. 5 – Transformação de dados no Oracle Data Miner

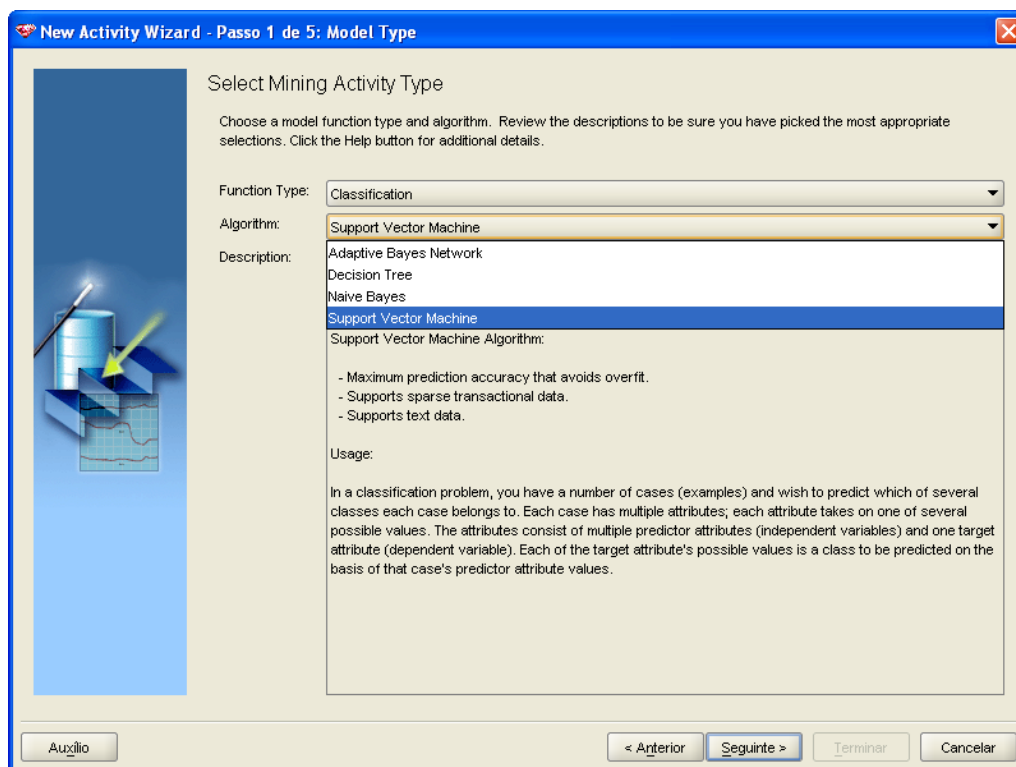


Fig. 6 – Wizard para construção de modelos

A construção dos modelos através dos *Wizards* disponíveis no Oracle Data Miner, é feita em duas partes: A 1ª parte refere-se à selecção das condições gerais que são exigidas pela técnica escolhida e que são comuns aos vários algoritmos dessa técnica; na 2ª parte são escolhidos os parâmetros específicos do algoritmo a aplicar no estudo em causa. Os modelos construídos têm uma visualização própria de acordo com as técnicas e algoritmos escolhidos. Os diversos resultados são apresentados de forma amigável através de gráficos ou de tabelas, sempre com a possibilidade de exportação para ficheiro.

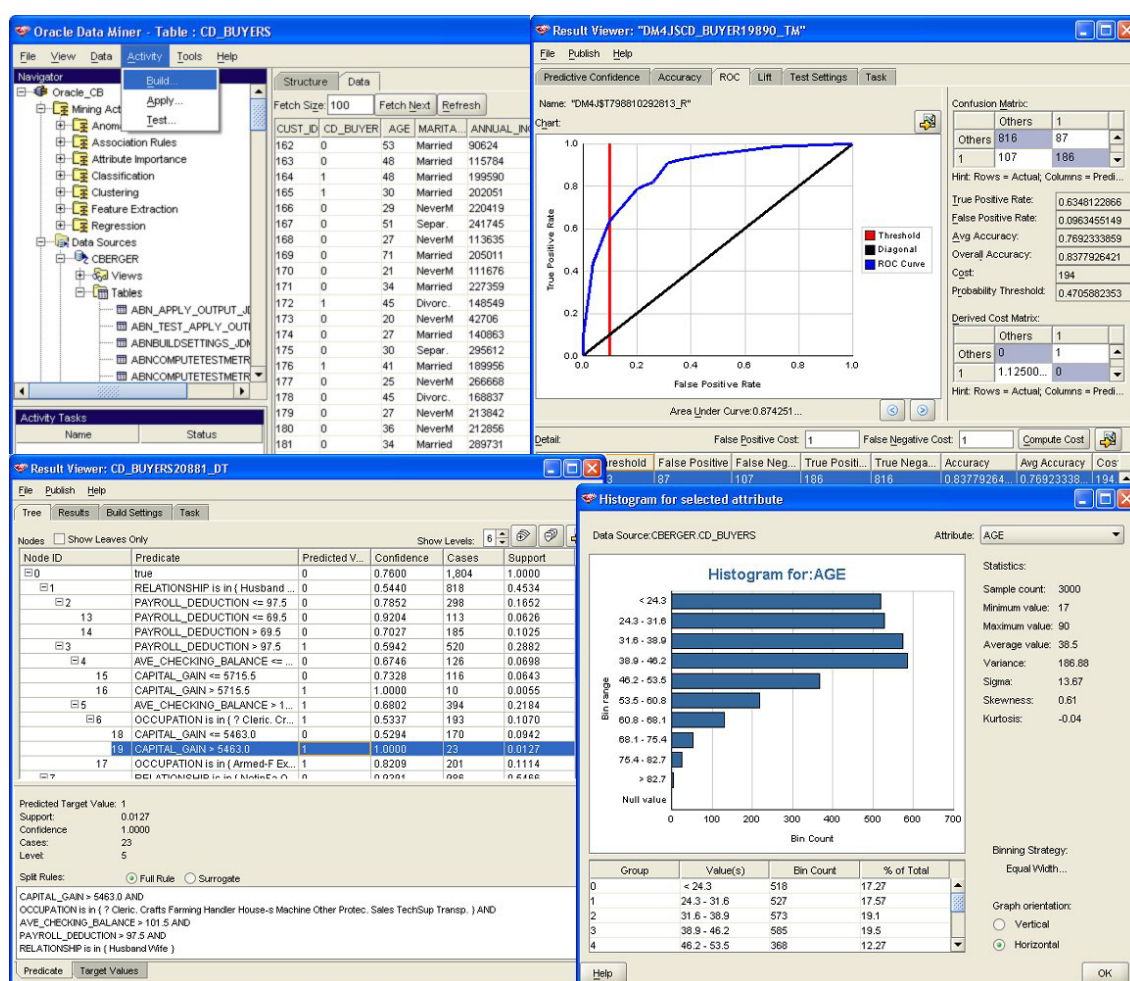


Fig. 7 – Várias apresentações dos resultados no Oracle Data Miner

2.4.1. Métodos de Classificação no Oracle Data Miner

Para os modelos de classificação, a construção inicia-se com a identificação do conjunto de dados e a especificação dos atributos. O conjunto de dados está disponível numa tabela ou *view*¹. Deve ser escolhido qual o repositório do conjunto de dados e identificado qual a *primary key*, isto é, o identificador único deste conjunto. Por defeito, todos os atributos estão seleccionados e deve ser confirmada a sua necessidade, rejeitando os que não são necessários ao estudo. Deve também ser confirmado o tipo dos atributos, categóricos ou numéricos, e escolhido o atributo alvo, isto é, o atributo que identifica as classes e qual o valor preferencial deste atributo. A escolha dum nome para identificação posterior do modelo criado é obrigatória.

Numa 2ª fase, agimos sobre a escolha dos dados. Podemos definir o tamanho do conjunto de dados, isto é, quantos registos são usados para o estudo indicando a contagem ou a percentagem de registos a considerar. É possível definir o modo de selecção, aleatório ou estratificado. A amostragem aleatória escolhe o número de casos específicos com aproximadamente a mesma distribuição de valores de alvo que nos dados originais. A amostragem estratificada escolhe casos para conduzir aproximadamente aos dados com o mesmo número de casos com cada valor de alvo. A amostragem estratificada é recomendada nas situações onde a distribuição das classes não é homogénea. Em termos de acerto, podemos ter um modelo que maximize o acerto global ou que seja bom a prever todas as classes. Esta indicação é dada pela escolha de uma das opções “*Maximum Overall Accuracy*” ou “*Maximum Average Accuracy*”. Já vimos anteriormente (secção 2.3) que a medida *Accuracy* é o acerto do sistema considerando as instâncias correctamente classificadas no total dos registos. Esta medida é calculada através da fórmula:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

¹ Já referimos, anteriormente, que o Oracle Data Miner é uma interface com o Oracle Data Mining que assenta sobre uma base de dados relacional, o Oracle Database.

O *Average Accuracy* é a média de acerto do sistema considerando as instâncias correctamente classificadas para a classe positiva e para a classe negativa. Esta medida é conseguida através da fórmula:

$$Avg\ Accuracy = \frac{\frac{TP}{TP + FN} + \frac{TN}{TN + FP}}{2}$$

Na fase seguinte são definidas as condições de divisão do conjunto de dados em treino e teste através da indicação da percentagem de dados para treino e teste respectivamente. São também definidas as métricas de teste. As métricas disponíveis são, entre outras, a Matriz de Confusão, os valores de *Recall* e de *Precision*, *Lift*, Curvas ROC, *Area Under Curve* (AUC) e as taxas de acerto. Há também a possibilidade de obter a matriz de custos se previamente forem introduzidos os seus valores.

Os algoritmos de classificação referidos anteriormente, Árvore de Decisão, *Naive Bayes* e *Support Vector Machines* estão disponíveis no Oracle Data Miner. Os parâmetros específicos para cada algoritmo são escolhidos na última fase deste processo de construção do modelo e estão de acordo com as particularidades de cada algoritmo. Para as Árvore de decisão, é escolhida a métrica de homogeneidade e as condições de paragem na construção do modelo. Para o *Naive Bayes*, são escolhidas as probabilidades mínimas, probabilidade condicional e à priori, e que condicionam a paragem na construção do modelo. No caso das SVMs, o critério de paragem é o valor de tolerância. Escolhemos o factor de complexidade e se pretendemos aprendizagem activa. É nesta fase que é identificada a função de *kernel* que o modelo vai usar.

Para além dos três algoritmos referidos, o Oracle Data Mining disponibiliza, a partir do Oracle Data Miner, um algoritmo de classificação que proporciona meios rápidos, evolutivos e não-paramétricos de extrair informação para previsão de um atributo alvo – *Adaptive Bayes Network* (ABN). O ABN é um algoritmo proprietário da Oracle (Oracle Corporation, 2008) que faz previsões quer para atributos binários quer para atributos multi-classe. Há três modalidades para os modelos construídos com ABN:

- Pruned Naive Bayes (Naives Bayes Model)
- Simplified Decision Trees (Single Feature Build)
- Boosted (Multi Feature Build)

ABN selecciona o atributo mais importante e usa-o para construir o modelo. Regra geral, este algoritmo não tem um bom desempenho para pequenos conjuntos de dados. Este algoritmo é gerador de regras só no caso de modelos construídos para *Single Feature Build*. Neste caso, o ABN não gera regras interessantes se o conjunto de dados for inferior a 20.000 exemplos.

Ao nível dos parâmetros específicos, no caso das *Adaptive Bayes Network*, são identificados, para a modalidade escolhida, quantos atributos a considerar na previsão para construção dos modelos. O critério de paragem para este modelo é definido através de um tempo máximo, em minutos.

O Oracle Data Miner disponibiliza os resultados através de um quadro resumo onde temos disponíveis a matriz de confusão, os valores de *recall* e *precision*, os custos e podemos obter as percentagens de acerto do modelo.

Análise Curvas ROC no Oracle Data Miner

O Oracle Data Miner disponibiliza um *viewer* que permite avaliar o impacto da variação deste valor.

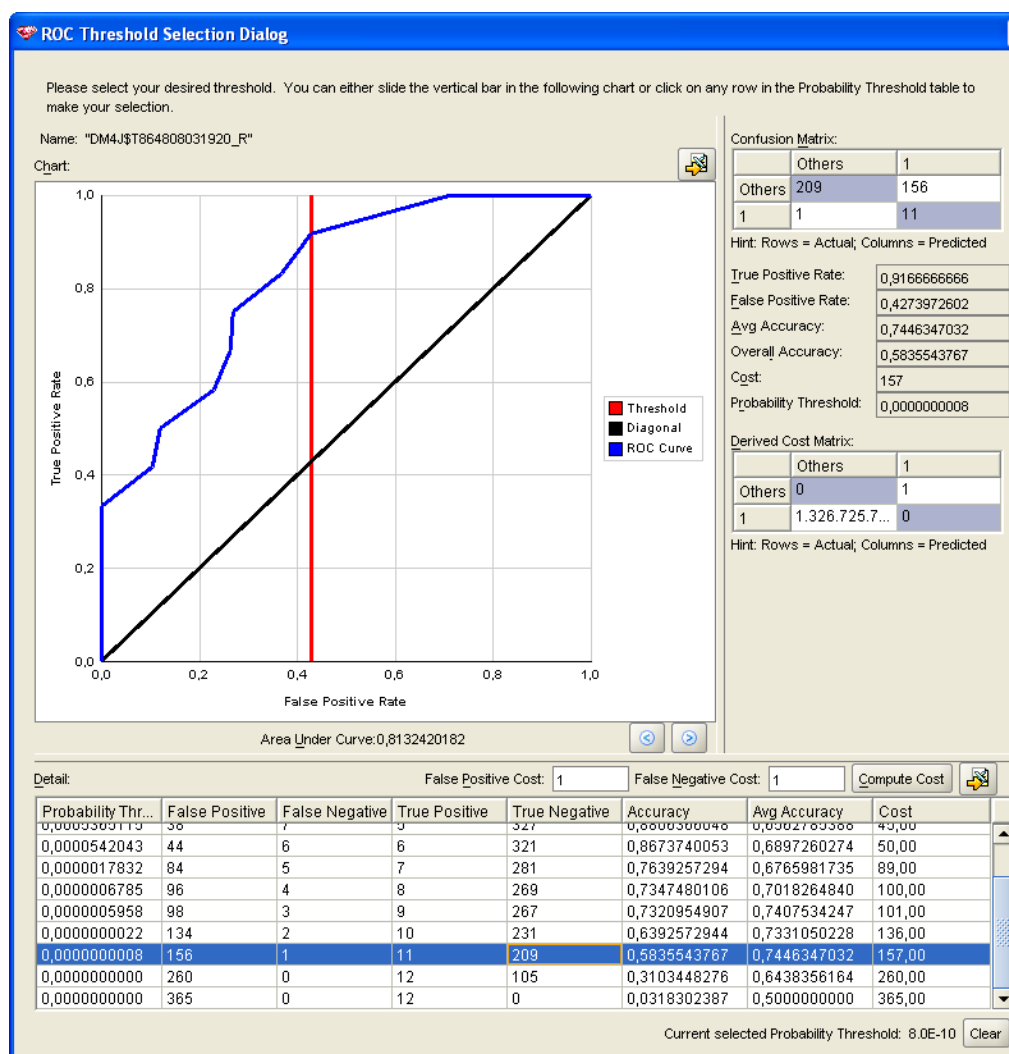


Fig. 8 – Viewer para análise Curvas ROC no Oracle Data Miner

É possível avaliar a variação do *Threshold* através da análise gráfica das Curvas ROC – *Receiver Operating Characteristic Curves* – disponíveis no Oracle Data Miner. As análises das curvas ROC são úteis para avaliar a exactidão das previsões e têm sido muito usadas na teoria de detecção de sinal para comparar os graus de sucesso e as taxas de falsos alarmes.

3. Extracção de Informação

3.1. *Extracção de Informação*

O objectivo dos sistemas de extracção de informação (Sitter, et al., 2003) é extrair informação relevante de um documento ou conjunto de documentos através da descoberta de padrões. Estes padrões podem ter origem léxica, sintáctica ou semântica em texto escrito em linguagem natural (Sun, et al., 2003) ou podem ter origem na forma, frequência da informação e relação entre os dados (Moens, et al., 2009). Desta forma, a extracção de informação pode ser vista como uma actividade de estruturar a informação a partir de informação não estruturada (Gaizauskas, et al., 1998). Um sistema de extracção de informação deve ser capaz de reconhecer informação relevante numa sucessão de eventos (Turmo, et al., 2006) e identificar qual a informação que está relacionada com o mesmo evento.

O interesse nesta área manifestou-se logo após o nascimento da Internet na década de 1960. No entanto, o seu impulso foi nos anos 80 quando DARPA (*Defense Advanced Research Projects Agency*), investiu em pesquisas na área da Inteligência Artificial. Na sequência destes desenvolvimentos, para além dos projectos iniciais nesta área, tem, nos anos 80, a primeira versão comercial dum sistema de extracção de informação (Gaizauskas, et al., 1998). Este sistema, ATRANS (*Automatic Processing of Money Transfer Messages*), foi desenvolvido para o processamento automático de mensagens para as transferências bancárias. ATRANS adopta um método de aproximação baseada em *scripts* para processamento de texto, usando previsões *script-driven* para identificar os intervenientes da transacção, a fim de preencher um *template* que fosse usado, após a verificação humana, para iniciar transferências de dinheiro automáticas (Gaizauskas, et al., 1998). No final desta década, foi também comercializado pela GE – General Electric Company, o SCISOR para extracção de informação de notícias on-line. Em 1990 DARPA lançou TIPSTER (Turmo, et al., 2006). Este trabalhava em duas fases: a primeira através do retorno de documentos relevantes e a segunda para extracção de dados. A metodologia associada a esta avaliação foi definida como "uma sequência de

módulos que, em cada passo, adiciona estrutura aos documentos e, às vezes, filtra informação relevante pela aplicação de regras" (Turmo, et al., 2006). Foi este o tópico da MUC-5 em 1993 mas desenvolvia-se desde 1987 (Sundheim, 1993), quando iniciaram as MUC's (*Message Understand Conferences*). No início da década de 90, foram definidas algumas das principais medidas de avaliação, *Recall* e *Precision* em 1991 e *F-Measure* em 1992.

Actualmente, os sistemas de extracção de informação são mais complexos, resultado das necessidades actuais de processamento de informação em larga escala. O desenvolvimento tecnológico na área dos sistemas de informação disponibiliza rápidos processadores e uma capacidade de armazenamento de dados e de documentos quase ilimitada, permite manter e gerir grandes bases de dados e é uma contribuição fundamental para a evolução dos trabalhos nesta área.

3.2. *Extracção de Informação de Documentos*

O tema da extracção de informação a partir de documentos em papel ou ficheiros tem sido amplamente estudado e a sua aplicação destina-se às mais diversas áreas. As áreas alvo de estudo são seleccionadas de forma natural mediante a importância e a necessidade em compreender a informação que tratam. São áreas como a matemática, a química e a física, a medicina, a biologia e a economia que têm permitido o avanço destes estudos à medida que impõem desenvolvimentos científicos mais elaborados e onde a perda de tempo e o rigor com extracção de dados de documentos não são compatíveis com os progressos exigidos. Áreas intervenientes na melhoria de processos *Workflow*, também contribuem para diversificar as abordagens e os progressos no âmbito da extracção de dados. A portabilidade destes sistemas também é fundamental para aplicação a novas situações (Turmo, et al., 2006). A extracção de dados relevantes e automatizados a partir da Internet tem sido fundamental para a evolução desta temática uma vez que esta é o maior repositório de dados e de qualquer tipo de informação.

3.2.1. Extracção via Dupla Classificação

Algumas abordagens (Sitter, et al., 2003) propõem uma aproximação inspirada no comportamento humano para a extracção de informação de documentos. Isto é, a maioria das vezes e perante um documento, a análise deste ou de parte deste, não é feita em detalhe de uma só vez. É feita uma leitura geral e, caso se justifique, esta será mais profunda numa 2ª leitura. Só nesta fase é que a informação a extrair é identificada. Na prática significa que se faz uma classificação dos documentos em duas fases e que se pode dividir os documentos em frases, analisá-las e, se interessar, dividimo-la em palavras.

Os autores testaram o sistema em anúncios de emprego. Neste caso, para cada frase do anúncio, foi usado o classificador *Naive Bayes* trabalhando sobre o conjunto de palavras que a compõem, para decidir se esta é relevante ou não. Desta forma, consegue-se um conjunto significativo de frases para cada documento. Um dos problemas encontrados resulta da existência de apenas uma frase relevante em cada documento, o que resulta num conjunto de dados não-balanceados entre as classes positiva e negativa e torna difícil a aplicação do *Naive Bayes*. Para contornar esta dificuldade, foram seleccionadas, aleatoriamente, entre as frases não relevantes, o mesmo número de frases relevantes. Para cada frase e já ao nível das palavras, foi usado um algoritmo baseado em regras de classificação – RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*). O RIPPER é algoritmo de aprendizagem proposicional desenhado para uma performance eficiente em grandes conjuntos de dados e que apresentam bastante ruído.

Cada palavra é identificada com uma marcação (*tag*) indicando se está *in* ou *out* da entidade a extrair. O conjunto de dados foi construído da seguinte forma: a palavra a ser classificada e respectiva *tag*, um conjunto de três palavras antes e respectivas *tags* e um conjunto de três palavras após e respectivas *tags*.

Para melhor avaliação foram feitas duas aproximações, uma para encontrar todas as ocorrências (AO – *All Occurrences*) e outra para encontrar uma ocorrência por cada documento (OBD – *One Best per Document*).

As medidas usadas para avaliação do sistema são o *Recall*, *Precision* e *F1-Measure* calculados a partir duma *Matriz de Confusão* e o método aplicado foi o de validação cruzada com 10 grupos (com excepção de alguns casos raros em que a validação cruzada foi feita com 5 grupos).

	AO			OBD		
	recall	precision	F_1	recall	precision	F_1
ID	98%	97%	97%	96%	99%	97%
country	98%	92%	95%	94%	91%	92%
state	97%	77%	86%	95%	93%	94%
city	95%	84%	89%	92%	90%	91%
company	78%	45%	57%	74%	26%	38%
title	43%	31%	36%	35%	35%	35%
salary	70%	56%	62%	72%	62%	67%
recruiter	79%	40%	53%	74%	44%	55%
post-date	99%	84%	91%	97%	99%	98%
desired degree	45%	28%	35%	37%	29%	33%
required degree	43%	29%	35%	51%	41%	45%
desired years experience	55%	33%	41%	66%	36%	47%
required years experience	80%	50%	62%	81%	72%	76%
platform	34%	31%	32%	38%	35%	36%
application	29%	32%	30%	30%	31%	30%
area	17%	16%	16%	18%	16%	17%
language	27%	25%	26%	34%	33%	33%
overall	63.9%	49.9%	55.5%	63.7%	54.8%	58.0%

Tabela 2 – Resultados do método aplicado às entidades individualmente

A Tabela 2 resume os resultados obtidos pelo método para cada entidade individualmente.

