Universidade do Minho

Escola de Engenharia

Departamento de Informática

Previsão de *Churn* em Companhias de Seguros

Bruno Miguel Viana Gomes

Dissertação de Mestrado

2011

Previsão de Churn em Companhias de Seguros

Bruno Gomes

Dissertação apresentada à Universidade do Minho para obtenção do grau de Mestre em Engenharia Informática, na especialidade de Sistema de Suporte à Decisão, elaborada sob orientação do Professor Doutor Orlando Manuel de Oliveira Belo.

2011

Dedicado à minha mãe, que me ensinou a não desistir.

"Our greatest weakness lies in giving up.
The most certain way to succeed is always
to try just one more time."
Thomas A. Edison

Agradecimentos

À minha família, pelo apoio psicológico.

Aos meus colegas e amigos, que facilitaram a tarefa.

Ao meu professor e orientador Orlando Belo, pela paciência.

À Deloitte Consultores S.A., pelo apoio no tema e pela disponibilização dos meios.

Resumo

Previsão de *churn* em companhias de seguros

Transversal a qualquer indústria, a retenção de clientes é um aspecto de elevada importância e a que se deve dar toda a atenção possível. O abandono de um produto ou de um serviço por parte de um cliente, situação usualmente denominada por *churn*, é cada vez mais um indicador a ter em atenção por parte das empresas prestadoras de serviços. Juntamente com técnicas de *Customer Relationship Management* (CRM), a previsão de *churn*, oferece às empresas uma forte vantagem competitiva, uma vez que lhes permite obter melhores resultados na fidelização dos seus clientes. Com o constante crescimento e amadurecimento dos sistemas de informação, torna-se cada vez mais viável a utilização de técnicas de *Data Mining*, capazes de extrair padrões de comportamento que forneçam, entre outros, informação intrínseca nos dados, com sentido e viável no domínio do negócio em questão. O trabalho desta dissertação foca-se na utilização de técnicas de *Data Mining* para a previsão de situação de *churn* dos clientes no ramo das seguradoras, tendo como o objectivo principal a previsão de casos de *churn* e, assim, possibilitar informação suficiente para a tomada de acções que visem prever o abandono de clientes. Nesse sentido, foi desenvolvido nesta dissertação um conjunto de modelos preditivos de *churn*, estes modelos foram implementados utilizando diferentes técnicas de *data mining*. Foram estudadas as técnicas de árvores de decisão, redes neuronais, regressão logística e SVM. A implementação de vários modelos usando este conjunto de técnicas permitiu concluir que as árvores de decisão e as regressões logísticas são as técnicas mais adequadas para realizar a previsão de *churn* em companhias de seguros, e adicionalmente fazer um levantamento e análise sobre os indicadores mais relevantes para a sua previsão.

Abstract

Insurance churn prediction

Transversal to any industry, customer retention is a highly important aspect and that we should give all possible attention. The abandonment of a product or a service by a customer, a situation usually referred to as churn, is an indicator that the service provider company should take in attention. Along with techniques of Customer Relationship Management (CRM), the churn prediction offers to companies a strong competitive advantage since it allows them to get better results in customer retention. With the constant growth and maturity of information systems, it becomes more feasible to use data mining techniques, which can extract behavior patterns that provide intrinsic information hided in the data. This dissertation focuses on using data mining techniques for predicting customer churn situations in insurance companies, having as main objective the prediction of cases of churn and thereby allow information gathering that can be used to take actions to avoid the customer desertion. In this dissertation we develop a set of predictive churn models using different data mining techniques. We studied the following techniques: decision trees, neural networks, logistic regression and SVM. The implementation of various models using this set of techniques allowed us to conclude that the most suitable techniques to predict churn in an insurance company are decision trees and logistic regression, in addiction we did a study about the most relevant churn indicators.

Índice

Índice de Figuras

Índice de Tabelas

Capítulo 1

1 Introdução

1.1 Enquadramento

Com o crescimento tecnológico exponencial que se registou nas últimas décadas, é cada vez mais fácil capturar e armazenar dados de negócio. Na verdade, é praticamente impossível encontrar actualmente uma empresa de média ou de grande dimensão que não use sistemas avançados de informação para a melhoria da gestão do seu negócio. No entanto, este aumento significativo na facilidade em obter e armazenar grandes volumes de dados não foi devidamente acompanhado pela capacidade para interpretar esses mesmos dados [Fayyad, et. al., 1996a]. Apesar destes serem armazenados, na maioria dos casos de forma bem estruturada e orientada ao negócio, não são, numa grande maioria dos casos, de fácil interpretação, ao se utilizar meios de análise convencionais. É portanto comum, existirem casos reportando um grande volume de dados históricos, com informação válida, que poderá potencialmente possuir um grande valor comercial, mas que no entanto não se sabe como extrair e gerar desses dados alguma informação válida e útil para as actividades empresariais correntes [Fayyad, et. al., 1996a]. As situações referidas anteriormente possuem frequentemente no seu histórico de dados, padrões implícitos que se encontram por descobrir, padrões de informação estes que podem ter grande valor, mas que no entanto, devido ao seu grande volume e complexidade, não são capazes de ser lidos e obtidos por

meio de técnicas de análise convencionais [Lloyd-Williams, et. al., 1995]. É neste contexto que surge em 1989 o conceito de KDD (*Knowledge-Discovery in Databases*). O KDD surge como um ramo da computação, que tem como principal objectivo a extracção de conhecimento útil a partir de dados, conhecimento este que não seria capaz de ser detectado usando as técnicas de análise usadas até ao momento. Conceito idêntico, o *Data Mining*, não deve ser confundido com o processo de KDD, pois representa apenas uma das fases do KDD. Enquanto o KDD é visto como o processo de descoberta de informação útil a partir de dados, o *Data Mining* é interpretado como sendo a aplicação de determinados algoritmos para a extracção de padrões, sem as fases adicionais que o KDD implica [Fayyad, et. al., 1996b].

1.2 Data Mining

Nos últimos anos o *Data Mining* tornou-se cada vez mais uma opção viável e de grande valor, tendo em conta, os referidos avanços nas tecnologias de exploração de dados, tanto no que toca a capacidade de processamento e de armazenamento de dados, como também ao aumento progressivo dos repositórios de dados. Estes avanços, em conjunto com algoritmos e técnicas estatísticas, deram origem ao processo de obtenção de conhecimento de forma automática, que hoje é sobejamente reconhecido por Data Mining (Figura 1). Assim sendo, o *Data Mining* aparece como uma solução para o problema da obtenção de informação a partir de grandes volumes de dado, colocando à disposição dos analistas de dados várias técnicas que lhes permitem extrair dos dados à sua disposição, padrões de comportamento relevantes para o negócio ou área em questão [Kantardzic, 2003].

Figura 1 – Elementos que contribuíram para a origem do *Data Mining*

O *Data Mining* é, por definição, o processo de extrair padrões implícitos nos dados operacionais [Thearling, 1999]. O *Data Mining* é nos dias de hoje, uma ferramenta imprescindível nos modelos de negócio actuais, permitindo transformar os dados de negócio em informação concreta e fiável para suporte a processos de tomada de decisão, traduzindo-se, na prática, numa vantagem estratégica que não deve ser ignorada. A utilização de técnicas de *Data Mining* permite a extracção de informação a partir de grandes volumes de dados nos quais os métodos tradicionais não se revelam muito fiáveis. Posteriormente, esta informação pode ser usada num largo espectro de aplicações práticas, que podem ser aplicadas em situações de previsão de eventos específicos, de segmentação de clientes ou mesmo na detecção de fraude em variadíssimas áreas [Kantardzic, 2003].

Essencialmente, as técnicas de *Data Mining* podem ser agrupadas em quatro classes distintas, a saber: algoritmos de *clustering*, algoritmos de classificação, algoritmos de regressão, e regras de associação. Os algoritmos de *clustering* permitem organizar automaticamente os dados, em sub-sets (*clusters*), para que os dados pertencentes a um determinado sub-set apresentem características idênticas. Os algoritmos de classificação são uma técnica de *Data Mining*, que permitem através de um grupo de dados inferir regras de classificação, de forma a ser possível classificar novos dados. As regras de classificação contrastam com as de *clustering*, uma vez que nas técnicas de *clustering* é analisado um só conjunto de dados, sendo determinado automaticamente o número de grupos a criar e as características que vão determinar essa mesma classificação, enquanto nas regras de classificação, é definido pelo utilizador, quais as classes em que os dados originais se vão dividir. Por estas razões, o *clustering* é classificado como um método de aprendizagem não supervisionada (*unsupervised learning*), enquanto os algoritmos de associação, são classificados como métodos de aprendizagem supervisionada (*supervised learning*). Quanto aos algoritmos de regressão, estes permitem explorar e inferir a relação de uma variável dependente (variável que queremos prever) com variáveis independentes. Para além de ajudar a compreender a relação existente entre as variáveis em relação à variável dependente, ou variável *target*, permite também construir modelos de previsão e de *forecasting*, capazes de prever, a partir das variáveis independentes, qual será o valor que a variável dependente irá tomar. Já os algoritmos de associação, conhecidos simplesmente por regras de associação, são uma técnica bastante conhecida e bastante utilizada para extrair relações entre variáveis em bases de dados de grande dimensão [Han, 1996].

1.3 Churn e CRM nas Companhias de Seguros

Churn, *customer attrition, customer turnover* ou *customer defection,* são diferentes termos com o mesmo significado: taxa de abandono de um determinado produto ou serviço pelos seus clientes. O *churn* é um indicador de negócio que não se deve subestimar. Num mercado onde a competição é cada vez mais implacável, as empresas tem que centrar o seu modelo de negócios no cliente. De forma a proteger o seu bem mais precioso – o próprio cliente. Em comparação com outros ramos de actividades, as seguradoras tem que lidar com uma elevada taxa de abandono de clientes. Em comparação à indústria bancária, por exemplo, onde a taxa de *churn* é em média de 7%, nos seguros no ramo automóvel a média chega aos 18%. Estes rácios de abandono podem ser extremamente comprometedores para o crescimento, manutenção e sobrevivência de uma companhia de seguros. Na verdade, num estudo de 2004, levado a cabo pela McKinsey & Company, previa-se que uma taxa de *churn* de cerca de 10% concentrada entre os clientes mais valiosos de uma seguradora poderia implicar uma redução dos lucros na ordem dos cerca de 40% [Giuliani, et. al., 2004]. O mesmo estudo, permitia concluir ainda que, se a anterior taxa fosse verificada entre os clientes menos lucrativos, a companhia melhoraria o seu *combined racio* em 1% e os seus lucros em cerca de 25%.

Depois de analisados os exemplos anteriores, torna-se clara a importância de saber distinguir em quais os clientes em que se deve investir, e também quanto se deve investir em cada um desses clientes de modo a impedir o seu abandono. Esta necessidade premente, de manter a carteira de clientes, gerindo o negócio com ênfase no cliente, traduziu-se por parte das empresas numa maior procura e implementação de sistemas e estratégias de CRM (*Customer Relationship Management*), ou se preferirmos na terminologia portuguesa, gestão do relacionamento com o cliente.

Um sistema de CRM tem como objectivo auxiliar a companhia a criar e manter um bom relacionamento com os seus clientes, colocando-os no centro do modelo de negócios. Estes sistemas permitem armazenar o histórico das actividades e da relação do cliente com a companhia, permitindo perceber e antecipar as suas necessidades. O CRM, juntamente com o problema da fidelização e obtenção de novos clientes, tem sido um aspecto que as companhias dedicam cada vez mais atenção. É assim possível com as actuais ferramentas de CRM, operar de acordo com o valor que o cliente representa para a empresa ao longo da sua vida, esta métrica de *marketing* é conhecida como CLV (*Custumer Lifetime Value*). O CLV permite a partir da informação da relação entre a empresa e o cliente, estimar o valor do cliente durante o seu tempo de vida, esta métrica permite em termos teóricos, representar em termos monetários o valor de um cliente. O cálculo do

CLV tem em consideração os seguintes indicadores: a) o tempo estimado de vida restante do cliente; b) a previsão dos lucros gerados pelo cliente nos próximos anos, tendo em conta o histórico de compras do cliente; c) o custo de produção e entrega dos produtos anteriores; d) cálculo do valor presente líquido dos futuros ganhos. Ao empregar métricas como o CLV é possível estimar quanto se irá perder devido ao abandono de determinados clientes, e de quanto deverá ser o esforço de manter esses mesmos clientes. É portando, muito importante criar mecanismos para que seja possível implementar de forma efectiva um sistema de *churn management,* que vise prevenir o abandono de clientes com algum tempo de antecedência, para que a companhia possa reagir de forma atempada, no sentido de contrariar, se possível, esse abandono. O termo *churn* [Berson, et. al., 2000] foi utilizado pela primeira vez no contexto de abandono de clientes, para descrever o abonado dos mesmos na indústria de serviços de comunicação móveis. Já o termo *churn management* é o processo de reter os clientes mais rentáveis [Kentrias, 2001]. Em essência, um correcto uso de um sistema de *churn management*, permitirá a uma empresa prever o abandono dos clientes e agir mediante o seu valor, ou seja, mediante o que a sua perda representa para a empresa, tomando medidas estratégicas com o intuito de combater esse abandono [Lariviere & Van den Poel, 2004]. As companhias de seguros oferecem, na generalidade dos casos, inúmeros tipos de produtos e soluções aos seus clientes, produtos estes que variam tanto no objecto seguro como nas modalidades possíveis de adquirir pelo cliente. De forma simplificada e em termos de introdução ao domínio em estudo, de seguida apresentamos uma breve descrição dos tipos de seguros com que vamos, de alguma forma, lidar neste trabalho.

Tipicamente, uma companhia de seguros divide a sua actividade em dois grandes grupos de ramos:

- **Ramo vida** – O "objecto" seguro é uma pessoa e, dependendo da modalidade, o pagamento é devido em caso de vida ou de morte da pessoa segura. Este tipo de seguro não tem carácter indemnizatório.
- **Ramo não-vida** – compreende os seguros que cobrem riscos de acidentes envolvendo pessoas e/ou de danos em coisas, sendo um tipo de seguro com carácter indemnizatório.

O ramo vida compreende produtos como: seguros de vida, risco e produtos financeiros, como por exemplo PPR (Planos Poupança Reforma), fundos de investimento e operações de capitalização. Já o ramo não-vida engloba um grande leque de produtos, como acidentes pessoais, seguros de saúde, seguros multirrisco (fogo, inundações, etc.), seguros automóveis, seguros de transporte,

responsabilidade civil, entre outros. Devido às inúmeras diferenças ao nível da lógica do negócio existente nos diferentes tipos de seguros dos vários ramos, seria de elevada complexidade estudar e implementar um modelo de previsão para *churn* que fosse transversal ao ramo, isto é, que fosse capaz de se adaptar aos clientes independentemente do tipo de seguros que possuam. Como tal, o trabalho desenvolvido nesta dissertação de mestrado, focar-se-á, na previsão de *churn* apenas em clientes que possuem produtos do ramo automóvel.

1.4 Motivação e Objectivos

No cerne da motivação para a realização deste trabalho está a crescente urgência do uso da informação de gestão das empresas e organizações, com o intuito de melhorar os seus processos de gestão e atingir melhores resultados. Exemplo disso é a possibilidade de usar estes dados históricos, para extrair informação de valor que permita implementar modelos preditivos, com o intuito de obter melhores resultados em áreas vitais como por exemplo a retenção de clientes. É nesta linha de raciocínio que assenta o trabalho desenvolvido nesta dissertação.

O principal objectivo desta dissertação é portanto analisar a viabilidade da implementação de modelos preditivos de *churn* orientados às companhias de seguros. Para esta análise torna-se imperativo estudar e fazer uma comparação crítica das diferentes técnicas à disposição para a implementação do dito modelo preditivo. Em última análise irá se tentar determinar qual a técnica mais eficaz, e em que condição é atingida essa eficácia. Para isto foram implementados vários modelos preditivos de *churn* empregando diferentes técnicas de *data mining*, e os resultados foram comparados à luz de algumas métricas de desempenho, de forma a ser possível fazer uma análise crítica dos modelos implementados.

1.5 Organização do Documento

A parte restante deste documento encontra-se organizada da seguinte maneira: no capítulo 2 é apresentada a metodologia usada na implementação do projecto de *data mining* desenvolvido nesta dissertação, mais concretamente, a metodologia CRISP-DM, uma metodologia *standard* para o desenvolvimento de projectos de *data mining*. Além disso, apresenta-se o âmbito do projecto, no qual foram apresentados os dados usados e várias definições técnicas necessárias para o desenvolvimento do trabalho, bem como uma pequena reflexão acerca de como foram realizadas as etapas de análise e tratamento dos dados, também é abordado o tema das métricas usadas para a avaliação dos modelos implementados e ainda o *software* usado no desenvolvimento dos

modelos. De seguida, no capítulo 3, são apresentadas as quatro famílias de algoritmos de *data mining* com maior influência no domínio do problema em estudo nesta dissertação, bem como os algoritmos seleccionados para o desenvolvimento dos modelos testados. No capítulo 4 apresenta-se o caso de estudo de *churn* trabalhado nesta dissertação, os dados utilizados, os resultados da análise obtidos, o tratamento de dados realizado e também são apresentados os modelos implementados, bem como as suas várias configurações. No capítulo 5 são apresentados os resultados dos modelos e é feita uma análise crítica e comparativa entre os vários modelos. Neste capítulo são também apresentados dois casos de aplicação dos modelos seleccionados como sendo os mais eficazes no tipo de problema trabalhado. Para concluir, no capítulo 6 são apresentadas as conclusões sobre o trabalho desenvolvido ao longo desta dissertação, bem como os principais desafios encontrados e possíveis pontos de melhoria a ter em conta no futuro.

Capítulo 2

2 A Metodologia de Implementação

2.1 A Metodologia CRISP-DM

2.1.1 Introdução

No desenvolvimento desta dissertação foi seguida uma metodologia baseada no *standard* para a implementação de projectos de *data mining:* o CRISP-DM (*Cross-Industry Standard Process for Data Mining*). O CRISP-DM é uma metodologia de DM (*data mining*) que define uma abordagem a seguir na implementação de projectos de *data mining,* ajudando também na resolução de problemas tipo típicos em projectos de DM [Chapman, et. al., 2000].

Em 1996, quando o DM era uma área ainda pouco conhecida, mas que ia ganhando adeptos a cada dia que passava, não existia uma metodologia de implementação de processos de DM bem definida e documentada. A necessidade de um *standard*, que fosse independente da indústria, gratuito, sem proprietário e capaz de auxiliar as industria no desenvolvimento dos seus projectos de DM seguindo boas práticas, impeliu quatro líderes da área de DM a desenvolver o CRISP-DM. Estes quatro líderes foram: a Daimler-Benz, a Integral Solutions Ltd, a NCR e a OHRA. No ano seguinte vir-se-ia a formar um consórcio – CRISP-DM SIG (CRISP-DM *Special Interest Group*) – com o objectivo de aperfeiçoar a metodologia. O SIG foi aperfeiçoando a metodologia

nos anos seguintes, tendo em consideração opiniões dada por vários profissionais, bem como de vendedores de software para *data warehouses*, com o objectivo de garantir a independência de indústria e ferramentas. Em 2000, quatro anos depois de ver a luz do dia, foi apresentada a versão 1.0 do CRISP-DM, espelhando as melhorias que foram sendo adoptadas desde a apresentação da metodologia original. Em Julho de 2006 foi anunciado pelo CRISP-DM SIG que se iriam iniciar os trabalhos para o planeamento da segunda versão da metodologia, e já em Setembro do mesmo ano o consórcio reuniu-se para realização do planeamento do mesmo. No entanto, até ao momento, não foi avançada qualquer novidade sobre o desenvolvimento da nova versão [Shearer, 2000].

O CRISP-DM é actualmente a metodologia mais seguida pelos especialistas de DM, inclusive, em votações realizadas pela KDNuggets [5] em 2002, 2004 e 2007, o CRISP-DM apresentava-se como a metodologia mais usada, sempre com larga vantagem em relação às suas concorrentes.

2.1.2 Vantagens do CRISP-DM

A utilização da metodologia CRISP-DM permite várias vantagens, nomeadamente: torna a implementação de projectos de DM mais rápida, mais simples, mais barata, e mais fácil de gerir; é independente da indústria e da ferramenta de mineração de dados; é ainda idêntica à filosofia presente no ramo KDD. Ao definir uma metodologia a seguir, bem documentada e facilmente aplicável, permite-se simplificar a implementação de projectos de DM, tornando-os mais rápidos, mais baratos e mais fáceis de gerir. A CRISP-DM é também independente da indústria e da ferramenta usada para a implementação. Isto quer isto dizer que pode ser usada independentemente do negócio em causa (saúde, comercio, retalho, etc.), bem como fazendo uso de qualquer ferramenta de *data mining*. A metodologia relaciona-se ainda com o conhecido paradigma KDD, já familiar aos profissionais de *data mining*, o que facilita ainda mais a adopção da metodologia.

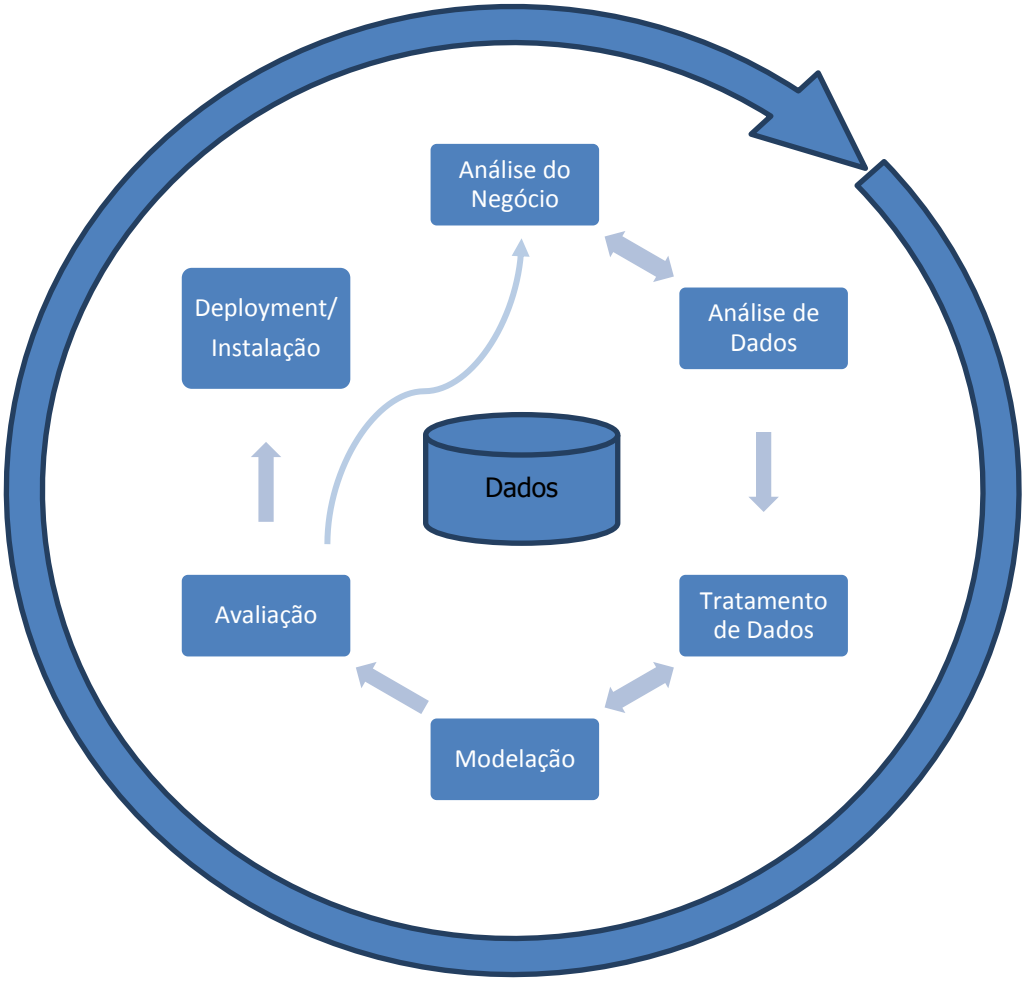

Figura 2 - Fases do CRISP-DM

O processo de CRISP-DM divide-se em várias fases. Na Figura 2 é possível observar a natureza cíclica do processo, as dependências entre as várias fases, e a possibilidade de a execução de uma fase encadear novas questões, mais focadas e específicas, que originam um regresso à fase anterior. As várias fases da CRISP-DM são apresentadas com mais detalhe na secção seguinte [Chapman, et. al., 2000].

2.1.3 Fases do CRISP-DM

A CRISP-DM é uma metodologia de *data mining*, que permite a todos os profissionais de *data mining*, do mais inexperiente até ao utilizador mais especialista*,* seguir um mapa, ou desenho técnico, que lhe permite conduzir todo o projecto de *data mining*, de forma simples, eficiente e com sucesso. O CRISP-DM divide o processo em seis fases distintas:

1. Análise do Negócio.
2. Análise dos Dados.
3. Tratamento dos Dados.
4. Modelação.
5. Avaliação.
6. *Deployment*/Instalação.

Cada uma destas fases compreende ainda um conjunto de subfases ou tarefas, apresentadas na Tabela 1. De seguida é feita uma breve descrição de cada uma das fases e das tarefas que as compõem.

A primeira fase, **análise do negócio**, é possivelmente a fase mais importante de todo o processo, já que uma má implementação desta fase implicará maus resultados em todas as outras fases, podendo por em causa o sucesso de todo o projecto. Esta fase divide-se em várias subfases. Iniciamos o trabalho com a análise do problema do ponto de vista do negócio ou funcional e a compreensão dos objectivos do ponto de vista do cliente, pois só após uma compreensão profunda do tema em questão e só após se compreender o que o cliente realmente pretende, é que se poder avançar para a próxima fase do projecto. Nesta fase é também definido os objectivos do ponto de vista lógicos, tal como por exemplo, no caso de um projecto para a redução do abandono de clientes, o objectivo poderia ser por exemplo, diminuir a perda de clientes em 20%. Depois de conhecido o negócio, são avaliadas as condições para a realização do projecto, entre as quais se faz uma avaliação dos recursos humanos e tecnológicos disponíveis, dos dados disponíveis, etc.. Aqui é, portanto, analisada a viabilidade do que se pretende fazer. De seguida esta informação é transformada num problema de *data mining*, no qual é definido qual será o objectivo do ponte de vista técnico, no caso em estudo, poderia ser por exemplo, modelar um modelo preditivo com uma precisão de pelo menos 80%. Após a compreensão do problema do ponto de vista do negócio, de analisada a viabilidade e de definidos os objectivos a alcançar, é elaborado um plano de projecto.

No plano de projecto deve-se especificar como se pretende obter os objectivos técnicos definidos anteriormente, este plano inclui o *timeline* do projecto, uma análise dos potenciais riscos, e um levantamento das ferramentas e técnicas a usar para atingir os objectivos.

Fases	Subfases
Análise do Negócio	Análise do problema do ponto de vista funcional.
	Avaliação da situação.
	Definição dos objectivos de *Data Mining.*
	Planeamento do projecto.
Análise dos Dados	Recolha de dados iniciais.
	Descrição dos dados.
	Exploração dos dados.
	Qualidade de dados.
Tratamento dos Dados	Selecção de dados.
	Limpeza de dados.
	Transformação de dados.
	Integração dos dados.
	Formatação de dados.
Modelação	Selecção das técnicas.
	Planificação da avaliação.
	Modelação.
	Avaliação.
Avaliação	Avaliação dos resultados.
	Reavaliação dos resultados.
	Decisão das próximas tarefas.
***Deployment*/Instalação**	Planeamento da instalação.
	Planeamento da monitorização e manutenção.
	Realização do relatório final.
	Revisão do projecto.

Tabela 1 – CRISP-DM: Fases e as suas tarefas

A segunda fase, a **análise dos dados**, é onde o analista se familiariza com os dados com que vai trabalhar. Compreende uma primeira etapa – a recolha de dados iniciais – que consiste em

carregar, e por vezes integrar de forma uniforme, os dados provenientes de diferentes fontes. Após esta recolha dos dados iniciais, é feita uma descrição dos dados, esta fase consiste numa análise superficial dos dados, de modo a recolher informação sobre os mesmos. É recolhida informação sobre os formatos, a quantidade dos dados, o número de campos e registos em cada tabela, entre outro tipo de informação. O mais importante desta fase é determinar se os dados obtidos satisfazem os requisitos para a modelação que se segue, e também, durante o processo, ficar a conhecer as características dos dados que se irá usar. Após a descrição dos dados, onde é feito um levantamento sobre as características dos dados, é realizada a fase de exploração dos dados, em que são explorados os dados de forma mais profunda, realizando-se pesquisas aos dados, visualizando-os e gerando relatórios, de maneira a compreender as suas características não superficiais. Nesta fase o analista deve-se concentrar em analisar os dados de forma orientada à temática do *data mining*, tentando descobrir, de forma preliminar, padrões ou relações entre os dados. A última etapa da análise dos dados consiste em estudar a qualidade dos dados. Nesta etapa são analisadas questões como a existência de valores a nulo e a branco (*missing values* e *blank fields*), muitas vezes resultantes de recolha de dados ao longo de um longo período temporal. É também analisado o universo de valores possíveis para cada atributo, ou seja, se os valores que um atributo toma fazem sentido (por exemplo uma variável que indique a idade contendo valores negativos não faz sentido). É ainda estudada nesta fase a existência de atributos com o mesmo significado, a existência de *outliers*, e também a existência de valores que contradigam o senso comum, por exemplo, idades demasiado elevadas, ou jovens com rendimentos demasiado altos.

Depois de compreendido o negócio e de analisado os dados, o analista tem condições para passar à terceira fase: o **tratamento dos dados**. Nesta fase, o analista usa o conhecimento adquirido nas duas fases anteriores para preparar os dados, de modo a estes poderem ser sujeitos à ferramenta de modelação. Esta fase compreende várias tarefas sobre os dados, entre elas a selecção de dados, a sua limpeza, a sua transformação, a sua integração e a sua formatação. A selecção de dados consiste, como a própria palavra indica, em seleccionar os dados que serão utilizados de facto no projecto e aqueles que serão descartados. Esta selecção é feita tendo em conta os objectivos da modelação e as restrições técnicas, isto é, deveremos descartar os atributos que nada ou pouco estejam relacionados com o objectivo do projecto, bem como as variáveis que, apesar de poderem ser úteis, a sua volumetria torna o seu uso proibitivo. Esta selecção de atributos deve ser bem documentada e justificada. Por seu lado, a limpeza dos dados consiste em tratar casos de valores em falta e valores em branco, bem como casos de *outliers*. Esta fase é feita

tendo em conta a fase final de análise dos dados na qual ocorre a verificação da qualidade dos dados. Após a limpeza dos dados ter sido executada, é feita a construção de dados, nesta fase são construídos registos completamente novos ou são gerados outros atributos derivados. Um exemplo de um registo novo é, por exemplo, a criação de uma venda vazia, para os clientes que não efectuaram encomendas no último ano. Para um exemplo de criação de um atributo derivado poderemos apresentar o número de dias que passaram desde a última compra ter sido efectuada. Outro tipo de atributo derivado pode ser a transformação de apenas um dos atributos já existentes, de maneira a adaptar-se aos requisitos das ferramentas de *data mining* usadas. Exemplo disto é a transformação de atributos contínuos em atributos categóricos, na qual se faz uma divisão do atributo em intervalos, ou também da transformação de atributos categóricos, para atributos numéricos com valor explícito, como por exemplo um atributo constituído pelos valores 'Não Satisfaz', 'Satisfaz', 'Satisfaz bastante', ou 'Excelente', para um atributo numérico com valores de 1 a 5. A fase de integração dos dados consiste em integrar num só local, os dados provenientes de fontes diferentes, mas que se referem à mesma entidade. Por exemplo, o caso de uma tabela com informação das vendas num armazém X e outra com informação das vendas de um outro armazém Y. Apesar de conterem ambas informação referente às vendas, encontram-se em localizações diferentes e, como tal, podem ter estruturas diferentes, tornando-se necessário integrar e agregar os dados das duas tabelas. Por vezes é também necessário realizar a fase de formatação de dados, que ocorre quando, por alguma razão, é necessário formatar os dados existentes, tais como alterar o tamanho, ou remover algum tipo de carácter, de forma a tornar os dados viáveis para uso.

A fase de **modelação** apenas se pode iniciar após a fase de tratamento de dados. No entanto pode-se voltar para a fase anterior de modo a realizar tarefas de tratamento de dados, com o intuito de melhorar os modelos gerados. Aliás, deve-se salientar que este é o comportamento típico na maioria dos projectos de *data mining*. Nesta fase são seleccionados várias técnicas de *data mining* capazes de lidar com o problema em questão. Estas técnicas são depois aplicadas sendo os seus parâmetros afinados de forma a tentar encontrar valores óptimos para o problema. Esta fase é composta por quatro subfases: a selecção das técnicas de modelação, a geração do *designe* de teste, a criação dos modelos, e a avaliação dos modelos. A selecção das técnicas de modelação consiste em seleccionar entre as várias técnicas de *data mining* existentes, as que permitem resolver o problema em questão. A geração do *designe* de teste consiste na definição do método usado para testar os modelos após a sua implementação. Após a construção dos modelos, é necessário avaliar o seu desempenho e a sua qualidade, o que

nem sempre é trivial. Por exemplo, uma das métricas utilizadas para fazer a avaliação de modelos de classificação, é o rácio de erros nas classificações. Neste caso, o que normalmente se faz, é definir um conjunto de dados que serão usados para a modelação e outro conjunto de dados para o teste, de forma a determinar a capacidade que um modelo tem de prever o passado, antes de o usar para prever o futuro. Assim sendo, é importante que se defina o método de teste a utilizar nos modelos, antes da sua implementação. De seguida é feito a criação dos modelos, através da execução das técnicas seleccionadas na ferramenta de *data mining* seleccionada.

