Machine Learning: um estudo sobre conceitos, tarefas e algoritmos relacionados com predição e recomendação

José Saias*, Miguel Maia, Luis Rato e Teresa Gonçalves

Dep. de Informática, ECT, Universidade de Évora

2018-12-23

Resumo

A atual abundância de dados, por um lado, e a complexidade da sua análise e processamento, por outro lado, tornam indispensáveis os sistemas de tratamento automático de dados, para apoio nas mais diversas tarefas, incluindo a tomada de decisão. Este documento procura apresentar e descrever alguma da terminologia referida em publicações científicas e outros textos noticiosos, a propósito de trabalhos que envolvem análise preditiva e recomendação.
Distinguem-se várias atividades de *Data Mining* que envolvem *Machine Learning*, designadamente Classificação, Regressão e *Clustering*, e são ainda enumerados alguns métodos ou algoritmos para cada uma delas, juntamente com as métricas de avaliação de desempenho mais comuns. Este levantamento termina com a apresentação das principais abordagens para um motor de recomendação.

1 Introdução

A abundância de dados é uma realidade atual. A geração automática de dados e a interoperabilidade entre sistemas têm facilitado o acesso a volumes cada vez maiores de dados sobre pessoas e os seus interesses, mas também metadados relacionados com os sistemas, a sua evolução e a forma como são utilizados.
Por um lado a quantidade, por outro lado a complexidade da informação implícita e distribuída em diferentes segmentos de dados, tornam complicada a extração de conhecimento que permita apoiar a tomada de decisão. São necessários mecanismos de tratamento automático dos dados, que permitam obter resultados corretos e em tempo útil. Nos últimos tempos têm surgido cada vez mais notícias sobre a utilização de Inteligência Artificial no desempenho das mais variadas tarefas. Este documento procura apresentar e descrever alguma da terminologia referida em publicações científicas e outros textos noticiosos, a propósito de trabalhos que envolvem análise preditiva e recomendação.

2 *Machine Learning*

Data Mining (DM, ou em Português: Mineração de Dados) foi definida nos anos 90 como um processo de aplicação de algoritmos para a extração de padrões dos dados [5], enquadrando-se dentro de um conceito mais abrangente: *Knowledge Discovery in Databases* (KDD). A

*jsaias@uevora.pt

literatura apresenta também definições um pouco diferentes, nomeadamente sobre o facto do processo poder combinar a Análise Estatística com Inteligência Artificial e *Machine Learning* para procurar padrões, eventualmente implícitos ou desconhecidos.
Machine Learning (ML, ou em Português: Aprendizagem Automática) é uma área dedicada ao desenvolvimento de teorias computacionais e construção de sistemas de aprendizagem, onde a aprendizagem envolve a construção ou modificação automáticas de representações de dados/cenários/objetos em avaliação [16]. Em geral, o objetivo será uma solução de software capaz de realizar uma tarefa, ou de otimizar o desempenho de uma tarefa, com base na experiência existente/anterior e com a mínima intervenção humana.
Podemos identificar duas formas distintas de aprendizagem: supervisionada e não supervisionada. A diferença tem a ver com a inclusão ou não do valor de saída esperado, juntamente com os restantes dados relativos a cada instância de treino. Podemos fazer a analogia com a intervenção de um professor ou supervisor que diz qual é o resultado certo para cada caso, ou a ausência dessa indicação. Em **aprendizagem supervisionada**, para cada instância de treino, para além dos valores de input, é indicado o valor com o output que lhe corresponde, para que a correspondência seja induzida pelo mecanismo de aprendizagem. Duas atividades que se encaixam dentro da aprendizagem supervisionada são a classificação, onde o valor de output é discreto/categórico, e a regressão, com valor de saída contínuo.
Na **aprendizagem não supervisionada** as instâncias de treino têm apenas os dados de input, sem indicação do valor de output esperado. O processo de aprendizagem depende da análise de similaridades dos dados fornecidos, sem aquelas pistas prévias de onde se quer chegar. Uma aplicação típica deste processo é o agrupamento ou *clustering*.
Para além destes tipos fundamentais de aprendizagem, têm surgido outros conceitos relacionados. O termo aprendizagem **semi-supervisionada** é atribuído a uma abordagem mista, tipicamente motivada por existirem dados anotados e dados não anotados no conjunto de treino. Então aplica-se uma estratégia supervisionada para rotular as restantes instâncias com um valor de output. *Reinforcement learning*, ou **aprendizagem por reforço**, é um tipo de aprendizagem automática em que um agente evolui ou aprende progressivamente através da interação com um ambiente, de onde recebe feedback com penalização ou recompensa pelas suas ações ou respostas. *Deep learning*, ou **aprendizagem profunda**, é um modelo de aprendizagem automática baseado em várias camadas de redes neuronais, que por sua vez usam modelos inspirados no funcionamento do cérebro humano. As redes neuronais são usadas há muitos anos, mas só com o poder computacional das plataformas recentes se conseguiu articular de forma eficaz o treino destes modelos multicamada.
Algumas atividades de *Data Mining* que envolvem aprendizagem automática são:

- **Classificação**: uma tarefa que visa prever a categoria de um item, a partir de um modelo baseado numa ou mais variáveis numéricas e/ou categóricas de input, que são também designadas de atributos preditores ou características. O ouput do classificador é um atributo categórico, com a classe ou categoria atribuída à observação.
  Uma possível aplicação deste procedimento é a previsão do tipo de atividade desportiva (descanso, caminhada, corrida) a partir de dados de um acelerómetro.

- **Regressão**: procura modelar a relação entre uma ou mais variáveis independentes e uma variável dependente, para prever um resultado numérico [18]. Um exemplo é a estimativa do valor a gastar em compras num determinado dia, a partir de dados demográficos e histórico de compras de uma pessoa.

- ***Clustering***: ou agrupamento, em Português, é uma tarefa de análise dos dados para, com base em semelhança de características, os dividir em subconjuntos "naturais" (isto

é, que lhes são intrínsecos). Estes subconjuntos são chamados *clusters*. Dois itens no mesmo *cluster* têm determinado grau de semelhança. Itens pertencentes a *clusters* distintos apresentam diferenças mais acentuadas. Em *clustering* usa-se aprendizagem não supervisionada para a procura das pistas de similitude que caracterizam os *clusters* implícitos no conjunto de dados.

- ***Frequent Pattern Mining***: é uma tarefa de extração de padrões úteis, com valor informativo, desde repositórios de dados. Os padrões podem ser de coocorrência de itens, ou padrões complexos de sequência de ocorrência de itens, com ou sem indicação temporal, ou sobre grafos [18]. Como exemplo, o primeiro tipo de processo com padrões de coocorrência é aplicável sobre dados de grandes retalhistas, em *basket case analysis*, para determinar conjuntos de produtos que são comprados conjuntamente, com muita frequência.

Análise Preditiva envolve a descoberta de padrões no histórico de dados, através de processos estatísticos de análise multivariada e/ou processos de *Machine Learning*, que suportem a previsão de resultados futuros. Algumas técnicas de predição de Churn envolvem várias tarefas de DM, como o *clustering* e a classificação.
A criação de um modelo com aprendizagem automática pode envolver parâmetros e hiperparâmetros. Um parâmetro é uma configuração interna ao modelo, que é derivada dos dados, e que influencia estimativas para novos dados. Um hiperparâmetro é uma configuração prévia, externa ao modelo, que é usada para inicializar ou gerir aspetos do processo de ajustamento, como o ritmo de consumo dos dados de treino.

3 Tipologia e pré-processamento de dados

Uma coleção de dados tem um determinado tamanho. O conceito de tamanho tem a ver com o número de instâncias representadas nos dados. Uma **instância** pode corresponder a um produto, ou a uma transação, a uma pessoa ou a qualquer objeto do domínio em que estamos a trabalhar. Cada instância é representada por um conjunto de atributos, ou propriedades ou **características**, que numa representação matricial correspondem às colunas de uma tabela, onde cada linha contém uma instância.
Um processo de análise que considera mais que duas características é designado de análise multivariada [18], em oposição a análise univariada, se analisarmos um só atributo (correspondente a uma só coluna na representação em tabela).
Existem atributos **numéricos** e **categóricos**. Os atributos numéricos podem ter valores inteiros ou reais. Os atributos categóricos tomam valores pertencentes a um conjunto de etiquetas possíveis. Como exemplo: Sim e Não para os booleanos; nome dos dias para um atributo com o dia da semana. Há dois tipos de atributos categóricos. Os **nominais** são atributos cujos valores possíveis não têm relação de ordem entre si. Exemplo: género. Os atributos **ordinais** têm uma estrutura onde existe a noção de ordem, e poderá haver noção de distância ou não.