3.2.2. Extracção de Títulos

Na sequência de uma visita à Microsoft Research na Ásia, foi desenvolvido um método de extracção automática de títulos para documentos através de *Machine Learning* (Hu, et al., 2005). Este trabalho é importante na medida em que pode contribuir para promover o nº e qualidade de documentos retornados em pesquisas. Estes documentos são os disponibilizados via Internet ou intranet e incluem apresentações, artigos técnicos, brochuras, relatórios, cartas, livros, especificações técnicas entre outros. No Gráfico 3, temos uma estimativa da distribuição dos formatos dos ficheiros na Internet e na intranet.

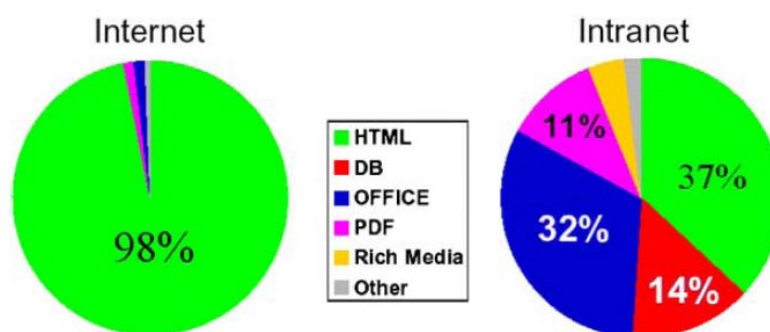


Gráfico 3 – Distribuição de formatos de ficheiros na Internet e na intranet

Excluindo os ficheiros com formato HTML e que naturalmente são os que aparecem em maior número na Internet e na intranet, são os ficheiros do Office os mais disponibilizados. Seguindo esta orientação e como caso de estudo, foi desenvolvido um método para extracção de títulos para alguns documentos do Office – os documentos de Word e os PowerPoint. Nos documentos do Office, os utilizadores podem definir o título com uma propriedade do documento. Contudo, pela recolha efectuada verificou-se, surpreendentemente, que esta propriedade quase nunca está de acordo com o verdadeiro título do documento. Nos documentos de Word, o título pode aparecer no topo da primeira página e no PowerPoint no primeiro slide. O título pode ser constituído por um título principal e por subtítulos. Neste caso são considerados apenas os títulos principais. Após um pré-processamento de dados a extracção de títulos é baseada em *Machine Learning* que consiste em treino e na respectiva extracção. Na recolha e pré-

processamento dos dados, é usado o conhecimento intuitivo sobre a estrutura dos títulos:

- São linhas consecutivas com o mesmo formato
- Um documento pode não ter título
- Há palavras que não fazem parte do título, por exemplo “rascunho”
- O título é o que aparece com a letra de maior tamanho

Num documento, para cada linha, se esta for separada por um parágrafo ou por outra linha de formato diferente, é considerada uma *unit*. Cada *unit* é etiquetada e tratada como uma instância de aprendizagem. Nesta aprendizagem, o input é a sequência de *units* onde cada sequência corresponde a um documento. Foram aplicados cinco tipos de modelos:

- Perceptron with Uneven Margins
- Maximum Entropy (ME)
- Maximum Entropy Markov Model (MEMM)
- Voted Perceptron Model (VP)
- Conditional Random Fields (CRF)

Estes modelos aplicados em conjunto aos dados conseguem promover os resultados na extracção dos títulos e ultrapassar a dificuldade inerente à grande diferença existente entre número de instancias positivas e negativas. Para os resultados foram consideradas as medidas de *Precision*, *Recall* e *F-Measure* e apresentados na tabela seguinte:

	Word			PowerPoint		
Model	Precision	Recall	F1	Precision	Recall	F1
Perceptron	0.810	0.837	0.823	0.875	0.895	0.885
ME	0.801	0.621	0.699	0.753	0.766	0.759
MEMM	0.797	0.824	0.810	0.841	0.861	0.851
VP	0.827	0.823	0.825	0.873	0.896	0.885
CRF	0.834	0.843	0.838	0.880	0.891	0.886

Tabela 3 – Resultados dos diferentes métodos à extracção de títulos

3.2.3. Extracção de Citações

A integração de informação bibliográfica das publicações académicas disponíveis na Internet é uma importante tarefa nas pesquisas académicas. O nº crescente de publicações exige uma eficiente organização dos recursos para proporcionar o rápido acesso a necessidades específicas. A partir destas necessidades (Day, et al., 2005) e (Day, et al., 2007), foi desenvolvido um método de aproximação baseado no conhecimento para extracção de citações. A extracção automática de citações é difícil devido à grande variedade dos campos, sub-campos e respectivos os separadores, consequência dos diferentes estilos e formatos com que são apresentados. O INFOMAP, um software desenvolvido em C, é um esquema de representação do conhecimento e serve de estrutura ontológica a este trabalho. O INFOMAP proporciona o conhecimento necessário para compreender a linguagem natural relacionada com um certo domínio do conhecimento. As citações são compostas por diversos campos delimitados por separadores. Os campos que normalmente as compõem são o(s) autor(es), o título, o ano, a publicação, o volume e as páginas e os separadores podem ser vírgulas (,), ponto e vírgula (;), dois pontos (:), traço (-), aspas (“”), entre outros.

Os dados de teste consistem num conjunto de 500 registos bibliográficos que foram escolhidos, aleatoriamente, de um total de 907 registos recolhidos da Web. Existem vários estilos de apresentação bibliográfica, sendo que este estudo incide apenas em seis estilos: APA, IEEE, ACM, MISQ, JMIS e ISR.

Nesta experiência, só é considerado que um campo é extraído correctamente quando o valor desse campo é extraído correctamente dos dados de teste. A medida de avaliação é calculada da seguinte forma:

$$Accuracy = \frac{n^{\circ} \text{ de campos extraídos correctamente}}{n^{\circ} \text{ total de campos}}$$

Na Tabela 4 temos os resultados experimentais da aplicação do método e apresentados no Gráfico 4.

Accuracy	Author (%)	Title (%)	Journal (%)	Volume (%)	Issue (%)	Year (%)	Pages (%)	Overall average (%)
APA	92.32	71.80	94.33	97.39	84.92	96.48	95.09	90.33
IEEE	94.17	89.05	92.07	95.45	84.49	97.18	89.81	91.75
ACM	88.36	91.10	99.41	80.28	87.73	96.47	83.95	89.61
ISR	91.93	78.33	95.32	95.28	87.00	96.34	90.61	90.69
MISQ	97.73	97.92	100.00	99.99	99.98	99.94	99.64	99.31
JMIS	76.55	72.57	99.99	99.98	99.97	99.93	99.69	92.67
Average	90.18	83.46	96.85	94.73	90.68	97.72	93.13	92.39

Tabela 4 – Resultados experimentais da extracção de citações

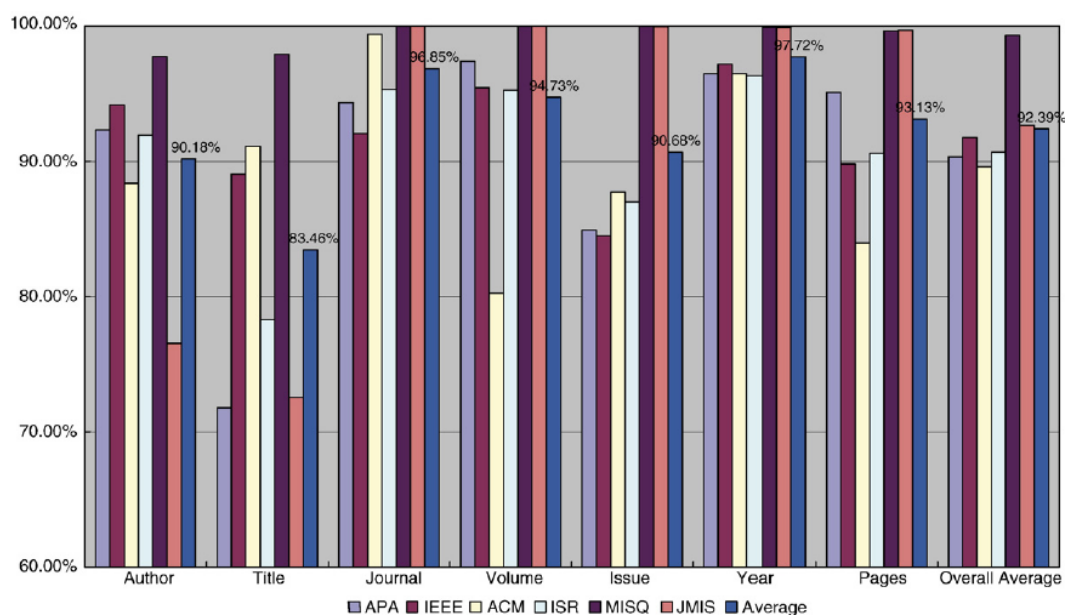


Gráfico 4 – Resultados experimentais da extracção de citações

Ainda neste contexto, foi desenvolvido um processo de segmentação de citações, BibPro, com base em duas ideias (Chen, et al., 2008):

- 1 – Uma citação pode ser representada como uma sequência de proteínas. A citação é separada em *tokens* (palavras) e é usada a simbologia dos aminoácidos para representar cada *token*. Podemos ver um exemplo na Fig. 9.
- 2 – Só a ordem dos separadores e as palavras reservadas é que são transformadas. A informação redundante é filtrada para simplificar o problema.

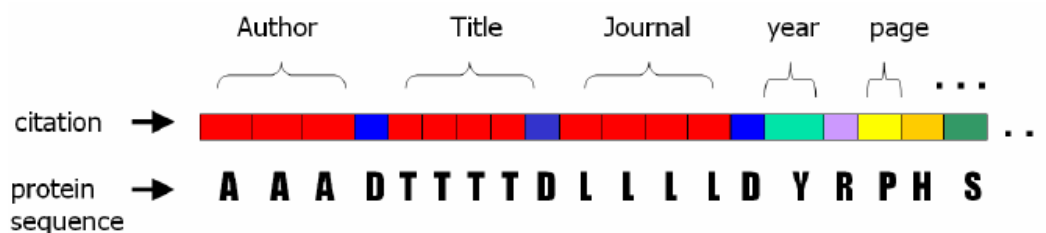


Fig. 9 – Representação da transformação da citação numa sequência de proteínas

Foram seleccionadas aleatoriamente 10.000 citações para treino e 10.000 para teste, dum grupo de 160.000 citações recolhidas dos seis estilos anteriormente referidos. A medida de avaliação é comum ao estudo anterior e vai servir de comparação dos dois estudos. A Tabela 5 mostra os resultados da comparação entre o BibPro e o Infomap:

	Style	author	title	journal	volume	issue	year	page	overall avg
Bib Pro	APA	99.67%	96.38%	97.06%	98.99%	98.12%	99.42%	98.71%	98.33%
	IEEE	98.72%	98.12%	99.12%	99.30%	98.39%	99.40%	98.40%	98.78%
	ACM	97.14%	95.01%	93.93%	97.19%	97.03%	98.88%	97.92%	96.73%
	ISR	99.48%	96.17%	96.96%	99.15%	98.39%	99.35%	98.55%	98.29%
	MISQ	98.59%	97.99%	98.98%	99.41%	98.61%	99.54%	98.83%	98.85%
	JMIS	91.95%	87.90%	90.46%	99.23%	98.03%	99.46%	98.76%	95.11%
	Avg.	97.59%	95.26%	96.09%	98.88%	98.09%	99.34%	98.53%	97.68%
INFO MAP	APA	92.32%	71.80%	94.33%	97.39%	84.92%	96.48%	95.09%	90.33%
	IEEE	94.17%	89.05%	92.07%	95.45%	84.49%	97.18%	89.81%	91.75%
	ACM	88.36%	91.10%	99.41%	80.28%	87.73%	96.47%	83.95%	89.61%
	ISR	91.93%	78.33%	95.32%	95.28%	87.00%	96.34%	90.61%	90.69%
	MISQ	97.73%	97.92%	100%	99.99%	99.98%	99.94%	99.64%	99.31%
	JMIS	76.55%	72.57%	99.99%	99.98%	99.97%	99.93%	99.69%	92.67%
	Avg.	90.18%	83.46%	96.85%	94.73%	90.68%	97.72%	93.13%	92.39%

Tabela 5 – Comparação dos resultados obtidos pelo BibPro e o Infomap

Duma forma geral, verifica-se que este método tem um melhor desempenho relativamente ao anterior em cerca de 5%.

3.2.4. Extracção de Artigos de Facturas

Num trabalho desenvolvido para análise de facturas (Belaïd, et al., 2004), é sugerido um método de codificação morfológica deste tipo de documentos. Este método investe na detecção dos artigos constantes da factura e os seus diferentes campos. Este método tem por base o conceito de marcação de texto – *part-of-speech tagging (PoS)*, mas não pode ser aplicado directamente. PoS assume que o texto é escrito em linguagem natural e, no caso particular dos artigos descritos nas facturas, isto não se verifica. Desta forma, este método teve de sofrer uma alteração e adaptação a esta estrutura. Na fase de recolha dos dados da factura e sua posterior segmentação foi feita a partir de uma digitalização e aplicação do OCR. Foram encontrados alguns problemas no reconhecimento da informação devido a ruído que resultam em erros no OCR: notas escritas manualmente, logótipos, carimbos, estrutura heterogénea, existência de cores de fundo, etc. De uma forma geral, os artigos compõem blocos de linha sucessivas, Cada linha é constituída pelos mesmos atributos que regularmente se repetem na mesma posição. Uma vez encontradas, são consideradas linhas principais. As linhas que não têm os mesmos atributos, em particular o preço, não correspondem a artigos e são eliminadas. A marcação é feita através da associação de um *token* (palavra) e organizados em linhas e colunas. Estuda-se a estrutura de cada artigo e determina-se o modelo para a estrutura mais frequente. Houve, contudo, um cuidado especial em tornar este método genérico para se adaptar a qualquer estrutura de uma nova factura. Foram testados 1704 artigos de 276 facturas. Na Tabela 6 temos o detalhe da informação reconhecida:

Headings	Recognized	Present	Percentage
Code	180	185	97,30%
Designation	262	274	95,62%
Quantity	250	273	91,57%
UP-discount	67	75	89,33%
Discount	31	39	79,49%
Unitary Price	250	274	91,24%
Gencode	52	59	88,13%
Amount	253	274	92,34%

Tabela 6 – Resultados experimentais do método

3.3. *Extracção de informação – World Wide Web*

A extracção de informação da Web é o problema de extrair *items* de informação alvo das páginas da Web (Liu, 2006). A Internet é, sem dúvida, o maior repositório de dados que existe actualmente (Chaudhuri, 2007). A fácil utilização e a rápida difusão de informação contribuíram para este crescimento. De acordo com as estatísticas², o crescimento dos utilizadores da Web durante esta década foi superior a 200% e existem mais de 1 bilião de utilizadores distribuídos por mais de 233 países (Lam, et al., 2008). Na Internet e à “distância de um *click*”, temos disponível qualquer tipo de informação. Isto é possível graças ao constante desenvolvimento na área da pesquisa rápida de informação. Contudo, a heterogeneidade dessa informação e a variedade de formatos disponíveis, limitam as pesquisas feitas através dos motores de busca (Laender, et al., 2002). Estas pesquisas retornam, normalmente, grandes volumes de dados e hiperligações, provocando uma árdua tarefa em sintetizar a informação considerada importante. Numa pesquisa, temos todo o tipo de informação existente, mas é necessária uma análise para verificar o que interessa e um processamento manual no caso de extracção e processamento dos dados retornados. Ferramentas de extracção de dados da Web tentam minimizar esta situação e para este efeito são usadas sofisticadas aplicações de Web Mining que requerem constante actualização e altos custos de manutenção (Chang, et al., 2006). Uma vez tomada consciência de que um repositório de dados onde as pesquisas fossem lentas e difíceis não teria qualquer interesse, o empenho para resolver este problema não tem tido limites e existe um crescente desenvolvimento na área de extracção de dados a partir da *World Wide Web*. Vários trabalhos tentam automatizar a extracção de dados da Internet através da construção de aplicações específicas para esse fim – *Wrappers*.

Os métodos de extracção de dados da Web, designados por *Wrapper Induction*, são sistemas de aprendizagem que extraem regras de um conjunto de exemplos de treino previamente etiquetados (Liu, 2006). A etiquetagem é, normalmente, manual e envolve a marcação dos *items* nas páginas ou exemplos de treino a extrair. Este cenário envolve um trabalho manual intenso acompanhado de muito tempo dispendido e de possíveis

² Miniwatts Marking Group em <http://www.internetworldstats.com/>

erros introduzidos no processo. As regras são depois aplicadas a um conjunto de outras páginas/exemplos com a mesma marcação ou formato. Exemplos que representam excepções são informativos na medida em que são diferentes dos exemplos previamente marcados. A aprendizagem activa – *Active Learning* – é um método que ajuda a identificar, automaticamente, exemplos informativos não etiquetados. Num contexto *Wrapper Induction*, dado um conjunto de exemplos U , não etiquetados, o método trabalha da seguinte forma:

1. Seleccionar, aleatoriamente, um subconjunto L de exemplos não etiquetados de U
2. Etiquetar, manualmente, os exemplo de L e $U = U - L$
3. Criar as regras baseadas em L
4. Aplicar as regras a U para encontrar exemplos informativos L
5. O processo pára se $L = \emptyset$. Caso contrário, voltamos ao passo 2.

Fig. 10 – Fases da aprendizagem activa

O método proposto para encontrar exemplos informativos, descritos no passo 4, é o *Co-Testing* (Muslea, et al., 2006). O *Co-Testing* explora o facto de, frequentemente, existirem vários métodos de extrair o mesmo *item*. No acto da extracção do item, se os diferentes métodos concordarem, é muito provável que a extracção esteja correcta e se não concordarem, este exemplo é seleccionado para que seja etiquetado.

Estudo comparativos de aplicativos para extracção de dados têm sido feitos (Laender, et al., 2002) a fim de concluir qual a performance de cada um e qual o mais adequado em cada situação conforme objectivo proposto. Critérios como o grau de automatização, facilidade de utilização e se tem interface gráfica para utilização, se o output é XML, se os conteúdos aceites são dados ou texto semi-estruturados, se lê outros formatos que não HTML e suporta objectos complexo são avaliados após classificação dos aplicativos por categorias. Podemos ter um resumo do resultado da comparação das diversas ferramentas na tabela seguinte:

Tools		Degree of Automation	Support for Complex Objects	GUI	XML Output	Support for Non-HTML Sources	Type of Page Contents
Languages	Minerva	Manual	Coding	No	Yes	Partial	SD
	TSIMMIS	Manual	Coding	No	No	Partial	SD
	Web-OQL	Manual	Coding	No	No	None	SD
HTML-aware	W4F	Semi-Automatic	Coding	Yes	Yes	None	SD
	XWRAP	Automatic	Yes	Yes	Yes	None	SD
	RoadRunner	Automatic	Yes	Yes	No	None	SD
NLP-based	WHISK	Semi-Automatic	No	Yes	No	Full	ST
	RAPIER	Semi-Automatic	No	Yes	No	Full	ST
	SRV	Semi-Automatic	No	Yes	No	Full	ST
Induction	WIEN	Semi-Automatic	No	Yes	No	Partial	SD
	SoftMealy	Semi-Automatic	Partial	Yes	No	Partial	SD
	STALKER	Semi-Automatic	Yes	Yes	No	Partial	SD
Modeling-based	NoDoSE	Semi-Automatic	Yes	Yes	Yes	Partial	SD
	DEByE	Semi-Automatic	Yes	Yes	Yes	Partial	SD
Ontology-based	BYU	Manual	Coding	Yes	No	Full	ST/SD

Tabela 7 – Resumo da análise comparativa

Podemos ter uma perspectiva gráfica dos resultados na figura seguinte:

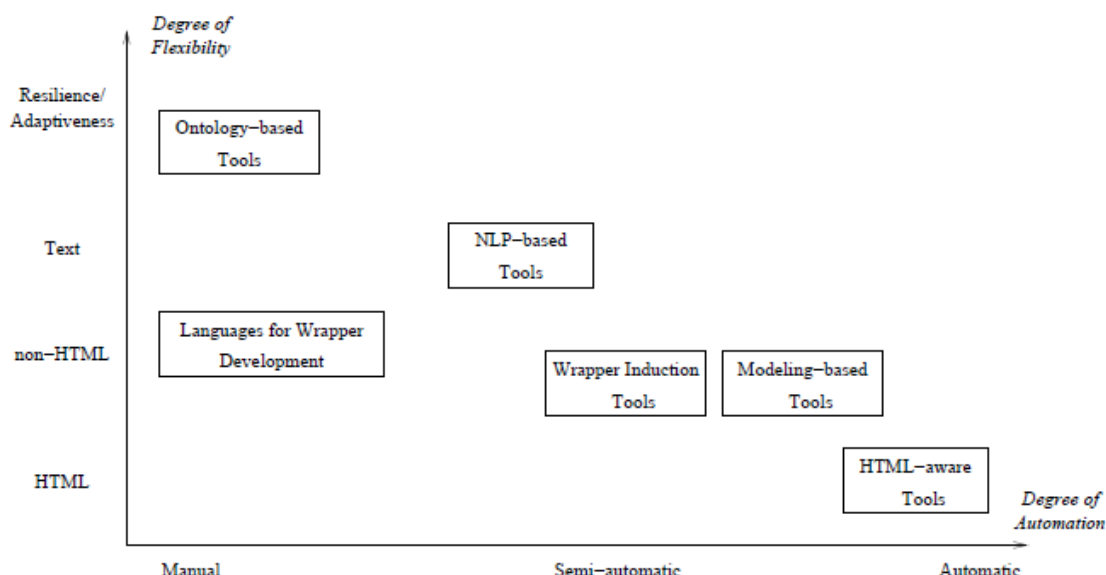


Gráfico 5 – Perspectiva gráfica da análise comparativa

Verificamos que o grau de automatismo decorre de forma inversa ao grau e flexibilidade, isto é, o grau de automatismo é maior no caso de informação formatada e, ao contrário, as intervenções manuais fomentam a flexibilidade e capacidade de adaptação dos aplicativos. Isto explica o facto de que aplicativos que apresentam um grau elevado de automatismo são implementados com heurísticas baseadas em hipóteses sobre as características das páginas alvo. Por outro lado, não formular hipóteses e a adaptação à estrutura das páginas diminuem a flexibilidade.

Noutro estudo (Chang, et al., 2006) foi feita a comparação dos sistemas de extracção de informação nas suas três dimensões. A primeira avalia a dificuldade de aplicação dos sistemas, a segunda compara as técnicas usadas e a terceira avalia o esforço de treinar o sistema e aplicá-lo em diferentes domínios. Verifica-se que, de certa forma, estes critérios estão interligados. Por exemplo, um sistema de extracção de informação que, em certo domínio, use uma boa técnica pode não ter a mesma avaliação quando aplicada a outro domínio. Isto explica porque é que alguns sistemas falham em sites com certas particularidades.