Após a criação dos modelos é chegada a hora de os testar. O analista avalia os modelos, tendo em atenção o seu conhecimento em relação ao negócio, os objectivos definidos para o projecto, as condições de sucesso definidas na primeira fase e o *designe* de teste previamente definido. No entanto, esta tarefa deve ser feita com o auxílio de especialistas no negócio, de forma a ajudarem a interpretar os resultados obtidos. Aliás, é aconselhada a introdução destes especialistas na fase de criação dos modelos, de forma a puderem ser identificados eventuais problemas de dados que não sejam óbvios e que, de outra forma, passariam provavelmente despercebidos. Nesta fase também é feita uma classificação comparativa dos vários modelos gerados, usando-se normalmente várias instâncias de uma técnica ou de modelos usando técnicas diferentes. Este conjunto de modelos gerados é então classificado, sendo gerado um *ranking* tendo em conta os critérios de avaliação definidos.

A quinta fase é a **avaliação**, nesta fase é feita uma análise mais cuidada do modelo escolhido e implementado, sendo avaliado se ele cumpre todos os objectivos de negócio previamente definidos e se nenhum pormenor de negócio foi descorado. No fim desta fase, o analista deverá decidir como usará os resultados obtidos. A primeira tarefa desta fase é a avaliação dos resultados. É neste momento que os resultados são avaliados em relação aos objectivos de negócio e se verifica se existe alguma consideração em termos de negócio que torna o modelo não adequado. Esta tarefa não deve ser confundida com a fase de comparação e avaliação que é realizada na fase de modelação, pois nessa fase os modelos são avaliados e comparados entre si, tendo em conta métricas bem definidas, como a precisão dos modelos.

Uma opção a considerar, caso o orçamento e a limitação de tempo o permitam, é o teste do modelo em ambiente real. No final o analista deverá documentar os resultados atingidos em termos de negócio, bem como uma justificação sobre o cumprimento ou não dos objectivos definidos. A seguir a esta etapa é o momento de fazer uma reavaliação, na qual é estudado se algum pormenor técnico ou tarefa realizada durante o processo foi descorado. Depois da avaliação e da reavaliação do modelo, é chegada a hora de se definir as próximas tarefas. É nesta altura que

se decide se se deve terminar o projecto e avançar para o *deployment* da solução ou efectuar novas iterações no modelo.

A última fase é o ***deployment* ou instalação** da solução implementada. O modelo final implementado não representa o final do projecto, os resultados obtidos devem ser organizados e apresentados ao cliente, de forma a serem compreendidos e correctamente utilizados. Esta fase tanto pode ser um processo muito simples, como gerar um relatório para o cliente, como complexo, como implementar um processo de *data mining* capaz de ser repetido por toda a empresa. Muitas vezes é o cliente a realizar esta tarefa. No entanto o analista tem o dever de o instruir de quais tarefas a realizar, de modo a utilizar de forma correcta os modelos. A primeira tarefa a executar na fase de instalação, é a planificação da instalação. Nesta fase é planeado todo o processo de instalação. Depois disso, é feito um planeamento da monitorização e manutenção dos modelos criados. Esta fase é especialmente importante, caso os resultados sejam planeados para usar no dia-a-dia da empresa. Após estas duas tarefas de planeamento, é realizado o relatório final. Dependendo do que foi definido no plano de instalação, este relatório poderá apenas incluir um resumo do projecto e algumas notas que não tenham sido documentadas durante o processo, ou então um relatório final detalhado, com uma apresentação exaustiva dos resultados obtidos. A última tarefa consiste em fazer uma revisão do projecto, onde é avaliada os pontos positivos e negativos do projecto, bem como os pontos de melhoria a ter em atenção em projectos a realizar no futuro. Deverá ser realizado também um breve resumo sobre as capacidades adquiridas durante o projecto e apresentar algumas dicas a usar no futuro sobre as técnicas a usar ou sobre os erros a evitar [Shearer, 2000].

2.2 Definição do Âmbito

2.2.1 Dados Utilizados

Para o trabalho desenvolvido nesta dissertação, foram usados dados provenientes do sistema operacional de uma seguradora nacional de média dimensão. Os dados foram filtrados por clientes do ramo automóvel, ou seja, clientes possuidores de pelo menos um produto do ramo automóvel. A extracção da informação desta subpopulação de clientes da seguradora deu origem a um conjunto de dados referentes a cerca de 24.000 clientes, todos clientes possuidores de pelo menos um produto do ramo automóvel. Dos dados disponíveis, foram extraídos vários indicadores de diferentes tipos de informação: demográfica, socioeconómica, específica do negócio e relativa à

relação entre a empresa e o cliente. Para além destes indicadores, foram ainda criados alguns atributos, derivados dos já extraídos anteriormente. O tema dos dados utilizados é abordado com mais detalhe na secção em que é apresentado o caso de estudo.

2.2.2 Intervalo de Tempo em Análise

Relativamente ao intervalo de tempo dos dados extraídos, foi usada a informação dos clientes relativa aos últimos seis meses, uma vez que o objectivo era o de construir um modelo preditivo, capaz de a partir dos dados operacionais dos últimos seis meses, prever o abandono dos clientes no próximo mês. Esta análise será efectuada todos os meses, com o intuito de se identificar o volume de clientes desertores estimados para o próximo mês.

2.2.3 Definição de Cliente "Churner"

Para a classificação dos clientes como clientes *churners* ou *não-churners*, foi necessários definir de forma rígida o conceito de cliente *churner*. Assim sendo, um cliente do ramo automóvel é considerado um cliente *churner*, quando:

- Quando o cliente anula todos os contratos associados ao produto automóvel, sendo que existe indicação da data de anulação (deve-se ignorar os casos, em que o motivo da anulação é o falecimento do cliente, não considerando caso de *churn*).
- Quando o cliente deixa de cumprir com as suas obrigações contratuais, com as quais se comprometeu com a entidade fornecedora do serviço, relativamente às apólices de que é tomador, durante um período de, no mínimo, 6 meses. A partir deste prazo, caso a situação não seja regularizada, o cliente é considerado um cliente *churner.*

2.2.4 EDA – Análise Exploratória dos Dados

Na fase de análise dos dados foi realizada uma análise exploratória dos dados - EDA (*Exloratory Data Analysis*). A EDA é uma abordagem preliminar realizada antes do início da modelação de modelos preditivos. Este procedimento tem como objectivo analisar detalhadamente os dados disponíveis, de forma a inferir algumas hipóteses que sejam válidas para o problema em questão [Fernholz, 2000]. Nesta fase foi analisada a base de dados operacional com o objectivo de procurar alguma informação acerca dos clientes que pudesse ser considerada útil para o problema

em questão, para o correcto desempenho desta tarefa. Para a correcta execução desta tarefa consultaram-se alguns especialistas na área dos seguros, que tinham uma visão mais alargada sobre os problemas a tratar. Fruto desta parceria foi possível identificar os atributos e os indicadores que permitem identificar uma possível intenção de abandono pelo cliente. Alguns destes indicadores estão relacionados, por exemplo, com reclamações ou pedidos de informação de como cancelar o serviço por parte dos clientes.

2.2.5 Selecção de Variáveis, Limpeza, Transformação e Formatação dos Dados

Nesta fase do processo, de entre o conjunto de variáveis à disposição nos dados extraídos, foi feita uma selecção de apenas aquelas que se considerou possuírem algum valor preditivo relativamente ao possível abandono de um cliente. Foi efectuada, ainda, uma limpeza e transformação dos dados seleccionados, entre estas operações de limpeza e transformação, foram ainda criados alguns indicadores com base em outros já existentes, e foi realizado o tratamento dos *outliers* e de valores nulos, que prejudicavam a qualidade dos dados. Para além destas operações, foi ainda realizada a categorização de algumas varáveis contínuas, com o intuito de facilitar o posterior processo de modelação.

2.3 Avaliação dos Modelos

O processo de avaliação dos modelos implementados é uma das fases mais importantes do processo de mineração de dados [Souza, et. al., 2002]. Para o processo de avaliação e comparação da qualidade dos modelos criados foi necessário definir os métodos de avaliação a serem utilizados. Para fazer esta avaliação foram usadas como métricas de desempenho a taxa de previsões correctas (ou *hit-ratio*) e o *lift*.

O *hit-ratio* é definido como: $\frac{A}{A+B}$, onde A representa o número de casos de *churn* classificados como *churn,* e B o número de casos de *churn* classificados, erradamente, como *não-churn*. Esta métrica permite avaliar os modelos, dando ênfase à capacidade de previsão dos casos de *churn*. Ao invés de usar a precisão, que indica os casos classificados correctamente independente da sua classe, esta métrica permite portanto, determinar qual dos modelos se comportará melhor na previsão dos utilizadores *churners*.

O *lift* é uma das métricas mais importantes na avaliação de modelos preditivos. O *lift* de um modelo não é mais do que o rácio entre o *hit-ratio* para um determinando segmento da população e a resposta da variável dependente quando analisada toda a população. Para o cálculo do *lift* é normalmente efectuada a divisão da população analisada em dez grupos de igual dimensão, de forma ordenada pela sua probabilidade de ser um caso de *churn* (probabilidade estimado pelo modelo preditivo que se está a avaliar)*,* sendo que o primeiro grupo possui os elementos com maior probabilidade de *churn*. Assim sendo, tomemos como exemplo, um modelo de previsão de *churn*, com 1000 registos (clientes) em que 50 clientes são casos de *churn,* ou seja, existe um rácio de *churn* de 5%.

Grupo	#Registos	Casos detectados	Lift
1	100	16	3.2
2	100	12	2.4
3	100	8	1.6
4	100	5	1
5	100	3	0.6
6	100	2	0.4
7	100	1	0.2
8	100	1	0.2
9	100	1	0.2
10	100	1	0.2
Total	**1000**	**50**	

Tabela 2 – Lift - Exemplo

Na Tabela 2 é apresentada os valores de *lift* para cada um dos grupos da população apresentada anteriormente. Como se pode observar, o valor do *lift* é maior no grupo 1, e vai diminuindo progressivamente até ao grupo 4, que é quando atinge o valor 1. Ou seja, para este grupo, a percentagem de casos de *churn* detectados correctamente é igual ao rácio de casos de *churn* (5%). Esta análise é bastante útil para detectar que subgrupo da população analisada deve ser tomada em conta. Por exemplo, no caso da previsão de *churn* em clientes de seguradoras*.* Pretende-se que após a classificação dos clientes, esta informação seja usada para implementar medidas que permitam evitar o efectivo abandono. No entanto, pôr essas mesmas medidas em prática implica um custo proporcional ao número de clientes detectado. A partir desta análise é

possível, mediante os custos e o retorno esperado, determinar que grupos é que se devem escolher para aplicar as medidas, tomando como inicio o grupo 1, até ao grupo que se entender não justificar o custo.

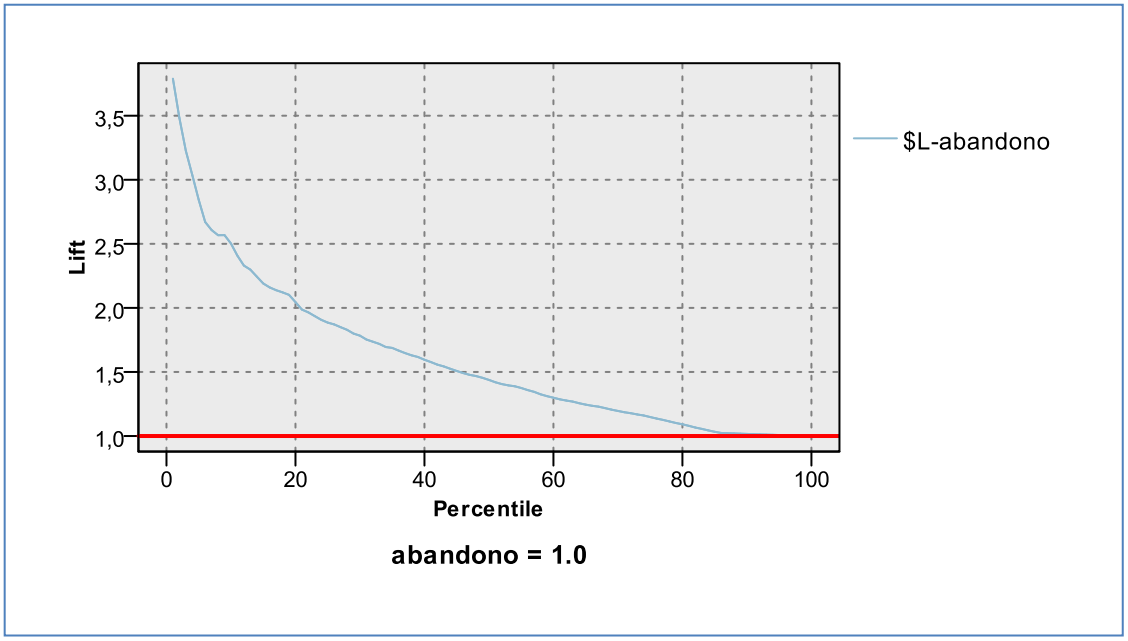

Figura 1 – Exemplo de Curva Lift

A informação apresentada na Tabela 2 representa uma possível visualização do valor do *lift*. No entanto, esta informação é normalmente visualizada em forma de curva – curva de *lift* –, permitindo ao utilizador analisar o valor do *lift* ao longo dos vários grupos criados (Na Figura 1, apresentamos um exemplo de uma curva *lift* resultante de um modelo preditivo de *churn* implementado no âmbito desta dissertação). Resta referir, que o *lift* não deve ser usado para determinar a utilidade de um modelo, pois os valores de *lift* podem variar muito de problema para problema, o uso desta métrica tem portanto especial utilidade na comparação de modelos com os mesmos objectivos [Coppock, 2002].

2.4 Ferramenta de Data Mining Utilizada

Como ferramenta de *data mining* para o desenvolvimento dos trabalhos desta dissertação foi escolhido o SPSS Modeler 14.1 [2]. O SPSS Modeler é um *software* de análise estatística e de *data*

mining desenvolvido pela SPSS Inc., actualmente uma companhia da IBM [1]. O SPSS Modeler, originalmente chamado de SPSS Clementine pela SPSS, teve o seu nome alterado após a aquisição da SPSS Inc. pela IBM, em 2009, o que originou um *rebrand* a deste produto.

O SPSS Modeler é uma ferramenta que permite, respeitando a metodologia CRISP-DM (secção 2.1), realizar todo o processo de implementação de um modelo preditivo, desde o carregamento e tratamento dos dados, até à geração dos resultados finais. O SPSS Modeler apresenta-se assim como uma plataforma analítica completa. Constituindo uma solução de *data mining* integrada e escalável, suportando um leque alargado de técnicas, algoritmos e ferramentas estatísticas, permitindo desenvolver soluções preditivas completas de forma simples, rápida e eficaz. Por estas razões, foi em 2010 considerado pelo *Rexer's Annual Data Miner Survey,* uma das ferramentas de DM com maior índice de satisfação entre os seus utilizadores [Rexer, et. al., 2010].

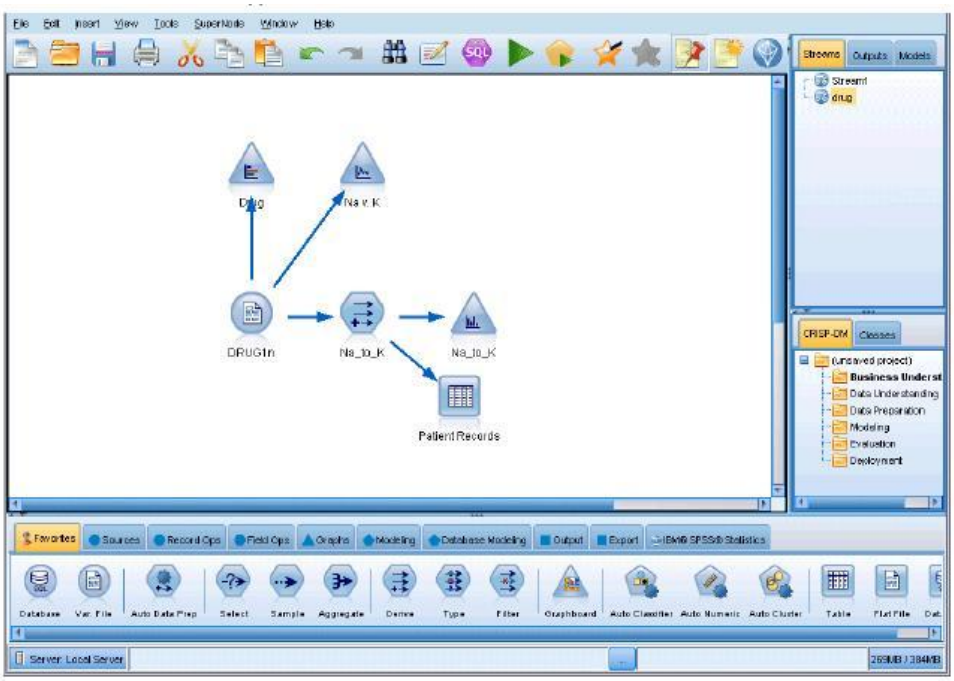

Figura 2 – Exemplo Projecto SPSS Modeler

O SPSS Modeler apresenta-se ao utilizador com uma interface simples e intuitiva. O utilizador é capaz de criar projectos de *data mining* em forma de um fluxo constituído por nodos de processamento. Estes nodos, os elementos básicos de um projecto, são responsáveis por efectuar as mais variadas tarefas, sejam elas a aplicação de algum tipo de tratamento de dados, criação de

um modelo preditivo, ou apresentação de resultados. Nesta ferramenta é oferecido ao utilizador um leque alargado de operadores capazes de realizar inúmeros tipos de tarefas, desde o simples acesso aos dados, o seu tratamento e manipulação, utilização de algoritmos de DM ou estatísticos, até à geração de resultados num determinado formato. O utilizador pode fazer a modelação do processo de DM, de forma simples e iterativa, ao adicionar nodos ao processo, com o objectivo de tratar e manipular o fluxo de dados em função do objectivo requerido. Na Figura 2 é apresentado um exemplo de um projecto de demonstração do SPSS Modeler, em que é visível um pequeno projecto no qual é feito o acesso a uma fonte de dados, o tratamento dos dados acedidos e a geração de vários *outputs* em forma de gráficos e tabelas [SPSS Inc. 2010b] [SPSS Inc. 2010c].

Capítulo 3

3 Mineração de Dados para Churn

3.1 Mineração de Dados

O *Data Mining*, tal como apresentado anteriormente (secção 1.2), é nos dias de hoje, uma ferramenta imprescindível no apoio à tomada de decisões de negócio. Quando se aborda o tema da detecção do abandono de clientes (*churn*), independentemente do contexto de negócio em que se insere, é fácil compreender a enorme vantagem que a utilização destas técnicas oferecem para realizar esta mesma detecção. Com o uso de técnicas de DM, é possível, tomando como fonte os dados operacionais, desenvolver modelos preditivos de *churn,* capazes de detectar padrões de comportamento, que aliados a determinadas características do cliente, poderão ser um indício de um futuro abandono do serviço por parte do cliente.

Esta necessidade de ser capaz de prever o abandono de clientes é facilmente estendida ao contexto das seguradoras. Num modelo de negócios onde a competição é cada vez maior, e onde a captação de novos clientes implica investimentos relativamente elevados, a manutenção e fidelização dos clientes actuais torna-se fulcral para uma boa performance financeira da empresa.

O tema da detecção de *churn* usando técnicas de DM é um tema que tem vindo a ser estudado desde o ano 2000, aquando a introdução do termo *churn*, para definir o abandono de clientes na área das telecomunicações [Berson, et. al., 2000]. Desde então, têm sido desenvolvidos inúmeros trabalhos na área da detecção de *churn* com técnicas de *data mining,* alguns exemplos

disso é o trabalho desenvolvido por Hung na detecção de *churn* em companhias de telecomunicações [Hung, Yen, & Wang, 2006], o trabalho de Archaux na detecção de *churn* nos clientes de serviços móveis pré-pagos [Archaux, Martin & Khenchaf, 2004], o desenvolvido por Qian onde é abordado o problema do *chun* no contexto da criação de perfis de utilizadores (*profiling*), para a aplicação de boas práticas de CRM [Quian, Jiang & Tsui, 2006], ou ainda o de Morik e Hanna na previsão de *churn* entre os clientes de uma seguradora da área da saúde [Morik, Kopcke, 2004].

3.2 Abordagens Algorítmicas

Como sabemos, existe um grande leque de técnicas de *Data Mining à disposição*. Apesar da grande diversidade de técnicas de DM, estas podem ser agrupadas em apenas quatro grupos, ou classes de algoritmos, que são: Classificação, Regressão, Associação e Segmentação (*Clustering*). Cada uma das classes de algoritmos tem as suas vantagens e desvantagens, tem os seus requisitos, e áreas de aplicabilidade, por vezes não coincidente. Por esta razão, no desenvolvimento desta dissertação de mestrado foram seleccionados vários algoritmos para a implementação de modelos de *churn,* no entanto no conjunto de algoritmos seleccionados não se encontram representados as 4 classes anteriormente enunciados, isto deve-se ao facto de nem todas as classes de algoritmos serem aproprias ao problema de detecção de *churn.* As técnicas usadas serão apresentadas em detalhe na secção 3.3 juntamente com a justificação para a sua selecção. Apesar de no grupo de algoritmos seleccionados não estarem representados todas as classes, as 4 classes são de seguida apresentadas de forma breve.

3.2.1 Algoritmos de Classificação

Os algoritmos de classificação permitem resolver o problema de classificar novas observações mediante uma subpopulação. Isto é, são usados quando é necessário identificar a que subpopulações pertencem determinadas novas observações desconhecidas, tendo como *input*, um conjunto de dados (*dataset*) de treino com as populações de registos previamente rotulados em subpopulações. Na base da classificação, está um conjunto de observações – o conjunto de dados de treino ou *dataset* de treino – em que a sua população está classificada em subpopulações, os métodos de classificação são capazes a partir deste *dataset* de treino, de criar um modelo de

previsão capaz de determinar a que subpopulações pertencem as novas observações. Os algoritmos de classificação contrastam com os algoritmos de *clustering (*ver secção *3.2.4)*, pois ao contrário destes, são métodos supervisionados de aprendizagem, pois é o utilizador que classifica os casos de estudo e os fornece ao algoritmo como *input* para serem processados. Existem vários tipos de algoritmos de classificação, sendo os mais conhecidos e maioritariamente usados, as árvores de decisão, as RN (Redes Neuronais) e os SVM (*Support Vector Machine*).

Existem vários trabalhos publicados sobre algoritmos de classificação para a detecção de *churn.* Alias, no decorrer da realização desta dissertação e aquando o levantamento de bibliografia, foi constatado que os algoritmos mais utilizados para a detecção de *churn*, são exactamente os algoritmos de classificação, com as árvores de classificação em primeiro no que toca a vezes ao número de artigos em que é utilizada, e com as RN e os SVM a ganhar terreno nos últimos anos, por serem técnicas mais sofisticadas e ainda em desenvolvimento. A título de exemplo de alguns destes trabalhos, podem se indicar trabalhos como o de Morik e Kopcke que estuda a viabilidade da detecção de *churn* em companhias de seguros usando árvores de decisão [Morik, Kopcke, 2004], ou ainda trabalhos como o de Au, Chan e Yao em 2003 em que fazem uma comparação critica entre dois modelos preditivos de *churn,* um usando árvores de decisão, e o outro usando RN [Au, Chan & Yao, 2003], ou ainda o trabalho sobre *churn* nas companhias de telecomunicações de Hung [Hung, Yen & Wang, 2006] que apresenta também uma comparação critica entre as árvores de decisão e as RN. Para além destes trabalhos que exploram algoritmos de árvores de decisão e de RN, existem vários trabalhos sobre a detecção de *churn* usando SVM. Exemplo destes trabalhos é por exemplo o artigo sobre a detecção de *churn* em clientes de telecomunicações móveis pré-pagas [Archaux, Martin & Khenchaf, 2004], ou ainda os trabalhos desenvolvidos por Wei-yun [Wei-yun, Zheng, Yu, Bing, Xiu, 2007] e por Xia e Jin [Xia & Jin, 2008] que estudam a viabilidade do uso de SVM. Em relação às Redes Neuronais, também é elevada a existência de estudos publicados sobre a sua aplicabilidade na detecção de *churn,* alguns desses estudos são por exemplo o trabalho de Pendharkar, sobre a detecção de *churn* em redes *wireless* moveis usando RN [Pendharkar, 2009] ou o trabalho de Song sobre um algoritmo hibrido de RN aplicada à detecção de *churn* em comunicações móveis [Song, et. al., 2006].

Neste trabalho foram estudados 3 dos algoritmos referidos anteriormente – árvores de decisão, RN e SVM – estes algoritmos foram escolhidos devido à sua relevância no uso para a previsão de *churn.* Estes algoritmos serão apresentados em mais pormenor posteriormente na secção 3.3.

3.2.2 Algoritmos de Regressão

As técnicas de regressão têm como objectivo definir uma função capaz de prever o resultado de uma variável, com o menor erro possível. Estas técnicas permitem modelar e analisar a relação existente entre uma variável dependente e outra variável, ou grupo de variáveis, vulgarmente designadas por variáveis independentes. Ou seja, é possível a partir de técnicas de regressão analisar o comportamento que uma dada variável dependente toma, tendo em atenção os valores das variáveis independentes. O resultado de um algoritmo de regressão, deverá ser uma função, denominada de função de regressão, que modela os valores possíveis da variável dependente em relação as variáveis independentes. As análises de regressão são vulgarmente usadas para a previsão e *forecasting* [Agresti, 2007].

Basicamente existem dois tipos básicos de regressões, regressão linear e regressão não linear. Nas regressões lineares é estimada a previsão do valor da variável dependente tendo em atenção os valores dados das variáveis independentes. O nome "linear" deve-se ao facto, de se partir do pressuposto que a função de resposta (a função de regressão) é uma função linear das variáveis independentes.

a) $\mathrm{Y_i} = \beta_0 + \beta_1 \mathrm{X_i} + \varepsilon_\mathrm{i},\ \mathrm{i} = 1, .., \mathrm{n}.$

b) $Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i,\ i = 1, .., n.$

Em a) é exemplificada uma equação de uma regressão linear simples com uma variável independente $\mathrm{X_i}$ e dois parâmetros β_0 e β_1. Em b) é apresentada uma equação linear múltipla, onde existem várias variáveis independentes. A variável ε_i representa todos os factores residuais mais os possíveis erros de medição. Já uma regressão não linear representa os casos em que os dados são modelados por uma função que combina os parâmetros do modelo de forma não linear, neste caso os dados necessitam de ser ajustados por sucessivos métodos de aproximação [Weisberg, 2005].

Apesar de ser menos utilizado que as técnicas de classificação, é possível encontrar vários trabalhos em que se estuda a aplicabilidade das técnicas de regressão ao problema da previsão de *churn.* Alguns desses trabalhos são, a título de exemplo, o trabalho de Mozer, e também o trabalho de Neslin, onde são comparados alguns métodos, entre eles o uso de regressão logística para a detecção de *churn* [Neslin, 2006] [Mozer, et. al., 2000].

Neste trabalho foi estudada a regressão logística binomial para a implementação do modelo preditivo de *churn*, este algoritmo será apresentado mais pormenorizadamente posteriormente na secção 3.3.2.

3.2.3 Algoritmos de Associação

As técnicas de associação são um conjunto de técnicas de DM que permitem descobrir relações entre variáveis. Existem vários algoritmos de associação, estes algoritmos têm em comum o seu *output*, constituído por um conjunto de regras, chamadas de regras de associação. Este conjunto permite expressar relações entre as diferentes variáveis, classificando ainda essas mesmas relações mediante algumas métricas de interesse que permitem classificar a qualidade da regra [Agrawal, et. al., 1993]. Agrawal introduziu em 1993 o conceito de regras de associação com o intuito de descobrir padrões de comportamento nas transacções de um supermercado. As regras de associação permitem exprimir relações entre itens de um conjunto de dados da seguinte forma: $\{cebolas, batatas\} \rightarrow \{\text{hambúrgeres}\}$. A regra apresentada anteriormente indica que um cliente que compre cebolas e batatas, tem propensão a comprar também hambúrguer. No entanto, esta regra tem pouco, ou nenhum valor, se não vier acompanhada de métricas de qualidade. Métricas tais como o suporte, a confiança e o *lift* da regra são o que classificam e quantificam a qualidade e utilidade da regra gerada. Por exemplo, se a regra anterior tiver um suporte de 90% e uma confiança de 70%, isto quereria dizer que 90% da totalidade dos clientes compraram cebolas, batatas e hambúrgueres, e que 70% dos clientes que compraram cebolas e batatas compraram também hambúrgueres.

O suporte de uma regra indica a proporção de transacções que contêm os itens da regra. Já a confiança de uma regra indica a percentagem de vezes que a regra está correcta quando o precedente da regra se verifica, a expressão da confiança de uma regra é definido como:

$conf(X \Rightarrow Y) = \sup(X \cup Y) / \sup(X)$;

É portanto clara a importância de saber filtrar as regras tendo em atenção aos valores destas duas métricas. Para além destas duas métricas, é possível classificar as regras mediante o seu *lift,* o *lift* é definida por:

$Lift\ (X \Rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(Y) \times \sup(X)}$;

O *lift* definida permite classificar as regras mediante o suporte em relação ao suporte esperado caso as varáveis fossem independentes. Ou seja, o *lift* não é mais do que o rácio entre a confiança real e a confiança esperada.

Os algoritmos de associação são um tipo de técnicas de DM que não se adequam ao problema de detecção de *churn,* pois, como já foi mencionado, o objectivo dos algoritmos de

associação é descobrir relações entre um grande número de variáveis, e não prever o valor de uma determinada variável. Por esta razão o uso deste tipo de técnicas para a previsão de *churn* é quase nula. No entanto, é possível utilizar este tipo de técnicas, não como método de modelação final do modelo preditivo de *churn*, mas sim como auxiliar ou, como é exemplo do trabalho de Chiang, onde é usado o algoritmo de associação para desenvolver um novo algoritmo hibrido que tem como base o princípio das regras de associação [Chiang, et. al., 2003]. Ou ainda o trabalho de Tsai e Chen, que usa regras de associação no processo de preparação de dados, de forma a optimizar o treino do modelo [Tsai, Chen, 2009]. Por este tipo de técnica não se adequar ao problema abordado no trabalho desenvolvido nesta dissertação a mesma não foi testada.

3.2.4 Algoritmos de *Clustering*

As técnicas de *clustering* (ou de segmentação) são usadas quando se pretende dividir um grupo observações em pequenos subgrupos, denominados por *clusters*. Esta divisão deverá ser feita de modo a que as observações de cada *cluster* possuam características similares [Klosgen & Zytkow, 1996]. As técnicas de *clustering* são largamente utilizadas em várias áreas para além do Data Mining, a sua utilização estende-se a áreas como: análise de imagem, reconhecimento de padrões, bio-informática, inteligência artificial, entre outras. O *Clustering* é um método não supervisionado (*unsupervised learning*), ou seja, ao contrário dos métodos supervisionados (*supervised learning*) nos quais é fornecido ao algoritmo um conjunto de dados, conhecidos como dados de treino, a partir do qual o algoritmo infere as regras necessárias para chegar ao *output* desejado, no *clustering* o algoritmo parte de dados não catalogados, ou seja, não usa dados de treino previamente classificados [Xu & Wunsch, 2005].

Existem vários tipos de algoritmos de *clustering*, *clustering* hierárquico, *clustering* de particionamento, *clustering density-based* e *sub-space clustering*. Dos tipos de algoritmos de *clustering* mencionados anteriormente, são mais comuns e utilizados os hierárquicos e os de particionamento [Xu & Wunsch, 2005]. Por esta razão, de seguida, será feita uma breve e superficial apresentação dos mesmos sendo que os restantes não serão abordados. Os algoritmos de *clustering* hierárquico realizam o processamento dos clusters partindo sempre dos clusters previamente formados. Estes algoritmos hierárquicos podem ser de dois tipos, aglomeradores, se usarem uma filosofia *bottom-up* para o particionamento, ou podem ser divisivos, se usarem uma abordagem *top-down*. Os algoritmos hierárquicos aglomeradores iniciam o processamento tomando cada observação como sendo um *cluster*, e a partir daí vai os agrupando mediante os

seus graus de semelhança. Já os algoritmos hierárquicos divisivos, assumem inicialmente todo o *dataset* como sendo o *cluster* inicial e vai aplicando o particionamento repetidamente até não ser possível dividi-lo mais. Os algoritmos de particionamento, ao contrário dos hierárquicos, que operam recursivamente sobre os *clusters* que vão estabelecendo, são capazes de determinar todos os clusters de uma só vez. Um dos algoritmos de *clustering* de particionamento mais utilizado é o *k-means* [MacQueen, 1967] [Anderberg, 1973].