O pré-processamento de dados inclui procedimentos de limpeza e transformação, contribuindo para um eficaz e eficiente uso futuro. A limpeza de dados visa eliminar erros, tratar os casos de *missing values* (ou casos de valores não preenchidos), ou deteção de inconsistências em registos. Perante uma observação com valores não preenchidos, pode optar-se entre descartar esse registo, ou atribuir-lhe automaticamente um valor, com um valor médio para o atributo, por exemplo. A transformação de dados inclui tarefas relativamente à forma, como a normalização, para uniformizar unidades de medida, maiúsculas, etc... E inclui também

operações de redução de dados, como a discretização.

4 Seleção de Características

O sucesso da aprendizagem automática depende em grande medida da qualidade e adequação dos dados escolhidos [7]. O objetivo da seleção de características é reduzir leque de atributos a considerar nos registos ou instâncias, sem perder informação útil para a tarefa a realizar. Pretende-se remover partes desnecessárias, ou atributos fortemente correlacionados entre si, no sentido de apurar o menor conjunto de atributos que permitam obter o mesmo resultado que se obteria com todos os atributos, mas com ganhos de eficiência e tratabilidade computacional. Para um processo de classificação com aprendizagem supervisionada, pretendem-se as características que minimizem o erro. Em tarefas de *clustering*, onde é comum usar-se aprendizagem não supervisionada, esta fase procura selecionar o menor subconjunto de características que permita identificar os *clusters* [10].

Em trabalhos mais antigos, a seleção de características era muitas vezes realizada por especialistas, com base no seu conhecimento do negócio, e não resultante de um processo de análise quantitativa sobre os dados. A abundância de dados nos sistemas de hoje, por um lado, e a constatação de que o sucesso da análise depende fortemente de uma seleção de características adequada, por outro lado, conduziram a uma diminuição da prática de seleção manual, ou holística, em favor de métodos analíticos automatizados. Ao contrário do que sucedia na área das telecomunicações, onde antes dominavam os critérios de escolha dos especialistas, o sector bancário desde há muito que emprega métodos analíticos para seleção de características. Temos hoje vários trabalhos em seleção de características especificamente para predição de Churn, como [24].

A seleção de características pode envolver uma abordagem do tipo *Filter*, *Wrapper* [11, 13], ou *Embedded*. O método *Filter* efetua a seleção com base em indicadores estatísticos sobre os atributos. Pode envolver análise univariada ou multivariada, resultando numa pontuação para cada atributo, que por sua vez servirá para ordenar e orientar o processo de filtragem. O indicador pode ser Qui-quadrado, entropia relativa (ou *information gain*), ou um coeficiente de correlação entre variáveis. Um método desta família, que usa o último tipo de indicador é o *Correlation-based Feature Subset Selection* (**CFS**).
Com o modo *Wrapper*, a seleção é um processo iterativo de pesquisa, que vai tratar várias hipóteses de subconjuntos de atributos. Para cada subconjunto usa-se um modelo preditivo, que depois de avaliado resulta numa medida (como a exatidão), que é associada à combinação de atributos usada. O valor pode servir para orientar a pesquisa, ou para possível decisão de paragem, por estagnação ou não existência de melhoria. A procura dos subconjuntos pode ser exaustiva, orientada por heurísticas, aleatória, ou seguir um método como *Best-first*.
Os métodos *Embedded* visam aprender quais os atributos que mais contribuem para o desempenho do modelo enquanto o mesmo é ajustado ou criado.