A comparação directa da performance dos diferentes métodos de *Machine Learning* é impossível se estes forem aplicados a diferentes domínios (Turmo, et al., 2006). Têm sido propostos cenários standard em certos domínios, tais como as MUC's e seminários sobre anúncios, que permitem uma avaliação comparativa para um significativo conjunto de autores. Neste contexto, foi feita uma análise comparativa dos métodos aplicados a anúncios de seminários universitários³ publicados online. O problema consistia na extracção da hora de início (*stime*), hora de fim (*etime*), o orador (*speaker*) e o local de realização (*location*). A Tabela 8 mostra os resultados dos métodos de *Machine Learning* aplicados aos anúncios de seminários.

APPROACH	stime	etime	location	speaker
Naive Bayes [Roth and Yih 2001]	98.3	94.6	68.6	35.5
SNOW-IE [Roth and Yih 2001]	99.6	96.3	75.2	73.8
ELIE _{L1} [Finn and Kushmerick 2004]	96.6	87.0	84.8	84.9
ELIE _{L2} [Finn and Kushmerick 2004]	98.5	96.4	86.5	88.5
HMM1 [Freitag and McCallum 1999]	99.1	59.5	83.9	71.1
HMM2 [Freitag and McCallum 2000]	—	—	87.5	76.9
ME ₂ [Chieu and Ng 2002]	99.6	94.2	82.6	72.6
BIEN [Peshkin and Pfeffer 2003]	96.0	98.8	87.1	76.9
SRV [Freitag 1998a]	98.5	77.9	72.2	56.3
RAPIER [Califf 1998]	95.9	94.6	73.4	53.1
EVIUS [Turmo and Rodríguez 2002]	96.1	94.8	73.3	55.3
WHISK [Soderland 1999]	92.6	86.1	66.6	18.3
LP ² [Ciravegna 2001]	99.0	95.5	75.1	77.6

Tabela 8 – Resultados dos métodos aplicados a anúncios de seminários

³ <http://www.cs.cmu.edu/dayne/SeminarAnnouncements>

Os resultados indicam que os métodos estatísticos (indicados no primeiro bloco da tabela) superam os resultados dos métodos de aprendizagem por regras. A investigação na automatização e portabilidade dos métodos tem tido um grande interesse e desencadeado vários trabalhos nesta área. Foi desenvolvido um software em Java – ANDES – com significativos avanços na resolução de problemas de extracção de informação e que disponibiliza uma plataforma para construção de processos para extracção de dados da Web (Myllymaki, 2002). A maioria da informação disponibilizada na Web é acedida através de *browsers* e encontra-se em documentos com formato HTML (*Hypertext Markup Language*), tal como verificamos no Gráfico 3. A chave deste software é que usa tecnologia XML e que é absolutamente essencial para permitir trocas e integração entres sistemas de extracção de informação incompatíveis. O XML é uma linguagem de marcação *standard* e a combinação desta com outras tecnologias, tais como XHTML (*Extensible HTML*) e XSLT (*Extensible Stylesheet Language Transformations*) proporciona o acesso às “profundezas da Web”, isto é, à “*deep Web*”. Outro trabalho inspirado nesta tecnologia (Lam, et al., 2008) desenvolveu um sistema que funciona em duas fases – a fase de pré-processamento e a fase de extracção. Na fase de pré processamento, os documentos são transformados em formato XHTML para contornar as más representações dos documentos HTML. Depois são aplicados os adequados processos de treino para obtenção de regras necessárias à fase de extracção de dados

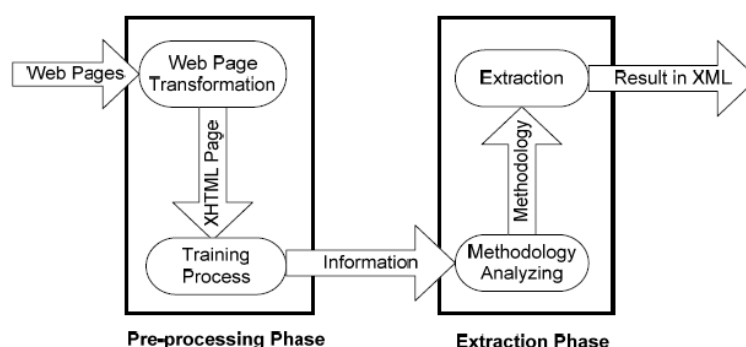


Fig. 11 – Arquitectura do sistema proposto

A Fig. 11 mostra, sucintamente, a arquitectura deste sistema. Contudo, este sistema tem limitações no sentido em que a extracção de dados é feita em cada página individualmente e a informação a extrair pode estar dispersa por vários documentos.

3.4. Adaptabilidade da Extracção de informação

Tradicionalmente, a aquisição de informação era manual e feita em colaboração com especialistas na área de sistemas de extracção de conhecimento (Turmo, et al., 2006) e dos especialistas de negócio da área em questão. A tentativa de automatização destes processos tem sido fundamental para eliminar a dependência que existe dos especialistas na área dos sistemas de extracção de informação. A aplicação dos métodos de extracção de informação é acompanhada por resultados que traduzem alguma certeza. A propagação de erro ao longo deste processos são um problema, uma vez que um pequeno erro pode tornar-se um grande erro à medida que o processo de extracção evolui. Por outro lado, a necessidade de portabilidade dos sistemas de extracção de informação e a sua adaptação a novos cenários implica novos conceitos que podem ir para além das capacidades dos actuais sistemas. Estes dois problemas são a causa das dificuldades no avanço dos estudos das tecnologias de extracção de informação. Nas últimas duas décadas, tem ido feito um esforço no sentido de ultrapassar estas dificuldades e a “adaptabilidade” das tecnologias de extracção de informação tenta aproveitar este investimento.

A aplicação de métodos combinados, num cenário multi-estratégia, ao contrário da tradicional aplicação de um só método, é uma possibilidade para minimizar esta situação. Contudo, isto torna a metodologia menos amigável em certos problemas e introduz desvios, dando origem a novos paradigmas e reformulação das abordagens aos problemas.

3.5. Caso Particular de Documentos em Papel

A extracção de dados de documentos em papel é um caso particular da extracção de dados de documentos. Documentos são suportes de informação cuja apresentação física em papel está fortemente enraizada. Esta informação aparece-nos sob a forma de contratos, orçamentos, publicações periódicas, documentos contabilísticos tais como facturas, recibos, vendas a dinheiro, notas de crédito e débito, declarações fiscais, relatórios, resultados de exames médicos, autorizações e muitos outros. A insegurança e alguma desconfiança que ainda existe nos sistemas de informação ou uma implementação desajustada destes, contribui em larga medida para manter este cenário. Há, ainda, a necessidade de documentos que têm de ser assinados, carimbados, reconhecidos ou autenticados e que de outra forma não são considerados válidos. A desejada era *paper-free* está, aparentemente, longe de atingir. Contudo, a maior parte das vezes, os documentos em papel estão também disponíveis em versão digital. Os documentos em formato digital estão disponíveis na Internet ou Intranets ou em repositórios estruturados de gestão documental. A extracção de dados de documentos em papel exige, numa primeira fase, a transformação destes em documentos digitais. A digitalização dos documentos é a forma de converter os documentos em papel em documentos digitais. Com a digitalização é possível a criação de ficheiros com vários formatos, por exemplo, formato .Pdf, .Tiff ou .Jpeg. Nestes formatos os documentos não são editáveis. No caso da extracção de informação os ficheiros devem ser editáveis para permitir a recolha de dados. Isto é possível através da aplicação de um software de OCR aos ficheiros digitalizados. A conversão de ficheiros não editáveis em ficheiros editáveis torna-os, na maioria das vezes, ficheiros sem estrutura. A conquista da estrutura necessária para recolha de dados pode ser obtida através da aplicação de técnicas disponibilizadas pelas tecnologias XML.

3.5.1. OCR – Optical Character Recognition

Esta tecnologia permite o reconhecimento de caracteres automaticamente através de um mecanismo óptico. Quando uma folha de papel com informação é digitalizada, converte-se um objecto físico num objecto digital, o qual é gravado sob a forma de imagem. A imagem pode ser manipulada como um todo, mas o texto não pode ser manipulado separadamente. Para que isto seja possível, é necessário que o computador reconheça o texto como tal e que o torne possível de manipular como se fosse originalmente escrito em ficheiro de texto. Graficamente e, passo a passo, podemos perceber melhor como se processa o OCR:

Passo 1: O 1º passo é a digitalização dos documentos



Passo 2: O documento é convertido em formato digital sob uma imagem



Passo 3: A tecnologia OCR é aplicada à imagem previamente criada



Passo 4: É criado um ficheiro de texto editável e possível de manipular



Fig. 12 – Etapas da tecnologia base do OCR

Isto é permitido graças ao OCR: reconhece o texto e torna-o editável, permitindo, por exemplo, pesquisas. Contudo, existem vários factores capazes de alterar o desempenho do OCR. Para um resultado tão bom quanto possível, a qualidade do documento inicial é crucial. Se o *input* é mau, o *output* será igualmente mau. A partir dum bom original, o OCR consegue arquivar com 99% de correcção. Desta forma, muitos dos sistemas de OCR tentam combinar técnicas de análise léxica para aumentar o seu desempenho. O OCR disponibiliza técnicas para análise das propriedades dos caracteres e que permitem escolher qual é o carácter mais adequado através de algoritmos matemáticos. Na figura seguinte percebemos melhor este conceito:

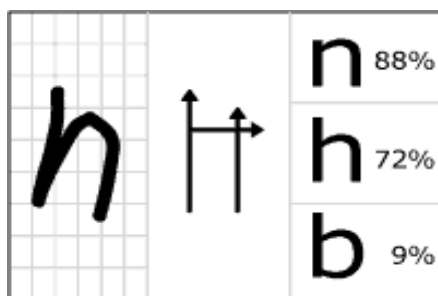


Fig. 13 – Carácter digitalizado vs candidatos a integrar o texto

Neste caso, o OCR identifica o “n” como o mais aceitável e é o melhor candidato a ocupar o lugar no texto. Isto, numa análise do carácter de forma isolada e fora do contexto do resto do texto.

Os contrastes existentes nos documentos, devido a cores de fundo, inclusão de logótipos ou imagens também são fundamentais para um bom resultado do OCR. Texto em preto sob um fundo branco garante 100% de contraste. Alguns sistemas requerem um

contraste mínimo de 50 a 60%. Abaixo disto o OCR não se mostra capaz de identificar o texto. Na Fig. 14, temos contrastes entre as cores de fundo e as cores do texto quer em branco quer em preto e é significativamente mais elevado quando temos uma cor de fundo clara com texto em cor escura.

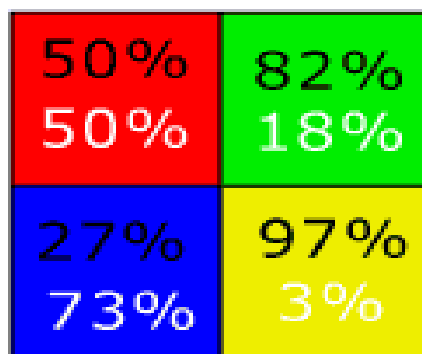


Fig. 14 – Contrastes entre cores de fundo e texto

Todos trabalhos de extracção de informação a partir de documentos em papel iniciam com a sua digitalização e respectiva aplicação do OCR. Desta forma, o OCR determina a qualidade dos *inputs* para aplicação dos métodos. A correcta identificação dos caracteres e sua formatação é essencial. Casos há em que esta identificação não é correcta o que provoca a inclusão de erros no ficheiro a estudar. A estes erros designamos por “ruído”. O ruído introduzido pode ter várias origens: caracteres trocados por outros semelhantes, leituras mal efectuadas devido a fracos contrastes, escrita manual (o OCR não é adequado à identificação de texto com letra escrita à mão (Chaudhuri, 2007)) e estrutura difícil de identificar. A estrutura do documentos e o tipo de letra usado tem também uma forte influência por ruído introduzido pelo OCR (Chaudhuri, 2007). O desempenho do OCR pode ser responsável pelo aumento da dificuldade ao problema inicial.

Sob este pretexto, grande parte dos trabalhos sobre extracção de informação concentram-se no resultado do OCR e vários métodos têm sido desenvolvidos para melhorar o seu desempenho. Grande parte da investigação existente nesta área tenta dotar o OCR com capacidades de análise léxica, semântica e gramatical para que, percebendo o contexto em que se insere, seja menos falível. A análise léxica prende-se com a composição das palavras e das letras que a constituem. Impede, por exemplo, a

troca da letra “O” por um “0” (zero) ainda que muito semelhantes em alguns tipos de letra. Incluída numa palavra não existe um zero, mas convém perceber se se trata de uma palavra ou não. A análise semântica e gramatical procura compreender o sentido em que a palavra se insere e perante várias alternativas escolha a mais adequada. Há, também, trabalho efectuado para tratamento e compreensão de palavras ou expressões raras ou ambíguas. Em resumo, tenta-se usar um conhecimento *à priori* sobre a linguagem e sobre relação e associação de palavras promovendo, desta forma, a precisão da informação extraída.

3.5.2. XML – Extensible Markup Language

A conversão de ficheiros digitais não editáveis em ficheiros editáveis, torna-os sem estrutura. A formatação destes ficheiros é essencial para possibilitar uma rápida e flexível recolha de dados. Esta formatação pode ser feita através de uma linguagem de marcação standard – o XML.

O XML é uma linguagem de marcação que permite uma formatação estruturada de dados através de marcas – *tags* – atribuídas aos dados. O XML veio colmatar algumas falhas que não eram resolvidas pelo formato HTML (HyperText Markup Language), também esta uma linguagem de marcação. Por um lado o XML é mais rígido e não tolera erros de sintaxe, mas a maior inovação prende-se com o facto de não ter um limite ou uma predefinição quanto ao número de *tags* utilizados e quanto à sua designação, ou seja, desde que se definam correctamente, podem ser usados quantos os necessários e com qualquer nome. São estas características que lhe atribuem a qualificação de “*Extensible*”, para além da simplicidade, legibilidade e portabilidade que tem. A representação e a consulta de dados semi-estruturados é, hoje em dia, um desafio para a comunidade científica e o XML tornou-se, recentemente, um novo padrão para representação e troca de dados e, provavelmente, tornar-se-á um formato universal. Aproveitando as potencialidades do XML, muitas linguagens de programação estão a criar variantes a fim de proporcionar um melhor desempenho das aplicações desenvolvidas sob essa linguagem.

As potencialidades do XML são fundamentais para a extracção de informação de documentos em papel. Os ficheiros ASCII criados através do OCR não têm estrutura e a sua estruturação permite uma recolha facilitada dos dados.

4. Extracção de Elementos de Facturas: Caso de estudo

“If I had one hour to save the world, I would spend fifty-five minutes defining the problem and only five minutes finding the solution.”

Albert Einstein

Este trabalho tem o seu início num conjunto de facturas em papel. Da informação que as empresas necessitam, a comum a todas elas e também a estrutura base é a facturação. As exigências legais obrigam à troca de vários tipos de documentos em papel (Belaïd, et al., 2004) para oficializar as transacções efectuadas entre empresas qualquer que seja o ramo de actividade a que se dedicam. Os dados desses documentos têm de ser processados rapidamente e por vários motivos: definição de estratégias comerciais, apresentação de contas e resultados aos sócios ou accionistas, pagamento a fornecedores, cumprimento de prazos legais para preenchimento e entrega de declarações fiscais, facturação aos clientes e ainda por questões logísticas, tais como falta de espaço para manter em cima da secretária tudo o que está pendente de lançar. Interessa, por isso, processar estes dados de forma rápida. Os registos físicos serão guardados em arquivos durante vários anos consoante o que estiver legalmente definido.

É de referir que os erros ao longo deste processo atrasam o desempenho das empresas e a sua obrigatória correcção torna este circuito ainda mais complexo e moroso. Dada a delicadeza desta informação, nestes casos, não pode haver lugar a enganos e tudo terá de ser corrigido a fim de não haver qualquer problema futuro.

Duma forma geral, estes documentos têm uma estrutura similar e a informação apresentada é do mesmo tipo (Belaïd, et al., 2004). Tanto esta estrutura como os elementos apresentados resultam, em primeira-mão, de imposições legais. Esta estrutura é, em traços muito gerais, composta por várias regiões (Chaudhuri, 2007):

- Uma delas contém dados sobre o fornecedor, isto é, nome do fornecedor, morada e contactos e número de identificação fiscal. Podemos ter outras informações tais como o montante do capital social, o registo na conservatória e

uma referência ou pequena publicidade aos serviços prestados por esse fornecedor. Neste caso o interesse incide sobre a primeira parte.

- Informação sobre o tipo de documento, isto é, se corresponde a uma factura, venda a dinheiro ou outro. O número e data do documento podem estar ou não associados à região que contém os dados do fornecedor pelo que vamos considerar como uma região distinta. Podemos ainda ter informação sobre o modo e condições de pagamento.
- Outra das regiões contém os dados do cliente. À semelhança da região que contém dados do fornecedor, também teremos o nome do cliente, a morada e número de identificação fiscal. Por norma não temos mais dados nesta região.
- Os documentos disponibilizam uma zona com a descrição detalhada dos artigos. Teremos, então, referência e designação do artigo, quantidade adquirida pelo cliente e valor unitário, a taxa de imposto a aplicar, desconto e valor total. Esta informação aparece linha a linha correspondente a cada tipo de artigo adquirido. Visualmente, estes dados aparecem, normalmente, sob a forma de uma tabela onde cada coluna corresponde a um atributo. Falta ainda referir que adjacente a esta informação temos ainda os dados sobre o total ilíquido, os descontos globais ao documento, as taxas a aplicar, outras despesas globais, tais como, despesas de cobranças ou de transportes e o total líquido do documento.
- Por último, podemos ainda ter um rodapé com informações diversas, tais como, contactos do fornecedor, publicidade, registo da conservatória e outras.

Podemos verificar estas diversas regiões através dum exemplo da Fig. 15:

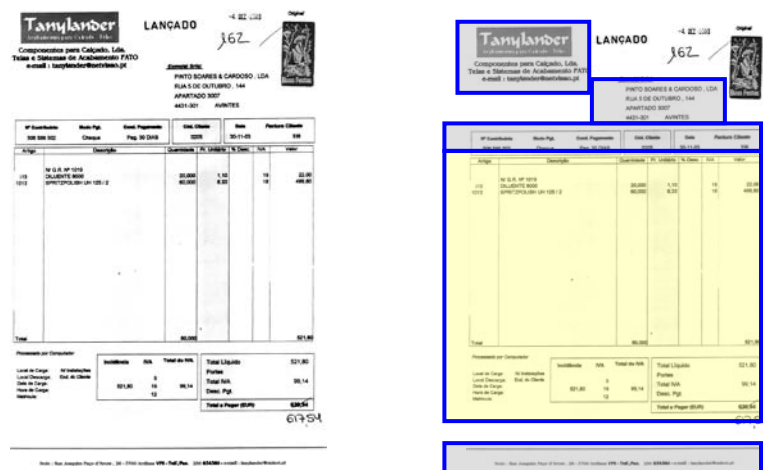


Fig. 15 – Exemplo de uma factura e sua correspondente segmentação em regiões

O seu registo contabilístico implica identificar qual a informação útil exigida para efeitos fiscais e de gestão. Da divisão dos documentos em regiões, resulta, quase intuitivamente, quais aos elementos a recolher. Sob o ponto de vista do cliente, interessa saber o seguinte:

- Confirmar os dados do cliente, isto é, verificar se o documento lhe é destinado.
- De que fornecedor lhe chega este documento. Neste caso podemos ter fornecedores habituais ou um novo fornecedor.
- Tipo, data e número do documento. Podemos ter procedimentos diferentes para diferentes tipos de documentos.
- Quais os artigos adquiridos, quantidades e valores, uma vez que tem implicações quer nos *stocks* quer nos preços de custo.
- Qual o valor total do documento.

Automatizar a recolha desta informação seria um avanço, na medida em que acelerava o processo de registo dos dados e minimizava possíveis erros ao longo desse processo de registo. Desta forma, o desafio proposto para este trabalho, é a aplicação de **técnicas de Data Mining num problema de extracção de elementos de documentos comerciais**, com a perspectiva de tornar este processo mais simples, célere e menos falível. O objectivo é aproveitar o facto de estes documentos terem uma estrutura tipo semelhante e os elementos a recolher serem comuns em todos eles.

4.1. Definição do Problema

Manter a controlo da facturação, através do constante registo contabilístico dos documentos comerciais, é um objectivo de todas as empresas e é parte integrante da sua actividade. Compromissos legais e sociais obrigam esta situação. Para responder positivamente a esta exigência, as empresas apoiam-se em recursos humanos com diversos conhecimentos na área financeira e da actividade da empresa. O recurso a sistemas de informação fiáveis, adequados à dimensão da empresa e ao seu ramo de actividade, é também essencial para cumprir estas obrigações. O registo das facturas é um processo manual e, por isso, requer um elevado número de recursos humanos e muito tempo de dedicação a esta tarefa. Também já foi referido que os erros que resultam deste processo impedem o bom desempenho das empresas e atrasam consideravelmente todo este processo. Isto traduz-se em elevados custos para as empresas.

Aqui reside o problema: elevados custos administrativos para identificar e registar todos os elementos dos documentos comerciais necessários a uma contabilidade organizada e actualizada. A identificação rápida e automática dos elementos em documentos comerciais permitiria uma redução de tempo e de recursos humanos necessários a esta tarefa e, consequentemente, a redução dos custos deste serviço.

Em termos de Data Mining, o problema é classificar os elementos recolhidos de documentos comerciais como sendo informação relevante. No entanto os documentos contêm muitos elementos importantes os quais obrigariam, em princípio, a abordagens distintas para cada um deles. Optámos por dirigir o trabalho para um único elemento. No caso do nosso estudo, a informação a encontrar é o “número de documento” e queremos classificar se um determinado elemento é ou não “número de documento”.

As medidas de avaliação dos resultados que vamos considerar são o *Recall*, o *Precision*, o *False Positive Rate* (FPT), o *F1-Measure* e o *Average Accuracy*. Vamos analisar a Matriz de Confusão, as Curvas ROC e respectivas AUC. No problema em estudo partimos do princípio que iniciamos com um custo máximo porque, em contexto real de

trabalho, a descoberta do número de documento é feita manualmente. Desta forma, como critério de sucesso, vamos considerar o melhor modelo aquele que permita uma maior redução de custos, isto é, um maior ganho.

4.2. Digitalização dos Documentos

A primeira etapa deste processo é conseguir recolher as palavras, sob a forma de texto, dos documentos em papel. O objectivo é construir um conjunto de dados, com as palavras dos documentos, que servirão de base à construção dos modelos para identificar o “número de documento”. Esta recolha de informação começa com a digitalização dos documentos. Dos documentos a digitalizar, são incluídos apenas os que são processados por computador. Um simples processo de digitalização não é suficiente para conseguirmos a informação desejada. Este apenas converte os documentos em papel em documentos digitais com formato *.pdf* ou em forma de imagem e que são pouco acessíveis para o objectivo proposto, ou seja, extrair informação. A partir da digitalização do documento, a ideia é conseguir obter os caracteres nele contidos através de um formato digital acessível. Neste caso, a melhor forma é convertê-lo num ficheiro ASCII⁴, cujo formato está normalmente associado aos ficheiros com extensão *txt*.