Relativamente à eficiência e à qualidade de cada um dos tipos de *clustering* abordados anteriormente, o *clustering* hierárquico é muitas vezes apontado como um algoritmo produtor de melhores resultados. No entanto é limitado pela sua complexidade quadrática que implica requisitos de processamento elevados. Por outro lado, o *k-means* e as suas variantes têm uma complexidade linear, sendo pouco dispendiosos ao nível de tempo do processamento, e produzem usualmente *clusters* de inferior qualidade [Steinbach, et. al., 2000]. Observa-se que nos últimos anos, tem sido estudado em vários trabalhos, a combinação dos princípios do *k-means* e dos algoritmos de *clustering* hierárquico aglomerador, de maneira a obter um algoritmo capaz de tirar partido do melhor dos "dois mundos", exemplo disso é o trabalho de D. Cutting [Cutting, et. al., 1992], no qual se utiliza o *k-means*, devido à sua eficiência no que diz respeito ao tempo de processamento, e o *clustering* hierárquico aglomerador devido à sua capacidade de produzir resultados com elevada qualidade.

Tal como os algoritmos de associação, os algoritmos de *clustering,* não são adequados para a previsão de *churn*, pois o seu objectivo não é prever o valor de uma variável, mas sim, segmentar registos, em grupos idênticos. Por esta razão esta técnica é pouco utilizada na detecção de *churn*. No entanto, apesar de pouco utilizada, não é de todo impossível o seu uso para este fim, tal como é prova o trabalho de Popović, onde é utilizado o algoritmo de *clustering fuzzy c-means* para a implementação de um modelo preditivo de *churn* na área da banca de retalho [Popović, Bašić, 2009], ou ainda o trabalho de Bose, em que o autor implementa e testa vários algoritmos híbridos usando *clustering* e árvores de decisão como base. É importante realçar também, que apesar de os algoritmos de *clustering* não serem vulgarmente utilizados na previsão de *churn*, o seu uso é extremamente valorizado na implementação de boas técnicas de CRM, nomeadamente na gestão dos clientes identificados como *churners.* Exemplo disto é o trabalho de Karahoca, no qual é estudado a utilização de técnicas de *clustering*, para fazer a correcta gestão de *churn* nas telecomunicações, a ideia núcleo neste trabalho, é que ao aplicar *clustering*, de maneira a identificar os clientes mais valiosos para a companhia, é possível aplicar medidas anti-*churn* diferenciadamente, de acordo com o valor do cliente [Karahoca, Kara, 2006].

3.3 Técnicas para Mineração de Dados

Observou-se nos últimos anos um grande avanço na investigação da temática da mineração de dados, nomeadamente no que diz respeito às técnicas de modelação de modelos preditivos. Foram publicados até à data, inúmeros trabalhos sobre as diversas técnicas de *Data Mining* que se podem utilizar para a previsão de resultados. No entanto, apesar do vasto leque de algoritmos disponíveis para a implementação de modelos preditivos, chegada a hora da escolha da técnica a usar na implementação, essa escolha costuma recair num pequeno conjunto de técnicas, conjunto de técnicas estas, que são reconhecidas como sendo mais eficazes e apresentam a melhor relação entre simplicidade de implementação e eficácia, e como tal são as mais usados [Au, et. al., 2003]. Estas técnicas são respectivamente as **redes neuronais** [Datta, et. al., 2001] [Boone & Roehm, 2002] [Vellido, et. al., 1999] [Kavzoglu & Mather, 2001] [Meyer-Base & Watzel, 1998], as **árvores de decisão** [Ho Ha, et. al., 2002] e os algoritmos de **regressão logísticas** [Bloemer, et. al., 2002] [Chen & Dey, 2003].

Para a realização do trabalho apresentado nesta dissertação, foram escolhidos 4 algoritmos: 3 pertencentes à classe dos algoritmos de classificação (secção 3.2.1), e um algoritmo de regressão (secção 3.2.2). Os algoritmos escolhidos foram, nomeadamente:

- Árvores de decisão.
- Regressão Logística Binomial.
- Redes Neuronais.
- *SVM (Support Vector Machine)*.

A escolha dos três primeiros algoritmos (árvores de decisão, regressão logística e redes neuronais) deveu-se ao facto de serem os três algoritmos que reúnem consenso geral entre a comunidade científica e entre os analistas de *data mining*, como sendo os mais viáveis para a modelação de modelos preditivos, prova disso é o grande número de publicações existentes sobre modelos preditivos usando as técnicas referidas. Já o SVM foi seleccionado, pois é um método eficiente e adequado para dados com ruído. Além disso também já foram publicados diversos trabalhos, em que se provou que o SVM é por vezes o método mais eficaz na detecção de *churn* [Archaux, et. al., 2004].

Partindo das 4 técnicas seleccionadas anteriormente, foram implementados 11 modelos preditivos de *churn*. Os modelos seleccionados são de seguida apresentados e classificados segundo as seguintes características:

1. **ID do Modelo** – representará o identificador do modelo a usar neste documento.
2. **Tipo de algoritmo** – indica a classe de algoritmos (secção 3.2) a que pertence (classificação, regressão, associação ou *clustering)*.
3. **Algoritmo Usado** – o algoritmo usado para a implementação do modelo.
4. **Algoritmos/Características Específicas** – Indica os algoritmos/técnicas/especificações do Algoritmo usado, que distingue o modelo em questão dos restantes modelos que usam o mesmo Algoritmo.

Nº	ID Modelo	Tipo de Algoritmo	Algoritmo Usado	Algoritmos Específicos
1	A1	Classificação	Árvores de Decisão	Exhaustive CHAID
2	A1.1	Classificação	Árvores de Decisão	Exhaustive CHAID c\ Bagging
3	A1.2	Classificação	Árvores de Decisão	Exhaustive CHAID c\ Boosting
4	A2	Classificação	Árvores de Decisão	C5.0
5	A2.1	Classificação	Árvores de Decisão	C5.0 c\ Boosting
6	RL1	Regressão	Regressão Logística Binomial	Forwards
7	RL2	Regressão	Regressão Logística Binomial	Backwards
8	RN1	Classificação	Rede Neuronal	Normal
9	RN2	Classificação	Rede Neuronal	c\ Bagging
10	RN3	Classificação	Rede Neuronal	c\ Boosting
11	SVM	Classificação	SVM	Normal

Tabela 3 – Os Modelos Implementados

Na secção seguinte serão apresentados os algoritmos usados no desenvolvimento dos trabalhos desta dissertação.

3.3.1 Árvores de Decisão

Apresentação Geral

Umas das técnicas analisadas neste trabalho para a implementação do modelo de previsão de *churn* foram as árvores de decisão. As árvores de decisão são das técnicas de DM mais utilizadas, para a sua enorme popularidade contribui a sua simplicidade de implementação, e principalmente a facilidade de compreensão do resultado final [Quinlan, 1986]. As árvores de decisão apresentam os seus resultados numa estrutura recursiva onde é expressa o processo de classificação sequencial de um item/observação que é caracterizado por um grupo de atributos e é classificado como pertence a uma classe de um grupo de classes disjuntas. Numa árvore de decisão, cada folha (nodo terminal) representa uma classe e cada nodo intermédio representa um teste, envolvendo uma ou mais variáveis/atributos e cada possível resultado do teste origina uma nova subárvore [Quinlan, 1987]. Após a implementação de uma árvore de decisão, a classificação de um novo item é feito navegando pela árvore, desde a raiz *(root*) da árvore, até uma folha, sendo então o novo item classificado segundo a classe representada por essa folha.

Na Figura 3 é apresentado um exemplo de uma árvore de decisão. Neste exemplo é modelada a decisão de "*se num determinado dia, se deve jogar um jogo de futebol ou não*", mediante 2 variáveis/atributos desse dia em questão, as classes possíveis são constituídas portanto pelo conjunto {Jogar, Não Jogar}, e as variáveis independentes, ou atributos, são compostas pelo conjunto {Temperatura, Humidade}. O processo de classificação, ou neste caso, de decisão, iria começar pela raiz da árvore, ou seja, pelo primeiro nodo da árvore. Neste nodo é analisada o valor da temperatura, sendo descriminados dois possíveis *outputs,* temperatura <35ºC ou Temperatura ≥ 35ºC. Sendo que no caso de a temperatura ser <35ºC o caso em análise é imediatamente classificado como "Jogar", caso seja ≥35ºC o processo de classificação avança para a subárvore seguinte, onde é analisada o valor da variável humidade, neste caso é feito uma diferenciação entre 35% ≤ humidade <35%, onde é feita a classificação como "Jogar" caso seja ≥355 e como "Não Jogar" caso contrario.

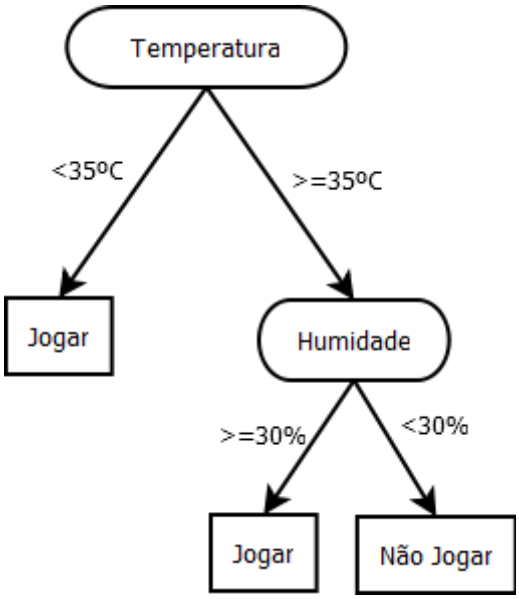

Figura 3 – Exemplo de uma Árvore de Decisão.

De realçar, que o exemplo apresentado representa uma árvore muito simplista e não modela um problema real. As árvores podem atingir um elevado nível de complexidade quando modelam problemas com um elevado número de variáveis e várias classes. De realçar que cada nodo pode dar origem a mais do que dois *outputs,* apesar de tal não estar expresso no exemplo. As árvores de decisão podem ser apresentadas com *layouts* díspares do apresentado neste exemplo, por exemplo, os nodos são frequentemente apresentadas com o valor relativo de cada classe, indicando a probabilidade que uma nova observação que satisfaz o nodo em que se encontra tem de pertencer a cada uma das classes.

Um modelo de árvores de decisão é construído, a partir de um *data-set* de treino, ou seja, é um método de aprendizagem supervisionada (*supervised learning*). O processo ocorre de forma iterativa ao dividir o *data-set* baseado no resultado de um determinado teste a uma das suas variáveis. O processo de divisão do *data-set* ocorre de forma recursiva em cada sub-partição (subárvore) formada na divisão anterior e termina quando todas as observações de um nodo pertencem á mesma classe (tem o mesmo valor para a variável alvo), ou quando o processo de divisão já não origina um aumento de valor nas previsões.

Tipos de Árvores de Decisão

As árvores de decisão dividem-se em dois grandes tipos, dependendo do tipo de output esperado:

- Árvores de Classificação.
- Árvores de Regressão.

São chamadas de árvores de classificação quando o objectivo da previsão é determinar a classe a que a observação pertence. Por outro lado, quando o *output* é um número real, é usada a nomenclatura de árvore de regressão.

O termo CART (*Classification and Regression Tree*) foi introduzido pela primeira em 1984 por Breiman, no seu trabalho sobre árvores de decisão [Breiman, 1984]. Estes dois tipos de árvores possuem várias semelhanças, mas também diferem em pormenores fulcrais, nomeadamente no modo com que calculam os "valores de corte", ou seja, no modo que definem os valores da variável, onde deve ser realizado o corte, originando duas novas subárvores.

O processo de implementação de uma CART pode ser resumido em três fases:

1. **Construção e crescimento da árvore**: a construção é feita de forma recursiva, neste processo os nodos são repetidamente divididos, até se chegar às folhas (nodos terminais da árvore), em que é atribuído uma classe à folha em questão. A divisão recursiva inicia-se na raiz da árvore, e começa pela selecção da melhor variável para usar na divisão, para de seguida ser determinado os pontos de divisão ideais da variável anteriormente definida, a decisão da variável e dos pontos de corte ideais é dependente do algoritmo usado (o tema do algoritmo usado será abordado posteriormente neste capítulo). A atribuição das classes nos nodos terminais é baseada na distribuição das classes nos dados de treino, na matriz de decisão, e ocorre apenas caso o nodo não dê origem a uma nova subárvore. A cada nodo terminal, correspondente apenas uma classe, e cada nodo terminal é definido por um conjunto de regras único [Lewis, 2000].
2. **Paragem do processo de crescimento da árvore**: o processo de crescimento da árvore ocorre até não ser possível criar mais subárvores, ou, alternativamente, até atingir uma condição de paragem previamente definida. Normalmente são definidas condições de paragem, estas condições podem ser a profundidade da árvore ou a significância mínima que um novo corte – que cria uma nova subárvore — implicará no modelo preditivo. A definição de condições de paragem do processo de crescimento de uma árvore é um aspecto crucial para o bom desempenho do processo, pois caso não seja definido

nenhuma condição, a árvore iria crescer até ao máximo possível, originando uma "árvore maximal" que contemplaria todas as observações do data-set de treino, ou seja, para além de ser seria demasiado moldada para os dados de treino, originando assim um problema grave de *overfitting.* As regras de paragem usadas nos modelos implementados são apresentadas nas secções de apresentação das configurações dos modelos das árvores (secções 4.2.1 e 4.2.2).

3. **Pruning –** o processo de *pruning*, consiste na simplificação da árvore de decisão gerada previamente, esta simplificação é atingida pelo corte sucessivo dos nodos que representam uma relevância preditiva pouco acentuada [Helmbold & Schapire, 1995]. O *pruning* permite ao mesmo tempo diminuir o tamanho e a complexidade das árvores de decisão, diminuindo o risco de *overfitting*. Como mencionado anteriormente, a decisão de paragem do crescimento de uma árvore é um assunto a que se deve dar elevada importância, no entanto é impossível determinar quando se deve parar o processo de crescimento pois é impossível calcular se a adição de um nodo extra irá melhorar drasticamente a capacidade preditiva da árvore. Por estas razões, é frequentemente realizado o crescimento da árvore até um determinado tamanho que implique a inclusão de um pequeno número de casos em cada nodo terminal, e posteriormente é realizado o *pruning,* de forma a remover os nodos que não representam uma mais-valia para o modelo [Hastie, Tibshirani, 2001]. O *pruning* pode ser feito mediante diferentes algoritmos, sendo o objectivo, atingir a melhor relação possível entre a simplicidade e o poder de previsão da árvore, o que não é trivial. Existem vários estudos efectuados até ao momento sobre técnicas de *pruning*, sendo que o objectivo é comum a todos: desenvolver um algoritmo capaz de obter a melhor árvore possível após o *pruning*, ou seja, a árvore ideal [Mansour, 1997], [Kearns & Mansour, 1998].

Vantagens e Desvantagens das Árvores de Decisão

De uma forma geral, as árvores de decisão apresentam as seguintes vantagens em relação a outros métodos de classificação [Lewis, 2000]:

- Apresentam uma elevada simplicidade na sua compreensão e na análise e interpretação dos seus resultados.
- Seguem uma filosofia de modelo "*white box*", ou seja, um resultado observado num modelo, é facilmente explicada e comprovada por operações lógicas, em contraste com

outras técnicas, tal como as redes neuronais, nas quais é difícil compreender os resultados obtidos.

- Não requer um intensivo tratamento prévio para assegurar a qualidade dos dados (normalização dos dados, criação de variáveis *dummy*, tratamento de *outliers*, tratamento de nulos, etc.) ao contrário de outras técnicas de DM.
- Opera tanto sobre dados numéricos como sobre dados categóricos, ao contrário de algumas técnicas que apenas consegue lidar com alguns tipos de dados (por exemplo, as redes neuronais apenas podem usar variáveis numéricas).
- Bom desempenho a analisar largos volumes de dados.

No entanto, como qualquer algoritmo, os algoritmos das árvores de decisão apresentam também algumas desvantagens em relação a outros métodos de DM. As desvantagens mais notórias são:

- A construção de uma árvore de decisão óptima é considerada um problema NP completo. Os algoritmos utilizados nas árvores de decisão, operam ao nível do nodo em processamento, ou seja, tomam a decisão óptima para o nodo em questão, não havendo garantia que a árvore resultante no final seja óptima [Hyafil, et. al., 1976] [Murthy, 1998].
- Dados de treino com fraca qualidade tendem a originar árvores demasiado adaptadas aos dados em questão (*overfitting*) [Papagelis & Kalles, 2001].
- Nos modelos com dados que incluam variáveis categóricas com diferentes níveis de detalhe, existe uma tendência para dar mais ênfase ao ganho das variáveis com mais nível de detalhe [Deng, 2011].

Construção da Árvore de Decisão

Como já foi referido, a construção, ou crescimento da árvore de decisão, é feita de forma recursiva, tomando como inicio a raiz da árvore, até chegar as folhas da árvore. Este processo de crescimento da árvore depende do algoritmo escolhido para a sua implementação, que é o responsável por determinar o comportamento a adaptar durante o processo, nomeadamente a escolha da variável que deverá ser usada no processo de corte, e também a selecção do valor de corte a usar na variável. Apesar dos algoritmos usados nas árvores de decisão terem o mesmo objectivo: maximizar a exactidão dos seus resultados. Eles diferem na maneira como fazem o crescimento da árvore, ou seja, na métrica e na técnica usada para decidir qual a variável a usar, e qual o ponto de divisão a escolher. Aqui serão apresentados os dois algoritmos de árvores usados

para a implementação de dois dos modelos preditivos implementados e testados no âmbito do trabalho desenvolvido nesta dissertação (secção 3.3). Os algoritmos em questão são o algoritmo *Exhaustive* CHAID, e o algoritmo C5.0.

Algoritmo Exhaustive CHAID

O algoritmo *Exhaustive* CHAID é uma evolução do seu predecessor, o CHAID (*Chi-Squared Automatic Interaction Detection)*. Este algoritmo deve o seu nome ao teste estatístico que usa para calcular os pontos de corte ideais (*splits*) durante o seu processo de divisão. O algoritmo CHAID é um algoritmo de classificação para implementação de árvores de decisão, este algoritmo foi desenvolvido por Kass em 1980 [Kass, 1980], e em 1991 seria apresentado o *Exhaustive* CHAID, que representa uma evolução em relação ao CHAID, nomeadamente no que toca a qualidade da análise do processo de corte [Biggs, 1991]. O algoritmo CHAID começa o seu processamento por criar e analisar a tabela de contingência entre cada uma das variáveis de *input* e a variável dependente, de forma a analisar a relação entre as diferentes variáveis independentes e a variável dependente, de seguida é realizado um teste estatístico – teste *chi-square* – de forma a determinar a independência entre as variáveis. O algoritmo seleccione de seguida a variável de entrada mais significativa (com o menor *p-value* calculado pelo teste *chi-square*). Relativamente à junção das categorias das variáveis de entrada, o CHAID actua de forma distinta, dependendo do tipo de variável. Caso seja uma variável nominal, o algoritmo processa da seguinte maneira: se a variável tiver mais do que duas categorias, as categorias são comparadas e as que não apresentarem diferenças em relação à variável dependente são fundidas na mesma categoria. Este processo é realizado de forma iterativa, agregando todos os pares de categorias que apresentam um valor de significância inferior ao valor mínimo previamente definido pelo utilizador. Caso esteja perante uma variável ordinal, apenas as categorias contínuas podem ser agregadas. Muitas vezes o CHAID não é capaz de determinar o ponto de corte ideal de uma variável, pois o processo de junção das categorias é interrompido logo que o algoritmo identifique que as restantes categorias são estatisticamente diferentes. É nesta fase de análise de agregação das categorias, que o *Exhaustive* CHAID se impõe como uma evolução do CHAID, ao efectuar uma análise mais exaustiva dos possíveis pontos de corte. O *Exhaustive* CHAID não interrompe o processo de junção como o seu antecessor, ele continua o processo iterativo de junção até restar apenas duas "super-categorias", para depois analisar todas as series de categorias fundidas, de forma a determinar qual o conjunto de categorias que oferece uma melhor relação de associação com a variável dependente, calculando de seguido o *p-value* ajustado para o conjunto escolhido. Assim é possível

determinar o ponto de corte ideal para cada uma das variáveis independentes, e escolher a que oferece melhor qualidade preditiva ao comparar os valores de *p-value* ajustado para cada uma delas. Apesar da vantagem significativa que o *Exhaustive* CHAID oferece em relação ao seu antecessor, este processo exaustivo de análise requer também uma capacidade de processamento consideravelmente superior ao exigido pelo CHAID, facto que não deve ser ignorado [Biggs, 1991].

Algoritmo C5.0

O algoritmo C5.0 é um dos vários algoritmos existentes para gerar árvores de decisão. Representa uma melhoria do algoritmo C4.5, que é por sua vez uma extensão do algoritmo ID3. Todos eles desenvolvidos por Ross Quinlan [Quinlan, 1993]. O C4.5 apresenta melhorias significativas em relação ao ID3, nomeadamente no que toca à capacidade de manipulação de atributos discretos e atributos contínuos, a possibilidade de lidar com casos de *missing data*, a capacidade de lidar com atributos de diferentes custos, e ainda possibilitando a realização de *prunning* após a criação da árvore [Quinlan, 1996].

Já o C5.0 apresenta em relação ao seu antecessor C4.5, as seguintes vantagens:

- Velocidade no processamento, sendo que é notavelmente mais eficaz a processar a árvore do que o C4.5.
- Gestão de memória mais eficiente.
- Suporta *Boosting,* possibilitando resultados mais precisos.
- Produz árvores de menor dimensão.
- Permite *Winnowing* – permitindo eliminar do processo de forma automáticos atributos que poderão não ser relevantes. O C5.0 analisa a utilidade das variáveis independentes antes de iniciar o processo de construção do modelo, eliminando da equação as que considerar irrelevantes para o desempenho do processo. Este processo é útil principalmente quando existem inúmeras variáveis independentes, ajudando a evitar problemas de *overfitting*.

Como pontos fortes do C5.0, em relação a outros algoritmos usados na geração de árvores de decisão, destaca-se a sua robustez quando confrontado com problemas de "*missing values*" e de elevada cardinalidade de variáveis de entrada, e ainda, a sua eficácia no tempo de treino. O C5.0 permite ainda fazer uso da opção de *boosting*, permitindo atingir uma melhoria significativa na precisão dos resultados.

O funcionamento deste algoritmo é similar a outros do mesmo género. Ele começa pela raiz da árvore, e divide recursivamente os nodos. O critério de divisão é baseado no "rácio de ganho de informação" implicada pela escolha de um atributo. O conceito de "ganho de informação" (ou diferença de entropia), de forma simplificada, é a diferença da entropia entre um estado inicial e o estado final, em que se dá uma informação como certa, ou seja, será a diferença de entropia que a divisão da variável causará que irá determinar a sua potencial escolha ou não, sendo que será escolhida a variável que representar um maior ganho de informação [Quinlan, 1993].

Modelos com Boosting e Bagging

O *boosting* é um meta-algoritmo que permite obter melhorias significativas na capacidade de previsão dos modelos preditivos. O *boosting* está disponível nos dois algoritmos de árvores de decisão apresentados anteriormente e foi usado no trabalho desenvolvido nesta dissertação para a implementação de dois modelos extra, de forma a analisar a melhoria de precisão que esta técnica trazia a cada um dos modelos implementados. Analogamente ao *boosting* mencionado anteriormente, foi também implementado um modelo preditivo extra, usando o algoritmo *Exhaustive* CHAID juntamente com a técnica de *bagging*, de forma a analisar a melhoria que esta técnica induz na capacidade preditiva dos modelos.

3.3.2 Regressão Logística

Apresentação Geral

Para além das árvores de decisão foram também desenvolvidos dois modelos preditivos usando a regressão logística como técnica de DM. A regressão logística é uma técnica estatística usada para a previsão da ocorrência de um evento, tomando como dados de entrada um conjunto de variáveis independentes, e adaptando-as a uma função logística, definida por $f(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}}$ (Figura 4), na qual $f(z)$ representa a probabilidade de um possível evento acontecer, e z é a medida de contribuição de todas as variáveis independentes. O valor de z é calculado usando todas as variáveis e os seus respectivos coeficientes de regressão, e pode ser reduzida à seguinte fórmula: $z = \beta_0 + \beta_1\chi_1 + \beta_2\chi_2 + \beta_3\chi_3 + \beta_k\chi_k$, onde β_0 é a "intercepção", que representa o valor de z onde todos os valores das variáveis independentes é zero, $\chi_1..\chi_n$ representa as variáveis independentes e $\beta_1..\beta_n$ os seus coeficientes de regressão.

Como se pode observar a função logística pode receber como *input* qualquer valor desde menos infinito até mais infinito, devolvendo sempre como resultado um valor, correspondendo a uma probabilidade entre 0 e 1 [Agresti, 2007] [Jannedy, et. al., 2003].

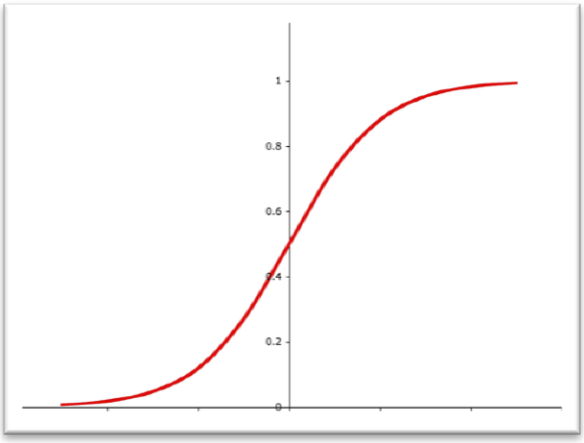

Figura 4 – Função Logística.

Relativamente à cardinalidade que a variável dependente pode assumir, as regressões logísticas são divididas em dois subtipos: regressões logísticas binomiais e regressões logísticas multinomiais. As regressões logísticas binomiais são usadas quando a variável de resposta, ou variável dependente, assume apenas dois valores, normalmente 1 ou 0, indicando verdadeiro/sucesso ou falso/insucesso respectivamente [Weisberg, 2005]. Como seria de esperar, as regressões logísticas multinomiais, são usadas quando a variável dependente é uma variável nominal (constituída por um conjunto de categorias que não podem ser ordenadas de uma forma com sentido).

Selecção das Variáveis

Existem vários métodos disponíveis para a selecção das variáveis independentes que devem ser incluídas no modelo de regressão logística. Estes métodos usam normalmente o teste estatístico *t-test* para fazer a selecção da variável mais significativa, no entanto podem usar outros testes estatísticos tais como o *R-square*, *Akaike information criterion*, *Bayesian information criterion*, *Mallows' Cp*, ou "*false discovery rate*" [Hocking, 1976] [Draper & Smith, 1981] [SAS Institute Inc., 1989]. Os métodos mais conhecidos e usados são conhecidos como *forward selection, backward selection, e stepwise regression*, existindo ainda mais uma serie de técnicas mais complexas e avançadas [Chen, et. al., 2003].

O *stepwise*, é um método para a selecção de variáveis, ele opera de forma iterativa partindo de um modelo sem nenhuma variável seleccionada, e em cada iteração (em cada *step)* todos os termos (variáveis independentes) que ainda não estão incluídas no modelo e que constituem o conjunto de variáveis candidatas, são analisadas, e a que cuja adição ao modelo implicar um maior ganho no poder de precisão do modelo é adicionada. Para além desta selecção, o algoritmo, em cada iteração, faz uma análise dos termos já seleccionados com o intuito de detectar se algum dos termos pode ser excluído sem implicar uma grande perda de precisão final.

O método *forwards* é bastante idêntico ao *stepwise,* ele actua partindo de um modelo extremamente simples sem nenhuma variável seleccionada, e em cada passo, ou iteração, é adicionado uma variável ainda não presente no modelo, esta variável é escolhida entre as variáveis candidatas mediante o valor que trará ao modelo, ou seja, o quanto a sua inclusão no modelo irá melhorar o resultado final do modelo de previsão, caso nenhuma das variáveis candidatas implique uma melhoria mínima, previamente definida, o processo termina, e o modelo final é construído.

O método *backwards* é o processo oposto ao do *forwards* apresentado anteriormente. Neste método parte-se do modelo mais complexo possível, ou seja, que contem todas as variáveis independentes, e de forma iterativa vão sendo excluídos do modelo as variáveis que contribuem pouco para a precisão do modelo final, quando não for possível eliminar mais termos sem prejudicar fortemente a precisão do modelo, o processo termina e o modelo final é construído [SPSS Inc., 2010a].

Para a testar o uso da técnica de regressão logística na previsão de *churn,* foram modelados dois modelos, utilizando respectivamente o algoritmo *forwards* e o *backwards* para a selecção das suas variáveis. A escolha destes dois algoritmos deveu-se ao facto de serem os mais conhecidos e amplamente utilizados. O *software* utilizado (secção 2.4) para a implementação dos modelos de regressão logística binomial oferece suporte para estes dois algoritmos e ainda para um método conhecido como *entry,* que funciona introduzindo todas as variáveis de forma indiscriminada no modelo, sem realizar, portanto, qualquer tipo de selecção. No entanto este método de selecção de variáveis não foi testado, por ser considerado ineficiente.

3.3.3 Redes Neuronais

Apresentação Geral

As redes neuronais (RN) são modelos matemáticos de previsão inspirados na estrutura e no funcionamento das redes neuronais biológicas. De forma análoga às redes neuronais biológicas,

um modelo de uma rede neuronal é constituída por um grupo de elementos base: os neurónios, este elemento é o elemento mais básico da rede neuronal, é um componente capaz de aceitar vários *inputs* e processa-los gerando um resultado (*output*) que pode ser uma previsão do modelo ou um input para outro neurónio. Os neurónios, ou nodos, são agrupados em camadas, ou *layers* de nodos, e encontram-se conectados entre si por um conjunto de ligações com um peso associado, é este peso que determina a influência que um neurónio tem com o outro, o modo como as *layers* e as ligações são definidas depende do tipo de rede neuronal utilizada [SPSS Inc., 2010a] [Byvatov, et. al., 2003].

Tipos de Redes Neuronais

Existem inúmeros tipos de redes neuronais, estes diferentes tipos distinguem-se entre si na forma com que os neurónios interagem entre si e como são organizados internamente no modelo. Alguns dos tipos de RN mais conhecidos são:

- Feedforward neural network [Frean, 1990] [Roman, et. al., 2007] [Auer, et. al., 2008].
- Radial basis function (RBF) network [Buhmann, 2003] [Yee & Haykin, 2001].
- Kohonen self-organizing network [Kohonen, 2007].
- Learning Vector Quantization [Kohonen, 1995].
- Recurrent neural network:
    - Fully recurrent network [Schmidhuber, 1992].
    - Hopfield network [Rojas, 1996].
    - Boltzmann machine [Hinton, 1986].
    - Simple recurrent networks [Rojas, 1996].
    - Echo state network [Jaeger & Haas, 2004].
    - Long short term memory network [Hochreiter & Schmidhuber, 1997].
    - Bi-directional RNN [Schuster & Paliwal, 1997].
    - Hierarchical RNN [Schmidhuber, 1992].
    - Stochastic neural networks [Rojas, 1996].
- Modular neural networks:
    - Committee of machines [Haykin, 1999].
    - Associative neural network (ASNN) [Haykin, 1999].
    - Physical neural network [Snider, 2008].
- Other types of networks.

- Holographic associative memory [Stoop, et. al., 2003].
- Alive networks [Haykin, 1999].
- Instantaneously trained networks [Haykin, 1999].
- Spiking neural networks [Gerstner, 2001].
- Dynamic neural networks [Haykin, 1999].
- Cascading neural networks [Fahlman & Lebiere, 1991].
- Neuro-fuzzy networks [Haykin, 1999].
- Compositional pattern-producing networks [Stanley, 2007].
- One-shot associative memory [Nasution & Khan, 2008].

Apesar do vasto leque de tipos de RN existente, não será feita uma explicação profunda de todos eles, pois não se enquadra nos objectivos desta dissertação, no entanto é apresentado o tipo usado na implementação (*Feedforward neural network*). Apesar de não serem apresentados os restantes tipos de RN, deixa-se referência a alguns dos principais trabalhos realizados a cerca da temática, para futura consulta, se desejado.

FeedForward Neuronal Network – Multilayer percepton

Para o trabalho desenvolvido nesta dissertação, foram implementados três modelos de previsão usando redes neuronais (secção 3.3), para a implementação dos três modelos. O tipo de RN usado foi um dos suportados pelo *software* disponível (secção 2.4): *Feedforward – multilayer perceptron*. O *software* utilizado suporta dois tipos de RN: *Multilayer Perceptron* ou MLP, e o *Radial Basis Function* ou RBF. O MLP apresenta, para casos com um elevado número de relações complexas, um elevado tempo de treino e de teste, por outro lado, o RBF apresenta tempos bastante inferiores ao MLP. No entanto, apesar dos tempos elevados impostos pelo MLP, o seu poder preditivo suplanta largamente o RBF, por esta razão, o tipo de rede escolhido para a implementação dos modelos foi o MLP.

O tipo de RN utilizado neste trabalho foi o conhecido como *Multilayer perceptron* (MLP)*,* que pertence ao tipo FeedForward. Estes tipos de RN são os mais simples que existem, neste tipo de RN a informação não forma ciclos, seguindo em apenas um sentido, partindo dos nodos de *input* (neurónios de *input*), passando para os – caso existam – nodos escondidos (*hidden nodes*), seguindo de seguida para os nodos finais, os nodos de *output* [Frean, 1990] [SPSS Inc., 2010a].