A conversão em ASCII é feita através da aplicação de um programa de reconhecimento óptico de caracteres – OCR. Desta forma, para cada factura é produzido um ficheiro ASCII com todas as palavras reconhecidas pelo OCR. Na Fig. 16, podemos verificar o resultado obtido pelo OCR após a digitalização de uma factura.

⁴ ASCII são as iniciais de **American Standard Code for Information Interchange** e disponibiliza um conjunto de códigos para o computador representar os caracteres disponibilizados pelo teclado, isto é, números, letras, pontuação e alguns caracteres especiais.

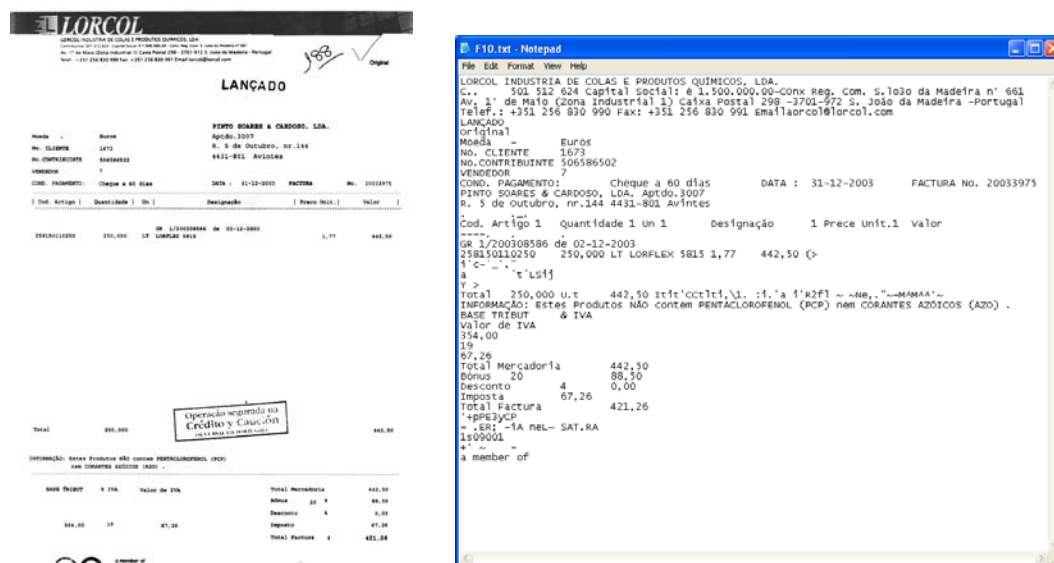


Fig. 16 – Exemplo de uma factura e sua conversão em formato ASCII

Verificamos que o ficheiro criado não tem qualquer semelhança com o original da factura em papel. É um texto desarranjado, confuso e sem uma estrutura que nos permita a obtenção de dados numa forma directa. É a partir deste ponto que podemos formatar e trabalhar a texto obtido, que nos é dada no ficheiro ASCII, e traduzi-la nos dados que queremos extrair. A formatação dos ficheiros de texto passa pela inclusão de *tags* no texto, recorrendo às capacidades e versatilidade do XML e cujo objectivo é identificar os dados a extrair. Esta formatação é manual e desta forma, passo a passo, vamos conseguir chegar à informação pretendida. Uma vez identificados e marcados os elementos do texto, podemos, através de um procedimento, construir um conjunto de dados que servirá de base à construção dos modelos. O conjunto a construir deve conter palavras candidatas a número de documento e suas palavras vizinha e será apresentado com mais detalhe na secção 4.3.1

4.3. Pré-processamento de Dados

De um grupo inicial de documentos em papel, foram seleccionados 61 documentos processados por computador. Destes 61 documentos separamos, aleatoriamente, 46 documentos para construção do conjunto de dados que servirá de base à construção dos modelos e reservamos 15 documentos para construção de outro conjunto para posterior aplicação do modelo que for eleito como o melhor segundo os critérios já definidos.

Da digitalização dos documentos e respectiva aplicação do OCR resultaram 46 ficheiros de texto identificados por F01.txt, F02.txt, F03.txt, ..., F46.txt, um para cada factura em papel. Podemos analisar mais um exemplo e comparar outra factura e o respectivo ficheiro ASCII.

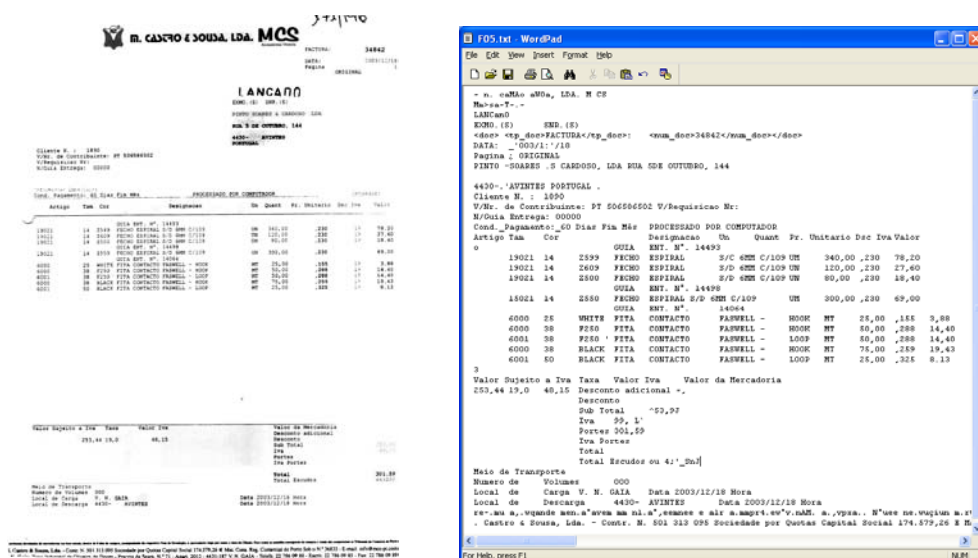


Fig. 17 – Outro exemplo de uma factura e sua conversão em formato ASCII

Em cada um destes ficheiros é necessário incluir os *tags* de XML, para marcar os dados a recolher.

Foi definida que a informação a marcar seria:

- Fornecedor – nome e nº de contribuinte
- Documento – tipo, data e número de documento
- Cliente – nº de contribuinte

- Artigos – para cada linha temos a referência e descrição do artigo, unidade de medida, quantidade, valor unitário, valor total
- Valor total do documento

O ficheiro ASCII criado é formatado através da inclusão de *tags* tal como identificado na figura seguinte:

```

F05.txt - WordPad
File Edit View Insert Format Help

<doc><tp_doc>FACTURA</tp_doc> <num_doc>34842</num_doc></doc>
DATA: <date>_003/12/18</date>
Pagina 1 ORIGINAL
PINTO-SOARES, S. CARDOSO, LDA RUA 5 DE OUTUBRO, 144
4430- AVINTES PORTUGAL
Cliente N.º: 1890
V/Nr. de Contribuinte: <nif_cliente>PT 506586502</nif_cliente> V/Requisicao Nr:
N/Guia Entrega: 00000
Cond. Pagamento: 60 Dias Fim Mês
PROCESSADO POR COMPUTADOR
Artigo Tam Cor Designacao Un Quant Pr Unitario Dsc Iva Valor
o
    <artigos><line> <item_id>19021</item_id> 14 Z599 <item_desc>FECHO ESPIRAL S/C 6MM C/109 </item_desc><units>UM</units> <qty>120,00</qty>
    <item_id>19021</item_id> 14 Z609 <item_desc>FECHO ESPIRAL S/D 6MM C/109 </item_desc><units>UN</units> <qty>80,00</qty>
    <item_id>19021</item_id> 14 Z500 <item_desc>FECHO ESPIRAL S/D 6MM C/109 </item_desc><units>UN</units> <qty>80,00</qty>
    <item_id>15021</item_id> 14 Z550 <item_desc>FECHO ESPIRAL S/D 6MM C/109 </item_desc><units>UM</units> <qty>300,00</qty>
    <item_id>6000</item_id> 25 <item_desc>WHITE FITA CONTACTO FASWELL - HOOK </item_desc><units>MT</units> <qty>25,00</qty>
    <item_id>6000</item_id> 38 F250 <item_desc>FITA CONTACTO FASWELL - HOOK </item_desc><units>MT</units> <qty>50,00</qty>
    <item_id>6001</item_id> 38 F250 <item_desc>FITA CONTACTO FASWELL - LOOP </item_desc><units>MT</units> <qty>50,00</qty>
    <item_id>6000</item_id> 38 <item_desc>BLACK FITA CONTACTO FASWELL - HOOK </item_desc><units>MT</units> <qty>75,00</qty>
    <item_id>6001</item_id> 50 <item_desc>BLACK FITA CONTACTO FASWELL - LOOP </item_desc><units>MT</units> <qty>25,00</qty>
3
Valor Sujeito a Iva Taxa Valor Iva Valor da Mercadoria
253,44 19,0 48,15 Desconto adicional -,
Desconto
Sub Total 253,90
Iva 99, L
Portes <total_value>301,59</total_value>
Iva Portes
Total
Total Escudos ou 4, SnJ

Meio de Transporte
Numero de Volumes 000
Local de Carga V. N. GAIA Data 2003/12/18 Hora
Local de Descarga 4430- AVINTES Data 2003/12/18 Hora
re- mu a, wqande men a'avem mm nl a°, eemnee e alr a mmpir4 ew°v nAM. a, ypxa. N°uee ne wuqun m rt mn. 14. w emb ° n me ° c ~ ~ e ° tr. e n, ° a... ymde. mn{ nu. a. mmmv.
. <fornecedor>Castro & Sousa, Lda.</fornecedor> - Contr. N. <nif_fornecedor>501 313 095</nif_fornecedor> Sociedade por Quotas Capital Social 174.579,26 E Mat. Cons.
For Help, press F1
NUM
    
```

Fig. 18 – Ficheiro marcado com os *tags* do XML

Desta forma, manualmente e ficheiro a ficheiro, são marcados os elementos a extrair e os ficheiros, inicialmente sem estrutura, têm agora a estrutura adequada ao trabalho a desenvolver. Os *items* podem ser marcados individualmente ou podem ser agrupados por tipo para uma melhor identificação dos dados. Esquemáticamente, temos uma estrutura em árvore, representada na Fig. 19:

```

<factura>
  <fornecedor>
    <nome>
    <nif_fornecedor>
  <doc>
    <tp_doc>
    <num_doc>
    <date>
    <nif_cliente>
  <artigos>
    <line>
      <item_id>
      <item_desc>
      <units>
      <qtd>
      <unit_value>
      <value>
    <total_value>

```

Fig. 19 – Estrutura em árvore das *tags* XML

A marcação rigorosa com esta estrutura em árvore nem sempre é possível. Por um lado temos casos em que a informação não existe no documento original ou não foi reconhecida pelo OCR. Por exemplo, para o caso em que as facturas se referem a prestações de serviços, as quantidades e valores unitários não existem registando-se apenas os valores totais. Por outro lado, a informação do mesmo tipo pode aparecer separada não sendo possível a inclusão de *tags* que agrupam essa informação. Essa separação da informação pode ser de origem ou pode resultar da aplicação do OCR não manter a estrutura original da informação não suportada pelos ficheiros em formato ASCII.

Por exemplo, no ficheiro ilustrado na Fig. 20, temos a informação do fornecedor intercalada com a informação do tipo de documento, não permitindo a marcação inicialmente prevista.

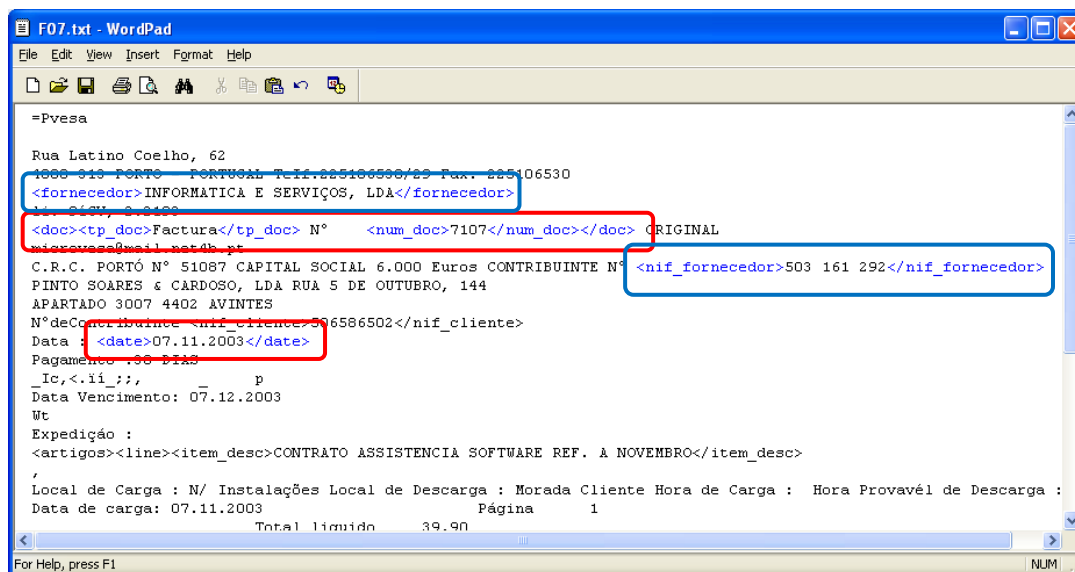


Fig. 20 – Ficheiro com marcação intercalada dos tags do XML

Desta forma, quando não é possível, é dada preferência à correcta marcação dos elementos individuais.

Após a marcação, é efectuada uma avaliação dos *tags* existentes, para decidir qual dos elementos vai ser o alvo do nosso estudo. Para cada ficheiro foi identificada a presença de cada *tag* e feita uma contagem. O objectivo é saber em quanto ficheiros aparece cada *tag* e qual o total de vezes que este aparece. No gráfico seguinte é feita essa comparação:

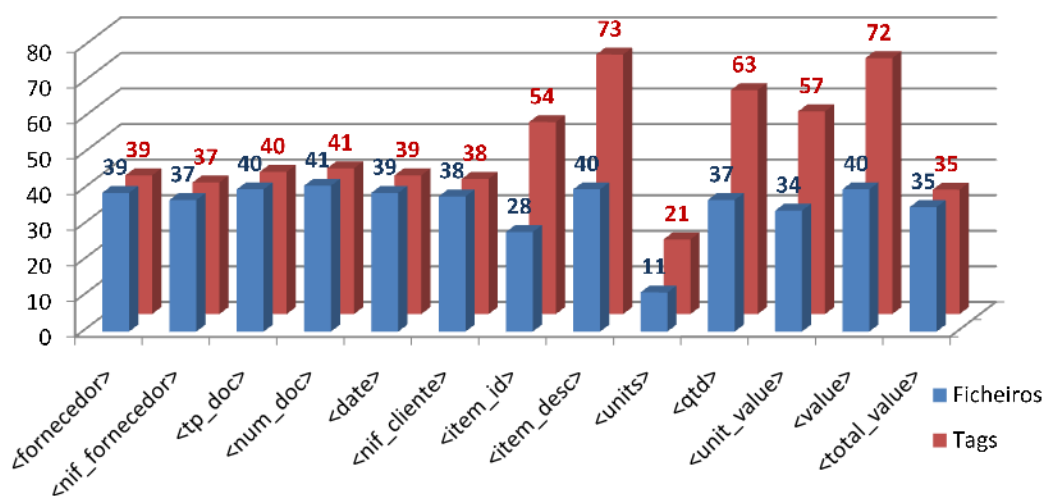


Gráfico 6 – Comparação tags por ficheiro e total de tags

Nesta comparação conseguimos perceber que há ficheiros em que os *tags* aparecem uma vez, enquanto outros aparecem várias vezes. O *tag* `<item_desc>` que corresponde à identificação da descrição do artigo é o que aparece mais vezes. São 73 vezes no total, distribuídos por 40 ficheiros e o que corresponde a uma média de 1,83 descrições de artigos por ficheiro. O *tag* com menos representação é o que corresponde às unidades de medida que aparece 21 vezes em 11 ficheiros. O *tag* `<num_doc>`, que identifica o nº do documento, é o que aparece em mais ficheiros. Sabemos que o nº do documento aparece uma vez em cada ficheiro e neste caso está em 41 ficheiros dos 46 inicialmente digitalizados. Para o estudo em questão, interessa-nos considerar o maior nº de ficheiros para termos um conjunto de exemplos mais alargado.

No gráfico seguinte temos, em termos de percentagens, a presença dos *tags* relativamente aos 46 ficheiros iniciais:

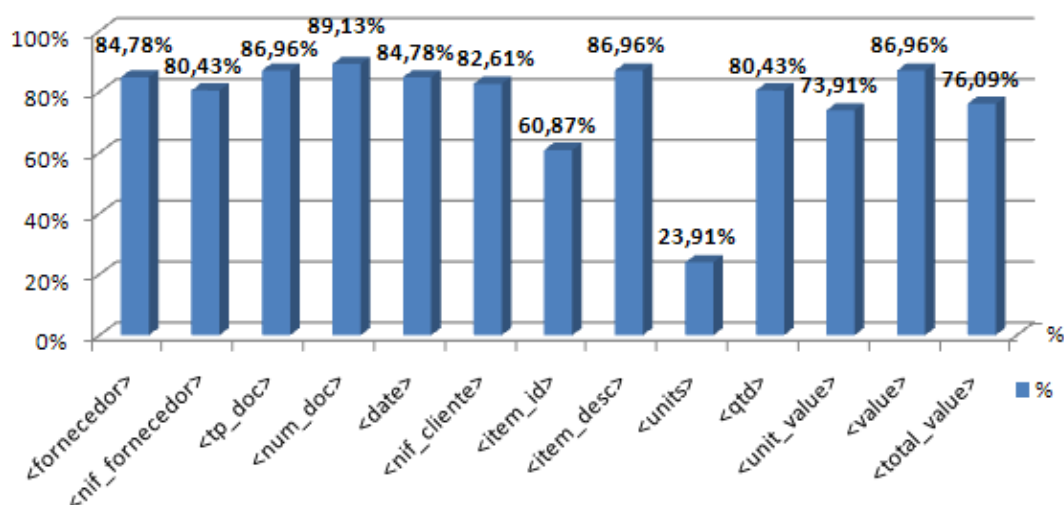


Gráfico 7 – Percentagem de Tags no total dos ficheiros

O *tag* `<num_doc>` aparece em 89,13% dos ficheiros. Uma vez que queremos estudar o maior nº de ficheiros possível, encontramos o elemento sobre o qual vamos focar o nosso estudo. Como já verificamos, temos 41 ficheiros onde o “número de documento” foi marcado com a *tag* `<num_doc>` e são esses 41 ficheiros que vamos usar.

4.3.1. Extracção do Número do Documento

Neste momento temos definido qual o elemento sobre o qual vai incidir este trabalho. O número do documento é usado para o identificar o próprio documento. Do ponto de vista de quem o emite, é um identificador único para esse tipo de documento. Do ponto de vista de quem o recebe, o número de documento associado ao fornecedor, tipo e data de documento identificam de forma única a transacção em causa.

No documento, o número aparece, em geral, no cabeçalho deste, assumindo uma posição de destaque. Normalmente o nº de documento é numérico, mas há casos em que a sua representação envolve caracteres especiais tais como as barras (/), asteriscos (*), menos (-), entre outros.

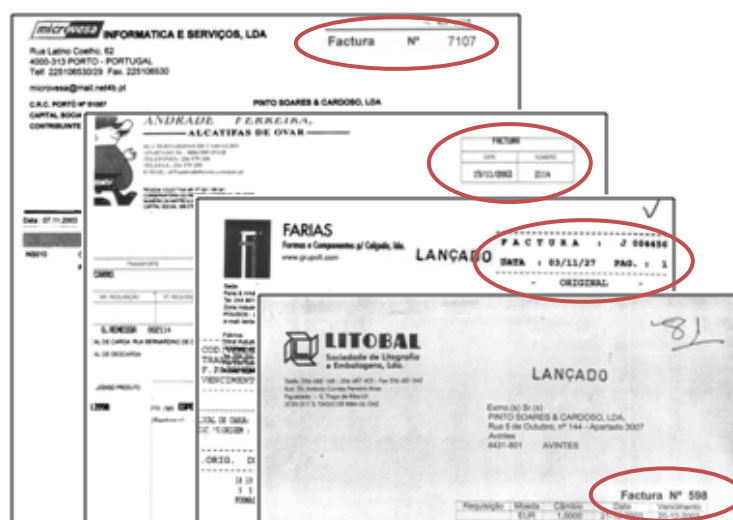


Fig. 21 – Cabeçalho de facturas

Nos ficheiros de texto com as marcações, o número do documento encontra-se entre as tags que o identifica, isto é, entre <num_doc> e </num_doc>.

4.3.2. Construção do Conjunto de Dados

O desenvolvimento do nosso trabalho vai basear-se na análise do contexto que envolve o número do documento, ou seja, do que antecede e segue este identificador. A ideia é, para cada ficheiro, pesquisar palavras⁵ que sejam candidatas a número de documento e recolher as *strings* vizinhas (Sitter, et al., 2003), isto é, as que aparecem imediatamente antes e as que aparecem imediatamente depois.

Com este intuito, o conteúdo dos ficheiros de texto foram inseridos numa base de dados. A estrutura da tabela que vai guardar este conteúdo é muito simples: tem uma coluna para identificação do ficheiro e outra coluna com o texto. Foi criado um procedimento⁶ para extrair de cada texto as palavras candidatas a número de documento e respectivas *strings* vizinhas. O algoritmo deste procedimento está descrito na Fig. 22.

O procedimento foi criado de forma a ser usado de forma dinâmica. Podemos extrair as palavras de um ficheiro ou, em alternativa, extrair em todos simultaneamente. O nº de palavras vizinhas a extrair é variável e a *tag* é parametrizável, neste caso estamos a pesquisar o *<num_doc>*, mas pode ser escolhida outra *tag* se necessário. Uma vez que o número de documento aparece, em geral, no cabeçalho, foi decidido que as palavras a extrair seriam as palavras com posição no texto inferior a 1000. A ideia é diminuir o volume de dados que não têm informação útil, uma vez que o detalhe da factura onde estão representados os artigos e o rodapé da factura contribuem com exemplos negativos para o conjunto de dados. O problema de conjuntos de dados com classes não-balanceadas (Sitter, et al., 2003) não é novo e já foi referido anteriormente assim como algumas estratégias para resolver a grande diferença entre as instâncias positivas e negativas (Hu, et al., 2005).

O algoritmo para recolha das palavras é descrito na Fig. 22. Como dados de entrada para este algoritmo são indicados o ficheiro para recolha dos registos para construção do

⁵ Vamos considerar que uma palavra candidata a número de documento, é uma sequência de caracteres, com pelo menos um carácter numérico.