As RN conhecidas como *multilayer perceptrons,* são do tipo de RN *FeedForward,* neste tipo de modelo, os neurónios estão organizados em *layers,* normalmente em três grupos de *layers*, o

primeiro grupo é constituído por apenas uma *layer,* a layer de entrada (*input layer*), o segundo grupo é constituído por uma ou mais *layers,* e são conhecidas como "*layers escondidas",* ou no idioma original: *hidden layers.* Por fim, o terceiro grupo, analogamente à *layer* de entrada, é constituído apenas por uma *layer*, a *layer* de saída (*output layer*). Os neurónios distribuídos pelas várias *layers* encontram-se completamente interligados, ou seja, todos os neurónios da *input layer* se encontram conectados a todos os neurónios da primeira *layer* da *hidden layer,* e por assim em diante, até à última *layer* da *hidden layer*, em que também, todos os neurónios estão conectados a todos os neurónios da *output layer*, um exemplo desta arquitectura é apresentada na Figura 5, onde é apresentado um exemplo de um diagrama de uma rede neuronal *Feedforward* constituída apenas por uma *hidden layer*, pode-se observar que neste tipo de rede, a informação apenas se move num sentido, nunca voltando para trás [Haykin, 1999] [Frean, 1990].

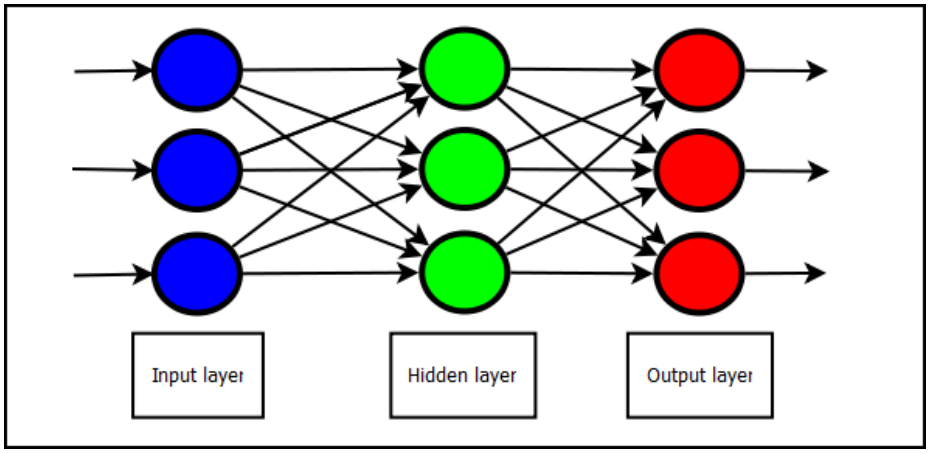

Figura 5 – Rede Neuronal - Feedforward

Como já foi referido anteriormente, as conexões entre neurónios possuem pesos associados, os quais determinam a influência que um neurónio induz no outro. Durante o processo de treino, a informação percorre o sentido *input layer* → *hidden layer(s)* → *output layer*, originando as previsões. O processo de aprendizagem da RN é feita durante o treino, sendo os registos analisados um a um, e conforme a informação flui pelas várias *layers*, é gerado uma previsão para esse registo baseado no valor da variável *target.* Durante este processo é ajustado também o valor do peso, em caso de previsão errada. O processo de treino é repetido inúmeras vezes, melhorando a capacidade de previsão do modelo, terminando logo que alguma das condições de paragem seja satisfeita. Inicialmente o valor de todos os pesos são atribuídos de forma aleatória,

sendo que apenas se uma rede neuronal com sentido após um treino inicial. Os dados de treino são então submetidos à rede iniciando-se o processo de treino. O resultado previsto pela RN é comparada com o resultado real do caso de treino, e o resultado desta comparação é submetido de novo à RN de forma a fazer a actualização dos pesos das ligações que estiveram envolvidas na previsão em causa. Desta forma, a RN torna-se cada vez mais precisa [SPSS Inc., 2010a] [Byvatov, et. al., 2003] [Frean, 1990].

Este tipo de modelo de previsão, possibilita também o uso de técnicas/algoritmos que permitem melhorar significativamente o poder preditivo dos modelos implementados, tal como o *boosting* e o *bagging* (secção 3.3.5), estes dois meta-algoritmos foram usados para implementar dois modelos preditivos extra (cada um deles fazendo uso de um dos meta-algoritmos em questão) de forma a analisar a melhoria na capacidade de previsão que cada um deles permitia obter em relação ao modelo original, que não faz uso de nenhum destes meta-algoritmos.

3.3.4 SVM

Apresentação Geral

O SVM, ou ***S**upport **V**ector **M**achine*, apresentado em 1995 por Vapnik e Cortes [Cortes & Vapnik, 1995], é uma técnica estatística que pode ser usada em problemas de classificação e de regressão. É um algoritmo supervisionado capaz de obter excelentes resultados, principalmente em casos em que os dados a analisar possuem uma elevada cardinalidade de classes independentes (na ordem das centenas) [SPSS Inc., 2010a]. O SVM é uma técnica utilizada em diversas áreas de conhecimento, nomeadamente: genética [Guyon, et. al., 2002], reconhecimento de padrões [Bin, et. al., 2000], optimização de *software* [Bradeley & Mangasarian, 2000] e bioinformática [Byvatov & Schneider, 2003].

Funcionamento da SVM

O SVM opera fazendo um mapeamento dos dados de entrada (Figura 6) para uma hiper-dimensão, de modo aos pontos poderem ser categorizados, e mesmo que os dados não estejam linearmente separados é encontrada uma separação entre as dimensões (Figura 7), de seguida é feita uma transformação aos dados para que a separação criada possa ser redefinida como um hiperplano (ver Figura 8), tendo ainda em atenção que a distância entre cada classe a esse hiperplano é maximizada [Scholkopf, et. al., 2000] [Karatzoglou, et. al., 2006].

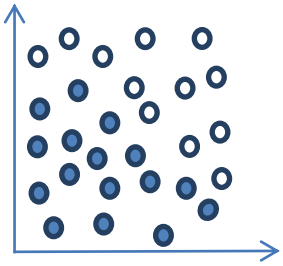

Figura 6 – Data-set Inicial

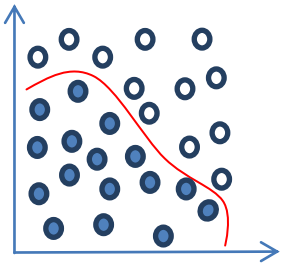

Figura 7 – Dados com separador definido

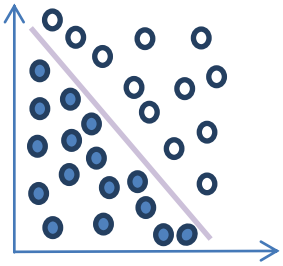

Figura 8 – Dados transformados

Para além do separador das classes, existe o conceito de linhas marginais, as linhas marginais são as linhas que definem a separação entre as duas classes. Cada linha marginal é definida pelos dois (ou mais) pontos mais próximos do separador, o vector definido pelos pontos anteriores é conhecido como vector de suporte. Quanto maior for a distância entre as duas linhas marginais, melhor vai ser a capacidade de previsão do modelo. Por esta razão, por vezes é aceitável induzir erros de classificação, se como resultado for obtido uma margem marginal maior que resultará numa melhor capacidade preditiva.

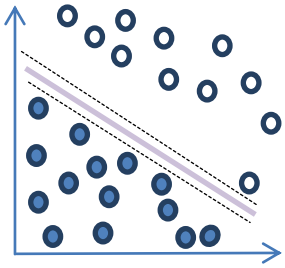

Figura 9 – Linhas marginais no modelo inicial

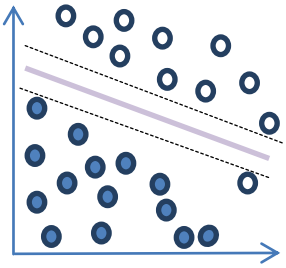

Figura 10 – Linhas marginais no modelo melhorado

Tendo isto em conta, tendo como exemplo o caso apresentado na Figura 9, onde é notável que a distância marginal é curta, ou seja, o modelo gerado irá ter uma fraca capacidade preditiva, seria viável induzir erros de classificação, ao alargar as linhas marginais de forma a aumentar a distância marginal, como é ilustrado na Figura 10 [Byvatov & Schneider, 2003] [SPSS Inc., 2010a].

Por vezes a separação linear não é facilmente realizada, como é o caso apresentado na Figura 11, nestes casos a táctica a seguir será encontrar o balanço ideal entre o valor da distância marginal e o número de casos mal classificados [SPSS Inc., 2010a].

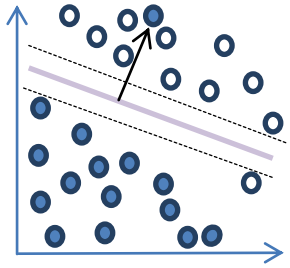

Figura 11 – Problema de separação linear

No processo de transformação dos dados é usada uma função conhecida como função *kernel*, a eficiência do SVM é altamente dependente do *kernel* escolhido [Karatzoglou, et. al., 2006]. Existem várias funções *kernel*, sendo que o software (secção 2.4) usado no desenvolvimento desta dissertação apenas suporta quatro das funções mais conhecidas e utilizadas, apresentadas de seguida: linear, polinomial, função *Radial Basis e* função *Sigmoid.* Estas quatro implementações representam as funções *kernel* mais simples que existem. A função linear deve ser utilizada quando a separação dos dados é relativamente simples e directa, caso contrário, deverá ser usada uma das outras funções [SPSS Inc., 2010a].

SVM Multi-Classe

Os algoritmos SVM foram originalmente desenhados para classificação binária [Cortes & Vapnik, 1995], não sendo capazes de actuar em casos em que existam mais do que duas classes. A sua adaptação a problemas de multi-classes continua a ser um tema em estudo, no entanto existem já várias abordagens ao problema da implementação eficiente de SVM's multi-classe [Hsu Lin, 2002]. Existem várias estratégias para a implementação de algoritmos SVM multi-classe, sendo que a mais popularmente seguida é realizar a redução do problema de classificação multi-classes a vários problemas de classificação bi-classe [Duan & Keerthi, 2005]. Algumas das metodologias mais populares que usam esta estratégia são [Duan & Keerthi, 2005] [Hsu & Lin, 2002]:

- Construção do modelo preditivo binário, fazendo a distinção entre:
    - Uma das classes e o resto das classes existentes, conhecida como *one-versus-all,* é a implementação mais antiga (exemplo de um destes algoritmos é o

usado no trabalho de reconhecimento de caligrafia em 1994 [Bottou, et. al., 1994]), neste tipo de implementação, para a classificação de novos casos é seguida uma estratégia *winner-takes-all*, em que o classificador com o output de maior confiança é a que é usado para fazer a classificação do novo caso.

  - Fazendo distinção entre cada um dos pares de classes possíveis, conhecido como *one-versus-one,* este tipo de implementação foi proposto por Knerr em 1990, e os primeiros trabalhos usando este método foram publicados por Friedman em 1996 [Friedman, 1996] e por Kreßel em 1999 [Kreßel, 1999]. Neste tipo de implementação, na classificação dos novos casos é usado um sistema de voto em que cada classificador determina a que classe pertence o novo caso, e no final a classe que reunir mais votos é a classe "vencedora", sendo o novo caso classificado como pertencendo a essa classe.

- DAGSVM – **d**irected **a**cyclic **g**raph **SVM** é um algoritmo apresentado em 2000 num trabalho de Platt [Platt, et. al., 2000]. O DAGSVM é idêntico à metodologia *one-versus-one,* pois tal como o anterior, na fase de treino para um problema de N classes, ele computa N(N-1)/2 classificadores, ou seja, um para cada par de classes. No entanto, o DAGSVM na fase de teste usa um grafo acíclico e direccionado e binário, em que cada nodo é uma SVM binária com 2 dos classificadores gerados na fase de treino. O processo de classificação é feito partindo da raiz do grafo, e avaliando a função de decisão é tomada a decisão de avançar tomando o caminho da esquerda ou para a direita, dependendo do resultado da função, este processo é efectuado até atingir uma folha do grafo que indicará a classe a que o novo caso pertence [Hsu & Lin, 2002]. Este algoritmo apresenta tempos de treino e de teste consideravelmente melhores que os métodos *one-against-one,* preservando no entanto a precisão das previsões [Platt, et. al., 2000].
- *Error-correction output codes* (ECOC) [Dietterich & Bakiri, 1995], a junção de ECOC e SVM binários, tal como os dois métodos apresentados anteriormente, permite resolver o problema dos SVM multi-classe ao implementar vários classificadores binários, usando-os em conjunto de forma a implementar o SVM multi-classe, a técnica ECOC foi usada no passado em algumas áreas de estudo, nomeadamente em processamento de linguagens e processamento de padrões, um exemplo do uso desta metodologia é o trabalho de Liu em reconhecimento de padrões textuais [Liu, 2006]. Esta metodologia cria uma matriz de códigos para cada classe, sendo que cada classe terá um código

único que a identifica constituído por 1's e 0's. Cada classificador gerado é treinado para fazer uma previsão binária, que originará um resultado que tomará os valores de 1 ou 0. Na fase de teste, é gerado um vector de resultados, resultante da aplicação dos vários classificadores. Este vector será comparado com os diferentes códigos das classes inicialmente definidos na matriz de códigos, e o que apresentar uma menor distância em relação ao vector de resultados, é o escolhido como hipótese da classe [Liu, 2006]. Esta metodologia apresenta duas temáticas sensíveis, um dos temas é a forma como é construída a matriz de códigos, e outro é a forma de cálculo usado para a distância entre os vectores. A ideia geralmente seguida na implementação da matriz de códigos é que se deve manter uma grande separação entre colunas e linhas entre as classes, pois assim, quanto maior for a distancia entre elas, maior a probabilidade de se obter a hipótese correcta, mesmo que alguns dos classificadores retornem erros. Idealmente pretende-se que os classificadores possuam uma baixa correlação entre eles e que produzam diferentes erros, de forma à sua junção no modelo geral ser o ideal. No entanto não existe uma medida bem definida capaz de lidar com os vários tipos de classificadores, principalmente quando os classificadores em questão realizam diferentes tarefas de classificação. Quanto ao cálculo da distância entre os vectores que determina a previsão da classe, existem dois métodos, um usa os dois vectores (o resultante gerado pelos vários classificadores e os vectores da matriz de códigos) e calcula a sua distância usando a fórmula da distância de Hamming. O outro método também usa os mesmos vectores, no entanto, realiza um cálculo intermédio com o vector de respostas. A distância é calculada segundo a seguinte formula: $d(i) = \sum_{k=1}^{N}(p(k,i) - M(k,i)$, em que N é o total dos classificadores, $p(k,i)$ é a probabilidade gerada pelo classificador S_k correspondente à classe C_i e $M(k,i)$ é o valor da matriz de resultados [Dietterich & Bakiri, 1995] [Liu, 2006].

O uso de vários classificadores binários agregados exige, por regra, mais capacidade de processamento. No entanto, o uso de uma função capaz de descriminar várias *labels* ao mesmo tempo elimina a necessidade de usar múltiplos classificadores, potenciando também o uso da dependência entre as várias *labels*. Foi esta linha de pensamento que levou em 2001, Crammer e Singer, a abordar o problema do SVM multi-classe de forma diferente, ao proporem uma metodologia que aborda o problema como um único problema de optimização em vez de o decompor em vários problemas de classificação binária [Crammer & Singer, 2001].

3.3.5 Meta-Algoritmos Auxiliares

Boosting

O *Boosting* é um meta-algoritmo usado em algoritmos preditivos, que tem como objectivo melhorar a precisão dos modelos. Esta técnica conhecida surgiu de uma questão colocada por Michael Kearns: "poderá um conjunto de modelos preditivos fracos, originar um modelo preditivo forte?" Entenda-se como modelo preditivo fraco um modelo que apresenta uma fraca correlação com a classificação correcta, fornecendo resultados apenas ligeiramente melhores do que a tentativa aleatória do resultado. Já um modelo preditivo forte é um modelo que apresenta uma elevada correlação com a classificação correcta [Kearns, 1988]. A questão de Kearns teve a sua resposta no trabalho de Schapire publicado em 1990, onde o autor apresenta o seu trabalho sobre a importância de modelos preditivos fracos em relação a modelos fortes. Este trabalho de Schapire esteve na origem da técnica *Boosting* tal como é conhecida nos dias de hoje [Schapire, 1990].

Existem vários algoritmos para a implementação de *Boosting*, não serão abordados as várias vertentes, apenas se fará uma descrição não exaustiva sobre o comportamento mais vulgarmente utilizado dos algoritmos de *Boosting*. De forma simplificada, os algoritmos de *boosting* permitem obter melhorias significativas na precisão dos modelos criados, ao criar de forma iterativa modelos preditivos – que isolados são considerados fracos – que vão sendo unificados num só modelo, que terá uma elevada precisão. Antes de se iniciar a geração do modelo seguinte, os registos são repesados, e às variáveis com menos expressão no modelo anterior, é dado um peso maior, para que o modelo seguinte se foque em prever eficazmente estes casos. Juntos, estes modelos formam um modelo capaz de classificar novos registos de forma eficaz, usando um conjunto de regras induzidos a partir de todos os modelos gerados [Freund, 1995] [Krause & Singer, 2004]. A forma como estas regras são induzidas dos vários modelos gerados é abordada posteriormente neste capítulo.

Bagging

A técnica de *bagging,* também conhecida como *bootstrap aggregating,* é um meta-algoritmo para modelos de classificação e regressão. Como o *boosting*, o *bagging* permite obter melhorias nos modelos no que toca à estabilidade e a exactidão das previsões. Para além da melhoria na estabilidade e na exactidão dos resultados, a técnica de *bagging* permite reduzir o *overfitting* e a variância dos resultados. Esta técnica foi proposta por Leo Breiman em 1996, com o intuito de

melhorar um modelo preditivo ao agregar classificadores modelados com *data-set* de treino gerados aleatoriamente [Breiman, 1996].

De forma simplificada, a técnica de *bagging* consiste em gerar várias versões de um modelo preditivo, e agregá-las de forma a criar um modelo de previsão agregado. O modelo agregado resultante faz as previsões dependendo do tipo da variável dependente, e dependendo também da forma como é definida o modo como é combinada as regras dos vários classificadores, normalmente para as variáveis numéricas é feita uma média sobre as previsões geradas pelos modelos agregados, caso seja uma classe, a classificação é feita por sistema de voto usando também as previsões dos modelos agregados, no entanto é possível usar diferentes técnicas de combinação das previsões [Breiman, 1996].

A geração das várias versões do modelo de previsão é feita da seguinte maneira, a partir de um *data-set* de treino D com tamanho n, são gerados m novos *data-sets* de treino ($D_1 .. D_m$), com tamanho igual a $n' \leq n$, estes novos *data-sets* são gerados escolhendo observações aleatórias de D e também por substituição. Quer isto dizer que algumas das observações de D podem estar repetidas em D_i. Se $n' = n$, para um valor de n elevado, o *data-set* D_i é esperado ter 63.2% de observações únicas de D, sendo as restantes 36.8% observações repetidas. A está forma de gerar os *data-sets* de treino, é dado o nome de *bootstrap sample* [Breiman, 1996].

Esta técnica permite obter melhorias significativas na exactidão das previsões na maioria dos casos, a factor chave para isto, é a instabilidade dos modelos preditivos, se um *data-set* de treino com ruído influenciar significativamente no modelo gerado, neste caso a técnica de *bagging* melhora a exactidão do modelo [Breiman, 1996]. O tema da instabilidade é estudado no trabalho de Breiman [1994], onde é referido que redes neuronais, árvores de classificação e regressão e um subconjunto de uma regressão linear são instáveis, enquanto o método *k-nearest neighbor* é estável.

Combinando Regras no Boosting e Bagging

Quando se usa técnicas como o *boosting* e o *bagging*, é necessário definir o modo como será feita a classificação dos novos registos, ou seja, como serão utilizados as previsões geradas pelos vários classificadores que constituem o modelo de *boosting/bagging*. Existem várias técnicas de implementar a forma com que as previsões dos vários modelos são combinadas, no entanto, geralmente, para as variáveis dependentes categóricas é usado um sistema de voto simples, voto ponderado ou é utilizado a previsão que oferece melhor probabilidade, já para as variáveis

dependentes contínuas, é usada a média ou mediana dos valores previstos pelos classificadores [Skurichina & Duin, 2000].

Com o sistema de voto simples, são analisadas as previsões dos vários classificadores, e a classe que reunir mais votos, ou seja, a que for dada como classe prevista por mais classificadores, é a classe seleccionada como previsão. Já o sistema de voto ponderado, leva em consideração a probabilidade de cada uma das regras que fez a previsão. Normalmente é usada como função estatística de análise a média das probabilidades. O novo registo será classificado como pertencendo à classe que apresentar uma maior probabilidade média. No caso da técnica "melhor probabilidade", é escolhida a classe indicada pelo classificador que possui a previsão com melhor probabilidade.

Classificador	Classe Prevista	Probabilidade
C1	A	80%
C2	B	79%
C3	A	80%
C4	A	90%
C5	B	89%

Tabela 4 – Resultados de Classificação

De forma a exemplificar as três técnicas referidos anteriormente, é apresentado de seguida um caso de exemplo. Tomemos como hipótese, a modelação de um modelo preditivo que faz uso da técnica de *bagging,* composto por cinco classificadores {C1..C5}, e que os novos registos pertencem a um conjunto possível de classes igual a {A,B}. Os resultados da classificação de um novo registo r são expressos na Tabela 4, onde é apresentada a classificação e a respectiva probabilidade, pelos vários classificadores que constituem o modelo. Pela observação da tabela, podemos concluir que caso seja utilizado o método de voto simples, o novo registo será classificação como pertencendo à classe A, pois é a que reúne mais votos (3 votos) enquanto a classe B só reúne 2 votos. Em caso de se usar o método de voto ponderado, o novo registo r seria classificado como pertencendo à classe B, pois apesar de A ter mais votos, a média das probabilidades para B é de 84% e a de A é de 83.33%. Por fim, se fosse utilizado a técnica "melhor probabilidade", o novo registo seria classificado como A, pois o classificador que possui a maior das probabilidades é o C4 com uma probabilidade de 90%, classificando o novo registo como pertencendo à classe A.

De uma forma geral, o uso do voto simples é normalmente a pior escolho, no que diz respeito à qualidade preditiva do modelo final. O voto ponderado é, na maioria dos casos, a melhor escolha tanto para *bagging* como para *boosting* [Skurichina & Duin, 2000].

Tipo de Variável	Bagging	Boosting
Variáveis Contínuas	Média Ponderada Mediana Ponderada	Mediana Ponderada
Variáveis Categóricas	Voto Simples "Melhor Probabilidade" Voto Ponderado	Voto Ponderado

Tabela 5 – Síntese da Combinação de Regras

Na Tabela 5, são resumidos os métodos de combinação de regras disponibilizados pelo *software* (secção 2.4) mediante o tipo de variável dependente, e o tipo de meta-algoritmo em questão (*bagging/boosting)* utilizado no desenvolvimento desta dissertação.

Capítulo 4

4 Churn em Seguros, Um Caso de Estudo

4.1 As Fontes de Informação

Para a implementação dos modelos desenvolvidos nesta dissertação foram usados dados reais de uma seguradora nacional de média dimensão. Os dados usados para a construção dos modelos correspondem a dados de 24.494 clientes do ramo automóvel. Entre toda a informação de negócio disponível nos sistemas operacionais da seguradora em estudo, foram seleccionados algumas dezenas de indicadores, de categorias díspares. Entre os indicadores destacam-se pela sua relevância para a previsão, indicadores como: o número de contratos em vigor, a antiguidade do cliente e a antiguidade do contrato, a idade, o tipo de fraccionamento, entre outros (secção 4.1.1).

Churn	Frequência	Percentagem	Percentagem Acumulada
Não	20.911	85,4	85,4
Sim	3.583	14,6	100,0
Total	**24.494**	**100,0**	-

Tabela 6 – Frequência de Churn

Dos dados usados, correspondentes a 24.494 clientes, apenas 3.583 desses clientes correspondem a clientes "*churners*" ou seja, clientes que abandonaram o serviço, sendo os restantes 20.911 clientes *não-churners* (Tabela 6).

4.1.1 Parâmetros Seleccionados

Análise Preliminar de Dados

Do conjunto de dados operacionais extraídos dos sistemas fonte, foi construída uma ABT (*Analytical Base Table*) com todos os potenciais indicadores de *churn* que seriam utilizados na modelação dos modelos preditivos. Esta tabela representa nesta fase, uma versão ainda muito básica do que será a tabela final que servirá para alimentar os modelos, pois nesta fase ainda não foi convenientemente analisada e tratada, possuindo um grande número de variáveis que terão que sofrer transformações e até excluídas da análise por não possuírem relevância na previsão suficiente.

À ABT inicial (Anexo A), formada por 104 atributos (incluindo a coluna indicadora de *churn*), foi efectuada uma análise de dados. Nesta análise de dados foram realizados vários testes estatísticos, nomeadamente: verificação de nulos e de valores vazios, existência de *outliers*, existência de variáveis redundantes (cujo significado já se encontra duplicado em outra variável) e ainda a possível combinação de variáveis para formar uma só variável (a título de exemplo, algumas variáveis representando o capital seguro nas diferentes coberturas do seguro automóvel, foram unificadas numa só variável, representando o capital seguro). Desta análise, resultou o seguinte:

- Exclusão de 5 indicadores por falta de qualidade de dados (poucos registos validos).
- Exclusão de 3 indicadores que segundo especialistas do negócio indicaram como não sendo relevantes.
- Criação de 4 novos indicadores a partir de indicadores já existentes, que foram de seguida excluídos (22 indicadores excluídos).
- Criação de uma nova variável binária (sinistro) a partir de outro indicador já existente, por se entender que implicaria vantagens na modelação.

No total, nesta primeira análise de dados, resultou na exclusão de 30 indicadores e na criação de 5 novos indicadores, resultando num universo de 79 variáveis, que, excluindo a coluna que nos indica *churn* ou não-*churn,* ficamos com 78 potenciais indicadores de *churn*. Esta análise preliminar encontra-se disponível no Anexo B onde são apresentadas os indicadores excluídos e indicadores criados nesta primeira análise, bem como a justificação para a acção em questão.

Análise Univariante

Às variáveis resultantes da análise preliminar de dados apresentada anteriormente foi realizada uma análise univariante com o objectivo de identificar as variáveis que individualmente possuem um maior poder discriminante, em relação à variável dependente. Esta análise permite portanto, eliminar indicadores que pouco ou em nada irão contribuir na correcta identificação de casos de *churn*. Este processo de selecção de variáveis mais significativas – que irá induzir uma maior capacidade preditiva aos modelos – é realizado pelos métodos de modelação, o que pode originar a questão: porque realizar esta análise univariante, se posteriormente este processo irá ser repetido, internamento na modelação de cada um dos modelos? A resposta a esta questão é simples, apesar das técnicas de DM oferecerem métodos específicos de selecção de variáveis, este processo pode ser muito pesado a nível de tempo de processamento se o número de variáveis de entrada for elevado. É portanto uma boa prática de modelação, fornecer aos algoritmos de modelação o menor número de variáveis possível, com vista a melhorar a performance do processo de construção dos modelos. Resumindo, esta análise univariante tem como objectivo eliminar as variáveis menos significativas, para que as fases de modelação futuras ocorrem de forma mais célere. Esta análise foi realizada em três fases:

1. Análise do poder discriminante de cada uma das variáveis independentes em relação à variável dependente.
2. Cálculo da correlação linear de cada uma das variáveis em relação à variável dependente.
3. Discretização das variáveis contínuas.

De seguida são explicadas as três fases indicadas anteriormente.

Poder Discriminante

Na análise do poder discriminante das variáveis, é utilizado um teste estatístico distinto mediante o tipo da variável. Para as variáveis nominais, foi aplicado um teste de contingência usando o teste

chi-quadrado [Spiegel & Liu, 1999], já para as variáveis contínuas foi aplicada o teste de Kruskal–Wallis [Kruskal & Wallis, 1952]. Estes dois testes foram utilizados por serem considerados os mais adequados neste tipo de análise [Boulesteix, 2006] [Rosner, 2010]. No caso do teste de contingência foi usado o teste chi-quadrado, onde a hipótese nula é que as variáveis em análise (a variável independente e a variável *target*) são independentes, não sendo o valor da variável *target* influenciada pela variável independente. O nível de significância (*p-value*) calculado pelo teste chi-quadrado indica-nos se devemos rejeitar ou aceitar a hipótese nula, convencionalmente toma-se como aceitáveis para o *p-value,* valores de 0.05 ou menos. Caso o valor da significância seja portanto, 0.05 ou menos, a hipótese nula é rejeitada, implicando que a variável *target* depende da variável independente em teste. Para as variáveis contínuas foi utilizado o teste de Krushal Wallis, este teste foi usado pois é considerado útil a sua utilização em variáveis contínuas. O seu método de funcionamento é idêntico ao anterior, neste teste a hipótese nula é que as variáveis em estudo são independentes uma da outra, e as condições para a rejeição da hipótese nula é um valor de significância (*p-value*) inferior a 0.05. Sendo que caso os testes retornarem valores de significância menores ou iguais a 5%, deve-se rejeitar a hipótese nula, indicando portanto que a variável *target* é de facto depende da variável em teste. O resultado da análise do poder discriminante pode ser analisado no Anexo C. De forma resumida, foram eliminadas por este teste 26 variáveis, todas elas variáveis nominais, por não passarem no teste do chi-quadrado, apresentando valores de significância superiores a 5%.

Correlação Linear

No cálculo da correlação linear, foi analisada a correlação existente entre os vários indicadores e a variável dependente, para esta analisa, foi utilizado o teste do coeficiente de correlação tau de Kendall [Prokhorov, 2002] para as variáveis contínuas, e o Coeficiente de contingência [Lauritzen, 1979] para as variáveis nominais. O coeficiente tau de Kendall permite medir a associação existente entre duas variáveis, o teste tau de Kendall, é assim um teste que usa este coeficiente para testar a dependência estatística entre duas variáveis. O teste devolve um valor entre -1 e 1, sendo que 1 indica uma correlação perfeita, e -1 uma correlação inversa perfeita, 0 indica ausência de correlação entre as variáveis. O teste do coeficiente de contingência aplicada nas variáveis nominais, é idêntico ao teste tau de Kendall, sendo que produz também como resultado um valor entre -1 e 1, em que tal como o anterior, 1 indica uma correlação perfeita, -1 uma correlação inversa perfeita, e 0 ausência de correlação entre as variáveis. Estes dois testes foram usados para medir a correlação entre cada uma das variáveis contínuas e a variável dependente, as variáveis

que apresentaram um valor absoluto de correlação inferior a 0.03 foram excluídas por apresentarem uma baixa correlação, esta regra resultou na exclusão de 12 variáveis. Os resultados destes testes podem ser observados no Anexo D.

Discretização das Variáveis Contínuas

Para além da selecção das variáveis usando o seu poder discriminante e o seu coeficiente de correlação, procedeu-se à discretização das variáveis contínuas [Liu, et. al., 2002]. Este processo consiste em transformar as variáveis contínuas em variáveis categóricas. A categorização das variáveis contínuas justifica-se pelos seguintes aspectos:

- O agrupamento de variáveis permite para a maior parte das variáveis, aumentar o seu poder discriminante.
- A partição das variáveis numéricas permite quantificar a percentagem de operações em cada grupo e determinar qual a tendência da variável dependente.
- Elimina problemas da existência de *outliers*.
- Facilita o tratamento de *missings*.
- Ao incluir as variáveis na modelização na forma categorizada, permite atribuir um peso diferente a cada uma das categorias, ao invés de com um único peso como aconteceria se fosse introduzida na forma de variável contínua. Isto é importante para os casos em que não existe um comportamento linear em relação à variável dependente, que é o caso para muitas das variáveis em questão.

Para a categorização das variáveis contínuas, foi usado um operador disponível no *software* usado para a modelação (secção 2.4), que permite fazer a discretização de variáveis contínuas. Este operador permite criar as categorias, usando várias técnicas, neste caso, foi usada a opção de criação das categorias de "forma óptima" em relação à variável dependente. Este método baseia-se na relação entre a variável a ser discretizada e a variável definida como *target,* permitindo criar as categorias de forma a aumentar o poder discriminante em relação à variável *target.* Este operador de nome "Binning" usa o algoritmo MDLP (*Minimal Description Length Principle*) [Fayyad, et. al., 1993] para a discretização das variáveis contínuas, este método divide uma variável escalar num pequeno número de intervalos *(bins)*, tendo em atenção a sua capacidade discriminante em relação ao *target*. A listagem das variáveis que foram sujeitas a este processo pode ser consultado no Anexo E.

Resultado da Análise Univariante

Resumindo, a análise univariante permitiu excluir de futuras operação, 38 variáveis, 26 das quais por não apresentarem poder discriminante mínimo, e as restantes 12, por não possuírem uma correlação linear em relação à variável dependente suficientemente forte. Para além desta selecção, foi ainda realizada a categorização das variáveis contínuas, de forma a permitir melhores performances quando usadas em determinadas técnicas de DM. Após esta exclusão, a ABT ficou reduzida a 41 variáveis, dos quais uma é a variável que nos indica a ocorrência de *churn*/não-*churn* e as restantes 40 são indicadores de *churn*. Este conjunto de indicadores representa o produto final do tratamento de dados, será este conjunto de dados que será fornecido aos algoritmos de modelação, a lista de variáveis finais pode ser consultada no Anexo F.

4.1.2 Problema de Classes não Balanceadas

Ao se observar a Tabela 6, apresentada anteriormente, constatamos que estamos perante um caso de classes não balanceadas. O problema das classes não balanceadas é um problema que afecta severamente a performance dos algoritmos de classificação actuais [Guo, et. al., 2008], este problema observa-se quando uma população de análise não se encontra classificada de forma balanceada por entre as classes possíveis. De forma mais explicativa, o problema das classes não balanceadas ocorre quando uma classe é representada por um elevado número de exemplos, enquanto, que a(s) outra(s) é/são representada(s) por um número consideravelmente inferior ao anterior. O problema das classes não balanceadas tem implicações graves quando se tem como objectivo maximizar a previsão da classe em menor representação.

A título de exemplo, se tivermos uma população de dados em que, por exemplo, 95% da população esteja classificada como X, e apenas 5% como Y, caso tenhamos como objectivo um modelo de previsão para classificar a população como X ou Y, esta operação pode se revelar extremamente complexa, pois os algoritmos de classificação, serão induzidos a classificar tudo como X devido à superioridade de casos de exemplo com valores X. A título de exemplo, neste caso, um modelo que classifique tudo como X, apresentará 95% de previsões correctas, como se pode observar, apesar de ser um modelo com uma elevada taxa de previsões correctas, é completamente inútil e sem valor preditivo, pois apresenta uma capacidade nula de classificar os casos Y.

O problema de classes não balanceadas tem vindo a ser intensivamente estudado [Kotsiantis, et. al., 2006], existindo actualmente várias abordagens para minimizar este problema. um dos métodos mais utilizados, por ser a abordagem que reúne mais consenso quanto à sua eficácia, é a realização de um novo balanceamento das classes dos dados de treino. Isto é, tendo em conta que o problema reside na diferença de observações entre as classes, procede-se a um balanceamento aleatório das mesmas. O balanceamento das classes pode ser feito de 3 maneiras distintas, por *over-sampling*, *under-sampling* ou por técnicas avançadas de balanceamento que usa uma mistura de conceitos das duas técnicas mencionadas anteriormente. Existem vários estudos sobre balanceamento de classes não balanceadas [Kotsiantis, et. al., 2006] [Weiss, 2003] [Japkowicz, 2000] [Kotsiantis & Pintelas, 2003] [Chawla, et. al., 2002] [Han, et. al., 2005]. No entanto, como o tema não está dentro do âmbito em estudo nesta dissertação de mestrado, será feita apenas uma breve definição e enquadramento das técnicas existentes de balanceamento de classes.

O *under-sampling* consiste em diminuir a classe em maioria até apresentar um volume idêntico ao número de casos da classe em minoria. É um método simplista, que tem como desvantagem a grande probabilidade de se descartar informação importante contida nos casos eliminados da classe em maioria. Por sua vez o *over-sampling* é uma técnica fortemente utilizada em teoria de detecção de sinais [Candy & Temes, 1991] que consiste em equilibrar as classes aumentando o número de casos da classe em minoria, replicando os mesmos. Tem como vantagem, não descartar casos que podem ser importantes para os classificadores, ao contrário do método anterior. A principal desvantagem, é que implica maior carga de processamento, o que pode ser problemático, se o *data-set* de treino já era volumoso antes da replicação da classe em minoria [Guo, et. al., 2008].

Técnicas avançadas de *sampling*, distinguem-se das anteriores, por fazerem uma mistura das duas, e por funcionarem iterativamente, isto é, baseiam-se em resultados de classificações anteriores, para fazer uma nova distribuição dos pesos das classes. Por exemplo, o *Boosting* é considerado um método avançado de *sampling*, onde são realizadas classificações iterativamente, e em cada classificação feita, é analisado os resultados para de seguida ser aumentado o peso da classe que apresentar menor resultados correctos, e diminuir o peso da classe que apresentar mais resultados correctos, para que na próxima iteração, o algoritmo apresente melhores resultados na previsão da classe que está a ser pior classificada.

No trabalho desenvolvido nesta dissertação, foi usada a técnica descrita anteriormente como *under-sampling*. Realizou-se, então, o *under-sampling* da classe em maioria (não-churn), a

técnica de *over-sampling* foi descartada, pois replicar o baixo número de casos de *churn*, até igualar o número de casos de não *churn*, implicaria dar demasiado ênfase aos casos seleccionados de *churn*, o que iria dar origem a modelos de previsão demasiados moldados para os dados de teste *(over-sampling*).

De forma a analisar o impacto que a diferença na proporção de casos de *churn* e *não-churn* implicam nos resultados dos modelos preditivos, foi realizado um teste preliminar de forma a testar duas configurações distintas de conjuntos de dados de treino de forma a determinar qual o mais indicado para o nosso caso de estudo. Antes de apresentar a constituição dos *data-sets,* convêm mencionar, que aos *data-sets* criados foi aplicada a seguinte regra para a modelação dos modelos: os dados de cada um dos *data-sets* em análise foram divididos, em cada um dos casos, por dois volumes de dados, um que servirá para a modelação do modelo – dados de treino – e outro que servirá para testar o modelo desenvolvido – dados de teste –, sendo que os dados de treino foram criados seleccionando aleatoriamente casos de *churn* e casos de não-*churn*, respeitando os rácios que serão de seguida apresentados, os restantes exemplos constituíram os dados de teste. Quanto à constituição dos *data-set* de teste, um deles foi construído de modo a possuir um rácio de 1 para 1 entre os exemplos com *churn* e não *churn* (conjunto este designado daqui para a frente como DS1), e o outro foi construído com um rácio 1 para 2 de casos de *churn* e não-*churn* (DS2). Na Tabela 7 é apresentada a constituição dos dois data-sets usados.

Caso	Data-set	Número de casos		TOTAL
		Churn	***Não-Churn***	
DS1	Treino	2.000	2.000	**4.000**
	Teste	1.583	18.911	**20.494**
DS2	Treino	2.000	4.000	**6.000**
	Teste	1.583	16.911	**18.494**

Tabela 7 – Constituição dos Data-Sets

Após uma simples analise, conclui-se que o uso do DS2 originava melhores resultados (Tabela 8), independentemente do algoritmo usado. Para esta análise foram modelados quatro modelos preditivos, usando diferentes técnicas de DM. Para todos os quatro modelos implementados, obteve-se resultados claramente superiores com o DS2. Este resultado explica-se ao facto de o DS1 não apresentar um rácio de classes realista. Da mesma maneira que os dados iniciais representam um problema de classes não balanceadas, em que o número de registos pertencentes à classe *não-churn* supera, largamente a classe *churn*, provocando caso o modelo seja treinado com dados respeitando este rácio, seja dada uma grande importância aos casos de *não churn,*

classificando praticamente tudo como não *churn*, apresentando o modelo uma exactidão para a classe de *churn* de zero, ou quase zero. Tal como este caso, se treinarmos os modelos com um rácio de 50-50, como é o caso do DS1, o modelo resultante, vai dar igual importância à previsão das duas classes, resultando em maus resultados na exactidão das precisões. A solução a seguir, é encontrar um ponto de equilíbrio, onde a percentagem de casos *churn* é considerada aceitável, sem prejudicar muito a classificação correcta dos casos de *não-churn.* Na Tabela 8 é apresentado os resultados do teste preliminar aos *data-sets* em estudo usando as várias técnicas de DM em análise.

Método	Data Set	Previsões Correctas	Previsões Correctas (abandono = 0)	Previsões Correctas (abandono = 1)
Árvore Decisão C5.0	**Data Set 1**	59.22%	58.61%	66.46%
	Data Set 2	85.33%	91.35%	20.97%
Regressão Logística Binomial (Forwards)	**Data Set 1**	47.7%	46.53%	61.72%
	Data Set 2	71.28%	75.57%	25.39%
Rede Neuronal	**Data Set 1**	46.75%	45.66%	59.76%
	Data Set 2	71.76%	76.67%	19.33%
SVM	**Data Set 1**	49.46%	49.16%	53%
	Data Set 2	57.07%	58.35%	43.34%

Tabela 8 – Resultados do estudo preliminar aos data-sets

4.2 Modelos Implementados

Tal como referido anteriormente foram usadas quatro técnicas distintas de algoritmos de modelação de modelos preditivos. A partir destas quatro técnicas, foram criados vários modelos fazendo variar, dependendo do algoritmo utilizado, opções internas do mesmo, tal como, por exemplo, o uso de meta-algoritmos para melhorar a capacidade preditiva (*boosting e bagging*), ou

ainda a técnica de selecção de variáveis usado, tal como é o caso dos modelos de regressão logística implementados. Os modelos implementados são de seguida listados.

Listagem dos modelos implementados:

1. Modelo A1 – Árvores de Decisão *Exhaustive* CHAID.
2. Modelo A1.1 – Árvore de Decisão *Exhaustive* CHAID com *Bagging*.
3. Modelo A1.2 – Árvore de Decisão *Exhaustive* CHAID com *Boosting*.
4. Modelo A2 – Árvore de Decisão C5.0.
5. Modelo A2.1 – Árvore de Decisão C5.0 com *Boosting*.
6. Modelo RL1 – Regressão Logística Binomial *Forward*.
7. Modelo RL2 – Regressão Logística Binomial *Backwards*.
8. Modelo RN1 – Rede Neuronal.
9. Modelo RN2 – Rede Neuronal com *Bagging*.
10. Modelo RN3 – Rede Neuronal com *Boosting*.
11. Modelo SVM – *Support Vector Machine*.

4.2.1 Modelos A1, A1.1 e A1.2 – Árvores de Decisão Exhaustive CHAID

Os modelos de previsão A1, A1.1 e A1.2 foram implementados usando árvores de decisão (secção 3.3.1). Os três modelos foram implementados usando como algoritmo de crescimento o algoritmo *Exhaustive CHAID*. Os modelos distinguem-se uns dos outros apenas pelo uso das técnicas de *bagging* no modelo A1.1 e *boosting* no modelo A1.2. As configurações usadas em cada um dos modelos em questão são apresentadas na Tabela 9 tendo em conta as opções que são disponibilizadas pelo *software* usado para a implementação dos modelos (secção 2.4).

De seguida é feita uma breve descrição e explicação do significado de cada uma das opções de configuração apresentadas na Tabela 9:

- **Objectivo** - representa o objectivo principal do modelo que se está a modelar. Entre as opções disponiveis encontram-se a possibilidade da construção de um novo modelo partindo do zero ou fazendo decontinuar o treino de um modelo já existente, neste caso a informação já processada mantem-se no modelo, e apenas os novos registos ou registos alterados são dados como *input* ao processo de treino da nova árvore. Para além do objectivo, é possivel definir se queremos criar um modelo preditivo composto por uma

única árvore (single tree), ou se queremos um modelo que usará uma das técnicas de melhoria do resultado que implica a construção de vários modelos, tal como o boosting e o bagging (secção 3.3.5).

Build Options		**Modelo**		
		A1	A1.1	A1.2
Objectivo		Single Tree (Default)	Bagging	Boosting
Algoritmo de Crescimento		Exhaustive CHAID	Exhaustive CHAID	Exhaustive CHAID
Regras de Paragem	Mínimo de Registos no Nodo Pai	0,2%	0,2%	0,2%
	Mínimo de Registos nos Nodos Filhos	0,1%	0,1%	0,1%
	Tamanho Máximo da Árvore	20	20	20
Custos		Default	Default	Default
Regra de previsão para *targets* Categóricos (bagging)		N/A	Voting	N/A
Nº de Componentes no Boosting/Bagging		N/A	10	10
Nível de significância para Merging		0,25	0,25	0,25
Nível de significância para Splitting		0,25	0,25	0,25
Teste Chi-square para variáveis Categóricas		Teste de Pearson	Teste de Pearson	Teste de Pearson
Alteração mínima da Frequência das Células		0,001	0,001	0,001
Iterações máximas por Convergência		100	100	100

Tabela 9 – Configuração dos Modelos de Árvores de Decisão Exhaustive CHAID

- **Algoritmo de crescimento** - Esta opção indica o algoritmo de crescimento da árvore utilizado. Como o *software* utilizado para a implementação destes três algoritmos fornece um operador (nodo) específico do algoritmo de árvores CHAID, apenas é possível neste operador, escolher entre o algoritmo CHAID e o algoritmo *Exhaustive* CHAID, sendo que a

escolha foi nos três casos do *Exhaustive* CHAID, pois apesar de ser um processo mais demorado, o caso de estudo em análise não apresenta dados de treino muito volumosos, que torne o uso deste algoritmo proibitivo.

- **Regras de paragem** - Indicam as regras de paragem do processo de crescimento da árvore, ou seja, definem os *triggers* que ao dispararem provocarão o terminar do processo de crescimento de um ramo específico. É possível definir as seguintes condições de paragem:
    - **Mínimo de registos no nodo pai -** Número mínimo de registos que um nodo pai tem que ter. Caso contrário o processo de divisão pára, não criando nodos filhos a partir deste nodo. Este valor é expresso em percentagem, representando o rácio de registos no nodo em relação ao total de registos no conjunto de dados de treino.
    - **Mínimo de registos no nodo filho -** Número mínimo de registos que um nodo filho tem que ter, caso contrário o nodo filho em questão não é criado, sendo as condições de corte alteradas de forma a incluir mais registos, ou caso não seja possível, terminando o processo de divisão desse ramo. Este valor é expresso em percentagem, representando o rácio de registos no nodo em relação ao total de registos no conjunto de dados de treino.
    - **Tamanho máximo da árvore -** Especifica o número máximo de níveis abaixo da raiz da árvore, ou seja, o número de vezes que os dados vão ser divididos recursivamente). O valor por omissão é 5, mas após alguns testes chegou-se à conclusão que o tamanho 20 seria mais adequado pois apresentava melhores resultados na capacidade de previsão, sem comprometer os resultados ao originar problemas de *overfitting* e sem tornar o processo demasiado pesado computacionalmente.
- **Custos -** Em alguns casos, certos tipos de erros implicam consequências piores que outros tipos de erros, e como tal, deve-se tentar evitar os erros considerados mais graves. Exemplo disto, é por exemplo os testes médicos em que se analisa a existência ou não de uma determinada doença, um falto negativo (o doente possui a doença mas é diagnosticado como não infectado) é considerado um erro mais grave que um falso positivo, em que o doente não é possuidor da doença em questão, mas é diagnosticado como estando doente. Os custos permitem portanto especificar a importância, ou o peso, que cada erro tem no modelo. Estes pesos são levados em atenção no processo de treino

e podem modificar o valor previsto, prevenindo contra erros que se podem revelar graves. Neste contexto é possível definir o custo de cada erro, ou seja, o custo dos falsos positivos e dos falsos negativos. No entanto, para a implementação dos modelos em questão foram utilizados os custos definidos por omissão, 1 para os dois tipos de erro.

- **Nº de componentes no boosting/bagging -** Indica o número de classificadores a criar, para incluir na construção do modelo de *boosting* ou de *bagging*. Este número depende, obviamente do caso em questão.
- **Nível de significância para *merging* -** Este valor é usado no processo de crescimento da árvore e indica o nível de significância mínimo necessário para se fazer a junção de duas categorias. O valor por omissão para este parâmetro é de 0,05. Após a realização de alguns testes, para se determinar qual o valor mais adequado, conclui-se que 0,05 era um valor muito baixo e originava problemas de *overfitting.* Como tal, escolheu-se o valor de 0,25 para este parâmetro.
- **Nível de significância para *splitting* -** Este valor é usado também no processo de crescimento da árvore e indica o nível de significância que uma variável dependente tem que ter para dar origem a uma divisão. Se o *p-value* ajustado da variável em questão for inferior ou igual ao valor definido neste parâmetro este campo é seleccionado como variável de corte - valores baixos tendem a originar uma árvore com poucos nodos. O valor por omissão para este parâmetro é de 0,05. De forma análoga ao parâmetro anterior, realizaram-se alguns testes para tentar determinar com que valor se obtinha os melhores resultados na capacidade de previsão, tendo-se obtido o valor de 0,25.
- **Teste chi-square para variáveis categóricas -** Especifica, para as variáveis categóricas, o método utilizado no teste Chi-square para o cálculo da significância. Aqui, é possível escolher entre dois métodos:
    - **Pearson –** Rápido e não recomendado para data-sets pequenos.
    - ***Likelihood Ratio* –** Apresenta mais robustez que o Pearson, mas implica mais tempo de computação. É aconselhado o seu uso em *data-sets* de teste pequenos, e em casos em que a variável dependente é contínua.

  O método utilizado foi o de Pearson, pois o *data-set* de treino não é considerado de pequena dimensão, e a variável dependente não é contínua. Logo não se justificava o uso do *Lekelihood Ratio*.
- **Alteração mínima da frequência das células -** O processo iterativo de cálculo dos pontos de corte ideais *(epsilon*) usa este parâmetro para determinar quando deve terminar

o processo. O processo épsilon calcula a quantidade de melhoria deve ocorrer para o processo terminar. Se a última iteração provocar uma melhoria inferior ao determinado por este parâmetro, então o processo termina.

- **Iterações máximas por convergência -** Número máximo de iterações antes de parar o processo de crescimento.

4.2.2 Modelos A2, A2.1 – Árvores de Decisão C5.0

Os modelos de previsão A2 e A2.1 foram implementados usando árvores de decisão. Estes dois modelos, ao contrário dos modelos apresentados em secções anteriores, foram implementados empregando outro tipo de algoritmo para a implementação da árvore: o algoritmo C5.0. Os dois modelos em questão são idênticos na sua configuração, apenas diferindo no uso da técnica de *boosting* no modelo A2.1. Este modelo foi implementado recorrendo a essa técnica no algoritmo de *boosting* de forma a analisar a melhoria de desempenho preditivo que a técnica em questão permite obter.

De seguida, na Tabela 10, estão apresentadas as configurações usadas nos modelos anteriores, tendo em conta as opções que são disponibilizadas pelo *software* usado para a implementação dos modelos (secção 2.4).

Build Options	**Modelo**	
	A2	A2.1
Tipo de Output	Árvore de Decisão	Árvore de Decisão
Uso de Boosting	Não	Sim
Nº de Componentes no Boosting	N/A	10
Severidade do Pruning	60	60
Mínimo de Registos nos Nodos Filhos	20	2
Pruning Global	Sim	Sim
Winnowing	Sim	Sim

Tabela 10 – Configuração dos Modelos de Árvores de Decisão C5.0

De seguida é feita uma breve descrição e explicação do significado de cada uma das opções de configuração expressas na Tabela 10:

- **Tipo de output -** O operador disponível no *software* (secção 2.4) usado para a modelação dos modelos preditivos oferece a possibilidade de apresentar os resultados do algoritmo de duas formas distintas: em formato de árvore de decisão ou num conjunto de regras. O tipo seleccionado foi a árvore de decisão, se bem que em termos práticos o resultado a nível de capacidade preditiva seria o mesmo, pois esta opção apenas define a forma com que o resultado é apresentado ao utilizador.
- **Uso de *boosting* -** Esta opção permite fazer uso do meta-algoritmo *boosting* (secção 3.3.5).
- **Nº de componentes do boosting -** Indica o número de classificadores a criar, para incluir na construção do modelo de *boosting.*
- **Severidade do pruning -** Determina a extensão máxima do *pruning* a aplicar. Este parâmetro apenas é usado no "*pruning* local". O aumento deste valor permite obter uma árvore menor e mais simples. Diminuir, normalmente, origina resultados mais precisos, mas pode originar, como sabemos, *overfitting.* O valor por omissão para este parâmetro é 75. No entanto, após alguns testes onde foram implementados alguns modelos, fazendo alternar o valor deste parâmetro, concluímos que 60 permitiria obter resultados mais precisos, sem aumentar muito a dimensão do modelo.
- **Mínimo de registos nos nodos filhos -** Número mínimo de registos que um nodo filho tem que ter, caso contrário o nodo filho em questão não é criado, sendo as condições de corte alteradas de forma a incluir mais registos, ou caso não seja possível, terminando o processo de divisão desse ramo. Após alguns testes de implementação, observou-se que 20 como valor para este parâmetro, juntamente com os parâmetros actuais, permitia obter bons resultados. No entanto isto só se verifica no modelo A2, pois no modelo A2.1 onde se faz uso do *boosting*, obtivemos melhores resultados com o valor de omissão 2.
- **Pruning global -** O *pruning* global permite realizar o processo de *pruning* em duas fases. Primeiro é realizado o "*pruning* local", ao nível das subárvores, colapsando ramos de forma a melhorar a exactidão do modelo. Depois, numa segunda fase em que se considera a árvore como um todo, realiza-se um "*pruning* global", no qual pode ocorrer o colapsar de subárvores consideradas fracas**.**
- **Winnowing -** Esta opção permite eliminar à cabeça, variáveis independentes que são consideradas irrelevantes. Não foi utilizado esta funcionalidade do C5.0, pois as variáveis usadas nos modelos já tinham sido submetidas, previamente, a um outro processo de

selecção e tratamento. Logo apenas um conjunto relativamente restrito de variáveis foi dado como *input* ao algoritmo, o que fez com que esta técnica não trouxesse grandes melhorias, já que não estávamos perante um problema com um número elevado de variáveis independentes. Para confirmar esta suspeita, foi implementado um modelo de teste, no qual foi usado esta opção, obtendo-se uma pequena melhoria na ordem das casas decimais na percentagem de previsões correctas. No entanto esta melhoria ocorreu devido a um aumento, também ligeiro na capacidade de previsão da classe não-churn, enquanto a classe *churn* sofreu uma diminuição na capacidade preditiva. Ou seja, apesar do aumento na capacidade preditivo geral, a capacidade de previsão da classe *churn* diminuiu, sendo compensada positivamente pelo aumento da capacidade preditiva da classe não-*churn*. Este não é no entanto um resultado aceitável, pois a capacidade preditiva da classe *churn* é prioritária em relação ao não-churn. Esta opção foi assim descartada**.**

4.2.3 Modelos RL1 e RL2 – Regressão Logística Binomial

Foram implementados nesta dissertação dois modelos de previsão de *churn* usando regressão logística binomial como técnica de DM. Os dois modelos implementados com esta técnica (RL1 e RL2) distinguem-se um do outro apenas no algoritmo que usam para fazer a selecção das variáveis independentes que devem ser incluídas. O modelo RL1 faz uso do algoritmo *Forwards* para a selecção das variáveis. Já o RL2 foi implementado utilizando o algoritmo *Backwards,* que funciona de modo oposto ao *Forwards*. Na Tabela 11 são apresentadas as configurações utilizadas nos dois modelos referidos anteriormente.

De seguida é feita uma breve descrição e explicação do significado de cada uma das opções de configuração expressas na Tabela 11:

- **Tipo -** Indica o tipo da regressão, pode ser binomial ou multinomial. As regressões binomiais são usadas quando a variável dependente, ou *target*, é uma *flag*, ou seja, assume apenas dois possíveis valores. A regressão multinomial é usada quando a variável *target* é nominal com mais de dois possíveis valores.
- **Algoritmo de selecção de variáveis -** Indica o algoritmo utilizado para a selecção das variáveis a utilizar no modelo. Para as regressões binomiais é possível escolher entre três algoritmos: *Enter*, *Forwards*, e *Backwards*.

- **Parâmetros de convergência -** Estes parâmetros permitem controlar o processo de convergência do modelo, ou seja, o número e o modo com que os parâmetros são processados de modo a analisar o seu desempenho. Quanto maior for a iteração de testes sobre os parâmetros, mais precisos irão ser os resultados.

    - **Nº máximo de iterações -** Define o número máximo de iterações que se irá realizar no processo de análise de adequação aos vários parâmetros. O valor deste parâmetro por omissão é de 20, no entanto, após alguns testes preliminares, concluiu-se que um valor apropriado para o nosso caso de teste seria 10, pois permitia obter resultados ligeiramente melhores, nomeadamente na previsão dos casos de *churn*.
    - **Convergência Log-Likelihood -** O processo de iteração pára se a diferença relativa entre o valor do *log-likelihood* for menor que o definido neste parâmetro. Este critério de paragem não é utilizado se for definido como 0.
    - **Convergência dos parâmetros - -** O processo de iteração pára se a diferença na estimativa do parâmetro for menor que o valor definido neste parâmetro. Este critério de paragem não é utilizado se for definido como 0. Após alguns testes concluiu-se que no nosso caso de estudo a manipulação deste parâmetro não alterava de forma significativa os resultados. Assim, considerámos o valor por omissão de 1.0E-6.
- **Critério de remoção -** Permite definir o critério usado durante processo de selecção das variáveis para fazer a sua exclusão ou introdução no modelo. Estão disponibilizados três testes: *Likelihood Ratio*, Teste de *Wald* e *Conditional Test.* O teste *Likelihood ratio* permite obter modelos robustos, enquanto o *Wald* não produz resultados tão fiáveis, mas no entanto permite obter tempos de treino muito inferiores ao teste anterior. Para as regressões logísticas binomiais, existe disponível ainda a opção: *Conditional*, que é idêntica à do teste *likelihood ratio,* exceptuando que usa estimativas como parâmetro. Para esta opção foi deixado ficar o valor por omissão, o *likelihood ratio*, por ser aquele que permite obter melhores resultados na capacidade preditiva do modelo.
- **Significância para entrada -** Indica o valor máximo da significância (*p-value*) que uma variável poderá apresentar para ser adicionada a um modelo, ou seja, apenas as variáveis com um *p-value* inferior ao definido neste parâmetro serão adicionadas. Foram feitos

alguns testes preliminares no qual se concluiu que para o nosso caso de teste o valor por omissão (0,05) usado neste parâmetro era o aconselhado.

- **Significância para remoção -** Indica o valor mínimo da significância (*p-value*) que uma variável terá que apresentar para era retirada do modelo, ou seja, uma variável será removida de um modelo apenas se o seu *p-value* for superior ao valor definido neste parâmetro. Como habitualmente, foram realizados alguns testes preliminares no qual se concluiu que para o nosso caso de teste o valor por omissão (0,1) usado neste parâmetro era o mais adequado.

Build Options		**Modelo**	
		RL1	RL2
Tipo		Binomial	Binomial
Algoritmo de Selecção de Variáveis		Forwards	Backwards
Parâmetros de Convergência	Nº Máximo de Iterações:	10	10
	Convergência Log-Likelihood	0	0
	Convergência dos Parâmetros	1.0E-6	1.0E-6
Critério de Remoção		Likelihood Ratio	Likelihood Ratio
Significância para Entrada		0,05	0,05
Significância para Remoção		0,1	0,1

Tabela 11 – Configuração dos Modelos de Regressão Logística

4.2.4 Modelos RN1, RN2 e RN3 – Rede Neuronal

Durante a realização dos trabalhos foram implementados mais três modelos de previsão de *churn* usando redes neuronais (secção 3.3.3). A implementação dos três modelos foi necessária devido à necessidade de comparar o desempenho do modelo de redes neuronais, com (e sem) o uso de meta-algoritmos auxiliares, como é o caso do *boosting* e do *bagging* (secção 3.3.5). Assim, foram implementados três algoritmos (RN1, RN2 e RN3). O RN1 é a implementação simples da rede neuronal, sem o uso de qualquer meta-algoritmo auxiliar, o RN2 difere do RN1 por fazer uso do *boosting*, e o RN3 por fazer uso do *bagging.* A configuração dos três modelos anteriores é apresentada na Tabela 12.

Build Options		Modelo		
		RN1	RN2	RN3
Objectivo		Standard Model	Bagging	Boosting
Tipo de Rede Neuronal		MLP	MLP	MLP
Hidden Layers		Automático	Automático	Automático
Regras de Paragem	Tempo de Treino	15	15	15
	Ciclos de Treino	-	-	-
	Precisão Mínima	-	-	-
Regra de para *targets* Categóricos (bagging)		N/A	Maior Probabilidade Média	N/A
Regra de para *targets* Contínuos (bagging)		N/A	Média	N/A
Nº de Componentes no Boosting/Bagging		N/A	10	10
Overfit Prevention Set		30%	30%	30%

Tabela 12 – Configuração dos Modelos de Redes Neuronais

De seguida é feita uma breve descrição e explicação do significado de cada uma das opções de configuração expressas na Tabela 12:

- **Objectivo** - Aqui é definido o objectivo do modelo que se quer modelar, é possível definir se se quer modelar um novo modelo ou continuar o treino de um modelo já existente. Neste caso a informação já processada mantém-se no modelo e apenas os novos registos ou registos alterados são dados como *input* ao processo de treino do novo modelo. Isto permite obter resultados de forma muito mais rápida, pois evita aceder e realizar processamento desnecessário. Para além disto, é possível definir se queremos criar um modelo *standard* ou se queremos um modelo que reúna vários, ao usar uma técnica de melhoria dos resultados, tendo disponíveis as opções de usar o meta-algoritmo *boosting* e o *bagging* (secção 3.3.5).
- **Tipo de rede neuronal -** Representa o tipo de rede neuronal que ser quer implementar, que, entre outras coisas, definirá a forma com que os neurónios interagem entre si e como são organizados internamente no modelo. Os tipos suportados pelo *software* são o

Multilayer Perceptron (MLP), e o *Radial Basis Function* (RBF). O MLP apresenta, para casos com um elevado número de relações complexas, um elevado tempo de treino e de teste. Por outro lado, o RBF apresenta tempos bastante inferiores ao MLP. No entanto, apesar dos tempos elevados impostos pelo MLP, o seu poder preditivo suplanta largamente o RBF. Por esta razão, o tipo de rede escolhido para a implementação dos modelos foi o MLP.

- ***Hidden layers* -** Este parâmetro permite indicar o número de neurónios em cada *hidden layer* do modelo. É possível definir se se quer que o número seja automaticamente calculado (automático) ou não, definindo-se neste caso o número de neurónios em cada uma das duas *layers.* Para este parâmetro foi usado o valor por omissão – automático – sendo que neste caso a RN é criada com apenas uma *hidden layer*, e o seu número de neurónios é calculado automaticamente durante o processo.
- **Regras de paragem –** Estas definem as condições de paragem de treino da rede neuronal:

    - **Tempo de Treino -** Permite definir o tempo máximo de treino de um modelo em minutos. Caso seja um modelo que usa *boosting* ou *bagging,* este tempo refere-se ao tempo máximo por componente. Para este parâmetro foi utilizado o valor por omissão de 15 minutos, pois é um valor que, devido a volumetria de dados com que se estava a trabalhar, se achou ser razoável para o processamento do modelo.
    - **Ciclos de treino -** Define o número máximo de ciclos de treino permitido. Esta condição de paragem está por omissão desactivada. Foi feito um pequeno teste preliminar e conclui-se que o uso desta condição não era necessária na implementação do modelo em questão, pelo que se manteve a opção inactiva.
    - **Precisão mínima -** Com esta condição activa, o processo de treino vai prosseguir até atingir o valor de precisão indicado pela condição. Esta condição de paragem está por omissão inactiva. Alguns testes que foram realizados permitiram concluir que o uso desta condição não era necessária na implementação do modelo em questão. Como tal, manteve-se a opção inactiva.

- **Regra para *targets* categóricos -** Este parâmetro apenas é usado no caso de uso da técnica de *bagging* e caso a variável dependente seja categórica. Define a técnica que o modelo final irá usar para fazer a previsão da variável *target*. A classificação pode ser feita usando três técnicas distintas: votação, maior probabilidade ou maior probabilidade média.

Após alguns testes preliminares, concluiu-se que a técnica que melhores resultados oferecia era a da maior probabilidade média.

- **Regra para *targets* contínuos -** Parâmetro apenas usado no caso de uso da técnica de *bagging* e em situações em que a variável dependente seja contínua. Pode-se escolher duas formas distintas para este parâmetro: a mediana e a média dos valores previstos pelos vários modelos. O valor por omissão é a média (*mean*). Esta opção foi preservada, já que não é relevante para o nosso modelo, uma vez que a variável *target* do modelo implementado é categórica.
- **Nº de componentes no boosting/bagging -** Este parâmetro permite definir o número de modelos que se pretende construir, no caso de estarmos a usar uma das técnicas de *bagging* ou *boosting.* Este parâmetro foi definido com o valor de 10, pois é considerado um valor mediano, não implicando excesso de processamento, e uma vez que oferece uma melhoria significativa na precisão do modelo final.
- **Overfit prevention set -** Nas redes neuronais, o *data-set* de treino é internamente separado de forma a criar dois data-sets distintos, um para a construção do modelo, e outro chamado de *overfit prevention set*. Este último data-set é usado para identificar possíveis erros durante o processo de treino. O valor de omissão de 30% foi usado.

Nota: As técnicas de combinação das regras dos vários modelos que constituem um modelo de *boosting* não são possíveis de especificar, pois ao contrário do *bagging* onde é possível definir estas técnicas, para o *boosting* é usado sempre, para as variáveis categóricas o voto ponderado, e para as variáveis contínuas a média ponderada.

4.2.5 Modelo SVM – Support Vector Machine

Foi também implementado um modelo com base na técnica *Support Vector Machine* (secção 3.3.4). Na Tabela 13 é feita uma apresentação da configuração utilizada no desenvolvimento do modelo SVM.

De seguida é feita uma breve descrição e explicação do significado de cada uma das opções de configuração expressas na Tabela 13:

- **Critério de paragem**: Permite definir quando parar o processo de optimização do SVM. Aceita valores entre 1.0E-1 até 1.0E-6, sendo que quanto menor for o valor, mais preciso

será o modelo resultante, mas maior será o tempo de treino. Após alguns testes preliminares chegou-se ao valor 1E-2 como sendo um valor aconselhável para o nosso modelo.