⁶ Ver Scripts em Anexo

conjunto de dados, a *tag* referente ao elemento que estamos a recolher os dados, neste caso `<num_doc>`, o nº de palavras vizinhas anteriores e o nº de palavras vizinhas posteriores que queremos recolher.

1. Para o texto de cada ficheiro
2. Encontra próxima string
3. Se string é uma palavra
4. Guarda string (KW)
5. Pesquisa e guarda pQP strings vizinhas anteriores
6. Pesquisa e guarda pQS strings vizinhas posteriores
7. Se KW marcada com tag pTag
8. É um exemplo positivo (ex=1)
9. Senão
10. É um exemplo negativo (ex=0)
11. Senão
12. Ignora string
13. Repetir os passos de 2. A 13. enquanto houver strings com posição menor que 1000

Fig. 22 – Algoritmo para recolha das palavras e *strings* vizinhas

A aplicação deste algoritmo preenche uma tabela de dados com a estrutura apresentada na Tabela 10. Para todos os documentos foi feita a pesquisa da *tag* `<num_doc>` e de 3 *strings* vizinhas anteriores e 3 *strings* vizinhas posteriores (Sitter, et al., 2003).

Foi feita uma avaliação à qualidade dos dados extraídos em comparação com os documentos originais em papel. Verificamos que os dados eram de má qualidade: havia dados em falta e as palavras estavam mal construídas tanto por omissão ou troca de caracteres como por inclusão de caracteres que não pertenciam às palavras. A inclusão de ruído nos dados foi resultado da aplicação do OCR na conversão do ficheiro digitalizado para ASCII. O OCR não teve, neste caso, um desempenho eficiente por motivos de má qualidade dos ficheiros originais, pela presença de contrastes devido a cores de fundo e de logótipos, a existência de carimbos e notas escritas manualmente e texto escrito na lombada do documento e que por isso aparece na vertical, em paralelo ao texto escrito na horizontal.

Apresentamos alguns casos:

Palavra(s) no documento	Conversão OCR
Factura	.FaCiUfa
CONTRIBUINTE	CQ`RPIBUINIE
de Outubro	déÓutubio:
PINTO SOARES & CARDOSO	.PINT030ARS3.&CARDOS
20-12-2003	20-12-2003
Telefs. 256 685 168	TeleCz:256685
518 * Apartado	510Apodado
PESSOA COLECTIVANR PT 501 195 041	PESSOACCLECTNANq.P!\$01195M1

Tabela 9 – Exemplos de conversão do OCR

Na Tabela 9, temos alguns exemplos do ruído introduzido com a aplicação do OCR aos ficheiros digitalizados. Este ruído afecta fortemente o nosso conjunto de dados e resolver este problema não é uma tarefa simples e não está no âmbito deste trabalho.

Um OCR perfeito que converta o texto dos documentos tal como é humanamente compreensível não existe. No entanto existem no mercado OCRs mais eficientes e capazes de gerar ficheiros com menos ruído. Não era objectivo deste trabalho investir na pesquisa e aquisição dum OCR e por este motivo assumimos que, na falta de um OCR perfeito que crie os documentos de forma perfeita, os dados deveriam ser corrigidos para aumentar o nível de qualidade destes. Partimos para uma árdua tarefa de corrigir manualmente os dados extraídos e acrescentar os que foram omitidos, simulando a aplicação de um OCR perfeito capaz de ter interpretação humana e de traduzir correctamente o conteúdo dos documentos.

A preparação dos dados foi feita com recurso ao PL/SQL, linguagem inata disponível no Oracle Database, através do Oracle SQL*Plus ou de outra interface gráfica que permita o acesso à base de dados. O Oracle SQL*Plus é um produto da família do Oracle Database que permite o acesso aos dados e sua manutenção. No entanto e neste caso, optamos por uma interface gráfica mais amigável – Toad V6.3. Esta interface é um *freeware* disponibilizado pela Quest Software. O Toad é uma plataforma de desenvolvimento que permite a administração, acesso e manipulação dos dados em diversas bases de dados. Os *packages* e os *queries* para pré-processamento dos dados

foram desenvolvidos em PL/SQL a partir desta plataforma. Apesar das possibilidades de pré-processamento de dados disponíveis no Oracle Data Miner, estas mostraram-se insuficientes para pré-processamento dos dados do problema em questão.

Do tratamento de dados ficamos com um conjunto cuja descrição e tipo estão resumidos na Tabela 10. O atributo “KW” contém a palavra candidata a número de documento e os atributos “A1”, “A2” e “A3” registam as *strings* imediatamente anteriores a “KW” e “D1”, “D2” e “D3” as *strings* imediatamente posteriores. “EX” é o atributo alvo, isto é, é o atributo que contém os valores da classe, 1 ou 0. O atributo “TAG_FHC” identifica a *tag* em estudo, neste caso a *tag* <num_doc>. “FCH” identifica o documento ao qual pertencem as palavras e as *strings*. “ID_PVH” é um identificador único desta tabela.

Atributo	Tipo	Descrição
ID_PVH	Number	Id do registo
FCH	Char	Ficheiro
TAG_FHC	Varchar	Tag em estudo
EX	Number	EX = 1, se KW é numero de documento EX = 0, caso contrário
A3	Varchar	3ª Palavra vizinha anterior
A2	Varchar	2ª Palavra vizinha anterior
A1	Varchar	1ª Palavra vizinha anterior
KW	Varchar	Candidata a numero de documento
D1	Varchar	1ª Palavra vizinha seguinte
D2	Varchar	2ª Palavra vizinha seguinte
D3	Varchar	3ª Palavra vizinha seguinte

Tabela 10 – Descrição e tipo dos atributos do conjunto de dados

Para o estudo em causa, retiramos a este conjunto o atributo “TAG_FHC” por ter um valor constante e o atributo “KW” por conter o candidato a número de documento. Sabemos que o número de documento é um identificador único para cada documento e queremos estudar a existência de um padrão nas *strings* vizinhas

4.4. Uniformização de Dados

Ao longo do pré-processamento dos dados foi notado que a representação dos conceitos, através das palavras, nem sempre é feita da mesma forma. Por exemplo, a representação para o “Imposto sobre o Valor Acrescentado” aparece como “IVA”, “I.V.A.” ou “%IVA”. Estas diferentes representações prendem-se com o *layout* que cada fornecedor adopta para os documentos emitidos.

São encontrados casos outros casos interessantes que vale a pena destacar. Na tabela seguinte temos alguns exemplos de representações da palavra que se encontra no topo da coluna:

CONTRIBUINTE		EXMO		NÚMERO		REFERÊNCIA		TELEFONE	
Representação	Qtd	Representação	Qtd	Representação	Qtd	Representação	Qtd	Representação	Qtd
CONT.Nº	3	EXMO.	2	N	4	REF	3	TEL	4
CONTRIB	4	EX.MOS	1	N.	3	REF.	4	TEL.	6
CONTRIB.	2	EXMO(S)	1	N.º	5	REF.:	3	TEL.:	4
CONTRIB.:	4	EXMO(S).	2	N.º	6	REF#.	4	TEL:	4
CONTRIBUINTE	6	EXMO.(S)	4	Nº	6	REFERENCIA	4	TELEF	2
CONTRIBUINTE:	4	EXMOS.	1	Nº.	3	REFERÊNCIA	5	TELEF.	4
NO.CONTRIBUINTE	6	EXMOS(AS)	1	Nº:	5	REFERÊNCIA:	1	TELEF.:	3
		EXMOS.SRS.	1	Nº	6			TELEFONE	4
				Nº:	2			TELEFONE:	3
				NO.	3			TELEFONES:	5
				NR	6			TELEFS.	4
				NR.	4			TELEFS.:	3
				NR.:	2			TELF	2
				NR:	2			TELF.	4
				NRO	1			TEL.(S):	4
				NRO.	3			TELS.:	3
				NUM.	2			TELS:	3
				NUMERO	4				
				NUMERO:	1				
				NÚMERO	5				
				NUMEROS	1				

Tabela 11 – Exemplos de palavras e diferentes representações

A observação das palavras indica que há certos factores implicam estas diferenças nas representações:

- Acentuação – O mesmo conceito pode aparecer representado por palavras com ou sem acentuação. Por exemplo, “NÚMERO” ou “NUMERO” e “REFERÊNCIA” ou “REFERENCIA”.

- Abreviaturas – Existem palavras que aparecem abreviadas. Para os dois exemplos referidos, encontramos “Nº”, “NUM”, “NRO” ou “NR” para abreviatura de “NUMERO” e “REF” ou “REF^a” para abreviatura de “REFERENCIA”.
- Caracteres especiais – A existência de caracteres especiais é frequente em documentos que não são escritos em linguagem natural. São caracteres como os dois pontos (“:”), o traço (“-“), o ponto (“.”), a barra (“/), etc. que servem para estruturar o documento, facilitando a sua compreensão.

A uniformização dos termos que representam o mesmo conceito pode trazer benefícios no sentido em que podemos diminuir a variedade de representação dos termos. Neste sentido, optamos por retirar às *strings* a acentuação e os caracteres especiais. Para ultrapassar o problema dos termos abreviados, optamos por considerar os 4 primeiros caracteres de cada termo cujo tamanho fosse superior a quatro.

Como já verificamos, esta decisão pode trazer benefícios pela diminuição da variedade dos termos, mas há casos em que pode uniformizar termos que não têm o mesmo conceito. Por exemplo, “DESC” resulta da uniformização das *strings* “DESCARGA”, “DESCONTO” e “DESCRICAO”, mas estas 3 *strings* não têm o mesmo conceito. Existem outros casos, tais como *strings* que representam as datas (p. ex., “03.12.2003”, “2003/12/18” e “21-11-2003”), que são apresentadas com formatos distintos mas que esta uniformização não é capaz de as identificar como uma única *string*.

Para a uniformização, foi construído um *query*⁷ e aplicado ao conjunto de dados para conseguir esta uniformização.

⁷ Ver Scripts em anexo

5. Modelos para Extracção do “Número de Factura”

Já foram referidas as vantagens do Oracle Database em disponibilizar dentro da sua plataforma o Oracle Data Mining, permitindo um eficiente aproveitamento dos recursos do sistema e garantindo a segurança das transacções efectuadas. No capítulo anterior, construímos um conjunto de dados que servirá de base à construção dos modelos. Neste capítulo, abordamos a construção dos modelos: na secção 5.1 é explicada qual a metodologia experimental, são definidos os custos a ter em conta na construção dos modelos e feita a construção dos modelos a partir dos métodos de classificação disponíveis no Oracle Data Miner. Na secção 5.2 testamos e comparamos os modelos e são avaliados os resultados. Na secção 5.3 aplicamos o modelo que obteve melhor desempenho a um novo conjunto de dados e avaliamos os resultados.

5.1. Conjuntos de Dados

Para armazenar os dados preparados anteriormente e descritos no capítulo anterior, foi criada uma base de dados e um *Schema* com a estrutura necessária. Temos um conjunto com 1261 exemplos recolhidos de 41 documentos.

A construção dos modelos vai ser feita sobre o conjunto de treino com os dados originais e sobre o conjunto de treino com os dados uniformizados. Aos modelos gerados com os dados originais vamos designá-los por **MDO** – Modelos gerados para os **Dados Originais**, e aos modelos gerados pelos dados uniformizados vamos designá-los por **MDU** – Modelos gerados para **Dados Uniformizados**.

Estamos perante um problema de classificação. Tal como referido na secção 4.3.2, temos um conjunto de registos com duas classes, em que o atributo “EX” assume o valor 1 se no exemplo o valor de “KW” for um número de documento e assume o valor 0 no caso contrário. Na Fig. 23, temos um pequena amostra dos registos e a respectiva classificação na classe:

	A	B	C	D	E	F	G	H	I	J
1	ID_PV	FCH	EX	A3	A2	A1	KW	D1	D2	D3
91	90	F03	0	C-3	CAPITAL	SOCIAL:	299,278,74€	FACTURA	DATA	NÚMERO
92	91	F03	0	FACTURA	DATA	NÚMERO	19-11-2003	2114	TRANSPORTE	MATRIC./SENHA
93	92	F03	1	DATA	NÚMERO	19-11-2003	2114	TRANSPORTE	MATRIC./SENHA	VENDEDOR
94	93	F03	0	MATRIC./SENHA	VENDEDOR	N/CARRO	1	NR.	REQUISICÃO	DT.
95	94	F03	0	ENCOMENDA	G.	REMESSA	2114	LOCAL	DE	CARGA
96	95	F03	0	CARDOSO,	LDA.	RUA	5	DE	OUTUBRO,	144
97	96	F03	0	5	DE	OUTUBRO,	144	APARTADO	3007	4430
98	97	F03	0	OUTUBRO,	144	APARTADO	3007	4430	795	AVINTES
99	98	F03	0	144	APARTADO	3007	4430	795	AVINTES	PESSOA
100	99	F03	0	APARTADO	3007	4430	795	AVINTES	PESSOA	COLECTIVA
101	100	F03	0	PESSOA	COLECTIVA	NR.	506586502	CÓDIGO	CLIENTE	NR.:
102	101	F03	0	CÓDIGO	CLIENTE	NR.:	2057	CÓDIGO	PRODUTO	DESCRIÇÃO
103	102	F03	0	COMERC.	LÍQUIDO	IVA	113998	FELTRO	ESPECIAL	132,48
104	103	F03	0	113998	FELTRO	ESPECIAL	132,48	M2	1,4	185,47
105	104	F05	1	Acessórios	Têxteis	FACTURA:	34842	DATA:	18-12-2003	Pagina
106	105	F05	0	FACTURA:	34842	DATA:	18-12-2003	Pagina	1	ORIGINAL
107	106	F05	0	DATA:	18-12-2003	Pagina	1	ORIGINAL	EXMO.(S)	SNR.(S)
108	107	F05	0	CARDOSO,	LDA.	RUA	5	DE	OUTUBRO,	144

Fig. 23 – Exemplos dos dados em análise e da separação em classes

O nosso objectivo é prever, para um novo conjunto de dados extraídos de um documento, qual a classe a que cada registo pertence. Em concreto, perante uma nova factura pretende-se encontrar qual o elemento que identifica o número de documento.

5.1.1. Metodologia Experimental

Os registos vão ser separados em 70% em dados para treino, isto é, para construção do modelo e os restantes 30% para teste do modelo construído. Ora, os registos não são independentes entre si. O atributo “FCH” identifica o documento de onde foi extraído esse registo. Uma palavra extraída como possível número de documento ou como palavra vizinha, é possível que seja extraída, nos registos seguintes como vizinha, mas noutra posição. Na Fig. 24 é visível esta situação. A construção do conjunto de treino deve manter juntos os registos que pertencem ao mesmo ficheiro. Os dados para treino devem corresponder a 70% dos documentos em análise e não a 70% dos registos. A divisão dos dados para treino e teste não pode ser obtida através da aplicação directa das opções disponíveis no Oracle Data Miner. As opções disponíveis apenas permitem que se seleccione 70% dos dados para treino. O que pretendemos é conseguir 70% dos documentos em análise. Foram feitas algumas tentativas para contornar esta situação usando os parâmetros disponíveis no Oracle Data Miner.

	A	B	C	D	E	F	G	H	I	J
1	ID_PV	FCH	EX	A3	A2	A1	KW	D1	D2	D3
162	161	F07	0	carga:	07.11.2003	Página	1	Tax	Base	de
163	162	F07	0	Valor	do	I.V.A.	19,00%	39,9	7,58	5,00%
164	163	F07	0	do	I.V.A.	19,00%	39,9	7,58	5,00%	17,00%
165	164	F07	0	I.V.A.	19,00%	39,9	7,58	5,00%	17,00%	39,9
166	165	F07	0	19,00%	39,9	7,58	5,00%	17,00%	39,9	7,58
167	166	F07	0	39,9	7,58	5,00%	17,00%	39,9	7,58	Quarenta
168	167	F09	0	GAIA	SOB	N.º	60	815	-	CAPITAL
169	168	F09	0	SOB	N.º	60	815	-	CAPITAL	SOCIAL
170	169	F09	0	CAPITAL	SOCIAL	REALIZADO	€249,398,94	-	CONTRIBUINTE	N.º
171	170	F09	0	-	CONTRIBUINTE	N.º	500	737	495	DOCUMENTO
172	171	F09	0	CONTRIBUINTE	N.º	500	737	495	DOCUMENTO	Factura
173	172	F09	0	N.º	500	737	495	DOCUMENTO	Factura	C
174	173	F09	1	Factura	C	NÚMERO	306796	DATA	(E)	25-11-2003
175	174	F09	0	306796	DATA	(E)	25-11-2003	DESTINATÁRIO	771	Nº
176	175	F09	0	(E)	25-11-2003	DESTINATÁRIO	771	Nº	CONTRIBUINTE	PT506586502
177	176	F09	0	771	Nº	CONTRIBUINTE	PT506586502	Nº	ENCOMENDA	10632/18
178	177	F09	0	PT506586502	Nº	ENCOMENDA	10632/18	AG	7	CARGA
179	178	F09	0	ENCOMENDA	10632/18	AG	7	CARGA	EXP.	LOCAL
180	179	F09	0	DATA	(E)	HORA	1	100	25-11-2003	14:43
181	180	F09	0	(E)	HORA	1	100	25-11-2003	14:43	R.
182	181	F09	0	HORA	1	100	25-11-2003	14:43	R.	Lages,
183	182	F09	0	1	100	25-11-2003	14:43	R.	Lages,	525
184	183	F09	0	14:43	R.	Lages,	525	-	Canelas	PÁG.
185	184	F09	0	-	Canelas	PÁG.	1	Exmos.Srs.	PINTO	SOARES
186	185	F09	0	CARDOSO,	LDA.	Rua	5	de	Outubro,	144
187	186	F09	0	5	de	Outubro,	144	Apartado	3007	4430-795
188	187	F09	0	Outubro,	144	Apartado	3007	4430-795	AVINTES	LOCAL
189	188	F09	0	144	Apartado	3007	4430-795	AVINTES	LOCAL	DE
190	189	F09	0	CLIENTE	VENCIMENTO	PAYMENT	24-01-2004	MATRÍCULA	:	-

Fig. 24 – Dados em análise recolhidos dos documentos

O problema é que a divisão é sempre feita sobre o conjunto final de dados e não sobre o conjunto dos ficheiros. A solução passou por criar um *query*⁸ em SQL que dividisse, aleatoriamente, 70% dos documentos para treino e 30% para teste. Ficamos com um conjunto de treino representado por 29 documentos e um conjunto de teste representado por 12 documentos. Divididos e identificados os documentos, um *join* com a tabela dos dados permite construir duas *views*⁹, uma para treino e outra para teste. Em termos práticos, vamos construir os modelos sobre a totalidade dos registos da *view* de treino e depois testá-los com os registos da *view* de teste. Ficamos com 884 registos de treino e 377 registos de teste.

⁸ Ver Scripts em anexo⁹ Ver Scripts em anexo

5.1.2. Definição dos custos

Já foi referido anteriormente, que temos a possibilidade de obter a matriz de custos, a partir do Oracle Data Miner, se previamente forem introduzidos os seus valores. Antes de iniciar a construção dos modelos, os custos devem ser definidos.

Para cada documento encontramos, em média, 30 palavras candidatas a número de documento. Em número de registos, isto equivale a termos, em média, 30 registos por cada valor do atributo “FCH”. Num cenário real, a procura do número de documento envolve a análise de todas as *strings* do documento. Sabemos e já foi referido que o número de documento aparece no cabeçalho do documento. Neste cenário os custos são máximos para cada documento, isto é, para cada documento a procura do número de documento envolve a análise, em média de 30 *strings*. É este o valor do custo que vamos considerar para cada caso em que não seja identificado o número de documento, isto é, para as instâncias pertencentes à classe positiva o custo de não a identificar correctamente é 30 pois obriga à procura nas 30 *strings* do documento. Para as instâncias classificadas erradamente como positivas o custo é 1 uma vez que basta a confirmação, pontual, desta situação.

Desta forma os custos são calculados em função dos falsos positivos (FP) e dos falsos negativos (FN) através da seguinte forma:

$$Cost = FP + 30 * FN$$

5.1.3. *Build Model by Oracle Data Miner*

Neste trabalho foram construídos modelos de classificação através dos algoritmos disponíveis no Oracle Data Miner e já abordados na secção 2.3. Os algoritmos aplicados foram as Árvores de Decisão, o *Naive Bayes*, *Adaptive Bayes Network* e as *Support Vector Machines*. Os modelos de classificação a seguir descritos foram construídos a partir do conjunto de treino previamente criado. É identificado o atributo alvo. O atributo alvo “EX” assume o valor 1 ou 0. É indicado qual o valor do atributo alvo que estamos a tentar identificar. A avaliação dos modelos deve ser medida em função da previsão do valor do atributo alvo. Neste caso, queremos identificar os registos cujo valor do atributo alvo é 1. Foram incluídos todos os atributos, excepto KW como já verificamos na secção 4.3.2. O passo de separação dos dados é ignorado. A separação dos dados é importante quando é necessário criar, um conjunto para treino e um conjunto para teste, a partir do conjunto de dados em análise. Neste caso em estudo, este passo não é necessário, uma vez que já temos construído o conjunto de treino e é dada a indicação para usar todos os registos deste.

O modelo a construir deve ser bom a prever todas as classes. Os registos mal classificados, isto é, classificados numa classe à qual não pertencem obrigam à sua confirmação manual. Por exemplo, um registo cuja classe é 0 e que seja classificado como pertencente à classe 1, obriga à sua confirmação manual no documento e rejeição deste. Por este motivo, é importante que o modelo preveja correctamente os registos da classe 1, mas também que classifique correctamente os registos da classe 0. Desta forma, é seleccionada a opção “*Maximum Average Accuracy*” para criação do modelo nestas condições. Nesta fase, são também escolhidas as métricas a disponibilizar pelo algoritmo na fase de teste. Queremos que todas as possíveis métricas em análise (referidas em 4.1) sejam disponibilizadas e, não havendo qualquer limitação ao nível de hardware, são marcadas todas as opções existentes no *Wizard* para a Matriz de Confusão, as Curvas ROC e a matriz de custos. Com estas opções teremos disponibilizados o *Recall*, o *Precision*, o *False Positive Rate (FPT)*, o *F1-Measure*, *Average Accuracy*, a Matriz de Confusão, as Curvas ROC e respectivas AUC.

Decision Trees

Na construção do modelo MDO (sobre os dados originais e sem uniformização), a partir do conjunto de dados originais, isto é, sem a uniformização dos dados, usando o algoritmo de Árvores de Decisão, foram usados os parâmetros gerais descritos para os métodos de classificação. Ao nível dos parâmetros específicos vão ser definidas as condições para construção da árvore.