- **Parâmetro de regulação (C):** Este parâmetro permite regular a relação entre a maximização da margem marginal e o número de casos mal classificados. Este parâmetro encontra-se entre 1 e 10, sendo o valor por omissão 10. Quanto maior for o valor deste parâmetro, maior será a capacidade preditiva do modelo, mas também, maior será a probabilidade de problemas de *overfitting.* Após alguns testes preliminares conclui-se que valores muito altos para este parâmetro induziam a grandes problemas de *overfitting,* a análise efectuada permitiu chegar ao valor 5 como parâmetro de regulação que permite obter bons resultados sem correr riscos elevados de *overfitting.*
- **Tipo de *kernel*:** Determina o tipo de função *kernel* a usar na transformação. É possível escolher entre os seguintes tipos de *kernel*: RBF, Linear, Polinomial e *Sigmoid* (secção 3.3.4). Os diferentes tipos de *kernel* implicam diferentes formas de calcular as divisões entre as classes, por esta razão foi necessário testar os quatro *kernels* disponíveis, de forma a verificar qual permitia melhores resultados para o caso de estudo em causa. Após o teste concluímos que o RBF *Radial Basis Function* era o que nos permitia obter melhores resultados.
- **RBF gama:** Usado quando o *kernel* escolhido é o RBF. Normalmente, valores aconselhados situam-se entre 3/k e 6/k onde k representa o número de variáveis dependentes. Quanto maior o valor deste parâmetro, maior a capacidade preditiva, mas também maior o risco de *overfitting.* Apôs alguns testes, conclui-se que 0.1 é um valor aconselhável para este parâmetro pois oferece um bom rácio entre a precisão e o *overfitting*.

Build Options	***Modelo***
	SVM
Critério de Paragem	1E-2
Parâmetro de Regulação (C)	5
Tipo de Kernel	RBF
RBF Gama	0.1

Tabela 13 – Configuração do Modelo SVM

Capítulo 5

5 Análise de Resultados

5.1 Sobre a Análise de Resultados

Este capítulo encontra-se dividido em duas secções distintas. Na primeira secção são apresentados os resultados individuais dos modelos implementados nesta dissertação. Os modelos foram avaliados tendo em conta as métricas de desempenho seleccionados no início do processo de implementação (secção 2.3). Na segunda secção será feita uma análise comparativa entre os modelos, realçando qual o(s) melhor(es) e em que condições essa superioridade é aplicável. São também apresentados os indicadores mais relevantes para o caso de estudo trabalhado, bem como dois casos de aplicabilidade dos modelos seleccionados.

5.2 Resultados Individuais dos Modelos

5.2.1 Árvores de Decisão CHAID

Para os modelos implementados usando árvores de decisão CHAID (secção 4.2.1) obteve-se *hit-rates* para os casos de *churn* de 27.54%, 28.42% e 37.21%, respectivamente para a versão simples, para a

versão com *bagging* e para a versão com *boosting*. Pela análise dos rácios de previsões correctas (Tabela 14), podemos concluir que de entre estes três modelos, o que se destaca pela positiva é o modelo A1.2. Este modelo apresenta resultados consideravelmente superiores para a detecção de casos de *churn,* do que os restantes dois e apesar do modelo com *bagging* – modelo A1.1 – apresentar a melhor capacidade preditiva geral (81.36%), a sua capacidade preditiva para os casos positivos de abandono é consideravelmente inferior em relação ao modelo A1.2, tornando-o numa escolha não viável, pois a correcta classificação dos casos de *churn* é de extrema importância.

Modelo	**Algoritmo**	Previsões Correctas		
		Geral	Abandono = 0	Abandono = 1
A1	CHAID (Normal)	78.64%	83.42%	27.54%
A1.1	CHAID (Bagging)	81.36%	86.41%	28.42%
A1.2	CHAID (Boosting)	72.97%	76.32%	37.21%

Tabela 14 – Resultados Árvores de Decisão CHAID

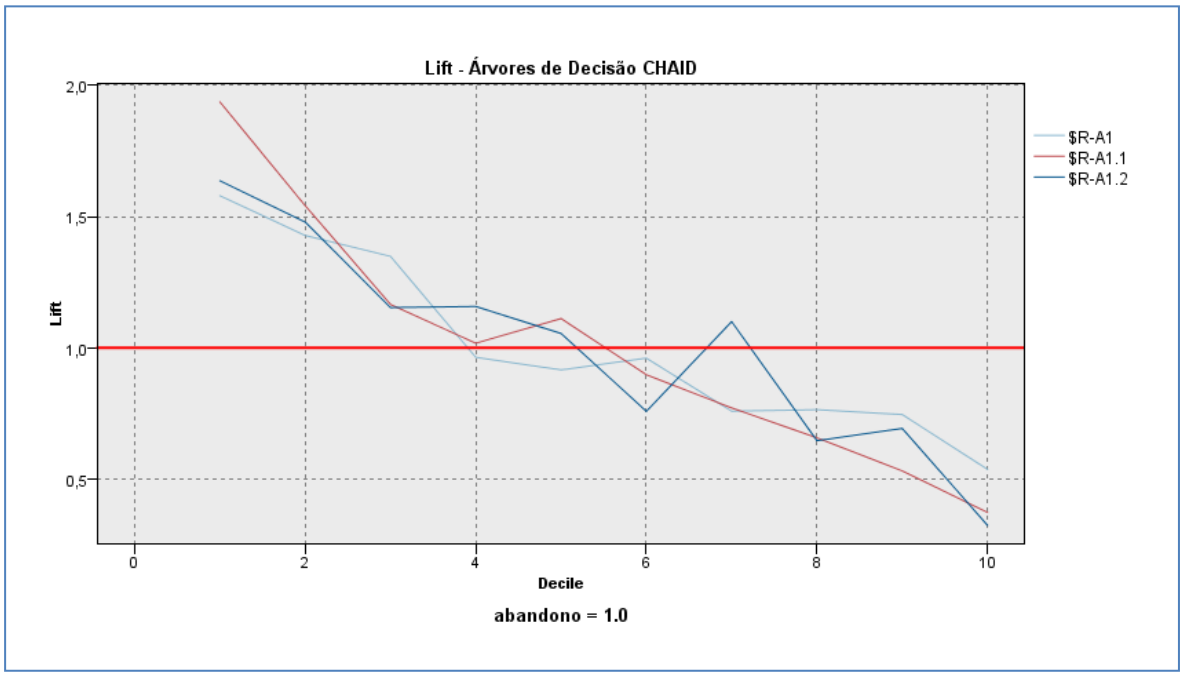

Figura 12 – Lift – Árvores de Decisão CHAID

Quando analisado os três modelos anteriores tomando como métrica de comparação o *hit-rate*, é clara a superioridade do modelo A1.2 (*boosting*). No entanto, quando analisadas as curvas de *lift*

(Figura 12) produzidas pelos três modelos, podemos observar que o modelo A1.1 produz valores superiores para os dois primeiros *decils*. Isto quer dizer que na previsão dos 20% de registos com maior probabilidade associada, o modelo A1.1 sai-se melhor que os outros dois. Porém, depois do segundo *decil*, o seu valor de *lift* cai para valores inferiores ao dos valores apresentados pelos outros dois modelos. Assim podemos concluir que a escolha entre os dois modelos anteriores dependeria do objectivo definido, podendo se optar entre os modelos A1.1 e o A1.2.

5.2.2 Árvores de Decisão C5.0

Para os dois modelos implementados usando o algoritmo de árvores de decisão C5.0 (secção 4.2.2), obteve-se resultados de *hit-rate* de 30.51% e 33.8% respectivamente para a versão simples do C5.0 e para a versão com *boosting*. Na Tabela 15 são apresentados os resultados das previsões dos modelos em questão.

Modelo	**Algoritmo**	Previsões Correctas		
		Geral	Abandono = 0	Abandono = 1
A2	C5.0 (Normal)	82.26%	87.1%	30.51%
A2.1	C5.0 (Boosting)	84.01%	88.71%	33.8%

Tabela 15 – Resultados Árvores de Decisão C5.0

Pela análise da Tabela 15 podemos concluir que o uso da técnica de *boosting* no algoritmo C5.0 é bastante benéfico, pois apesar de apenas aumentar a precisão do modelo em cerca de 2 casas decimais, de 82.26 para 84.01, permitiu um aumento na capacidade de prever os casos de *churn* (abandono = 1) superior a 10%, ao aumentar de 30.5%, para 33.8%.

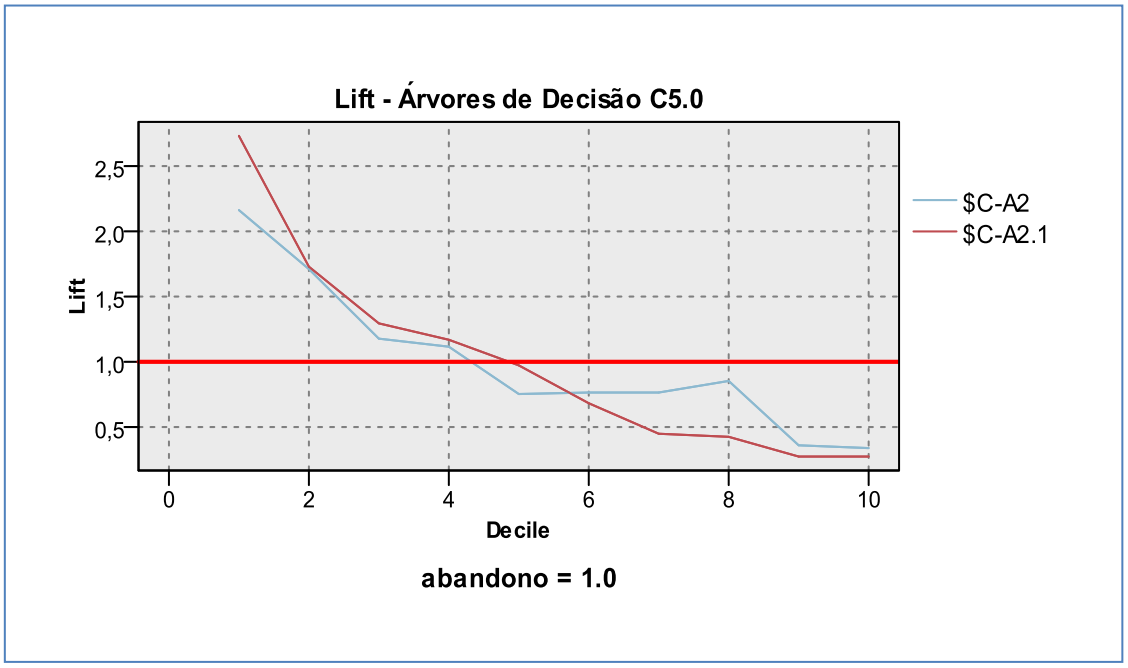

Figura 13 – Lift – Árvores de Decisão C5.0

Também quando comparados os resultados dos modelos, em relação aos valores do *lift* obtido, concluídos que o modelo que tira partido da técnica de *boosting* é ligeiramente melhor que a versão simples do C5.0. Pela análise da Figura 13 podemos observar que o modelo A2.1 oferece resultados substancialmente superiores no 1º e 2º *decil* em relação ao modelo A2, sendo que após o 2º *decil*, os resultados apresentam-se apenas ligeiramente melhores em relação ao modelo A2. O modelo A2.1 cruza a linha de *lift* = 1, para valores próximos do 5 *decil*, enquanto o modelo A2 cruza a mesma linha para valores de *lift* ligeiramente superiores a 4, indicando que o modelo A2.1 oferece vantagem de uso, em quase 50% da população, enquanto o modelo A2 em apenas pouco mais do que 40%.

5.2.3 Regressão Logística Binomial

Tal como apresentado na secção 4.2.3, foram implementados dois modelos usando a técnica de regressão logística binomial. Os rácios de previsões correctas foram de 72.42% e 71.95%, para o modelo RL1 e RL2 respectivamente. Pela análise da Tabela 16, podemos concluir que os modelos apresentam capacidade preditivas bastante idênticas, ambos apresentam uma percentagem de previsões correctas de cerca de 72%, apenas variando um pouco na capacidade de previsão dos

casos de *churn* (abandono = 1), onde o modelo RL2 conseguiu um desempenho ligeiramente superior ao modelo RL1, com uma precisão de 23.94% contra os 23.25% do RL1.

Modelo	**Algoritmo**	Previsões Correctas		
		Geral	Abandono = 0	Abandono = 1
RL1	Regressão Logística Binomial (Forwards)	72.42%	77.03%	23.25%
RL2	Regressão Logística Binomial (Backwards)	71.95%	76.44%	23.94%

Tabela 16 – Resultados Regressão Logística Binomial

Tal como na análise do rácio de previsões correctas dos dois modelos, também pela análise das curvas de *lift* apresentadas na Figura 14 concluímos que ambos os modelos são bastantes idênticos, apresentando valores de *lift* quase coincidentes para todos os dez *decils* analisados.

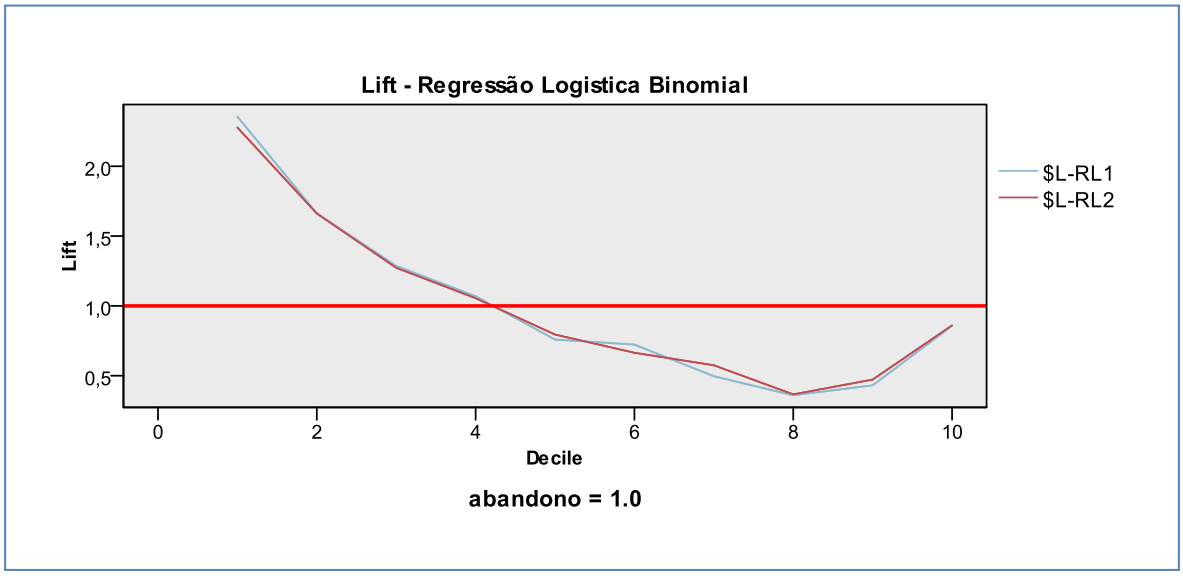

Figura 14 – Lift – Regressão Logística Binomial

5.2.4 Rede Neuronal

Foram implementados três modelos usando a técnica de redes neuronais (secção 4.2.4). Estes três modelos apresentam uma precisão geral de 69.42% para o modelo RN1, de 67.64% para o modelo

RN2 que tira partido da técnica de *bagging* e de 60.4% para o modelo RN3 que faz uso do *boosting* (Tabela 17). Quando comparado entre si tendo em conta o rácio de previsões correctas, constatasse que o modelo RN1 apesar de ter o melhor desempenho geral, prevendo correctamente 69.42% dos casos de teste, tem uma fraca capacidade preditiva dos casos de *churn,* com apenas 26.15% dos casos classificados correctamente, o que é um fraco desempenho comparado com os 30.26% do modelo RN2, e os 35.82% do RN3. O modelo RN3 (com *boosting*) apresenta-se assim como o melhor entre os três, quando tomando como medida de comparação a capacidade preditiva dos casos de *churn.*

Modelo	**Algoritmo**	Previsões Correctas		
		Geral	Abandono = 0	Abandono = 1
RN1	Normal	69.42%	73.47%	26.15%
RN2	Bagging	67.64%	71.14%	30.26%
RN3	Boosting	60.4%	62.7%	35.82%

Tabela 17 – Resultados Rede Neuronal

Apesar dos resultados obtidos no que toca à capacidade preditiva dos modelos, quando analisado os valores de *lift* dos modelos apresentados na Figura 15, constata-se que o modelo RN3 é entre os três, o modelo que apresenta menor *lift* para o primeiro *decil*, para depois a partir do segundo *decil* ser o que apresenta melhores resultados. O RN1 e o RN2 apresentam valores de lift bastante idênticos, apenas se destacando a superioridade do RN1 ao cruzar a linha de *lift* = 1 juntamente com o RN3 depois do quarto *decil*, enquanto o RN2 cruza-a pouco antes do quarto *decil*. Perante estes resultados, podemos concluir que entre estes três modelos, a escolha do mais adequado pode pender entre o RN2 e o RN3. Caso o objectivo seja fazer uma classificação de toda a população com o intuito de identificar o máximo de casos de *churn*, nesse caso o mais adequado seria o RN3, pois é q o que apresenta o melhor *hit-ratio* para os casos de *churn*, já se o objectivo for pôr em prática medidas *anti-churn*, aplicando essas mesmas medidas apenas aos clientes que apresentam uma maior probabilidade de *churn*, poderia ser usado o modelo RN2 sendo aplicadas essas mesmas medidas apenas ao top 20% (os dois primeiros *decils*) da população.

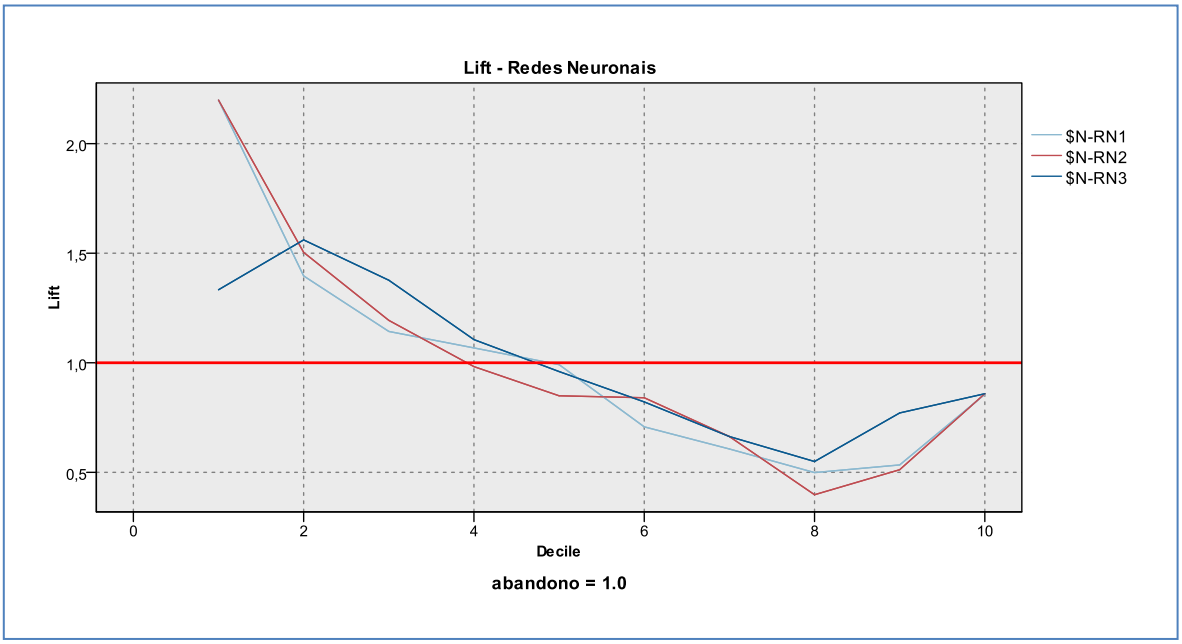

Figura 15 – Lift – Redes Neuronais

5.2.5 SVM

O modelo implementado usando a técnica de SVM (secção 4.2.5) obteve, como se pode observar na Tabela 18, resultados muito modestos, apenas apresentando 56.47% de registos classificados correctamente, mas por outro lado, conseguiu atingir um *hit-ratio* para os casos de *churn* de 45.17%, um valor bastante satisfatório.

Modelo	**Algoritmo**	Previsões Correctas		
		Geral	Abandono = 0	Abandono = 1
SVM	Normal	56.46%	57.52%	45.17%

Tabela 18 – Resultados SVM

Relativamente ao seu *lift,* como se pode observar na Figura 16, o modelo apresenta valores de *lift* satisfatórios para o primeiro *decil*, cruzando a linha *lift* = 1 pouco antes do quarto *decil*, querendo isto dizer, que não é vantajoso usar este modelo para a população que não pertence ao quarto *decil*.

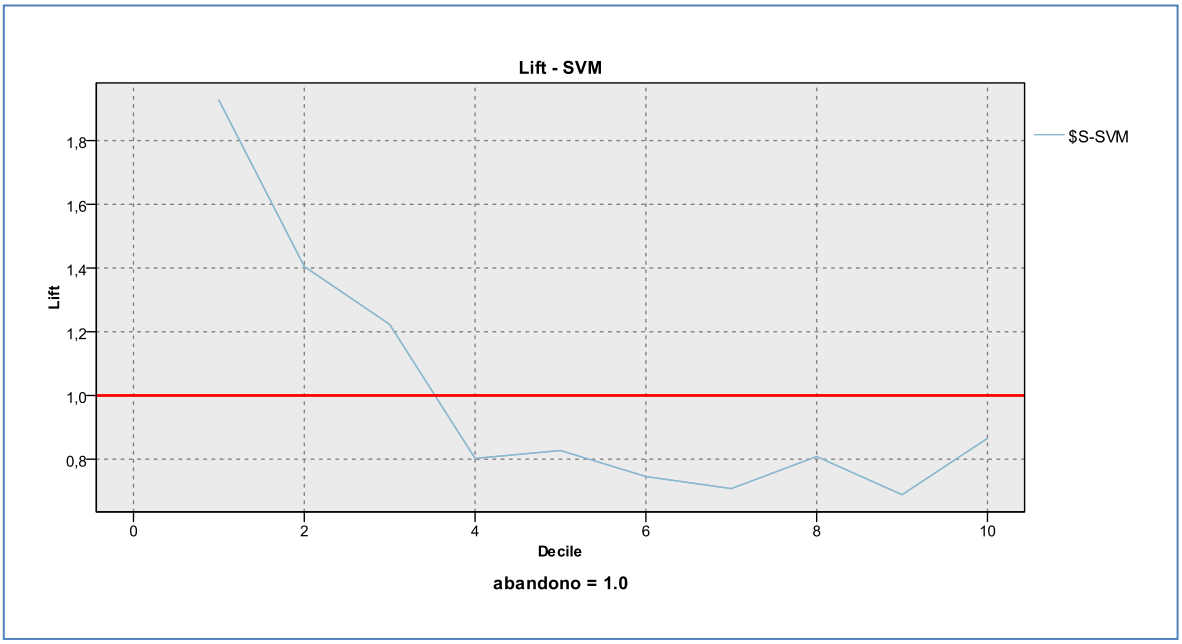

Figura 16 – Lift – SVM

5.3 Comparação de Modelos

5.3.1 Comparação de Resultados Entre Modelos

Na Tabela 19 é apresentada uma representação dos resultados, fazendo-se referência ao método utilizado, ao seu algoritmo auxiliar (*boosting*, *bagging*, algoritmo de selecção de variáveis, etc.), à percentagem de previsões correctas, e ainda à percentagem de previsões correctas descriminada por cada uma das classes da variável dependente (*churn/não-churn*).

Algumas conclusões rápidas que podemos extrair pela análise da Tabela 19 são:

1. O modelo que obteve um melhor desempenho geral foi o modelo A2.1 (árvore de decisão C5.0 com *boosting*) apresentando uma precisão de 84.01%, ou seja, classificou 84.01% dos casos de forma correcta.
2. O modelo que obteve um melhor desempenho na classificação de casos de *churn* foi o modelo SVM que apesar de apresentar apenas uma precisão geral de 56.46%, para os casos classificados como *churn*, apresenta uma precisão de 45.17%, a sua precisão geral

tão baixa deve-se ao facto de apresentar uma precisão muito pobre (57.52%) para a classe de *não-churn* que é a classe predominante.

Método	ID do Modelo	Algoritmo Auxiliar	Previsões Correctas	Previsões Correctas (abandono = 0)	Previsões Correctas (abandono = 1)
Árvore Decisão CHAID	A1	Normal	78.64%	83.42%	27.54%
	A1.1	Bagging	81.36%	86.41%	28.42%
	A1.2	Boosting	72.97%	76.32%	37.21%
Árvore Decisão C5.0	A2	Normal	82.26%	87.1%	30.51%
	A2.1	Boosting	84.01%	88.71%	33.8%
Regressão Logística Binomial	RL1	Forwards	72.42%	77.03%	23.25%
	RL2	Backwards	71.95%	76.44%	23.94%
Rede Neuronal	RN1	Normal	69.42%	73.47%	26.15%
	RN2	Bagging	67.64%	71.14%	30.26%
	RN3	Boosting	60.4%	62.7%	35.82%
SVM	SVM	Normal	56.46%	57.52%	45.17%

Tabela 19 – Resultados dos Modelos de Previsão

No entanto, um bom modelo de previsão não é um modelo que apresenta uma grande percentagem de previsões correctas, ou um grande valor de *hit-rate*. Um modelo para ser considerado eficaz tem que ser capaz de prever o melhor possível ambas as classes de *churn* e *não-churn*. É este *trade-off* que é necessário ter em atenção quando se avalia um modelo preditivo de *churn,* de forma a inferir qual dos modelos oferece o melhor "balanceamento" entre a precisão geral e a precisão dos casos de *churn.* Ao analisar os resultados segundo estas directivas, verificamos que se destacam os modelos A1.2, com uma precisão geral de 72.9% e um *hit-rate* dos casos de *churn* de 37.21%, o modelo A2.1, apresentando 84.01% e 33.8% respectivamente de precisão geral e de *hit-rate*. E ainda os modelos RL1 e Rl2, ambos com resultados satisfatórios e idênticos, apresentando precisões de 72.42% e 71.95%, e *hit-rates* de 23.25% e 23.94%.

De seguida é apresentado e analisado os valores de *lift* para os vários modelos, sendo apresentado os valores do *lift* dividido por dez *decils*, é também apresentado o valor do *lift* acumulado, nas mesmas circunstâncias.

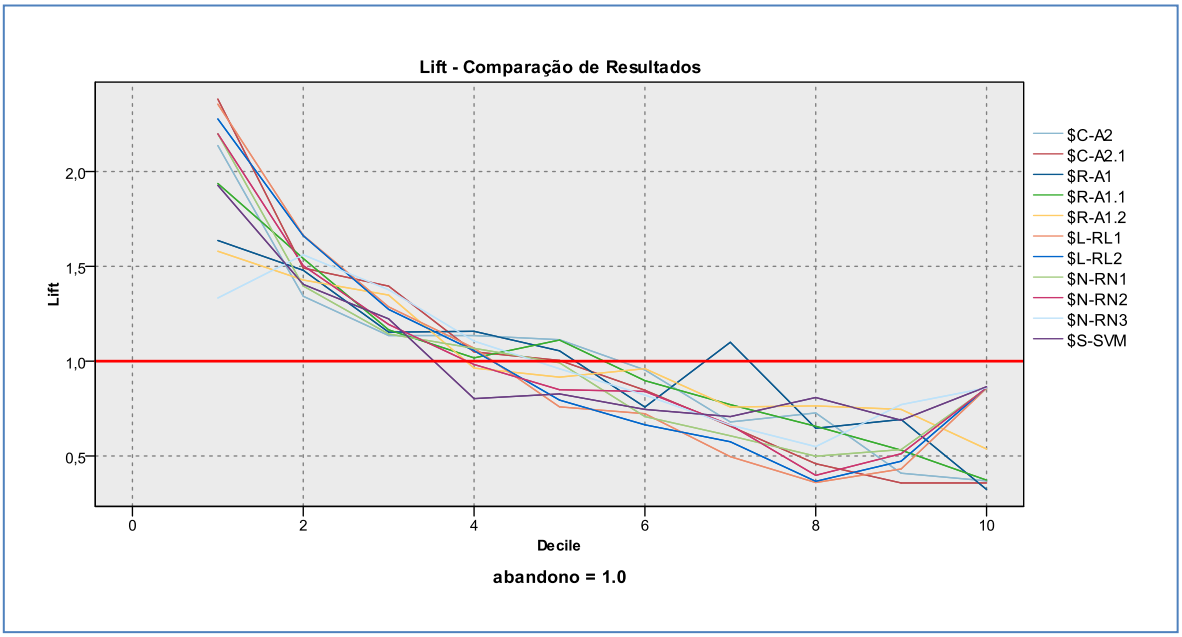

Figura 17 – Lift – Comparação entre os modelos

Pela análise da Figura 17 podemos ter uma visão do comportamento de todos os modelos em relação aos seus valores de *lift* dividido por *decil.* Podemos observar que os modelos A2.1, RL1 e o RL2 são os que apresentam melhores resultados para o primeiro *decil*, mantendo o RL1 e o RL2 bons desempenhos para o 2º *decil*, enquanto o A2.1 cai significativamente no 2º *decil*. Destes três modelos, o A2.1 cruza a linha *lift* = 1 depois do 5º *decil* (*lift* = 1.0041 para o 5º *decil*) destacando-se em relação ao RL1 e RL2 que atingem este valor antes de atingir o 5º *decil*, sendo os seus valores de *lift* para o 5º *decil*, 0.759 e 0,7948 respectivamente. Na Figura 18 é possível visualizar o valor do *lift* acumulado ao longo dos 10 *decils*, para os modelos implementados, podemos concluir que o RL1 é o que apresenta melhor *lift* acumulado até ao 4º *decil*, apenas superado pelo A2.1 no 1º *decil*, mas que por sua vez, assume a liderança a partir do 4º *decil*, ao apresentar melhores resultados de lift acumulado em todos os *decils*.

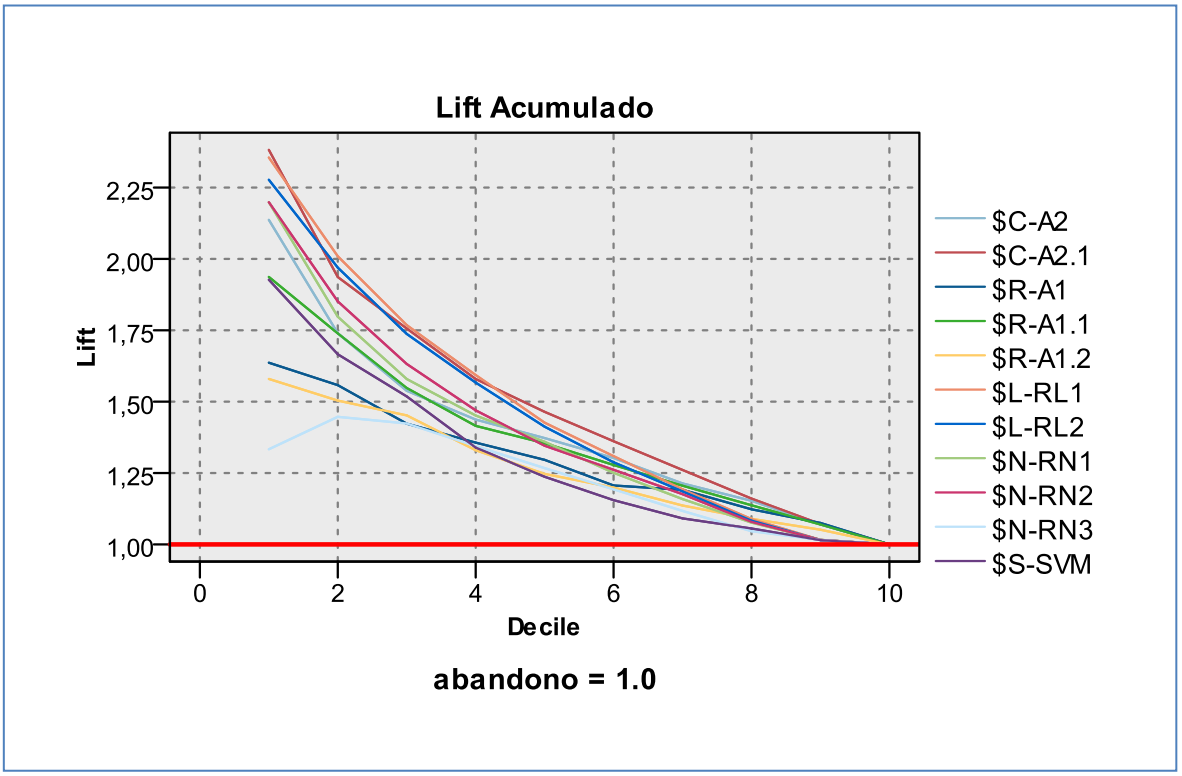

Figura 18 – Lift Acumulado

Anteriormente, pela análise das precisões dos modelos de previsão, concluímos que os modelos mais promissores eram os modelos A1.2, A2.1, RL1 e RL2. À luz dos resultados de *lift* apresentados anteriormente, é nos permitido concluir que de entre os modelos anteriores, são os modelos A2.1 e o RL1 que permitem obter melhores resultados. Para esta conclusão apoiou o facto do modelo A1.2 apresentar valores abaixo da média, não se afirmando como uma escolha viável, já o modelo RL2, apresenta um comportamento de *lift* semelhante ao RL1, mas mantendo-se em valores ligeiramente inferiores, o que originou a sua exclusão em detrimento do RL1. Os dois modelos restantes, além disso o A2.1 e o RL1, apresentam entre todos os modelos os mais elevados valores de *lift* para 1º *decil*, mantendo-se como lideres até ao sexto *decil*, apresentando ao longo de toda a análise valores de *lift* acumulado bastante superiores aos restantes modelos.

Podemos concluir então que as duas melhores técnicas para fazer a previsão de *churn* em companhias de seguros, são as árvores de decisão C5.0 com *boosting* e a regressão logística binomial usando o algoritmo *stepforwards.* A decisão de escolha entre estes dois algoritmos é apresentada na secção seguinte, onde são apresentados dois casos de aplicação e o procedimento que seria correcto adoptar mediante o resultado que se pretende obter.

5.3.2 Casos de Aplicação

De seguida são expostos dois possíveis casos de aplicação, onde será escolhido o algoritmo de entre os dois modelos preditivos identificados anteriormente, que se considera ser o mais adequado para aplicar ao caso em questão, esta escolha será justificada e fundamentada.

> **Caso 1**
> A companhia de seguros pretende realizar a previsão dos clientes *churners* do próximo mês. Com os resultados dessa previsão, pretende-se lançar uma campanha de fidelização que implicará um custo significativo por cada cliente a quem a campanha seja endereçada. Assim sendo decidiu-se que se concentrariam os esforços nos 20% dos clientes com mais probabilidade de abandonar o serviço (cliente *churner).*

Em resposta a esta necessidade, deveria ser utilizado o modelo de regressão logística *forwards* para fazer a previsão dos clientes *churners.* A selecção deste modelo deve-se ao facto de para o 2º *decil* (20% da população com maior probabilidade de ser *churner*), o seu valor de *lift,* ser consideravelmente superior ao do algoritmo de árvores de decisão C5.0 com *boosting*. Esta superioridade no valor de *lift* no 2º *decil* indica que o algoritmo é capaz de nos primeiros 20% dos clientes, ordenados pela probabilidade decrescente de ser *churn,* detectar correctamente mais casos de *churn* que o outro algoritmo. Logo é a escolha certa para o caso em análise. O valor de *lift* igual a aproximadamente 2 para o 2º *decil,* indica-nos que este modelo, para este conjunto de clientes, conseguirá prever os casos *churn* com cerca de 2 vezes mais precisão do que se fosse de forma aleatória. Ou seja, o modelo possuirá uma capacidade de previsão de cerca de 29.2% (14,6 x 2), supondo que o rácio de abandono é igual a 14.6%.

> **Caso 2**
> A companhia de seguros pretende fazer um estudo da probabilidade de abandono dos clientes no próximo período em análise. É pretendido portanto avaliar a probabilidade de abandono para todos os clientes activos em carteira.

Em resposta a esta necessidade deveria ser utilizado o modelo baseado em árvores de decisão C5.0 com *boosting.* O uso desta técnica permitiria classificar correctamente 84.01% da população, com um

hit-ratio de 33.8%, ou seja, com 33.8% dos casos de *churn* correctamente classificados. Como o objectivo é fazer a previsão para todos os clientes, este seria o modelo indicado em detrimento do modelo de regressão logística binomial que produziria valores inferiores para as duas métricas.

5.3.3 Indicadores mais Relevantes

Através da análise dos indicadores utilizadores por cada um dos modelos, bem como pela análise dos resultados obtidos fase da análise univariante, podemos determinar quais dos indicadores de *churn* utilizados para a modelação (Anexo F) são mais significativos para o processo preditivo.

Indicador	Descrição
Antiguidade_cliente	Antiguidade do cliente
Antiguidade_contrato	Antiguidade do contrato
Auto_FRB	Indicador de cobertura Automóvel 'Furto ou Roubo'
Auto_vig	Nº de Apólices 'Automóvel' em Vigor
Idade	Idade do cliente
N_cob_vig	Nº de coberturas (Automóvel) em vigor
N_cont_vig	Nº Contratos/Apólices em Vigor
PPR_apol_vig	Nº de Apólices 'Plano Poupança Reforma' em Vigor

Tabela 20 – Indicadores mais relevantes.

Os indicadores presentes na Tabela 20 são os considerados mais relevantes, esta selecção deveu-se ao facto dos indicadores partilharem duas particularidades, primeiro, todos foram seleccionados internamente para uso por todos os algoritmos de modelação estudados, e segundo, todos possuem uma elevada correlação linear com a variável dependente *churn.*

Relativamente à interpretação dos resultados, alguns dos indicadores são facilmente compreendidos, nomeadamente, os indicadores que representam a antiguidade de um cliente, a antiguidade do contrato e a idade do cliente. Estes indicadores estão fortemente relacionados com a probabilidade de abandono de forma inversamente proporcional. Sendo que quanto mais antigo for o cliente e/ou o contrato, menor será a probabilidade de o cliente abandonar o serviço. Este comportamento é explicável pela menor disposição à mudança dos clientes mais antigos, que ao se acostumarem a um serviço, o aceitam, não procurando alternativas e rejeitando a mudança. Por outro lado, os clientes mais jovens são clientes mais atentos e que procuram melhores oportunidades, estando mais dispostos a trocar de seguradora. Outros indicadores facilmente compreendidos são os indicadores de nº de coberturas automóvel em vigor, nº de apólices em vigor e número de apólices

automóveis em vigor (directamente relacionada com o nº de apólices em vigor). Estes três indicadores estão inversamente relacionados com a probabilidade de abandono de um cliente, visto que quanto maior for o número de coberturas/apólices/apólices automóveis em vigor, menor será a probabilidade de abandono por parte do cliente. Este comportamento também é explicado pelo factor psicológico da comodidade humana: um cliente que possua muitos produtos é mais relutante em abandonar o serviço, devido ao incómodo inerente ao processo. O indicador do número de planos poupança reforma é também um indicador inversamente relacionado com a probabilidade de abandono. No entanto, este poderá ser um falso indicador, do ponto de vista que não é o facto de um cliente ter muitos produtos deste tipo que determina que não irá abandonar o serviço, mas sim, o facto de estar satisfeito e confiante na seguradora, é que determina um elevado número deste tipo de produtos. A única surpresa nos indicadores é o aparente relacionamento do *churn* com a existência da cobertura automóvel contra roubos e furtos. Aparentemente, os clientes com este tipo de cobertura tem menor probabilidade de abandonar o serviço.

Capítulo 6

6 Conclusões e Trabalho Futuro

6.1 Apresentação de Resultados

6.1.1 Motivações e Análise Geral

Sendo o tema da fidelização de clientes, um tema cada vez mais tido em conta pelas companhias, é de elevado interesse o desenvolvimento de técnicas capazes de detectar e antecipar o possível abandono dos clientes em seguradoras. Este facto contribuiu para que a temática da detecção de *churn* tenha vindo, nos últimos tempos, a ganhar alguma visibilidade, tanto no meio académico como no próprio meio empresarial. O *churn* tem vindo a ser estudado principalmente nas áreas das telecomunicações, sendo possível encontrar um número considerável de bibliografia académica sobre o tema. O difundir desta preocupação para outro tipo de indústrias de negócios, tais como fornecedores de conteúdos, companhias de saúde, banca e as seguradoras foi um impulsionador do desenvolvimento de estudos e de trabalhos na área.

Esta dissertação de mestrado pretende assim apresentar algumas das mais robustas técnicas de *Data Mining* existentes, aplicadas à previsão de *churn* no domínio das seguradoras. As técnicas abordadas não foram apresentadas exaustivamente, uma vez que o objectivo deste trabalho não era o de realizar um *survey* sobre as técnicas usadas, mas sim efectuar uma comparação crítica entre as diferentes técnicas para a implementação de um modelo preditivo de *churn* em seguradoras. Apesar

disso, pensa-se que a extensa bibliografia citada no documento fornecerá, caso desejado, informação mais específica sobre todos os temas abordados nesta dissertação.

De seguida é realizado um apanhado sobre as conclusões extraídas dos resultados obtidos e na secção seguinte são apresentadas as considerações finais sobre o trabalho realizado, bem como sugestões de melhorias futuras.

6.1.2 Conclusão dos Resultados Obtidos

Tendo em consideração que os modelos não deverão ser analisados apenas tendo em conta métricas simples como o precisão geral ou o *hit-ratio*. Pois um bom modelo, não é o que apresenta um bom rácio geral de previsões correctas ou um bom *hit-rate* dos casos de *churn,* mas sim um correcto balanceamento entre as duas métricas anteriores. Tendo em conta esta métrica composta, podemos observar que os modelos que se destacam são respectivamente o modelo de árvore de decisão CHAID com *boosting*, o modelo de árvore de decisão C5.0 com *boosting* e ainda ambos os modelos implementados recorrendo à regressão logística binomial, que apresentam ambos resultados satisfatórios e bastante idênticos. Quando cruzada esta análise com a análise do *lift* dos modelos, concluímos que de entre os modelos anteriores, é o modelo de árvore de decisão C5.0 com *boosting* e o modelo da regressão logística binomial *forwards* que permite obter melhores resultados. Estes dois modelos apresentam os maiores valores de *lift* no 1º *decil*, mantendo-se como lideres até ao sexto *decil* e apresentando também ao longo de todos os *decils,* valores de *lift* acumulado bastante superiores aos restantes modelos, afirmando-se por tanto como os melhores modelos.

Em suma, ambos os modelos anteriores são viáveis para a modelação de um modelo preditivo de *churn* em seguradoras. A escolha entre os dois deverá ser feita mediante o objectivo a aplicar aos resultados da previsão efectuado. O modelo de árvore de decisão C5.0 com *boosting* seria indicado numa situação em que a seguradora pretendesse classificar e calcular a propensão ao abandono de todos os seus clientes com vista a aplicar medidas para todos os clientes identificados como potenciais *churners*. A escolha neste caso deveria cair sobre este modelo de árvores de decisão, pois é o que apresenta uma maior precisão geral e também o melhor *hit-rate*. No entanto, se o objectivo da seguradora fosse aplicar medidas anti-churn a um segmento da população dos clientes analisada, o mais correcto seria utilizar o modelo de regressão logística *forwards* para fazer a previsão. Após a previsão, a empresa poderia aplicar medidas de retenção, por exemplo, apenas aos quatro primeiros *decils* (40% da população com maior probabilidade de ser *churner*), garantindo que ia conseguir direccionar correctamente as suas medidas anti-churn a um grande número de clientes correctamente

classificados como *churners,* minimizando o efeito negativo de aplicar medidas preventivas de *churn* a clientes que na realidade não precisam das mesmas.

6.2 Considerações Finais

Durante o desenvolvimento desta dissertação de mestrado, foram várias as dificuldades encontradas. Alguns dos pontos mais desafiantes, foram a sequência de etapas que se iniciou com a necessidade de fazer um estudo relativamente profundo, sobre a temática da área seguradora. Para de seguida, após esse estudo, fazer um levantamento dos dados de negócios disponíveis. Foi entre estas duas etapas que surgiu um dos principais entraves ao trabalho proposto: a percepção da limitação imposta pelas diferenças a nível funcional, dos distintos mundos que representam os diferentes ramos de seguros. Este problema tornou claro que teria que se reduzir o raio de acção do estudo, focando-nos em previsão de *churn* em companhias de seguros, utilizando apenas um segmento no negócio – o automóvel. Para além deste entrave, existiram também outras limitações que se foram tornando evidentes ao longo do desenvolvimento dos trabalhos, entre elas:

- Os modelos implementados foram baseados na estrutura de negócio de uma organização específica. O que significa que os resultados obtidos não são directamente aplicáveis noutras organizações da industria seguradora, pois mesmo que sejam muito idênticas, existem diferenças a nível de negócio que podem comprometer os resultados/conclusões obtidos. Apesar de ser óbvio e de já ter sido referido anteriormente que os diferentes ramos de seguros representam realidades distintas, e consequentemente não se pode actuar da mesma forma. Mesmo que estejamos a falar de organizações idênticas, a operar no ramo dos seguros automóveis, as soluções estudadas neste trabalho podem não ser viáveis, devido às especificidades da organização em questão.
- Existe a necessidade de definir correctamente e de forma rígida, a que cliente classificados como futuros *churners* se deverá aplicar as medidas *anti-churn*. Estas medidas têm por normal um custo associado, logo é do interesse das companhias que o investimento neste tipo de campanha apenas seja direccionado para clientes com um valor de retorno associado que justifique o investimento. Apesar de no decorrer do trabalho ter sido referida a necessidade de se realizar esta selecção, não foram discutidos técnicas para o fazer. Mas a sua necessidade é real e a solução poderia passar por aliar técnicas de CRM, capazes de calcular métricas que indiquem o valor esperado do cliente, a médio e a longo prazo.

- A fiabilidade de qualquer tipo de modelo de previsão é posta em causa pelo simples facto, de existirem situações impossíveis de prever. No caso do *churn* nas seguradoras existe situações de abandono impossíveis de prever, tal como por exemplo mudanças súbitas no modo de vida dos clientes como: mudança de país de residência; desemprego; doença grave súbita; etc.. Este tipo de situações leva em muitas vezes ao abandono por parte do cliente, pois o serviço deixa de ser uma necessidade, ou deixa de ser prioritário. Estes casos são portanto impossíveis de prever e de evitar. Posto isto, qualquer métrica de qualidade de modelos de previsão e qualquer conclusão que se possa inferir, estará condicionada por estas situações.

O trabalho proposto e desenvolvido consistia no estudo de técnicas que permitissem a detecção de *churn* em seguradoras. Apesar das limitações e problemas identificados durante o processo de desenvolvimento dos trabalhos, foi possível concluir com sucesso o mesmo. Este trabalho vem assim contribuir para a promoção de algumas das técnicas disponíveis para a implementação deste tipo de modelos. Apresentando também uma metodologia que se deverá seguir, e expondo alguns dos problemas que se poderá encontrar.

6.3 Trabalho Futuro

O trabalho desenvolvido nesta dissertação teve como objectivo analisar a aplicabilidade de modelos preditivos de *churn* em companhias de seguros, bem como analisar as técnicas de *data mining* mais favoráveis para as implementar. Apesar das considerações finais apresentadas anteriormente, este trabalho não representa uma solução final. Várias considerações e melhorias poderiam ser de futuro realizadas. De seguida são apresentadas algumas possibilidades de melhoria que se pretende trabalhar:

- Estudo da viabilidade do alargamento do âmbito de acção do modelo predito, de modo a ser capaz de operar sobre toda a informação de negócio de uma seguradora, independentemente do ramo de negócio a que o cliente pertença. Ou seja, o desenvolvimento de um modelo capaz de fazer a previsão para o cliente independentemente dos produtos de que ele é tomador. Este tipo de implementação numa seguradora de grande dimensão, e com um grande leque de produtos de oferta, iria requerer um extenso levantamento de requisitos funcionais, de forma a ser possível recolher todos os indicadores de interesse de todas as áreas em questão. Para além deste esforço de levantamento de requisitos teria que ser

analisado e delineado o modo de actuação do modelo. Sendo que a solução final poderia passar pela implementação, não de um modelo preditivo simples, mas sim de uma infra-estrutura preditiva, composta por um conjunto de modelos preditivos. O intuito de tal plataforma seria disponibilizar a capacidade de fazer a previsão de clientes *churners*, independente dos produtos que o cliente possui. Isto permitiria que fossem aplicados modelos distintos, a clientes com perfis distintos, tendo em atenção quais os indicadores de interesse para o cliente em questão.

- Estudar a possibilidade do emprego de algoritmos de *data mining* híbridos que permitam fundir conceitos de diferentes técnicas, oferecendo a possibilidade de usar o melhor de cada um dos algoritmos. Este tipo de trabalho tem vindo a ser estudado, exemplo disso é o trabalho de Bose e Chen, onde os autores propõem uma abordagem preditiva de *churn* em telecomunicações que tira partido do *clustering* e das árvores de decisão. A sua abordagem consiste em submeter os dados dos clientes a um processo de *clustering*, para de seguida utilizar a classificação gerada pelo algoritmo de *clustering* como *input* para o algoritmo de árvores de decisão C5.0 juntamento com os restantes indicadores de negócio [Bem & Chen, 2009].
- Integrar o uso de técnicas de CRM, com o intuito de desenvolver uma metodologia capaz de determinar a que clientes classificados como *churners* aplicar as medidas anti-churn.

Os pontos de melhoria e trabalho futuro destacados anteriormente, são apenas um exemplo do que poderá ser feito, mas que evidencia claramente a possibilidade e espaço para melhorias que este trabalho representa.

Lista de Acrónimos e Siglas

Acrónimo/Sigla	Significado
ABT	Analytical Base Table
BD	Base de Dados
BI	Business Intelligence
CART	Classification and Regression Tree
CLV	Customer Lifetime Value
CRISP-DM	CRoss Industry Standard Process for Data Mining
CRM	Customer Relationship Management
DAGSVM	Directed Acyclic Graph SVM
DM	Data Mining
ECOC	Error-Correcting output codes
EDA	Exploratory Data Analysis
KDD	Knowledge-discovery in databases
MLDP	Minimal Description Length Principle
MLP	Multilayer perceptron
PPR	Produto Poupança Reforma
RBF	Radial Basis Function
RN	Rede Neuronal
SVM	Support Vector Machine

Tabela 21 – Lista de Acrónimos e Siglas

Bibliografia

1. Agrawal, R and Imielinski, T and Swami, A 1993, "Mining Association Rules Between Sets of Items in Large Databases", SIGMOD Conference, vol. 22, no. 2, pp. 207-216.
2. Agresti, A 2007, "Building and applying logistic regression models", An Introduction to Categorical Data Analysis, Hoboken, New Jersey: Wiley, pp. 138.
3. Anderberg, M 1973, "Cluster Analysis for Applications", Academic Press.
4. Archaux, C and Martin, A and Khenchaf, A, 2004, "An SVM based churn detector in prepaid mobile telephony", Information and Communication Technologies: From Theory to Applications, pp. 459-460.
5. Au, W and Chan, C. C. and Yao, X 2003, "A Novel Evolutionary Data Mining Algorithm with Applications to Churn Prediction", IEEE transactions on evolutionary computation, vol. 7, no. 6, pp. 532-545.
6. Auer, P and Burgsteiner, H and Maass, W 2008, "A learning rule for very simple universal approximators consisting of a single layer of perceptrons", Neural Networks, vol. 21, no. 5, pp. 786–795.
7. Baesens, B and Verstraeten, G and Van Den Poel, D and Egmont-Peterson, M and Van Kenhove, P and Vanthienen, J 2004, "Bayesian Network Classifiers for Identifying the Slope of the Customer Lifecycle of Long-Life Customers", European Journal of operational Research, vol. 156, pp. 508-523.
8. Balabin, R and Safieva, R and Lomakina, E 2007, "Comparison of linear and nonlinear calibration models based on near infrared (NIR) spectroscopy data for gasoline properties prediction", Chemometrics and intelligent laboratory systems, vol. 88, no. 2, pp. 183-188.

9. Berson, A, Smith, S & Thearling, K 2000, "Building data mining applications for CRM", New York, NY: McGraw-Hill.

10. Biggs, D 1991, "A method of choosing multiway partitions for classification and decision trees". Journal of Applied Statistics, vol. 18, pp. 49–62.

11. Bigss, D and Ville, B and Suen, E 1991, "A Method of Choosing Multiway Partitions for Classification and Decision Trees", Journal of Applied Statistics, vol. 18, no. 1, pp. 49-62.

12. Bin, Z and Yong, L and Shao-Wei, X 2000, "Support Vector Machine and its Application in Handwritten Numeral Recognition", 15th International Conference on Pattern Recognition (ICPR'00), vol. 2, pp. 2720-2734.

13. Bloemer, J and Brijis, T and Vanhoof, K and Swinnen, G 2002. "Comparing Complete and Partial Classification for Identifying Customers at Risk", International journal of research in marketing, vol. 20, pp. 117-131.

14. Bod, R and Hay, J and Jannedy, S 2003, "Probabilistic Linguistics", Cambridge, Massachusetts: MIT Press.

15. Boone, D and Roehm, M 2002, "Retail Segmentation Using Artificial Neural Networks", International journal of research in marketing, vol. 19, no. 3, pp. 287-301.

16. Bose, I and Chen, X, 2009, "Hybrid models using unsupervised clustering for prediction of customer churn", Proceedings of the International MultiConference of Engineers and Computer Scientists, vol. 1.

17. Bottou, L and Cortes, C and Denker, J and Drucker, H and Guyon, I and Jackel, L. and LeCun, Y and Muller, U and Sackinger, E and Simard, P and Vapnik, V 1994, "Comparison of classifier methods: A case study in handwriting digit recognition" in Proc. Int. Conf. Pattern Recognition, vol. 2, pp. 77–87.

18. Boulesteix, A, 2006, "Maximally Selected Chi-Square Statistics and Binary Splits of Nominal Variables", Biometrical Journal, vol. 48, no. 5, pp. 838-848.

19. Bradley, P and Mangasarian, O 2000, "Massive data discrimination via linear support vector machines", Optimization Methods and Software, vol. 13, pp. 1–10.

20. Breiman, L 1994, "Bagging Predictors", Technical Report 421, Department of Statistics, University of California Berkeley, CA.

21. Breiman, L 1996, "Bagging Predictors", Machine Learning, vol. 22, no. 2, pp. 123-140.

22. Breiman, L and Friedman, J and Olshen, R and Stone, C 1984, "Classification and regression trees".

23. Buhmann, M 2003, "Basis Functions: Theory and Implementations", Cambridge University, vol. 12.

24. Byvatov E and Schneider G 2003, "Support vector machine applications in bioinformatics", Applied Bioinformatics, vol. 2, no. 2, pp. 67-77.

25. Byvatov, E and Fechner, U and Sadowski, J and Schneider, G 2003, "Comparison of support vector machine and artificial neural network systems for drug/nondrug classification", in Journal of chemical information and computer sciences, vol. 6, pp. 1882-1889.

26. Candy, J.C. and Temes, G.C. and Institute of Electrical and Electronics Engineers and IEEE Circuits and Systems Society 1992, "Oversampling delta-sigma data converters: theory, design, and simulation", IEEE-PC0274-1, New York.

27. Candy, JC and Temes, GC 1991, "Oversampling methods for data conversion", Communications, Computers and Signal Processing, 1991., IEEE Pacific Rim Conference on, pp. 498-502.

28. Chapman, P and Clinton, J and Kerber, R and Khabaza, T and Reinartz, T and Shearer, C and Wirth, R 2000, "CRISP-DM 1.0", CRISP-DM Consortium.

29. Chawla, N.V. and Bowyer, K.W. and Hall, L.O. and Kegelmeyer, W.P. 2002 "SMOTE: Synthetic Minority Oversampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321-357.

30. Chen, M and Dey, D 2003, "Variable selection for multivariate logistic regression models", in Journal of Statistical Planning and Inference, vol. 111, pp. 37-55.

31. Chiang, D, Wang, Y, Lee, S, and Lin, C, 2003, "Goal-oriented sequential pattern for network banking churn analysis", Expert Systems with Applications, vol. 25, no. 3, pp. 293-302.

32. Coppock, D 2002, "Data Modelling and Mining: Why Lift?", Information Management Online. Disponível em: http://www.information-management.com/news/5329-1.html (acedido em 01/04/2011).

33. Cortes, C and Vapnik, V 1995, "Support-Vector Networks", Machine Learning, vol. 20, pp. 273-297.

34. Crammer, K and Singer, Y 2001, "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines", in Journal of Machine Learning Research, vol. 2, pp. 265–292.

35. Cutting, D and Karger, D and Pedersen, J and Tukey, J 1992, "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," SIGIR '92, pp. 318– 329.

36. Datta, P and Masand, B and Mani, D. R and Li, B 2001, "Automated Cellular Modeling and Prediction on a Large Scale", Issues on the application of data mining, vol. 14, pp. 485-502.

37. Deng, H and Runger, G and Tuv, E 2011, "Bias of importance measures for multi-valued attributes and solutions", Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN), pp. 293-300.

38. Dietterich, T and Bakiri, G 1995, "Solving Multiclass Learning Problems via Error-Correcting Output Codes", in Journal of Artificial Intelligence Research, vol. 2, pp. 263–286.

39. Draper, N and Smith, H 1981, "Applied Regression Analysis", 2d Edition, New York: John Wiley & Sons, Inc.

40. Duan, K and Keerthi, S 2005, "Which Is the Best Multiclass SVM Method? An Empirical Study", in Proceedings of the Sixth International Workshop on Multiple Classifier Systems, Springer, pp. 278-285.

41. Dubes, R and Jain, A 1988, "Algorithms for Clustering Data", Prentice Hall.

42. Dudoit, S and Van Der Laan, M 2003, "Asymptotics of Cross-Validated Risk Estimation in Estimator Selection and Performance Assessment", Statistical Methodology, vol. 2, pp. 131-154.

43. Fahlman, S and Lebiere, C 1991, "The Cascade-Correlation Learning Architecture".

44. Fayyad, U and K. Irani 1993, "Multi-interval discretization of continuous-value attributes for classification learning", In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, pp. 1022–1027.

45. Fayyad, U and Piatetsky-Shapiro, G and Smyth, P 1996b "From data mining to knowledge discovery in databases", in AI magazine, vol. 17, no. 3, pp. 37.

46. Fayyad, U and Piatetsky-Shapiro, G and Smyth, P and Uthurusamy, R 1996a, "Advances in knowledge discovery and data mining".

47. Fernholz, L and Morgenthaler, S. and Tukey, J and Tukey, E 2000, "A conversation with John W. Tukey and Elizabeth Tukey", Statistical Science, JSTOR, pp. 79-94.

48. Fong, J and Cheung, S 2004, "Translating Relational Schema into Xml Schema Definition with Data Semantic Preservation and Xsd Graph", Information and software technology, vol. 47, pp. 437-462.

49. Frean, M 1990, "The upstart algorithm: A method for constructing and training feedforward neural networks", Neural computation, vol. 2, pp. 198-209.

50. Freund, Y 1995, "Boosting a weak learning algorithm by majority", in Information and computation, vol. 121, no. 2, pp. 256-285.

51. Friedman, J 1996, "Another Approach to Polychotomous Classification", Dept. Statist., Stanford Univ., Stanford, CA. [Online].

52. G.M. Weiss 2003, "The Effect of Small Disjuncts and Class Distribution on Decision Tree Learning", Ph.D. Dissertation, Department of Computer Science, Rutgers University, New Brunswick, New Jersey.

53. Gerstner, W 2001, "Spiking Neurons", In Wolfgang Maass and Christopher M. Bishop. Pulsed Neural Networks, MIT Press.

54. Giuliani, G, Moretti, P and Piancastelli, A 2004, "Limiting churn In Insurance", McKinsey Quarterly, the business journal of McKinsey & Company.

55. Goodman, L 1979, "Simple Models for the Analysis of Association in Cross-Classifications Having Ordered Categories", in Journal of the American Statistical Association, vol. 74, pp. 537-552.

56. Guha, S and Rastogi, R and Shim, K 2000, "ROCK: A Robust Clustering Algorithm for Categorical Attributes", Information Systems, vol. 25, no. 5, pp. 345-366.

57. Guo, X and Yin, Y and Dong, C and Yang, G and Zhou, G 2008, "On the Class Imbalance Problem", Fourth International Conference on Natural Computation, pp. 192-201.

58. Guyon, I and Weston, J and Barnhill, S and Vapnik, V 2002, "Gene selection for cancer classification using support vector machines", Machine Learning, vol. 46, pp. 389–422.

59. Han, H. and Wang, W.Y. and Mao, B.H. 2005, "Borderline- SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning", Advances in Intelligent Computing, pp. 878-887.

60. Han, J 1996, "Data mining techniques", ACM SIGMOD Record, vol. 25, no. 2, pp. 545.

61. Hastie, T and Tibshirani, R and Friedman, J 2001, "The Elements of Statistical Learning", Springer: 2001, pp. 269-272.

62. Haykin, S 1999, "Neural Networks: A Comprehensive Foundation (2nd edition ed.)", Upper Saddle River, NJ: Prentice Hall.

63. Helmbold, D and Schapire, R 1995, "Predicting Nearly as Well as the Best Pruning of a Decision Tree" in Proceedings of the Eighth Annual Conference on Computational Learning Theory, ACM Press, pp. 61 - 68.

64. Hinton, G and Sejnowski, T 1986, "Learning and Relearning in Boltzmann Machines", in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 282–317.

65. Ho Ha, S and Min Bae, S and Chan Park, S 2002, "Customer's Time- Variant Purchase Behavior and Corresponding Marketing Strategies: An Online Retailer's Case", Computers and Industrial Engineering, vol. 43, pp. 801–820.

66. Hochreiter, S and Schmidhuber, J 1997, "Long short-term memory", Neural Computation, vol. 9, no. 8, pp. 1735–1780.

67. Hocking, R 1976, "The Analysis and Selection of Variables in Linear Regression," Biometrics, 32.

68. Hsu, C and Lin, C 2002, "A Comparison of Methods for Multiclass Support Vector Machines", IEEE Transactions on Neural Networks, vol.13, no. 2, pp. 415-425.

69. Hsu, C. and Chang, C and Lin, C and others 2004 "A practical guide to support vector classification", National Taiwan University.

70. Hung, S, Yen, D, and Wang, H 2006, "Applying data mining to telecom churn management", Expert Systems with Applications, vol. 31, no. 3, pp. 515-524.

71. Hyafil, L and Rivest, R 1976, "Constructing Optimal Binary Decision Trees is NP-complete", Information Processing Letters, vol. 5, no. 1, pp. 15–17.

72. Jaeger, H and Haas, H 2004, "Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication", Science 2 April 2004, vol. 304, no. 5667, pp. 78 – 80.

73. Japkowicz, N and others 2000, "Learning from Imbalanced Data Sets: a Comparison of Various Strategies", AAAI Workshop on Learning from Imbalanced Data Sets, Menlo Park, CA, AAAI Press.

74. Japkowicz, N and Stephen, S 2002, "The class imbalance problem: A systematic study", Intelligent Data Analysis, vol.6, no. 5, pp. 429-449.

75. Kantardzic, M 2003, "Data mining: concepts, models, methods, and algorithms", Wiley-Interscience.

76. Karatzoglou, A and Meyer, D and Hornik, K 2006, "Support Vector Machines in R", in Journal of Statistical Software, vol. 15, no. 9, pp. 1-28.

77. Kass, G 1980, "An Exploratory Technique for Investigating Large Quantities of Categorical Data", Applied Statistics, vol. 20, no. 2, pp. 119-127.

78. Kavzoglu, T and Mather, P 2001, "The Role of Feature Selection in Artificial Neural Network Applications", International Journal of Remote Sensing, vol. 23, no. 15, pp. 2919-2937.

79. Karahoca, A, Kara, A, 2006, "Comparing clustering techniques for telecom churn management", The 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, pp. 27-29.

80. Kearns, M 1988, "Thoughts on hypothesis boosting", Unpublished manuscript.

81. Kearns, M and Mansour, Y 1998, "A fast, bottom-up decision tree pruning algorithm with near-optimal generalization" in Proc. 15th Int. Conf. Machine Learning, J. Shavlik, Ed., 1998, pp. 269–277.

82. Klosgen, W and Zytkow, J. 1996, "Knowledge discovery in databases terminology". Advances in Knowledge Discovery and Data Mining, pp. 573-592.

83. Knerr, S and Personnaz, L and Dreyfus, G 1990, "Single-layer learning revisited: A stepwise procedure for building and training a neural network," in Neurocomputing: Algorithms, Architectures and Applications, J. Fogelman, Ed. New York: Springer-Verlag.

84. Kohonen, T 1995, "Learning vector quantization", M.A. Arbib, editor, The Handbook of Brain Theory and Neural Networks, pp. 537–540, MIT Press, Cambridge, MA.

85. Kohonen, T and Honkela, T 2007, "Kohonen network", Scholarpedia.

86. Kotsiantis, S 2007, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31, pp. 249-268.

87. Kotsiantis, S and Kanellopoulos, D and Pintelas, P 2006, "Handling imbalanced datasets: A review", GESTS International Transactions on Computer Science and Engineering 30 (1), pp. 25-36.

88. Kotsiantis, S and Pintelas, P 2003, "Mixture of Expert Agents for Handling Imbalanced Data Sets", Annals of Mathematics, Computing & TeleInformatics, vol. 1, pp. 46-55.

89. Krause, N and Singer, Y 2004, "Leveraging the margin more carefully", In Proceedings of the International Conference on Machine Learning (ICML).

90. Kreßel, U 1999, "Pairwise classification and support vector machines" in Advances in Kernel Methods, MIT Press, pp. 255–268.

91. Kruskal, W and Wallis, W 1952, "Use of ranks in one-criterion variance analysis", in Journal of the American statistical Association, pp. 583-621.

92. Lariviere, B and Van den Poel, D 2004, "Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services". Expert Systems with Applications, 27(2), pp. 277–285.

93. Lauritzen, S 1979, "Lectures on contingency tables".

94. Lee, H 1999, "Semantics of Recursive Relationships in Entity-Relationship Model", Information and software technology, vol. 41, pp. 877- 886.

95. Lewis, R 2000, "An introduction to classification and regression tree (CART) analysis", Annual Meeting of the Society for Academy Emergency Medicine, San Francisco, California. USA.

96. Liu, H and F. Hussain and C. L. Tan and M. Dash 2002, "Discretization: An Enabling Technique", Data Mining and Knowledge Discovery, vol. 6, pp. 393–423.