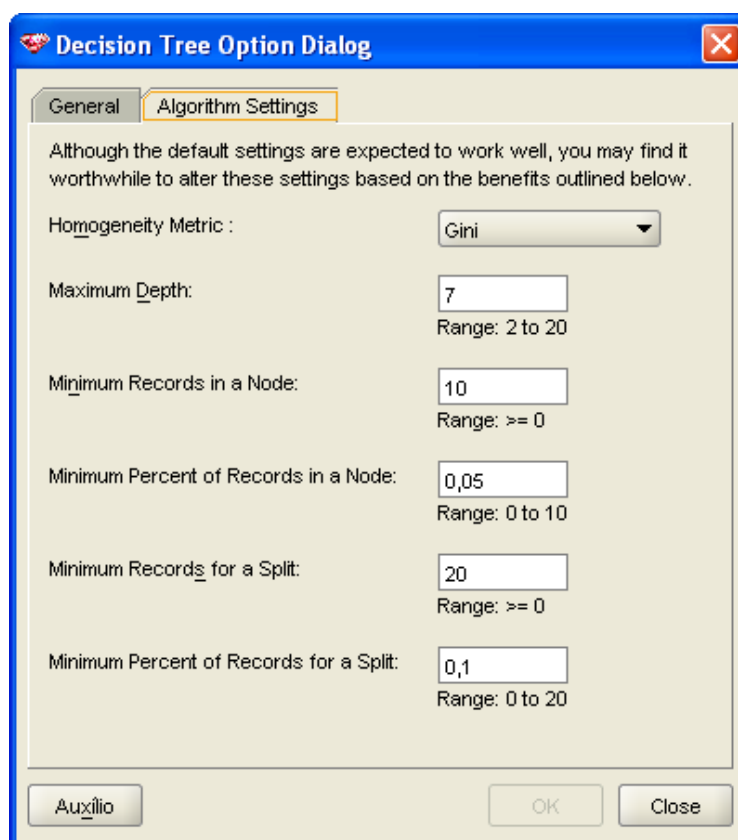


Fig. 25 – Opções construção do modelo de Árvores de Decisão

O algoritmo da Árvore de Decisão executa uma otimização interna para decidir quais os atributos a usar em cada nó e na separação dos ramos. Para isso o Oracle Data Miner dispõe de uma métrica de homogeneidade para determinar os valores dos atributos de cada lado dos ramos. Temos duas medidas de homogeneidade disponíveis – *Gini* e entropia. *Gini* atribui a um dos ramos a percentagem mais elevada de uma classe. A medida proposta por omissão é a *Gini* e vamos optar por essa proposta. São definidos os critérios de construção da árvore de decisão. A ramificação da árvore continua até que uma das condições de paragem seja atingida:

- A árvore a construir não deve ter mais do que 7 níveis
- Um nó não se divide mais se contém menos de 20 registos ou menos de 10% dos dados e, neste caso, transforma-se em folha
- Uma divisão regressa ao estado anterior se produzir um nó com menos de 10 registos ou menos de 5% dos dados.

O modelo MDO gerou uma árvore de decisão representada na Fig. 26:

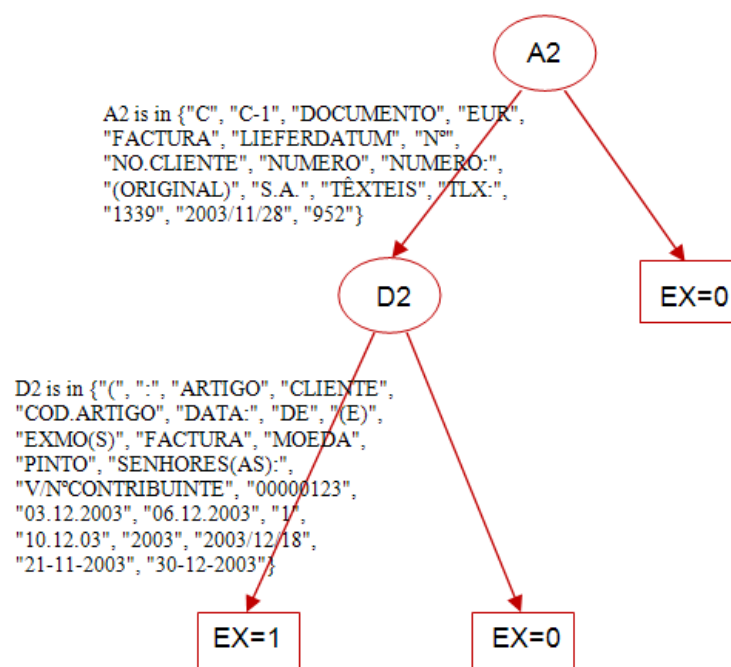


Fig. 26 – Árvore de decisão gerada no Oracle Data Miner

A árvore gerada tem dois nós e três folhas e para os registos da classe EX = 1, a árvore apresentada equivale à regra representada na condição visualizada na Fig. 27:

```

IF
D2 is in { ( : ARTIGO CLIENTE COD.ARTIGO DATA: DE (E) EXMO(S) FACTURA MOEDA PINTO SENHORES(AS):
V/NºCONTRIBUINTE 00000123 03.12.2003 06.12.2003 1 10.12.03 2003 2003/12/18 21-11-2003 30-12-2003 }
AND
A2 is in { C C-1 DOCUMENTO EUR FACTURA LIEFERDATUM Nº NO.CLIENTE NUMERO NUMERO: (ORIGINAL) S.A.
TÊXTEIS TLX: 1339 2003/11/28 952 }
THEN EX equal 1
Confidence (%)=1,0000
Support (%)=0,0328
  
```

Fig. 27 – Regra principal gerada pela modelo MDO de Árvores de Decisão

Já foi referido que uma das vantagens dos modelos das Árvores de Decisão era a sua fácil interpretação. Da regra criada por este modelo entendemos que, perante uma palavra candidata a número de documento¹⁰, se a 2ª *string* anterior a esta palavra for “C” ou “C-1” ou “DOCUMENTO” ou ... ou “NO.CLIENTE” ou “NUMERO” ou “NUMERO:” ou ... ou “952” e se a 2ª *string* após esta palavra for “(“ ou “:” ou “ARTIGO” ou “CLIENTE” ou ... ou “FACTURA” ou “MOEDA” ou ... ou “30-12-2003”, então a palavra candidata a número de documento é um “número de documento”.

Desta regra resulta a descoberta dos 29 casos positivos que fazem parte do conjunto de treino. Temos, então, um suporte de 3,28% ($29/884=0,0328$) e uma confiança de 100% ($29/29=1,0000$). Temos uma confiança elevada resultado da descoberta de todos os casos positivos e um suporte baixo que resulta do facto de existirem muitos registos negativos quando comparados com o número de casos positivos.

No Oracle Data Miner cada regra é acompanhada de uma regra alternativa. Esta regra alternativa é gerada para um atributo que pode ser usado, alternativamente, na aplicação do modelo aos dados de teste, caso falem o(s) atributo(s) da regra principal. A regra alternativa gerada foi a seguinte:

```
Se A3 is in { “**”, “ACESSÓRIOS”, “AVINTES”, “CALÇADO”, “COD.CLIENTE”,
“DATA”, “FACTURA”, “GRIFAGEM.PEDRO@CLIX.PT”, “INVOICE”,
“NR”, “PÁG.”, “(PORTUGAL)”, “PORTUGAL”, “VALUTA”,
“WWW.STUBBE-PORTUGUESA.PT”, “0,00”, “01”, “03/12/27”,
“189-LIVRO”, “226088692”, “248”, “292”, “31-12-2003”, “5340”}
AND D3 is in { “/”, “CÂMBIO”, “CIF:”, “CONTACTO:”, “CONTRIBUINTE”,
“DATA”, “DE”, “DESCRIÇÃO”, “DUPLICADO”, “EMISSAO”, “EXMO.”,
“PAGINA”, “PINTO”, “QUANTIDADE”, “SENHOR(ES)”, “SOARES”,
“SOC.”, “TECNICOURO”, “0042”, “03/11/27”, “2003/11/24”, “2003/11/25”,
“21.11.2003”, “24.11.2003”, “5.12.2003”}
Então EX = 1
```

Fig. 28 – Regra alternativa gerada pelas Árvores de Decisão

¹⁰ De acordo com o que foi definido na secção 4.3.2

As condições da regra alternativa foram geradas a partir do atributo que contém a 3ª *strings* antes e o atributo que contém a 3ª *strings* após a palavra candidata a número de documento.

Na regra principal, verificamos que esta contém *strings* diferentes mas cujo conceito é o mesmo. Temos o caso de “FACTURA” e “FACTURA:” que têm o mesmo significado da *string* “FACTURA” e “03.12.2003”, “2003/12/18” e “21-11-2003” que representam datas, mas que são apresentadas com formatos distintos. Verifica-se, então, que palavras diferentes e que representam o mesmo conceito têm influência diferente para as previsões.

No caso dos modelos MDU, modelos para dados uniformizados, gerados com o algoritmo das Árvores de Decisão, verificamos que a regra gerada incide sobre os atributos que contém as 1ªs e 2ªs *strings* antes da palavra candidata a número de documento. A regra gerada é visível na figura seguinte:

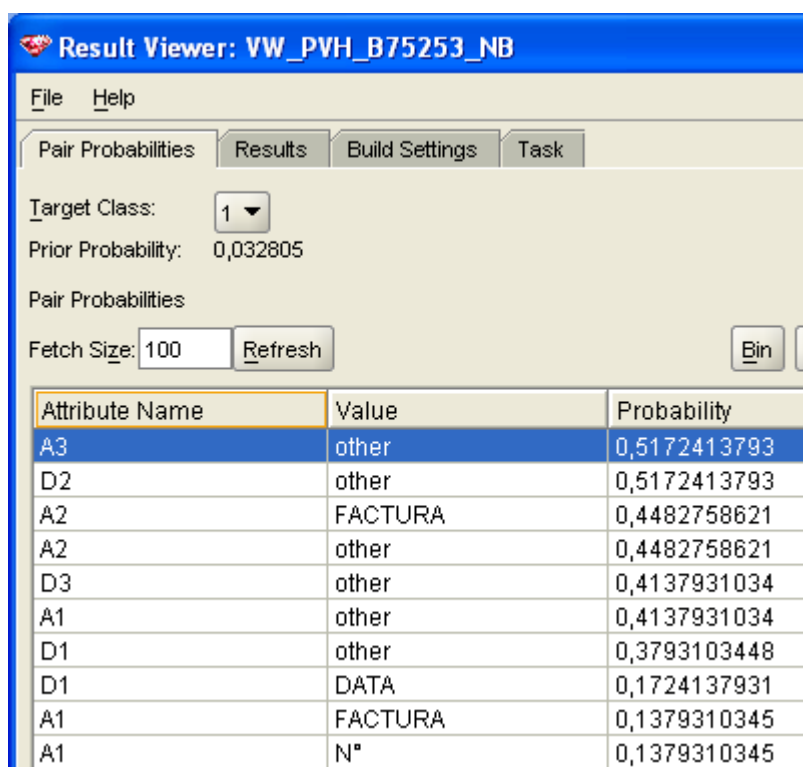
```
IF
A1 is in { ## DATA FACT FT INVO KUND N NO NRO NUM NUME VDIN VEND 1000 2811 }
AND
A2 is in { C C1 DOCU EUR FACT LIEF N NOCL NUME ORIG SA TEXT TLX 1339 2003 952 }
THEN EX equal 1
Confidence (%)=0,7838
Support (%)=0,0419
```

Fig. 29 – Regras geradas pelas Árvores de Decisão aplicado aos modelos MDO

Nesta regra verificamos que há situações em que a uniformização não resolveu, na totalidade, o facto de termos *strings* diferentes a representar o mesmo conceito. O caso do “NUMERO” aparece na regra de diversas formas: “N”, “NO”, “NRO”, “NUM” e “NUME”.

Naive Bayes

Para o modelo MDO e para os parâmetros específicos do algoritmo de *Naive Bayes*, podemos seleccionar o método de discretização e os valores mínimos a considerar para as probabilidades condicionais. No caso do método de discretização, o método escolhido foi o “Top N Binning” porque só existem atributos categóricos. Neste caso os atributos são divididos em grupos, um por cada valor com mais alta frequência. Os restantes valores, com baixa frequência no atributo, são recodificados e atribuídos a um grupo chamado “other”. Podemos, também, definir os mínimos para os valores das probabilidades condicionais. Os valores *default* estão a 0. Vamos mantê-los para não excluir condições. As probabilidades condicionais calculadas pelo modelo MDO são mostradas no *Viewer* de resultados. Verificámos que a probabilidade de $EX = 1$ é de 3,28%. Isto verifica-se porque o número de registos da classe $EX = 1$ é muito inferior ao número de registos para $EX = 0$. Na Fig. 30, para o caso de $EX = 1$, temos as probabilidades condicionadas com os valores mais elevados.



Attribute Name	Value	Probability
A3	other	0,5172413793
D2	other	0,5172413793
A2	FACTURA	0,4482758621
A2	other	0,4482758621
D3	other	0,4137931034
A1	other	0,4137931034
D1	other	0,3793103448
D1	DATA	0,1724137931
A1	FACTURA	0,1379310345
A1	N°	0,1379310345

Fig. 30 – Probabilidades geradas pelo modelo de *Naive Bayes*

Verificámos que a probabilidade do atributo “A2” assumir o valor “FACTURA” é de 44,82% e do atributo “D1” assumir o valor “DATA” desce para 17,24%. Com uma probabilidade de 13,79%, o atributo “A1” pode assumir igualmente o valor “FACTURA” e “Nº”. No entanto, verificamos que os atributos “A3” e “D2” são os que têm maior probabilidade (51,72%) quando assume o valor “other”.

Para o modelo MDU, verificámos que a probabilidade do atributo “A2” assumir o valor “FACT” é de 44,82% e do atributo “D1” assumir o valor “DATA” é, neste caso, 27,59%. O atributo “A1” assume valor “N” também com probabilidade 27,59%. No entanto, verificamos que os atributos “A3” e “D2” são os que têm a mesma probabilidade (51,72%) quando assumem o valor “other”. Em resumo, relativamente ao modelo anterior, a probabilidade de “A3” e “D2” assumir o valor “other” decresceu o que significa que a uniformização das palavras retirou elemento a este grupo e aumentaram a probabilidade de “D1” assumir o valor “DATA”, “A1” assumir o valor “N”. A probabilidade de “A1” assumir o valor “FACT”, abreviatura de “FACTURA” também aumentou.

Adaptive Bayes Network

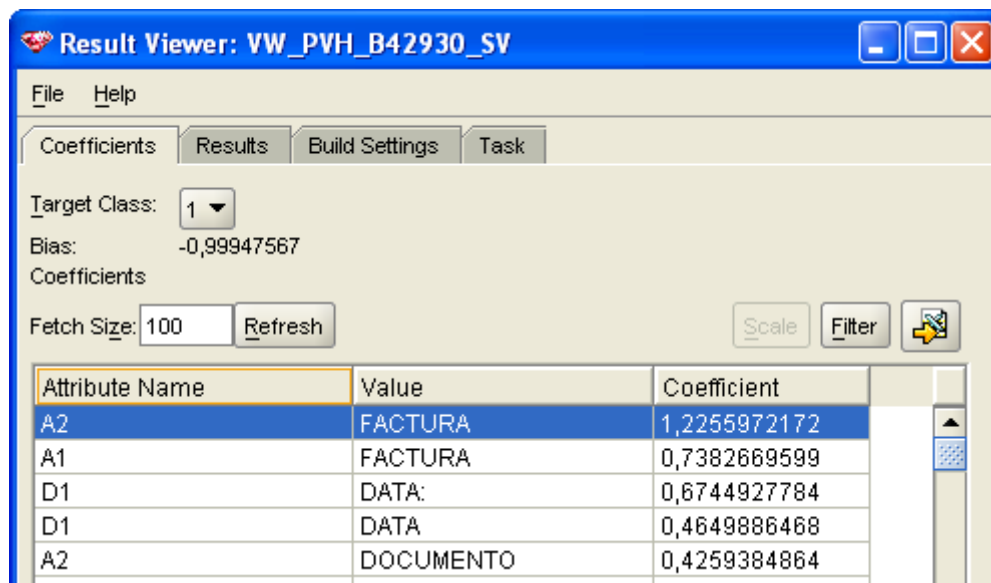
Para os parâmetros específicos do modelo MDO gerado para o algoritmo *Adaptive Bayes Network*, podemos definir qual a modalidade a usar na construção do modelo. Uma vez que o ABN não gera regras interessantes se o conjunto de dados for inferior a 20.000 exemplos, como já referido anteriormente (na secção 2.3), optamos por gerar o modelo MDO através de uma modalidade não geradora de regras, *Naive Bayes*. Por *default* o valor para a quantidade de atributos máximos a considerar como atributos de previsão (*predictors*) é de 25. O conjunto de treino tem 8 atributos, um nº menor que o proposto, pelo que foi aceite esta sugestão.

Na construção do modelo MDO através da modalidade *Naive Bayes* e, neste caso, é possível a construção do modelo, mas não é possível a sua visualização. O mesmo acontece para o modelo MDU construído através do algoritmo *Adaptive Bayes Network*, optamos por escolher a modalidade *Naive Bayes* e, tal como no caso anterior, não é possível a visualização deste modelo.

Support Vector Machines

Para os parâmetros específicos do modelo MDO gerado para o algoritmo SVMs a escolha começa por determinar qual a função de *Kernel* que o algoritmo das SVMs vai usar. Por *default*, o Oracle Data Miner propõe que o *Kernel* seja decidido pelo sistema. Vamos aceitar esta sugestão e deixar que o Oracle Data Miner determine qual a função de *Kernel* a usar. O valor de tolerância é uma medida que determina quando o algoritmo deve estar satisfeito com o resultado e considera que deve parar a construção deste modelo. É um mecanismo de paragem e vai assumir o valor 0,001. Um valor mais alto para a tolerância resulta num modelo de construção mais rápida mas com uma taxa de acerto inferior. O factor de complexidade é uma medida que previne o *Overfitting*. Quando não é definido, o algoritmo calcula e tenta otimizar este valor para encontrar o equilíbrio entre a simplicidade e a complexidade do modelo. Neste caso, este valor não vai ser definido e vamos deixar que o modelo encontre este valor. A aprendizagem activa é uma metodologia de optimização para controlo do crescimento do modelo e de diminuição do tempo de construção do mesmo. Esta metodologia permite a construção de modelos para conjuntos de treino de grandes dimensões. Para a construção deste modelo MDO, vamos manter a opção de aprendizagem activa ligada, apesar do nosso conjunto de treino ser de dimensões reduzidas.

No modelo construído verificamos que o *Kernel* eleito pelo sistema como adequado a este problema foi o *Kernel* linear e o factor de complexidade assumiu o valor 0,1667. Do modelo MDO construído, e para o caso do *Kernel* linear, resulta uma tabela com os coeficientes para os pares atributo/valor. Os valores dos atributos com elevados coeficientes, quer positivos quer negativos, indicam características com forte influência nas previsões.



Attribute Name	Value	Coefficient
A2	FACTURA	1,225972172
A1	FACTURA	0,7382669599
D1	DATA:	0,6744927784
D1	DATA	0,4649886468
A2	DOCUMENTO	0,4259384864

Fig. 31 – Coeficientes gerados pelo modelo MDO com SVMs

Como se pode ver na figura, o atributo “A2”, com o coeficiente de 1,226, é o que tem mais influência sobre as futuras previsões quando assume o valor “FACTURA”. Para o mesmo valor, o atributo “A1” tem também um forte contributo para efectuar previsões na aplicação a novos conjuntos de dados. O atributo “D1” com os valores “DATA:” e “DATA” são, de seguida, os que mais contribuem para as previsões.

Para o modelo MDU construído através do algoritmo SVM, verificamos que o atributo “A2”, com o coeficiente de 1,309, é o que tem mais influência sobre as futuras previsões quando assume o valor “FACT”. Para o mesmo valor, o atributo “A1” tem também um forte contributo (0,776) para efectuar previsões. Este coeficientes são mais elevados que o modelo anterior. Para o caso em que “A1” assume o valor “NUM” este coeficiente também subiu. Para o caso em que “D1” assume o valor “DATA” este coeficiente tem apenas um valor que substitui os dois valores do modelo anterior.

5.2. Resultados dos testes aos modelos MDO

A aplicação dos modelos MDO ao conjunto de testes é directa. Como já vimos anteriormente, temos um conjunto de teste e o Oracle Data Miner permite testar um conjunto de dados individualmente. O conjunto de teste tem a mesma estrutura que o conjunto de treino. E, lembrando, para teste dos modelos MDO temos um conjunto de dados que representam 12 documentos e 377 exemplos dos quais 12 são exemplos positivos e os restantes negativos.

Decision Trees

O resultado do teste ao modelo MDO gerado pelo algoritmo de árvores de decisão é apresentado num *viewer* onde estão disponíveis várias medidas de interesse. Este modelo é o que maximiza a média de acerto. Vemos estes resultados na Fig. 32:

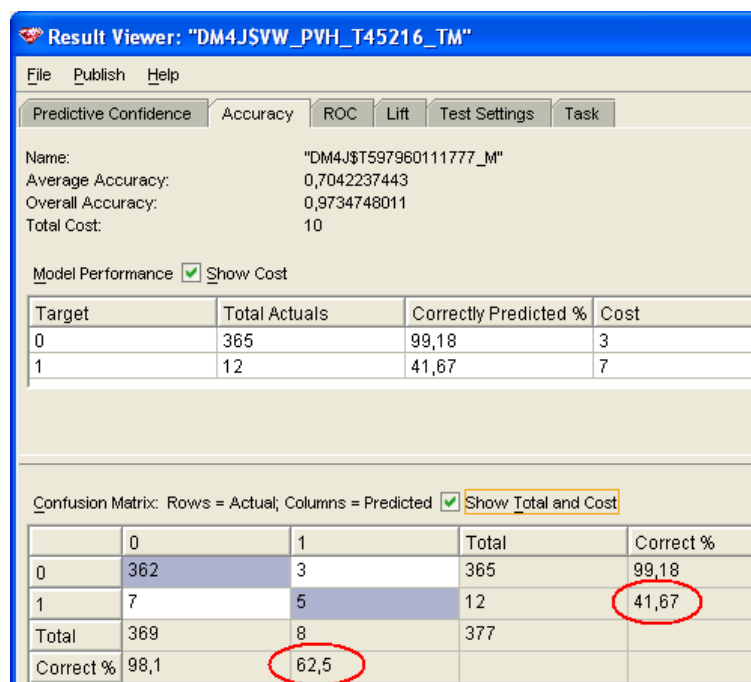


Fig. 32 – Resultados dos testes ao modelo de Árvores de Decisão

Na Matriz de Confusão estão quantificados os registos do conjunto de teste e separados pela sua classificação. Dos 12 registos cujo valor da classe é EX = 1, foram classificados correctamente 5 elementos, *Verdadeiros Positivos* = 5 e 7 foram mal

classificados por este modelo, *Falsos Negativos* = 7. Três registos foram classificados incorrectamente como positivos, *Falsos Positivos* = 3 e os restantes 362 registo foram correctamente identificados como sendo da classe EX = 0, Verdadeiros Positivos = 362. Dos resultados da Matriz de Confusão podemos obter os valores de *Recall* e *Precision* de 0,4167 e de 0,6250 respectivamente. A média de acerto, *Average Accuracy*, é de 70,43%.

No contexto do nosso problema estes resultados transmitem o seguinte: dos 12 documentos seleccionados para teste, em 5 foi descoberto, acertadamente, qual o elemento que identificava o número de documento. Erradamente foram sugeridos três elementos como número de documento. Em termos do esforço necessário para o utilizador em identificar o número de documento, este fica apenas ligeiramente reduzido. Apenas em 5 dos 12 documentos, é suficiente confirmar que os elementos sugeridos são os números de documento, sendo necessário rejeitar os 3 elementos que são erradamente sugeridos e procurar os números de documento nos restantes 7 documentos. Assim, podemos dizer que o resultado obtido é pouco satisfatório. Sob ponto de vista do negócio, uma equipa do departamento de contabilidade teria o trabalho facilitado em cerca de 40% dos documentos, mas 60% dos documentos exigiam o mesmo trabalho de procura do número de documento. Era desejável e mais interessante se conseguíssemos descobrir uma maior quantidade de número de documentos.

Probability Threshold	FP	FN	TP	TN	Recall	Precision	FPR	F1-measure	Accuracy	Avg Accuracy	Cost
1,0000	0	12	0	365	0,0000		0,0000		0,9682	0,5000	360
1,0000	2	9	3	363	0,2500	0,6000	0,0055	0,3529	0,9708	0,6223	272
0,7073	3	7	5	362	0,4167	0,6250	0,0082	0,5000	0,9735	0,7042	213
0,0000	365	0	12	0	1,0000	0,0318	1,0000	0,0617	0,0318	0,5000	365

Tabela 12 – Variação dos resultados em função do *Threshold* nas Árvores de Decisão

No caso dos resultados apresentados, o *Threshold* assume o valor 0,7073. A variação deste valor provoca o reajuste das previsões e nova classificação dos registos em teste. No entanto, não é possível baixar o *threshold*, dado que o valor seguinte seria 0, o que significa que todos os elementos seriam classificados como positivos. Por outro lado, dado que o número de falsos negativos é elevado, não faz sentido aumentar o *threshold*.

De qualquer forma, se isso fosse feito, o valor seguinte é 1.0, originando a classificação de todos os elementos como negativos.

Na Tabela 12 podemos avaliar esta variação de forma resumida. A linha da tabela cujo *Threshold* = 0,7073 é a que resulta num acerto médio mais elevado (*Average Accuracy* = 0,7042). Este resultado está de acordo com as condições iniciais definidas para a construção do modelo, isto é, maximizar o valor do *Average Accuracy*. Ao nível global a taxa de acerto é de 97,35%. Este resultado é o que apresenta uma maior redução de custos.

No cenário actual, onde a identificação do número de documento implica verificar todas as palavras¹¹ dos 12 documentos, temos um custo de 365. Este custo é o custo máximo e é calculado, de acordo com a fórmula anterior definida na secção 5.1.2:

$$Cost = 365 + 30 * 0 = 365$$

Essa realidade é representada quando o *Threshold* é 0. Para valores intermédios, interessa identificar o que permite uma significativa redução de custos. No caso do resultado apresentado, esta redução é de 152 (365-213 = 152).

¹¹ De acordo com as condições definidas neste problema, na secção 4.3.2

Naive Bayes

A análise ROC do teste ao modelo MDO de *Naive Bayes* permite-nos avaliar vários cenários. O cenário apresentado com maior taxa média de acerto, *Average Accuracy* = 75,15%, está assinalado na Tabela 13:

Probability Threshold	FP	FN	TP	TN	Recall	Precision	FPR	F1-measure	Accuracy	Avg Accuracy	Cost
1,0000	0	12	0	365	0,0000		0,0000		0,9682	0,5000	360
1,0000	0	9	3	365	0,2500	1,0000	0,0000	0,4000	0,9761	0,6250	270
0,1603	6	8	4	359	0,3333	0,4000	0,0164	0,3636	0,9629	0,6584	246
0,0062	45	6	6	320	0,5000	0,1176	0,1233	0,1905	0,8647	0,6884	225
0,0003	50	5	7	315	0,5833	0,1228	0,1370	0,2029	0,8541	0,7232	200
2,04E-05	83	4	8	282	0,6667	0,0879	0,2274	0,1553	0,7692	0,7196	203
8,19E-06	108	3	9	257	0,7500	0,0769	0,2959	0,1395	0,7056	0,7271	198
2,302E-06	134	2	10	231	0,8333	0,0694	0,3671	0,1282	0,6393	0,7331	194
1,331E-08	151	1	11	214	0,9167	0,0679	0,4137	0,1264	0,5968	0,7515	181
4,33E-10	206	0	12	159	1,0000	0,0550	0,5644	0,1043	0,4536	0,7178	206
0	365	0	12	0	1,0000	0,0318	1,0000	0,0617	0,0318	0,5000	365

Tabela 13 – Variação dos resultados em função do *Threshold* no *Naive Bayes*

Neste caso, o valor do *Recall* é de 0,9167, resultado de termos 11 verdadeiros positivos e apenas 1 falso negativo. No entanto, para este valor do *Threshold*, $Threshold = 1,33 * 10^{-8}$, temos 151 falsos positivos. Isto justifica o baixo valor do *Precision*, *Precision* = 0,0679. Temos em média, para cada documento, 13 elementos negativos, propostos como positivos e que precisam de ser identificados e rejeitados. Este cenário é o que apresenta o custo mais baixo. Neste caso, temos uma redução de custos de 184 relativamente ao custo inicial.

Adaptive Bayes Network

A aplicação do modelo MDO *Adaptive Bayes Network* ao conjunto de testes, resulta 11 verdadeiros positivos e 1 falso negativo. Nesta situação, o *Recall* assume o valor 91,67%. Foram sugeridos, erradamente, 262 registos como sendo registos positivos e 103 registos verdadeiros negativos. Nesta situação o *Precision* assume o valor 4,03%. A média de acerto é, neste caso, de 59,94% e o total de acerto é de 30,24%.

Na Tabela 14 temos estes valores resumidos:

Probability Threshold	FP	FN	TP	TN	Recall	Precision	FPR	F1-measure	Accuracy	Avg Accuracy	Cost
1,0000	0	12	0	365	0,0000		0,0000		0,9682	0,5000	360
0,5232	1	10	2	364	0,1667	0,6667	0,0027	0,2667	0,9708	0,5820	301
0,0556	262	1	11	103	0,9167	0,0403	0,7178	0,0772	0,3024	0,5994	292
0,0016	315	0	12	50	1,0000	0,0367	0,8630	0,0708	0,1645	0,5685	315
0,0000	365	0	12	0	1,0000	0,0318	1,0000	0,0617	0,0318	0,5000	365

Tabela 14 – Variação dos resultados em função do *Threshold* nas ABN

Estes resultados sugerem muitos falsos positivos. Da análise ROC, a maior taxa média de acerto é 59,94% e temos 11 elementos positivos identificados correctamente e 1 elemento positivo incorrectamente classificado. No entanto existe um maior número de elementos erradamente classificados como positivo comparativamente a algoritmo de *Naive Bayes*. Para cada documento, temos em média 23 elementos sugeridos como positivos, dos quais 22 são falsos. O valor de FPR = 71,78% indica o elevado rácio de falsos positivos. No problema em questão, o modelo identifica correctamente o número de documentos em 91,67% dos documentos, contudo apenas 3,99% dos elementos sugeridos como positivos são verdadeiros. Esta diferença é notada avaliando o *F1-Measure*. Este valor é baixo e indica que existe uma grande diferença entre os valores de *Recall* e *Precision*. Neste caso o *Recall* é elevado e o *Precision* é baixo. O elevado número de elementos erradamente sugeridos como positivos, obriga a um árduo trabalho de confirmação de cada um dos elementos. Por este motivo, temos neste cenário, uma taxa de acerto total de apenas 30,24% e o ganho obtido é de 73.

Support Vector Machines

Da análise ROC do teste ao modelo MDO das SVMs, podemos avaliar a variação do *Threshold* e permite-nos obter vários cenários. Na Tabela 15, a seguir apresentada, vemos essa variação:

Probability Threshold	FP	FN	TP	TN	Recall	Precision	FPR	F1-measure	Accuracy	Avg Accuracy	Cost
1,0000	0	12	0	365	0,0000		0,0000		0,9682	0,5000	360
0,7393	0	7	5	365	0,4167	1,0000	0,0000	0,5882	0,9814	0,7083	210
0,4999	1	5	7	364	0,5833	0,8750	0,0027	0,7000	0,9841	0,7903	151
0,4606	2	4	8	363	0,6667	0,8000	0,0055	0,7273	0,9841	0,8306	122
0,2928	10	3	9	355	0,7500	0,4737	0,0274	0,5806	0,9655	0,8613	100
0,2081	42	2	10	323	0,8333	0,1923	0,1151	0,3125	0,8833	0,8591	102
0,1933	49	1	11	316	0,9167	0,1833	0,1342	0,3056	0,8674	0,8912	79
0,1616	266	0	12	99	1,0000	0,0432	0,7288	0,0828	0,2944	0,6356	266
0,0000	365	0	12	0	1,0000	0,0318	1,0000	0,0617	0,0318	0,5000	365

Tabela 15 – Variação dos resultados em função do *Threshold* nas SVMs

O cenário apresentado com maior taxa média de acerto, *Average Accuracy* = 89,12%, está assinalado na tabela. Neste caso, o valor do *Recall* é de 0,9167, resultado de termos 11 verdadeiros positivos e apenas 1 falso negativo. Neste caso temos apenas 49 falsos positivos. Ao contrário do teste ao modelo de *Naive Bayes* e ao modelo *Adaptive Bayes Network*, que propõem um elevado número de registos incorrectamente classificados como positivos, 151 e 262 falsos positivos respectivamente, este modelo apresenta um reduzido número de falsos positivos. Consequentemente o valor de FPR é de 13,42%. Em média, para cada documento, temos 4 palavras erradamente propostas para número de documento. Estas palavras precisam de ser identificadas e rejeitadas. Este valor de *Threshold* = 0,1933, é o que apresenta o custo mais baixo. Neste caso, temos uma redução de custos de 184 relativamente ao custo inicial. O resultado obtido é notoriamente melhor que os resultados das Árvore de Decisão, *Naive Bayes* e do *Adaptive Bayes Network*.

Em termos de trabalho na identificação do número de documento para os 12 documentos, neste cenário, este esforço fica significativamente reduzido na medida em que, basta verificar e rejeitar os 4 elementos erradamente sugeridos para cada documento, e procurar o número de documento em apenas 1 documento.

5.2.1. Comparação dos Modelos

A análise individual dos resultados dos testes feitos a cada um dos modelos MDO permite avaliar algumas particularidades e os aspectos favoráveis e desfavoráveis de cada um. Contudo interessa saber qual o modelo que, em termos globais, é mais vantajoso. Os dados apresentados na Tabela 16, resumem o desempenho de cada um dos modelos e permite uma comparação dos mesmos.

	SVM	DT	NB	ANB
Recall	91,67%	41,67%	91,67%	91,67%
Precision	18,33%	62,50%	6,79%	4,03%
FPR	13,42%	0,82%	41,37%	71,78%
F1-measure	30,56%	50,00%	12,64%	7,72%
Avg Accuracy	89,12%	70,42%	75,15%	59,94%
AUC	0,9451	0,7041	0,8338	0,6634
Cost	79	213	181	292
Gain	286	152	184	73

Tabela 16 – Comparação dos indicadores nos testes aos 4 modelos MDO

Idealmente, num contexto de trabalho, a identificação do número de documento deve ser isenta de dúvidas e o mais célere possível. Isto significa que o modelo deve conseguir identificar, para cada documento, o elemento que representa o número de documento. No entanto, não deve sugerir elementos que não representam números de documentos e que obriguem a uma intervenção humana para sua confirmação com consequentes perdas de tempo e potenciais falhas neste processo.

Na ausência de um cenário ideal, interessa-nos o que apresenta menor risco e o que contribui para uma maior redução dos custos. O modelo cuja redução de custos é maior é o modelo gerado pelo algoritmo das *Support Vector Machines*. Neste caso, o custo é de 79 o que significa um ganho relativamente ao estado inicial de 286. Para além desta característica, este modelo é o que apresenta uma maior taxa média de acerto, *Average Accuracy* = 89,12%, o que demonstra que este modelo tem uma elevada taxa de acerto para todas as classes. O elevado valor do *Recall* que este modelo apresenta, *Recall* = 91,67% (neste caso para a classe positiva, $EX = 1$), significa que o modelo consegue identificar correctamente 91,67% dos exemplos positivos. Este valor de *Recall* é

igualado pelos modelos de *Naive Bayes* e *Adaptive Bayes Network*. No entanto, o baixo valor do *Precision* para estes dois modelos, 6,79% e 4,03% respectivamente, revela que os modelos propõem um elevado número de falsos positivos, isto é, propõem para número de documentos, elementos que não o são. O modelo que apresenta uma maior precisão, *Precision* = 65,50%, é o das Árvores de Decisão. No entanto este modelo apresenta um *Recall* baixo indicando que consegue identificar apenas 41,67% dos exemplos positivos. Nas SVMs, o valor do *Precision* é de 18,33%. Este valor representa, como já analisado anteriormente, uma média de 5 elementos, por documento, sugeridos como número de documento e que exigem uma confirmação. A confirmação destes 5 elementos é, neste contexto, razoável e aceitável. Representa 16,67% dos elementos a confirmar¹². Também nas SVMs, o valor de FPR é de 13,42% o que significa uma taxa reduzida de falsos positivos. O modelo que apresenta uma menor taxa de falsos positivos é o da Árvores de Decisão, FPR = 0,82% e o modelo que apresenta a maior taxa de falsos positivos é o *Adaptive Bayes Network* com FPR = 71,78%. Este último modelo é, também, o que apresenta uma menor média de acerto, *Average Accuracy* = 59,94% e custos mais elevados, *Cost* = 292, e consequentemente o que apresenta menor ganho, *Gain* = 73.

No Gráfico 8 temos uma comparação das curvas ROC para os testes aos modelos. A linha “vermelha” indica o ponto que representa um maior ganho para cada um dos modelos.

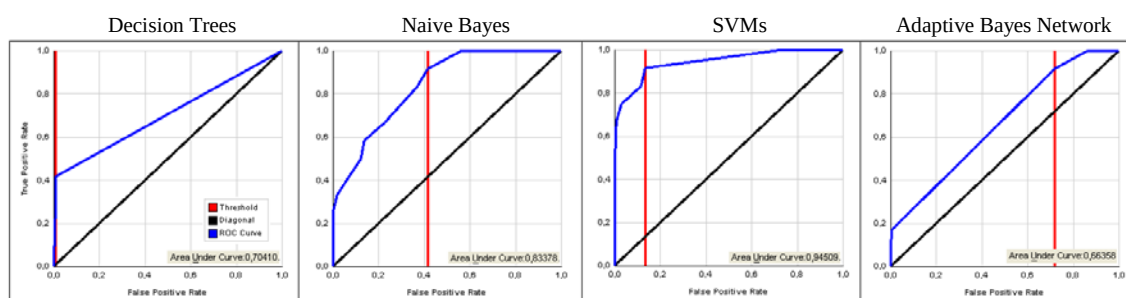


Gráfico 8 – Comparação análise Curvas ROC dos Modelos MDO

¹² Por documento, temos uma média de 30 elementos candidatos a número de documento

Podemos comparar as curvas ROC de forma mais objectiva através dos valores da *Area Under Curve* (AUC) já apresentados na Tabela 16. Este valor é um bom indicador da performance de um modelo. O modelo gerado pelas SVMs é o que apresenta este indicador mais elevado. A curva ROC da SVMs apresenta o valor para a AUC de 0,9451 e, em contrapartida, o modelo gerado pelo algoritmo da *Adaptive Bayes Network* é o que apresenta este indicador mais baixo, $AUC = 0,6634$.

5.2.2. Comparação com os Modelos Dados Uniformizados

A aplicação dos modelos MDU ao conjunto de testes, tal como para os modelos anteriores, é directa e não envolve a definição de novos parâmetros. O Oracle Data Miner assume, para o conjunto de teste, todos os valores definidos na construção do modelo MDU. O conjunto de teste mantém a mesma estrutura do conjunto de treino.

Da análise ROC do teste aos modelos MDU, escolhemos, para cada um dos algoritmos, a que nos permite um menor custo. Fizemos a comparação com a análise ROC do teste ao modelo MDO, também para os cenários que nos permitem um maior ganho. A Tabela 17 tem a comparação dos resultados obtidos:

		Probability Threshold	FP	FN	TP	TN	Recall	Precision	FPR	F1-measure	Avg Accuracy	Cost	Gain	
SVM	MDO	0,1933	49	1	11	316	0,9167	0,1833	0,1342	0,3056	0,8912	79	286	↓
	MDU	0,3830	5	3	9	360	0,7500	0,6429	0,0137	0,6923	0,8682	95	270	
DT	MDO	0,7073	3	7	5	362	0,4167	0,6250	0,0082	0,5000	0,7042	213	152	↑
	MDU	0,7838	5	6	6	360	0,5000	0,5455	0,0137	0,5217	0,7432	185	180	
NB	MDO	1,33E-08	151	1	11	214	0,9167	0,0679	0,4137	0,1264	0,7515	181	184	↓
	MDU	7,54E-10	156	1	11	209	0,9167	0,0659	0,4274	0,1229	0,7446	186	179	
ABN	MDO	0,0556	262	1	11	103	0,9167	0,0403	0,7178	0,0772	0,5994	292	73	↑
	MDU	0,0489	36	5	7	329	0,5833	0,1628	0,0986	0,2545	0,7424	186	179	

Tabela 17 – Comparação dos Modelos MDO e MDU

Verificamos que para os modelos MDU gerados a partir das Árvore de Decisão, do *Naive Bayes* e das *Adaptive Bayes Network* o custo conseguido é semelhante, 185 e 186 respectivamente, e a média de acerto é aproximadamente 74%. As SVMs são as que conseguem um menor custo (95) e, conseqüentemente, um maior ganho ($365-95 = 270$). Contudo, apesar do modelo MDU gerado pelas SVMs apresentar o maior ganho entre os modelos MDU, este não consegue melhorar o ganho obtido pelo modelo MDO gerado pelas SVMs. Notamos que houve uma ligeira melhoria no ganho para o modelo MDU gerado pelas Árvore de Decisão relativamente ao modelo MDO gerado pelo mesmo algoritmo (os modelos foram apresentados na secção 5.1.3) e que para o modelo MDU gerado pela *Adaptive Bayes Network*, cujo ganho é de 179, o ganho aumentou consideravelmente relativamente ao modelo MDO, cujo ganho é de 73.

O nosso ponto de partida tem um custo de 365. Isto acontece, como já verificamos, porque a identificação do número de documento é feita manualmente verificando todas as palavras. No gráfico seguinte temos a comparação dos custos e ganhos conseguidos pelos vários algoritmos dos modelos MDO e MDU.

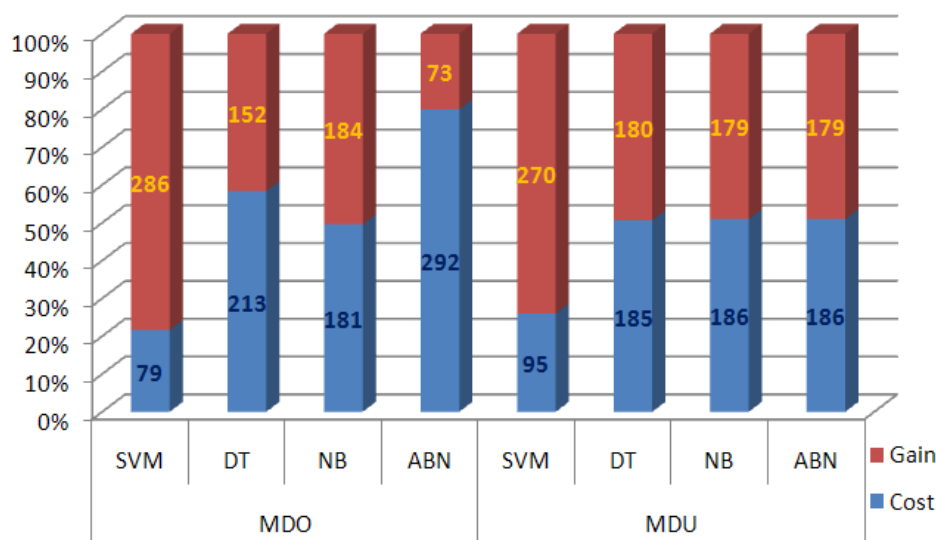


Gráfico 9 – Comparação de custos e ganhos dos Modelos MDO e MDU

Verificamos que a uniformização dos dados equilibrou os custos e ganhos para os algoritmos das *Árvores de Decisão*, *Naive Bayes* e *Adaptive Bayes Network* nos modelos MDU. Os algoritmos SVMs são os que conseguem um maior ganho para ambos os modelos. No entanto, é o modelo MDO gerado com as SVMs, o que consegue o maior ganho. A partir do gráfico verificamos que esse ganho, como já referido, é de 286 o que equivale a uma redução de custos de cerca de 78%.

De uma forma geral, a uniformização dos dados não trouxe substanciais mais-valias em termos de redução de custos e para dois dos modelos obtiveram-se menores ganhos relativamente a dados não uniformizados. Isto significa que, com a uniformização, houve introdução de ruído nos dados dando origem a perda de informação importante.

5.3. Aplicação a novos dados

No estudo experimental realizado, o algoritmo SVM obteve os melhores resultados na identificação do número de documento. Dado que foram realizadas várias experiências, é possível que esses resultados tenham sido obtidos por acaso. Para confirmarmos as conclusões desse estudo, aplicámos o mesmo algoritmo a um conjunto de 15 documentos que não tinha sido usado anteriormente.

Construímos um conjunto de dados a partir de 15 novas facturas, de acordo com a estrutura e os pressupostos do conjunto de treino e teste. Este novo conjunto de dados é constituído por 510 registos. Destes 510 registos queremos descobrir os 15 números de documento, um para cada novo documento. A partir do Oracle Data Miner temos uma opção que nos permite aplicar os modelos gerados, para previsão das classes, em novos conjuntos de dados. A partir de um *wizard*, a aplicação é directa e não envolve a definição de novos parâmetros. Deve ser identificada qual o valor do atributo alvo e neste caso procuramos classificar os registos para a classe $EX = 1$. Como referido, o modelo para previsão das classes no novo conjunto de dados vai ser o que resultou no maior ganho, isto é, o modelo MDO gerado a partir do algoritmo com SVMs. Vamos assumir o valor de *Threshold* que nos garantiu uma maior redução de custos na fase de teste ao modelo, isto é, $Threshold = 0,1933$.

Na Tabela 18 estão resumidos os resultados da aplicação do modelo MDO aos novos dados:

		EX Predict		
		0	1	Total
EX real	0	418	77	495
	1	1	14	15
	Total	419	91	510

Tabela 18 – Matriz de confusão da aplicação do modelo MDO aos novos dados

Verificamos que foram descobertos 14 dos 15 números de documentos. Foram erradamente classificados 77 elementos como sendo número de documento o que equivale a uma média de 5 elementos erradamente sugeridos por cada documento. Em termos de custos, o esforço inicial de procurar o número de documento em 15 documentos é de 495. Este é o custo máximo e é obtido quando todos os elementos são classificados como positivos ($\text{Cost}=495+30*0= 495$), isto é, para $\text{Threshold} = 0$. Com este resultado, temos um custo de 107 o que equivale a um ganho de 388.