97. Liu, Y 2006, "Using SVM and error-correcting codes for multiclass dialog act classification in meeting corpus" in Proc. Interspeech-ICSLP, pp. 1938–1941.

98. MacQueen, J. 1967, "Some methods for classification and analysis of multivariate observations", Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297.

99. Mansour, Y 1997, "Pessimistic decision tree pruning based on tree size" in Proc. 14th International Conference on Machine Learning, pp. 195–201.

100. Meyer, D and Leisch, F and Hornik. K 2003, "The support vector machine under test". Neurocomputing, vol. 55, no. (1–2), pp. 169–186.

101. Meyer-Base, A and Watzel, R 1998, "Transformational Radial Basis Neural Network for Relevant Feature Selection", Pattern Recognition, vol. 19, pp. 1301-1306.

102. Mozer, M, Wolniewicz, R, Grimes, D, Johnson, E and Kaushansky, H, 2000, "Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry", Neural Networks, IEEE Transactions on, vol. 11, no. 3, pp. 690-696.

103. Morik, K and Kopcke, H 2004, "Analysing customer churn in insurance data--a case study", Knowledge Discovery in Databases: PKDD 2004, Springer, pp. 325-33.

104. Murthy S 1998, "Automatic construction of decision trees from data: A multidisciplinary survey", Data Mining and Knowledge Discovery.

105. Nasution, B and Khan, A 2008, "A Hierarchical Graph Neuron Scheme for Real-Time Pattern Recognition", IEEE Transactions on Neural Networks, vol. 19, no. 2, pp. 212-229.

106. Neslin, S.A Gupta, S, Kamakura, W, Lu, J and Mason, C.H, 2006, "Defection detection: Measuring and understanding the predictive accuracy of customer churn models", Journal of Marketing Research, vol. 43, no. 2, pp. 204-211.

107. Papagelis A and Kalles D 2001, "Breeding Decision Trees Using Evolutionary Techniques", Proceedings of the Eighteenth International Conference on Machine Learning, pp. 393-400.

108. Pendharkar, P, 2009, "Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services", Expert Systems with Applications, vol. 36, no. 3, pp. 6714-6720.

109. Platt, J and Cristanini, N and Shawe-Taylor, J 2000, "Large margin DAGs for multiclass classification", in Advances in Neural Information Processing Systems (MIT Press), vol. 12, no. 3, pp. 547-553.

110. Popović, D, Bašić, B, 2009, "Churn Prediction Model in Retail Banking Using Fuzzy C-Means Algorithm"

111. Prokhorov, A 2002, "Kendall coefficient of rank correlation", Encyclopaedia of Mathematics.

112. Qian, Z, Jiang, W, and Tsui, K 2006, "Churn detection via customer profile modeling", International journal of production research, vol. 44, no. 14, pp. 2913-2933, Taylor and Francis Ltd.

113. Quinlan, J 1986, "Introduction of Decision Trees", Machine learning, vol. 1, no. 1, pp. 81-106.

114. Quinlan, J 1987, "Generating production rules from decision trees", Proceedings of the Tenth International Joint Conference on Artificial Intelligence, vol. 30107, pp. 304-307.

115. Quinlan, J 1987, "Simplifying Decision Trees", International journal of man-machine studies, vol. 27, no. 3, pp. 221--234.

116. Quinlan, J 1993, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers.

117. Quinlan, J 1996, "Improved use of continuous attributes in c4.5", Journal of Artificial Intelligence Research, vol. 4, pp. 77-90.

118. Rexer, K and Allen, H and Gearan, P 2010, "2010 Data Miner Survey Summary", presented at Predictive Analytics World, Oct. 2010.

119. Rojas, R 1996, "Neural networks: a systematic introduction", Springer.

120. Rosner, B, 2010, "Fundamentals of biostatistics", Duxbury Pr.

121. Rosset, S and Neumann, E and Uri, E and Vatnik, U and Idan, Y 2002, "Customer Lifetime Value Modelind and Its Use for Customer Retention Planning", Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 332-340.

122. SAS Institute Inc. 1989, "SAS/STAT User's Guide", Version 6, Fourth Edition, Volume 2, Cary, NC: SAS Institute Inc.

123. Schmidhuber, J 1992, "A fixed size storage O(n3) time complexity learning algorithm for fully recurrent continually running networks", Neural Computation, vol. 4, no. 2, pp. 243–248.

124. Schmidhuber, J 1992, "Learning complex, extended sequences using the principle of history compression", Neural Computation, vol. 4, no. 2, pp. 234-242.

125. Scholkopf, B and Smola, A and Williamson, R and Bartlett, P 2000, "New Support Vector Algorithms", Neural Computation, vol. 12, pp. 1207–1245.

126. Schuster, M and Paliwal, K 1997, "Bidirectional recurrent neural networks", IEEE Transactions on Signal Processing, vol. 45, pp. 2673–2681.

127. Shearer C 2000, "The CRISP-DM model: the new blueprint for data mining", in Journal of Data Warehousing, vol. 5, no. 4, pp. 13-22.

128. Shin, H and Sohn, S 2004, "Segmentation of Stock Trading Customers According to Potential Value", Expert systems with applications, vol. 27, pp. 27-33.

129. Shaw, R and Stone, M 1988. "Database Marketing", Gower, London.

130. Skurichina, M and Duin, R 2000, "The role of combining rules in bagging and boosting", Advances in Pattern Recognition, Springer, pp. 631-640.

131. Snider, G 2008, "Cortical computing with memristive nanodevices", Sci-DAC Review 10: 58–65.

132. Song, G, Yang, D, Wu, L, Wang, T and Tang, S, 2006, "A mixed process neural network and its application to churn prediction in mobile communications", Data Mining Workshops, ICDM Workshops 2006, Sixth IEEE International Conference, pp. 798-802.

133. Souza, J and Matwin, S and Japkowicz, N 2002, "Evaluating data mining models: A pattern language", Proceedings of the 9th Conference on Pattern Language of Programs, USA.

134. Spiegel, M and Liu, J 1999, "Mathematical handbook of formulas and tables", Schaum's Outline Series.

135. SPSS Inc. 2010a, an IBM Company, "IBM SPSS Modeler 14.1 Algorithms Guide", SPSS Inc.

136. SPSS Inc. 2010b, an IBM Company, "IBM SPSS Modeler 14.1 Applications Guide", SPSS Inc.

137. SPSS Inc. 2010c, an IBM Company, "IBM SPSS Modeler 14.1 User's Guide", SPSS Inc.

138. Stanley, K 2007, "Compositional Pattern Producing Networks: A Novel Abstraction of Development", Genetic Programming and Evolvable Machines Special Issue on Developmental Systems 8.

139. Steinbach, M and Karypis, G and Kumar, V 2000, "A comparison of document clustering techniques", KDD workshop on text mining, vol. 400, pp. 525-526.

140. Stoop, R and Buchli, J and Keller, G and Steeb, WH 2003, "Stochastic resonance in pattern recognition by a holographic neuron model", Physical Review E.

141. Thearling, K 1999, "An Introduction to Data Mining", Direct Marketing Magazine, pp. 28-31.

142. Van Den Poel, D and Lariviere, B 2003, "Customer Attrition Analysis for Financial Services Using Proportional Hazard Models, European journal of operational research, vol. 157, pp. 196-217.

143. Vellido, A and Lisboa, P and Meehan, K 1999, "Segmentation of the Online Shopping Market Using Neural Networks", Expert systems with applications, vol. 17, pp. 303-314.

144. Wei-yun, Y, Zheng, QIN, Yu, Z, Bing, LI and Xiu, LI, 2007, "Support Vector Machine and Its Application in Customer Churn Prediction", Systems Engineering-Theory & Practice, vol. 7.

145. Weisberg, S 2005, "Binomial Regression", Applied Linear Regression, Wiley-IEEE. pp. 253–254.

146. Wu, X and Kumar, V and Ross Quinlan, J and Ghosh, J and Yang, Q and Motoda, H and McLachlan, G.J. and Ng, A and Liu, B and Yu, P.S. and others 2008, "Top 10 algorithms in data mining", Knowledge and Information Systems, vol.14, no. 1, pp. 1-37.

147. Xia, G and Jin, W, 2008, "Model of Customer Churn Prediction on Support Vector Machine", Systems Engineering-Theory & Practice, vol. 28, no. 1, pp. 71-77.

148. Xu R and Wunsch D 2005, "Survey of clustering algorithms", IEEE Transactions on Neural Networks, 16(3), pp. 645–678.

149. Yee, P and Haykin, S 2001, "Regularized Radial Basis Function Networks: Theory and Applications", John Wiley.

Referências WWW

[1] Http://www.ibm.com/pt/pt/

Site oficial da companhia IBM, detentora do software usado no desenvolvimento desta dissertação.

[2] Http://www-01.ibm.com/software/analytics/spss/products/modeler/

Site de apresentação da feramenta SPSS Modeler, usado no desenvolvimento desta dissertação. Aqui encontra-se inúmera informação e documentação acerca da ferramenta.

[3] Http://www.java.sun.com/

Este *site* fornece uma elevada quantidade de informação acerca da linguagem orientada aos objectos Java. É possível encontrar uma vasta gama de recursos que vai desde as versões mais actuais do Java, passando por uma extensa documentação, até grupos de discussão e investigação.

[4] Http://www-db.stanford.edu/warehousing/warehouse.html

O projecto *WareHouse Information Prototype at Stanford* (WHIPS) teve como objectivo primordial analisar a criação e a manutenção de SDWs e desenvolver algoritmos e ferramentas que assegurem estas actividades. Aqui encontra-se toda a documentação disponível sobre este projecto.

[5] Http://www.kdnuggets.com/

Site da comunidade de Data Mining kdnuggets. Esta comunidade apresenta-se como a comunidade líder e de referência desde 1997, oferecendo noticias sobre *Data Mining*, *software*, empregos, cursos, e muito mais.

Anexo A Lista dos Indicadores Iniciais

#	Nome do Indicador	Descrição
1	Antiguidade_cliente	Antiguidade do cliente
2	Antiguidade_contrato	Antiguidade do contrato
3	AP_apol_anul	Nº de Apólices 'Acidentes Pessoais' Anuladas
4	AP_apol_vig	Nº de Apólices 'Acidentes Pessoais' em Vigor
5	AT_apol_anul	Nº de Apólices 'Acidentes de Trabalho' Anuladas
6	AT_apol_vig	Nº de Apólices 'Acidentes de Trabalho' em Vigor
7	Auto_ADV	Indicador de cobertura Automóvel 'Actos de Vandalismo'
8	Auto_anul	Nº de Apólices 'Automóvel' Anuladas
9	Auto_BAG	Indicador de cobertura Automóvel 'Bagagens'
10	Auto_CCC	Indicador de cobertura Automóvel 'Choque Colisão capotamento'
11	Auto_CRT	Indicador de cobertura Automóvel 'Colisão com responsabilidade de terceiros'
12	Auto_FDN	Indicador de cobertura Automóvel 'Forças da Natureza'
13	Auto_FRB	Indicador de cobertura Automóvel 'Furto ou Roubo'
14	Auto_IRE	Indicador de cobertura Automóvel 'Incêndio Raio ou Explosão'
15	Auto_PDU	Indicador de cobertura Automóvel 'Privação de uso'
16	Auto_PTJ	Indicador de cobertura Automóvel 'Protecção Jurídica'
17	Auto_QIV	Indicador de cobertura Automóvel 'Quebra isolado de vidros'
18	Auto_RCV	Indicador de cobertura Automóvel 'Responsabilidade Civil'
19	Auto_RSP	Indicador de cobertura Automóvel 'Riscos Sociais e Políticos'
20	Auto_VDS	Indicador de cobertura Automóvel 'Veiculo de Substituição'
21	Auto_vig	Nº de Apólices 'Automóvel' em Vigor
22	Capital_Auto_ADV_anul	Capital Associado às coberturas Automóvel 'Actos de Vandalismo' em contratos anulados
23	Capital_Auto_ADV_vig	Capital Associado às coberturas Automóvel 'Actos de Vandalismo' em contratos em vigor
24	Capital_Auto_BAG_anul	Capital Associado às coberturas Automóvel 'Bagagens' em contratos anulados
25	Capital_Auto_BAG_vig	Capital Associado às coberturas Automóvel 'Bagagens' em contratos em vigor
26	Capital_Auto_CCC_anul	Capital Associado às coberturas Automóvel 'Choque Colisão Capotamento' em contratos anulados
27	Capital_Auto_CCC_vig	Capital Associado às coberturas Automóvel 'Choque Colisão Capotamento' em contratos em vigor
28	Capital_Auto_FDN_anul	Capital Associado às coberturas Automóvel 'Forças da Natureza' em contratos anulados
29	Capital_Auto_FDN_vig	Capital Associado às coberturas Automóvel 'Forças da Natureza' em contratos em vigor
30	Capital_Auto_FRB_anul	Capital Associado às coberturas Automóvel 'Furto ou Roubo' em contratos anulados
31	Capital_Auto_FRB_vig	Capital Associado às coberturas Automóvel 'Furto ou Roubo' em contratos em vigor

#	Nome do Indicador	Descrição
32	Capital_Auto_IRE_anul	Capital Associado às coberturas Automóvel 'Incêndio Raio ou Explosão' em contratos anulados
33	Capital_Auto_IRE_vig	Capital Associado às coberturas Automóvel 'Incêndio Raio ou Explosão' em contratos em vigor
34	Capital_Auto_ODV_anul	Capital Associado às coberturas Automóvel '' em contratos anulados
35	Capital_Auto_ODV_vig	Capital Associado às coberturas Automóvel '' em contratos em vigor
36	Capital_Auto_QIV_anul	Capital Associado às coberturas Automóvel 'Quebra isolado de vidros' em contratos anulados
37	Capital_Auto_QIV_vig	Capital Associado às coberturas Automóvel 'Quebra isolado de vidros' em contratos em vigor
38	Capital_Auto_RCV_anul	Capital Associado às coberturas Automóvel 'Responsabilidade Civil' em contratos anulados
39	Capital_Auto_RCV_vig	Capital Associado às coberturas Automóvel 'Responsabilidade Civil' em contratos em vigor
40	Capital_Auto_RSP_anul	Capital Associado às coberturas Automóvel 'Riscos Sociais e Políticos' em contratos anulados
41	Capital_Auto_RSP_vig	Capital Associado às coberturas Automóvel 'Riscos Sociais e Políticos' em contratos em vigor
42	Class_Agente	Tipo de Classe do Agente
43	Cli_Nr	Número do cliente
44	Contencioso	Indicador de Situação Contenciosa
45	CP_apol_anul	Nº de Apólices 'Operações de Capitalização' Anuladas
46	CP_apol_vig	Nº de Apólices 'Operações de Capitalização' em Vigor
47	Custo_sin_anul	Custo Total dos Sinistros em Contratos Anulados
48	Custo_sin_Auto_anul	Custo dos sinistros Automoveis em Contratos Anulados
49	Custo_sin_Auto_vig	Custo dos sinistros Automoveis em Contratos em Vigor
50	Custo_sin_vig	Custo Total dos Sinistros em Contratos em Vigor
51	Data_Nascimento	Data de Nascimento do Cliente
52	Data_Participa__o	Data de participação do último Sinistro
53	Data_Sinistro	Data desde o último Sinistro
54	DI_apol_anul	Nº de Apólices 'Diversos' Anuladas
55	DI_apol_vig	Nº de Apólices 'Diversos' em Vigor
56	Distrito	Distrito do Cliente
57	Exclusividade_Agente	Tipo de exclusividade do Agente
58	Forma_cobr	Forma de Cobrança
59	Fraccionamento_Anulado	Tipo Fraccionamento Anulado
60	Fraccionamento_Vigor	Tipo de Fraccionamento em Vigor
61	IN_apol_anul	Nº de Apólices 'Inc Elementos natureza' Anuladas
62	IN_apol_vig	Nº de Apólices 'Inc Elementos natureza' em Vigor
63	Md_Custo_sin_Auto_anul	Custo Médio por Sinistro em Contratos Automóvel Anulado
64	Md_Custo_sin_Auto_vig	Custo Médio por Sinistro em Contratos Automóvel em Vigor
65	MRE_apol_anul	Nº de Apólices 'Multirrisco Empresarial' Anuladas
66	MRE_apol_vig	Nº de Apólices 'Multirrisco Empresarial' em Vigor
67	MRH_apol_anul	Nº de Apólices 'Multirrisco Habitação' Anuladas
68	MRH_apol_vig	Nº de Apólices 'Multirrisco Habitação' em Vigor
69	N_cob_anul	Nº de coberturas (Automóvel) anuladas

#	Nome do Indicador	Descrição
70	N_cob_vig	Nº de coberturas (Automóvel) em vigor
71	N_cont_anul	Nº Contratos/Apólices Anulados
72	N_cont_vig	Nº Contratos/Apólices em Vigor
73	N_sin_anul	Nº de Sinistros em Apólices Anuladas
74	N_sin_Auto_anul	Nº de Sinistros em Apólices Automóvel Anulados
75	N_sin_Auto_vig	Nº de Sinistros em Apólices Automóvel em Vigor
76	N_sin_vig	Nº de Sinistros em Apólices em Vigor
77	OutrosNV_apol_anul	Nº de Apólices 'Outros Não-Vida' Anuladas
78	OutrosNV_apol_vig	Nº de Apólices 'Outros Não-Vida' em Vigor
79	OutrosV_apol_anul	Nº de Apólices Outros Vida Anuladas
80	OutrosV_apol_vig	Nº de Apólices Outros Vida em Vigor
81	PPR_apol_anul	Nº de Apólices 'Plano Poupança Reforma' Anuladas
82	PPR_apol_vig	Nº de Apólices 'Plano Poupança Reforma' em Vigor
83	Pr_mio_Auto_anul	Prémio das Apólices Automóvel Anuladas
84	Pr_mio_Auto_vig	Prémio das Apólices Automóvel em Vigor
85	Premio_anul	Prémio Total das Apólices Anuladas
86	Premio_vig	Prémio Total das Apólices em Vigor
87	Prof_cod	Código da Profissão
88	RC_apol_anul	Nº de Apólices 'Responsabilidade Civil' Anuladas
89	RC_apol_vig	Nº de Apólices 'Responsabilidade Civil' em Vigor
90	RS_apol_anul	Nº de Apólices 'Risco Saúde' Anuladas
91	RS_apol_vig	Nº de Apólices 'Risco Saúde' em Vigor
92	Sexo	Sexo do Cliente
93	Tempo_Gest_o_Auto_anul	Tempo de Gestão das Apólices Automóvel Anuladas
94	Tempo_Gest_o_Auto_vig	Tempo de Gestão das Apólices Automóvel em Vigor
95	Var_Anonima_01	Variável Anonimizada
96	Var_Anonima_02	Variável Anonimizada
97	Var_Anonima_03	Variável Anonimizada
98	Var_Anonima_04	Variável Anonimizada
99	Var_Anonima_05	Variável Anonimizada
100	Var_Anonima_06	Variável Anonimizada
101	Var_Anonima_07	Variável Anonimizada
102	Var_Anonima_08	Variável Anonimizada
103	Var_Anonima_09	Variável Anonimizada

Tabela 22 – Lista de Indicadores Iniciais

Nota: Por imposição da entidade fornecedora dos dados utilizados nesta dissertação de mestrado, foi necessário anonimizar alguns dos indicadores utilizados.

Anexo B Análise Preliminar dos Dados

#	Variável	Acção	Razão Para a Eliminação
1	Cli_Nr	Eliminada	Considerada não relevante para o problema
2	Data_Nascimento	Eliminada	Substituída pela variável Idade
3	Distrito	Eliminada	Substituída pela variável Distrito_new
4	Prof_cod	Eliminada	Considerada não relevante para o problema
5	Capital_Auto_RCV_vig	Eliminada	Substituída pela variável Capital_vig
6	Capital_Auto_QIV_vig	Eliminada	Substituída pela variável Capital_vig
7	Capital_Auto_FRB_vig	Eliminada	Substituída pela variável Capital_vig
8	Capital_Auto_CCC_vig	Eliminada	Substituída pela variável Capital_vig
9	Capital_Auto_FDN_vig	Eliminada	Substituída pela variável Capital_vig
10	Capital_Auto_IRE_vig	Eliminada	Substituída pela variável Capital_vig
11	Capital_Auto_BAG_vig	Eliminada	Substituída pela variável Capital_vig
12	Capital_Auto_RSP_vig	Eliminada	Substituída pela variável Capital_vig
13	Capital_Auto_ADV_vig	Eliminada	Substituída pela variável Capital_vig
14	Capital_Auto_ODV_vig	Eliminada	Substituída pela variável Capital_vig
15	Capital_Auto_RCV_anul	Eliminada	Substituída pela variável Capital_anul
16	Capital_Auto_ADV_anul	Eliminada	Substituída pela variável Capital_anul
17	Capital_Auto_QIV_anul	Eliminada	Substituída pela variável Capital_anul
18	Capital_Auto_FRB_anul	Eliminada	Substituída pela variável Capital_anul
19	Capital_Auto_CCC_anul	Eliminada	Substituída pela variável Capital_anul
20	Capital_Auto_FDN_anul	Eliminada	Substituída pela variável Capital_anul
21	Capital_Auto_IRE_anul	Eliminada	Substituída pela variável Capital_anul
22	Capital_Auto_BAG_anul	Eliminada	Substituída pela variável Capital_anul
23	Capital_Auto_RSP_anul	Eliminada	Substituída pela variável Capital_anul
24	Capital_Auto_ODV_anul	Eliminada	Substituída pela variável Capital_anul
25	Data_Sinistro	Eliminada	Alto rácio de missings (14.15% not missing)
26	Data_Participa__o	Eliminada	Alto rácio de missings (14.15% not missing)
27	Tempo_Gest_o_Auto_vig	Eliminada	Alto rácio de missings (13.28% not missing)
28	Tempo_Gest_o_Auto_anul	Eliminada	Alto rácio de missings (1.07% not missing)
29	Var_Anonima_01	Eliminada	Considerada não relevante para o problema
30	Var_Anonima_02	Eliminada	Alto rácio de missings (5.5% not missing)
31	Capital_vig	Criada	Criada a partir de várias variáveis
32	Capital_anul	Criada	Criada a partir de várias variáveis
33	Idade	Criada	Criada a partir da Data_Nascimento
34	Distrito_new	Criada	Criada a partir da Distrito
35	sinistro	Criada	Criada a partir da N_sin_vig

Tabela 23 – Análise Preliminar dos Dados

Anexo C Análise Univariante – Capacidade Discriminante

#	Tipo Variável	Variável	Eliminada por teste de Krushal Wallis (p-value>0.05)	Eliminada por Teste de Contingência (Chi-square) (p-value>0.05)
1	Nominal	Sexo	N/A	Não
2	Nominal	Auto_vig	N/A	Não
3	Nominal	Auto_anul	N/A	Não
4	Nominal	AT_apol_vig	N/A	Não
5	Nominal	AT_apol_anul	N/A	**SIM**
6	Nominal	AP_apol_vig	N/A	**SIM**
7	Nominal	AP_apol_anul	N/A	**SIM**
8	Nominal	DI_apol_vig	N/A	**SIM**
9	Nominal	DI_apol_anul	N/A	**SIM**
10	Nominal	MRH_apol_vig	N/A	Não
11	Nominal	MRH_apol_anul	N/A	**SIM**
12	Nominal	MRE_apol_vig	N/A	**SIM**
13	Nominal	MRE_apol_anul	N/A	**SIM**
14	Nominal	IN_apol_vig	N/A	Nao
15	Nominal	IN_apol_anul	N/A	**SIM**
16	Nominal	RC_apol_vig	N/A	Nao
17	Nominal	RC_apol_anul	N/A	**SIM**
18	Nominal	OutrosNV_apol_vig	N/A	Não
19	Nominal	OutrosNV_apol_anul	N/A	**SIM**
20	Nominal	PPR_apol_vig	N/A	Não
21	Nominal	PPR_apol_anul	N/A	**SIM**
22	Nominal	CP_apol_vig	N/A	**SIM**
23	Nominal	CP_apol_anul	N/A	**SIM**
24	Nominal	Var_Anonima_08	N/A	**SIM**
25	Nominal	Var_Anonima_05	N/A	**SIM**
26	Nominal	RS_apol_vig	N/A	**SIM**
27	Nominal	RS_apol_anul	N/A	**SIM**
28	Nominal	OutrosV_apol_vig	N/A	**SIM**
29	Nominal	OutrosV_apol_anul	N/A	**SIM**
30	Nominal	N_cont_vig	N/A	Não
31	Nominal	N_cont_anul	N/A	Não

#	Tipo Variável	Variável	Eliminada por teste de Krushal Wallis (p-value>0.05)	Eliminada por Teste de Contingência (Chi-square) (p-value>0.05)
32	Nominal	Auto_RCV	N/A	**SIM**
33	Nominal	Var_Anonima_03	N/A	**SIM**
34	Nominal	Var_Anonima_04	N/A	**SIM**
35	Nominal	Auto_PTJ	N/A	Não
36	Nominal	Auto_QIV	N/A	**SIM**
37	Nominal	Auto_FRB	N/A	Não
38	Nominal	Auto_CCC	N/A	Não
39	Nominal	Auto_FDN	N/A	Não
40	Nominal	Auto_IRE	N/A	Não
41	Nominal	Auto_VDS	N/A	**SIM**
42	Nominal	Auto_BAG	N/A	Não
43	Nominal	Auto_RSP	N/A	Não
44	Nominal	Auto_ADV	N/A	**SIM**
45	Nominal	Auto_PDU	N/A	Não
46	Nominal	Auto_CRT	N/A	Não
47	Nominal	N_cob_vig	N/A	Não
48	Nominal	N_cob_anul	N/A	Não
49	Nominal	Fraccionamento_Vigor	N/A	Não
50	Nominal	Fraccionamento_Anulado	N/A	Não
51	Nominal	Class_Agente	N/A	Não
52	Nominal	Exclusividade_Agente	N/A	Não
53	Nominal	Var_Anonima_07	N/A	Não
54	Nominal	Var_Anonima_06	N/A	Não
55	Nominal	Forma_cobr	N/A	Não
56	Nominal	N_sin_Auto_vig	N/A	Não
57	Nominal	N_sin_Auto_anul	N/A	Não
58	Nominal	N_sin_vig	N/A	Não
59	Nominal	N_sin_anul	N/A	Não
60	Nominal	Contencioso	N/A	Não
61	Nominal	Distrito_new	N/A	Não
62	Nominal	sinistro	N/A	Não
63	Contínua	Idade	Não	N/A

#	Tipo Variável	Variável	Eliminada por teste de Krushal Wallis (p-value>0.05)	Eliminada por Teste de Contingência (Chi-square) (p-value>0.05)
64	Contínua	Antiguidade_cliente	Não	N/A
65	Contínua	Antiguidade_contrato	Não	N/A
66	Contínua	Pr_mio_Auto_vig	Não	N/A
67	Contínua	Pr_mio_Auto_anul	Não	N/A
68	Contínua	Premio_vig	Não	N/A
69	Contínua	Premio_anul	Não	N/A
70	Contínua	Var_Anonima_09	Não	N/A
71	Contínua	Custo_sin_Auto_vig	Não	N/A
72	Contínua	Custo_sin_Auto_anul	Não	N/A
73	Contínua	capital_vig	Não	N/A
74	Contínua	capital_anul	Não	N/A
75	Contínua	Custo_sin_vig	Não	N/A
76	Contínua	Custo_sin_anul	Não	N/A
77	Contínua	Md_Custo_sin_Auto_vig	Não	N/A
78	Contínua	Md_Custo_sin_Auto_anul	Não	N/A

Tabela 24 – Análise da Capacidade Discriminante

Anexo D Análise Univariante – Correlação Linear

#	Variável	Tipo Variável	Teste Kendall tau/Coeficiente de Contingência	Excluída pelo teste de correlação linear
1	N_cont_vig	Nominal	0.124	Não
2	Auto_vig	Nominal	0.123	Não
3	Antiguidade_contrato	Contínua	-0.112	Não
4	Antiguidade_cliente	Contínua	-0.111	Não
5	Idade	Contínua	-0.094	Não
6	MRH_apol_vig	Nominal	0.081	Não
7	Fraccionamento_Vigor	Nominal	0.071	Não
8	Var_Anonima_07	Nominal	0.068	Não
9	Var_Anonima_09	Contínua	0.065	Não
10	capital_vig	Contínua	-0.064	Não
11	N_sin_Auto_vig	Nominal	0.063	Não
12	N_cob_vig	Nominal	0.06	Não
13	N_cob_anul	Nominal	0.055	Não
14	Auto_PTJ	Nominal	0.054	Não
15	Distrito_new	Nominal	0.054	Não
16	Custo_sin_Auto_vig	Contínua	0.053	Não
17	Auto_anul	Nominal	0.052	Não
18	Fraccionamento_Anulado	Nominal	0.052	Não
19	Md_Custo_sin_Auto_vig	Contínua	0.052	Não
20	Contencioso	Nominal	0.049	Não
21	Pr_mio_Auto_anul	Contínua	0.048	Não
22	N_sin_vig	Nominal	0.046	Não
23	capital_anul	Contínua	0.044	Não
24	Custo_sin_vig	Contínua	0.044	Não
25	PPR_apol_vig	Nominal	0.043	Não
26	N_sin_Auto_anul	Nominal	0.043	Não
27	Auto_FRB	Nominal	0.04	Não
28	Auto_IRE	Nominal	0.039	Não
29	Auto_RSP	Nominal	0.036	Não
30	AT_apol_vig	Nominal	0.035	Não
31	Exclusividade_Agente	Nominal	0.035	Não
32	N_cont_anul	Nominal	0.034	Não
33	Sinistro	Nominal	0.034	Não
34	Auto_BAG	Nominal	0.033	Não
35	N_sin_anul	Nominal	0.033	Não

#	Variável	Tipo Variável	Teste Kendall tau/Coeficiente de Contingência	Excluída pelo teste de correlação linear
36	Sexo	Nominal	0.032	Não
37	Auto_CCC	Nominal	0.032	Não
38	Premio_anul	Contínua	0.032	Não
39	RC_apol_vig	Nominal	0.03	Não
40	Auto_PDU	Nominal	0.03	Não
41	Var_Anonima_06	Nominal	0.029	**Sim**
42	IN_apol_vig	Nominal	0.028	**Sim**
43	Auto_FDN	Nominal	0.028	**Sim**
44	OutrosNV_apol_vig	Nominal	0.026	**Sim**
46	Custo_sin_Auto_anul	Contínua	0.025	**Sim**
47	Md_Custo_sin_Auto_anul	Contínua	0.025	**Sim**
48	Class_Agente	Nominal	0.023	**Sim**
52	Auto_CRT	Nominal	0.021	**Sim**
54	Forma_cobr	Nominal	0.02	**Sim**
55	Custo_sin_anul	Contínua	0.017	**Sim**
57	Premio_vig	Contínua	-0.012	**Sim**
58	Pr_mio_Auto_vig	Contínua	0.011	**Sim**

Tabela 25 – Correlação Linear de cada uma das variáveis em relação à variável target

Anexo E Análise Univariante – Categorização das Variáveis Contínuas

#	Variável	Tipo Variável
1	Antiguidade_contrato	Contínua
2	Antiguidade_cliente	Contínua
3	Idade	Contínua
4	Var_Anonima_09	Contínua
5	capital_vig	Contínua
6	Custo_sin_Auto_vig	Contínua
7	Md_Custo_sin_Auto_vig	Contínua
8	Pr_mio_Auto_anul	Contínua
9	capital_anul	Contínua
10	Custo_sin_vig	Contínua
11	Premio_anul	Contínua
12	Custo_sin_Auto_anul	Contínua
13	Md_Custo_sin_Auto_anul	Contínua
14	Custo_sin_anul	Contínua
15	Premio_vig	Contínua
16	Pr_mio_Auto_vig	Contínua

Tabela 26 – Variáveis contínuas sujeitas a categorização

Anexo F Lista Final dos Indicadores

#	Variável	Tipo Variável
1	N_cont_vig	Nominal
2	Auto_vig	Nominal
3	Antiguidade_contrato	Categórica
4	Antiguidade_cliente	Categórica
5	Idade	Categórica
6	MRH_apol_vig	Nominal
7	Fraccionamento_Vigor	Nominal
8	Var_Anonima_07	Nominal
9	Var_Anonima_09	Categórica
10	capital_vig	Categórica
11	N_sin_Auto_vig	Nominal
12	N_cob_vig	Nominal
13	N_cob_anul	Nominal
14	Auto_PTJ	Nominal
15	Distrito_new	Nominal
16	Custo_sin_Auto_vig	Categórica
17	Auto_anul	Nominal
18	Fraccionamento_Anulado	Nominal
19	Md_Custo_sin_Auto_vig	Categórica
20	Contencioso	Nominal
21	Pr_mio_Auto_anul	Categórica
22	N_sin_vig	Nominal
23	capital_anul	Categórica
24	Custo_sin_vig	Categórica
25	PPR_apol_vig	Nominal
26	N_sin_Auto_anul	Nominal
27	Auto_FRB	Nominal
28	Auto_IRE	Nominal
29	Auto_RSP	Nominal
30	AT_apol_vig	Nominal
31	Exclusividade_Agente	Nominal
32	N_cont_anul	Nominal
33	Sinistro	Nominal
34	Auto_BAG	Nominal
35	N_sin_anul	Nominal
36	Sexo	Nominal
37	Auto_CCC	Nominal
38	Premio_anul	Categórica

#	Variável	Tipo Variável
39	RC_apol_vig	Nominal
40	Auto_PDU	Nominal

Tabela 27 – Lista final das variáveis usadas na modelação