Documents		FP	FN	TP	TN	FP/Doc	Cost	Gain	%Gain
Test	12	49	1	11	316	4,08	79	286	78,36%
New	15	77	1	14	418	5,13	107	388	78,38%

Tabela 19 – Comparação dos resultados de teste e da aplicação aos novos dados

Na Tabela 19 comparamos o resultado obtido com este novo conjunto e o resultado obtido com o conjunto de teste. Em termos de percentagem, temos uma redução de custos 78,38% para os resultados obtidos com os novos dados e uma redução de custos 78,36% para os resultados obtidos no teste ao modelo. Estas percentagens são obtidas pesando o ganho em cada um dos casos com o custo máximo respectivo. Este resultado confirma os resultados obtidos no teste ao modelo SVM. Isto significa que os ganhos esperados em termos do esforço necessário para a identificação dos números dos documentos, nesta aplicação em particular, são os obtidos nos dados de teste.

6. Conclusão

O trabalho desenvolvido ao longo deste projecto demonstra que é possível uma abordagem que automatize ou semi-automatize a extracção de informação de documentos comerciais. No caso concreto da facturação, sabemos da necessidade de automatizar tarefas, diminuindo os tempos de processamento e os erros de introdução de dados. Esta pesada tarefa administrativa de inserir manualmente no sistema a informação contida nos documentos em papel, pode ser simplificada através do investimento em novas abordagens de tratamento e análise de dados. As técnicas de Data Mining são uma opção a considerar e os ganhos podem ser substanciais.

Na abordagem seguida, recolhemos e estruturamos a informação contida nos documentos em papel com do recurso ao OCR e ao XML. Avaliamos, através de vários algoritmos, de que forma as palavras vizinhas, no caso concreto, as 3 palavras vizinhas anteriores e as 3 palavras vizinhas posteriores, influenciam a classificação dos termos como sendo ou não número de documento e concluímos que o algoritmo das *Support Vectors Machines* é capaz de discriminar palavras contendo o número de documento de outras palavras candidatas. Verificamos que, em termos de custos, podemos ter reduções na ordem dos 78%.

A opção pelas tecnologias Oracle, em particular, o Oracle Data Mining e Oracle Data Miner, foi uma motivação adicional para este estudo e permitiu aproximar este trabalho de um ambiente real pela aplicação de uma ferramenta disponível no mercado. A ausência da necessidade de escrever código fonte, para as diversas funções disponibilizadas, permitir uma interacção inicial quase imediata com o analista de dados. As várias opções estão acessíveis a partir de *Wizards*. Os *Wizards* disponíveis para preparação de dados, construção e testes de modelos e aplicação a novos conjuntos de dados, permitem um rápido e fácil interface com o utilizador. A disponibilização dos resultados através de *viewers* e a possibilidade de exportação de dados permite uma rápida e fácil avaliação dos resultados. Vimos, no entanto, que existem situações particulares, em especial na preparação dos dados, em que é necessário o recurso a ferramentas de apoio.

6.1. Contribuições deste trabalho

Como contribuições deste trabalho destacamos:

- Descrevemos, detalhadamente, uma metodologia de extracção de informação, usando uma ferramenta comercial existente no mercado, partindo de documento originais, em papel, até à fase de avaliação e aplicação a novos documentos.
- Usamos uma ferramenta disponível no mercado, o Oracle Data Miner que permite um rápido e fácil interface com o utilizador, não havendo necessidade de escrever código para criação dos modelos.
- Demonstramos que com um reduzido número de documentos (29 de treino e 12 de teste) conseguimos aplicar os métodos de Data Mining e gerar modelos com bom desempenho na aplicação a novos conjuntos de dados.

6.2. Limitações

Destacamos algumas limitações desta abordagem:

- A metodologia descrita envolve a conjugação de várias tecnologias (OCR, XML, PL/SQL, OMD e Excel), o que obriga a uma coordenada integração de aplicações.
- O OCR utilizado e a má qualidade dos documentos condicionaram a fase inicial desta abordagem e impediram a aplicação desta metodologia num conjunto com maior número de registos
- Foram encontradas algumas limitações quanto ao pré-processamento deste conjunto de dados utilizando o Oracle Data Miner, em particular na divisão do conjunto em dados treino e teste.

6.3. *Trabalho Futuro*

Como trabalho futuro, que complementaria o trabalho até aqui realizado destaco os seguintes pontos:

- **Modelos sobre um maior conjunto de dados**

Seria interessante a aplicação destes métodos a um conjunto de dados de maior dimensão. O OCR utilizado gerou documentos editáveis de baixa qualidade, o que implicou a correcção manual dos dados e, conseqüente, a impossibilidade de se obter um conjunto mais alargado de dados. A utilização de um OCR com melhor desempenho permitiria, com menor esforço, testar num maior volume de dados.

- **Outras combinações de palavras vizinhas**

Seria interessante avaliar o efeito, nos resultados, de usar uma vizinhança maior ou menor da que foi usada. Por exemplo, gerar modelos só com as palavras vizinhas anteriores ou com duas anteriores e duas posteriores. Neste caso concreto, temos de considerar que foram recolhidas apenas três palavras vizinhas anteriores e três palavras vizinhas posteriores da palavra candidata a “número de documento”.

- **Sub-samples do conjunto de dados**

O conjunto de dados que serviu de base a este trabalho era um conjunto cujas classes são não balanceadas, uma vez que, por cada documento, apenas encontramos um “número de documento”. Poderíamos criar modelos para subconjuntos do conjunto de dados de forma a garantir um equilíbrio entre classes positiva e negativa.

- **Descoberta de outros itens**

A extracção de dados de documentos comerciais não pode focar-se apenas na descoberta do “número de documento”. Existe outra informação relevante sobre a qual poderemos desenvolver métodos para extracção de informação. O

“número de contribuinte” do fornecedor assim como o “valor total” da factura parecem ser *items* importantes a considerar.

- **Melhor uniformização das palavras**

A uniformização dos dados é uma abordagem que, do meu ponto de vista, valeria a pena investir. Deveriam ser avaliados métodos mais sofisticados para uniformização das palavras que contemplem não só as *strings* que representam o mesmo conceito, mas que diferem na apresentação devido, por exemplo, às abreviaturas ou à acentuação, mas também para o caso das *strings* que diferem devido à formatação (por exemplo, as data ou valores monetário). Isto resultaria em ganho na qualidade dos dados e que podem permitir a construção de modelos com mais capacidade de generalização.

Bibliografia

Belaïd, Y. e Belaïd, A. (2004). *Morphological Tagging Approach in Document Analysis of Invoices*. France : 17th International Conference on Pattern Recognition (IEEE) pp 469-472.

Bradley, P. S., Fayyad, Usama e Mangasarian, O. L. (1999). *Data Mining: Overview and Optimization Opportunities*. USA : INFORMS Journal on Computing.

Chang, Chia-Hui, Kayed, Mohammed, Girgis, Moheb Ramzy e Shaalan, Khaled (2006). *A Survey of Web Information Extraction Systems*. USA : IEEE Educational Activities Department, IEEE Transactions on Knowledge and Data Engineering, Vol 18, pp 1411-1428.

Chaudhuri, Bidyut B. 2007. *Digital Document Processing*: Springer, 2007.

Chen, Chien-Chih, Yang, Kai-Hsiang, Kao, Hung-Yu e Ho, Jan-Ming (2008). *BibPro: A Citation Parser Based on Sequence Alignment Techniques*. USA : Proceedings of the 22nd International Conference on Advanced Information Networking and Applications - Workshops table, pages 1175-1180.

Day, Min-Yuh, Tsai, Tzong-Han, Sung, Cheng-Lung, Lee, Cheng-Wei, Wu, Shih-Hung, Ong, Chorng-Shyong e Hsu, Wen-Lian. (2005). *A Knowledge-based Approach to Citation Extraction*. USA : Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE), pp 50-55.

Day, Min-Yuh, Tsai, Richard Tzong-Han, Sung, Cheng-Lung, Hsieh, Chiu-Chen, Lee, Cheng-Wei, Wu, Shih-Hun, Wu, Kun-Pin, Ong, Chorng-Shyong e Hsu, Wen-Lian. (2007). *Reference metadata extraction using a hierarchical knowledge representation framework*. The Netherlands : Decision Support Systems, Vol 43, pp 152-167.

Fayyad, Usama. (1996). *Mining Databases: Towards Algorithms for Knowledge Discovery*. USA: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol 21 , nº1.

Fayyad, Usama, Piatetsky-Shapiro, Gregory e Smyth, Padhraic. (1996). *From Data Mining to Knowledge Discovery in Databases*. USA: American Association for Artificial Intelligence.

Fayyad, Usama, Piatetsky-Shapiro, Gregory e Smyth, Padhraic. (1996a). *Knowledge Discovery and Data Mining: Towards a Unifying Framework*. USA : AAAI KDD-96 Proceedings.

Gaizauskas, Robert e Wilks, Yorick. (1998). *Information Extraction: Beyond Document Retrieval*. Sheffield : Computational Linguistics Society of R.O.C., vol 3, nº 2, pp 17 - 66.

Han, Jiawei e Kamber, Micheline. (2006). *Data Mining: Concepts and Techniques*. Illinois : Morgan Kaufmann, Second edition.

Hu, Yunhua, Li, Hang, Cao, Yunbo, Teng, Li, Meyerzon, Dmitriy e Zheng, Qinghua. (2005). *Automatic extraction of titles from general documents using machine learning*. USA : Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries.

Laender, Alberto, Ribeiro-Neto, Berthier, Altigran, Silva e Teixeira, Juliana. (2002). *A Brief Survey of Web Data Extraction Tools*. Brazil : ACM SIGMOD Record, vol 31, pp 84-93.

Lam, Man, Gong, Zhiguo e Mueyba, Maybin. (2008). *A method for Web Information Extraction*. USA : Springer - Lecture Notes in Computer Science, Vol 4976, pp 383-394.

Liu, Bing. (2006). *Web Data Mining.*: Springer.

Moens, Marie-Francine e Hiemstra, Djoerd. (2009). *Information Extraction and Linking in a Retrieval Context*. France : Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, pp 810-813.

Muslea, Ion, Minton, Steven e Knoblock, Craig. (2006). *Active Learning with Multiple Views*. USA : Journal of Artificial Intelligence Research.

Mylymaki, Jussi. (2002). *Effective Web Data Extraction with Standard XML Technologies*. Hong-Kong : ACM, Proceedings of the 10th international conference on World Wide Web, pp 689 - 696.

Oracle Corporation. (2008). *Oracle Data Mining Administrator's Guide 11g Release 1 (11.1)*. USA : Oracle. B28130-04.

Oracle Corporation. (2008). *Oracle Data Mining Application Developer's Guide 11g Release 1 (11.1)*. USA. Oracle. B28131-04.

Oracle Corporation. (2008). *Oracle Data Mining Concepts 11g Release 1 (11.1)*. USA : Oracle. B28129-04.

Oracle Corporation. (2008). *Oracle® Data Miner Tutorial for Oracle Data Mining*. USA: Oracle.

Oracle Corporation. (2007). *Oracle Data Mining 11g - An White Paper*. USA : Oracle.

Sitter, An De e Daelemans, Walter. (2003). *Information Extraction via Double Classification*. Antwerp, Belgium : International Workshop & Tutorial on Adaptive Text Extraction.

SPSS. (2000). *CRISP-DM 1.0 Step-by-step Data Mining Guide*. USA : SPSS Inc.-
<http://www.crisp-dm.org>.

Sun, Aixin, Naing, Myo-Myo, Lim, Ee-Peng e Lam, Wai. (2003). *Using Support Vector Machines for Terrorism Information Extraction*. Singapore : Springer - Lecture Notes in Computer Science.

Sundheim, Beth M. (1993). *Information Extraction System Evaluation*. USA : Proceedings of the workshop on Human Language, pp 147-163.

Turmo, Jordi, Ageno, Alicia e Català, Neus. (2006). *Adaptive information extraction*. USA : ACM Computing Surveys, Vol 38, nº 4.

Anexos

Scripts

```

DROP TABLE FileText;
CREATE TABLE FileText(      -- Tabela com conteúdo dos ficheiros
  Fch    VARCHAR2 (5),      -- Ficheiro
  TextF  VARCHAR2 (4000),   -- Texto
  PRIMARY KEY (Fch) );

DROP TABLE FileKW (
CREATE TABLE FileKW (      -- Tabela com palavras
  ID_PV    NUMBER (5),
  Fch      VARCHAR2 (5),    -- Ficheiro
  Tag      VARCHAR2 (20),   -- Tag
  KW       VARCHAR2 (100),  -- Palavra
  PosKW    NUMBER (4),      -- Posição da palavra
  EX       NUMBER (1),      -- Exemplo: 0 - Negativo/ 1 - Positivo
  KWViz    VARCHAR2 (100),  -- Palavra Vizinha
  PosKWViz NUMBER (1),      -- Posição da palavra vizinha face a KW
  PRIMARY KEY (ID_PV) );

DROP TABLE TESTE_F;
CREATE TABLE TESTE_F ( -- Tabela com conteúdo dos ficheiros
  F    VARCHAR2 (5),  -- Ficheiro
  TEXT VARCHAR2 (4000), -- Texto
  ORD  NUMBER );     -- ordem

DROP TABLE TESTE_FT;
CREATE TABLE TESTE_FT ( -- Tabela com tags
  FT VARCHAR2 (20) );

DROP TABLE TESTE_FW;
CREATE TABLE TESTE_FW ( -- Tabela com palavras
  F    VARCHAR2 (5),  -- Ficheiro
  FT   VARCHAR2 (20), -- Tag
  KW   VARCHAR2 (100), -- Palavra
  PKW  NUMBER (4),    -- Posição da palavra
  EX   NUMBER (1),    -- Exemplo: 0 - Negativo/ 1 - Positivo
  KWV  VARCHAR2 (100), -- Palavra Vizinha
  PKWV NUMBER (1) );  -- Posição da palavra vizinha face a KW: 1 - 1ª palavra vizinha após /
                        -- 2 - 2ª palavra vizinha após / ...
                        -- / -1 - 1ª palavra vizinha antes /
                        -- -2 - 2ª palavra vizinha antes / ...

```

```

-- Procedimento para extrair as palavras.O parâmetros são:
-- pF - ficheiro
-- pQP - quantidade de vizinhas anteriores
-- pQP - quantidade de vizinhas posteriores
-- pTag – tag a pesquisar

PROCEDURE ExtString(pF VARCHAR2, pQP NUMBER, pQS NUMBER, pTag VARCHAR2);
AS

    strCS    VARCHAR2(1) := ' ';
    strLS    VARCHAR2(1) := ' ';
';

CURSOR C1 IS
SELECT F, REPLACE(REPLACE(REPLACE(MAX(text1) || MAX(text2), strCS, ' '), strLS, ''), '---') text
FROM (
    SELECT t1.F, DECODE(ord, 1, t1.text, '') text1, DECODE(ord, 2, t1.text, '') text2
    FROM TESTE_F t1
    WHERE F = DECODE(pF, 'T', F, pF) )
GROUP BY f
ORDER BY F;

CURSOR C2 IS
SELECT FT
FROM TESTE_FT;

nPi    NUMBER(4);
nPf    NUMBER(4);
strAux  VARCHAR2(100);

nPVAuxi NUMBER(4);
nPVAuxf NUMBER(4);
strAuxV VARCHAR2(100);

nAux    NUMBER(2);
nEx     NUMBER(1);
nT      NUMBER(4);

BEGIN

DELETE FROM teste_fw
WHERE F = DECODE(pF, 'T', F, pF);

FOR C IN C1 LOOP

    nT := LENGTH(C.text);
    nPi := 1;
    nPf := INSTR(C.text, ' ', nPi);

    WHILE nPf != 0 LOOP

        nPf := INSTR(C.text, ' ', nPi);
        strAux := SUBSTR(C.text, nPi, nPf - nPi);

        FOR i in 0..9 LOOP

            nAux := INSTR(strAux, TO_CHAR(i), 1);
            IF nAux != 0 then

                IF INSTR(strAux, pTag, 1) = 0 THEN
                    nEx := 0;
                ELSE
                    nEx := 1;
                END IF;
            END IF;
        END LOOP;
    END LOOP;
END LOOP;

```

```

END IF;

-- #Prefixos
nPVAuxi := nT - nPi + 3;
FOR J in 1..pQP LOOP

    nPVAuxf := INSTR(C.text, '', (-1)*nPVAuxi, 1);
    IF nPVAuxf = 0 THEN
        EXIT;
    END IF;
    strAuxV := SUBSTR(C.text, nPVAuxf + 1, nT - nPVAuxi - nPVAuxf + 1);

    FOR T IN C2 LOOP
        strAux := REPLACE(strAux, T.ft);
        strAuxV := REPLACE(strAuxV, T.ft);
    END LOOP;

    INSERT INTO teste_fw(f, kw, pkw, ex, kwv, pkwv)
    VALUES(C.f, strAux, nPi, nEx, strAuxV, (-1)*j);

    nPVAuxi := nT - nPVAuxf + 2;
END LOOP;

-- #Suffixos
nPVAuxi := nPf + 1;
FOR J in 1..pQS LOOP

    nPVAuxf := INSTR(C.text, '', nPVAuxi, 1);
    IF nPVAuxf = 0 THEN
        EXIT;
    END IF;
    strAuxV := SUBSTR(C.text, nPVAuxi, nPVAuxf - nPVAuxi);

    FOR T IN C2 LOOP
        strAux := REPLACE(strAux, T.ft);
        strAuxV := REPLACE(strAuxV, T.ft);
    END LOOP;

    INSERT INTO teste_fw(f, kw, pkw, ex, kwv, pkwv)
    VALUES(C.f, strAux, nPi, nEx, strAuxV, j);

    nPVAuxi := nPVAuxf + 1;
END LOOP;
EXIT;

END IF;

END LOOP;

nPi := nPf + 1;
COMMIT;
END LOOP;

END LOOP;

END;
```

```

DROP VIEW AA_PV_H;
CREATE VIEW AA_PV_H AS
SELECT ROWNUM id_pvh, f, ft, ex, MAX(KWV_A3) a3, MAX(KWV_A2) a2,
      MAX(KWV_A1) a1, kw, MAX(KWV_D1) d1, MAX(KWV_D2) d2, MAX(KWV_D3) d3
FROM ( SELECT fch f, tag ft, kw, poskw pkw, ex,
      DECODE(PosKWViz, -3, KWViz, '') KWV_A3,
      DECODE(PosKWViz, -2, KWViz, '') KWV_A2,
      DECODE(PosKWViz, -1, KWViz, '') KWV_A1,
      DECODE(PosKWViz, 1, KWViz, '') KWV_D1,
      DECODE(PosKWViz, 2, KWViz, '') KWV_D2,
      DECODE(PosKWViz, 3, KWViz, '') KWV_D3
    FROM FileKW )
GROUP BY f, ft, ex, kw
ORDER BY f, pkw;

```

--Tabela palavras e vizinhas

```

drop TABLE AA_PV_H ;
CREATE TABLE AA_PV_H (
  id_pvh number,
  fch      char(3),
  tag_fhc varchar2(12),
  ex       number,
  kw       varchar2(35),
  a1       varchar2(35),
  a2       varchar2(35),
  a3       varchar2(35),
  d1       varchar2(35),
  d2       varchar2(35),
  d3       varchar2(35),
  primary key (id_pvh) );

```

```

DROP VIEW VW_AA_PV_H;
CREATE VIEW VW_AA_PV_H AS
SELECT ID_PVH, FCH, TAG_FHC, EX, UPPER(KW) KW, UPPER(A1) A1, UPPER(A2) A2,
      UPPER(A3) A3,  UPPER(D1) D1, UPPER(D2) D2, UPPER(D3) D3
FROM AA_PV_H

```

--Tabela palavras e vizinhas classificadas em treino (B) e teste (T)

```

CREATE TABLE AA_PVH_BT (
  ID_PVH  NUMBER      NOT NULL,
  TIPO    CHAR (1),
  FCH     CHAR (3),
  TAG_FHC VARCHAR2 (12),
  EX      NUMBER,
  KW      VARCHAR2 (35),
  A1      VARCHAR2 (35),
  A2      VARCHAR2 (35),
  A3      VARCHAR2 (35),
  D1      VARCHAR2 (35),
  D2      VARCHAR2 (35),
  D3      VARCHAR2 (35),
  PRIMARY KEY ( ID_PVH ) );

```

--Divisão dos dados em treino e teste

```
INSERT INTO aa_pvh_bt
SELECT id_pvh, tipo, A.fch, tag_fhc, ex, kw, a1, a2, a3, d1, d2, d3
FROM vw_aa_pv_h a,
      (SELECT DECODE(SIGN(ROWNUM-30), -1, 'B', 'T') tipo, fch
       FROM ( SELECT fch
               FROM vw_fch
               ORDER BY dbms_random.value() )) b
WHERE a.fch = b.fch
```

-- Criação View de treino e view de teste

```
DROP VIEW VW_AA_PVH_B;
CREATE VIEW VW_AA_PVH_B AS
SELECT ID_PVH, FCH, TAG_FHC, EX, KW, A1, A2, A3, D1, D2, D3
FROM AA_PVH_BT
WHERE TIPO = 'B'
```

```
DROP VIEW VW_AA_PVH_T;
CREATE VIEW VW_AA_PVH_T AS
SELECT ID_PVH, FCH, TAG_FHC, EX, KW, A1, A2, A3, D1, D2, D3
FROM AA_PVH_BT
WHERE TIPO = 'T'
```

-- Uniformização das palavras através da eliminação de caracteres especiais e redução aos 4 1's caracteres

```
CREATE OR REPLACE VIEW VW_T4_B AS
SELECT ID_PVH, FCH, EX,
SUBSTR(NVL(TRANSLATE(a1, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a1,
SUBSTR(NVL(TRANSLATE(a2, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a2,
SUBSTR(NVL(TRANSLATE(a3, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a3,
SUBSTR(NVL(TRANSLATE(d1, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d1,
SUBSTR(NVL(TRANSLATE(d2, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d2,
SUBSTR(NVL(TRANSLATE(d3, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d3
FROM AA_PVH_BT
WHERE TIPO = 'B'
```

```
CREATE OR REPLACE VIEW VW_T4_T AS
SELECT ID_PVH, FCH, EX,
SUBSTR(NVL(TRANSLATE(a1, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a1,
SUBSTR(NVL(TRANSLATE(a2, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a2,
SUBSTR(NVL(TRANSLATE(a3, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) a3,
SUBSTR(NVL(TRANSLATE(d1, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d1,
SUBSTR(NVL(TRANSLATE(d2, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d2,
SUBSTR(NVL(TRANSLATE(d3, 'ÃÄÅÖËÊÇÚÍ1234567890)*+%-V:.,'aoO', 'AAAOOEECUI1234567890'), '##'),
1, 4) d3
FROM AA_PVH_BT
WHERE TIPO = 'T'
```