5 Algoritmos de Classificação

A técnica de classificação ***Naive Bayes*** (NB) é inspirada no teorema de Bayes, assumindo que as características seguem uma distribuição normal, ou a independência entre as características. O método é rápido e aplicável a cenários de múltiplas classes (mais que duas). ***Linear Discriminant Analysis*** (LDA) é outra técnica de base estatística que procura uma combinação linear nos dados que permita separar duas ou mais classes.
O método de regressão ***Logistic Regression*** (LR) é muito usado em classificação binária, onde a decisão entre classes pode fazer-se em função da probabilidade estimada pelo algoritmo para cada caso. É assumida uma relação linear entre variáveis independentes e variáveis dependentes. O algoritmo é referido também na secção seguinte, para regressão.
O algoritmo Máquina de Vetores de Suporte, ou ***Support Vector Machines*** (SVM), baseia-se num processo de aprendizagem supervisionada muito usado para classificação, mas também para regressão. Para classificação binária, onde o resultado deve ser uma de duas categorias, o modelo é baseado numa representação espacial, na qual é encontrado um hiperplano que separa a região das classes. O SVM usa uma função de kernel para mapear os dados dos atributos num espaço altamente dimensional, onde seja possível encontrar um separador linear ótimo [3]. O ajuste do modelo procura encontrar um hiperplano com máxima distância, ou margem, até aos pontos de cada classe. A capacidade de generalização do processo de classificação é tanto melhor quanto aquela distância de separação.
K-Nearest Neighbors (KNN) é um algoritmo usado para classificação, mas também para regressão. A classificação de novos casos é efetuada com base na semelhança com casos já vistos, ou treinados. A semelhança pode medir-se com a distância euclidiana ou outra. Para a tarefa de regressão, o output seria um valor médio da mesma propriedade, para os K vizinhos mais próximos.
O método de **Árvores de Decisão** (AD) é um dos mais usados em classificação, onde é gerada uma estrutura hierárquica, como uma árvore onde os nós correspondem a testes sobre critérios aprendidos nos dados. Uma vantagem das árvores de decisão é a sua relativa simplicidade, onde é possível interpretar a lógica de decisão (entre classes) na própria estrutura criada. O algoritmo **C4.5** é um método desta família [22]. O algoritmo **J48** é uma implementação específica de C4.5 numa solução de código aberto, disponibilizada à comunidade científica em várias ferramentas de análise de dados.
Os métodos ***Ensemble***, ou métodos combinados, procuram otimizar o desempenho da predição utilizando um conjunto de classificadores separados, cujas respostas são combinadas para gerar uma nova classificação. Os métodos *ensemble* homogéneos usam vários classificadores do mesmo tipo, treinados com frações diferentes, ou aleatórias, dos dados. Exemplos: *Boosting* e *Random Forest*. Os métodos *ensemble* heterogéneos recorrem a classificadores de natureza diferente, e determinam a resposta usando uma média, maioria, ou outro critério ponderação dos resultados parciais.
Bagging ou *Bootstrap aggregating* é uma técnica *ensemble* para classificação baseada na combinação de respostas de modelos treinados com subconjuntos de dados construídos de forma aleatória. O valor da classificação é a média, ou o resultado maioritário, das estimativas internas.
O método ***Random Forests*** (RF) é um *ensemble* de árvores de decisão. Segue uma abordagem idêntica a *Bagging*, em termos de divisão do conjunto de treino em partes geradas ao acaso, para treinar diferentes árvores de decisão e posteriormente agregar os resultados escolhendo o voto maioritário. Em relação a uma árvore de decisão simples, o RF é mais resistente a sobreajustamento.
Boosting é uma família de processos *ensemble* mais complexos que os anteriores. A ideia

geral é que a cada fase, um modelo fraco é melhorado, através do ajuste para os casos em que antes falhava.
O processo começa pelo treino de um modelo base, considerando todo o conjunto de treino de forma comum. Depois é feita uma avaliação de cada resposta desse modelo, e na fase seguinte é dado peso maior às instâncias que antes tinham erro de classificação. Estabelecido o critério de pesos, é treinado novo modelo. O processo pode repetir-se várias vezes, até estabilizar a avaliação. Dois dos algoritmos deste tipo são o ***AdaBoost*** e o ***Stochastic Gradient Boosting***. O algoritmo **XGBoost** [2] é um dos mais recentes na linha dos *Gradient Boosted Trees*, com ganhos de eficiência na procura dos critérios ideais para nova ramificação a introduzir na árvore, considerando a esparsidade dos dados. O algoritmo treina uma árvore de decisão em cada fase, de modo a melhorar ou corrigir erros da árvore treinada na fase anterior, e acrescenta fatores de correção na função objetivo, no sentido de reduzir o risco de sobreajustamento.
As **Redes Neuronais** (RN) são modelos matemáticos inspirados na estrutura biológica do cérebro, com as sinapses e neurónios, que usam uma abordagem de computação baseada em estruturas densamente interligadas. Uma RN tem camadas, usualmente a de entrada, a de saída e uma ou mais camadas intermédias, designadas *hidden layers*. Nas variantes recentes de *Deep Learning* as RN têm vindo a usar mais camadas, beneficiando da maior capacidade de cálculo das plataformas atuais, algumas recorrendo a GPUs. Em cada camada há neurónios artificiais, ou nós, que têm uma unidade de processamento com múltiplas entradas. A rede é treinada relativamente ao modo como se interligam os neurónios de camadas adjacentes, e como cada neurónio trata os sinais que recebe. Na prática, corresponde a afinar uma função onde os valores de entrada em cada neurónio são ponderados por um coeficiente. Para o ajuste destes pesos ou coeficientes, um dos métodos usados é o *Back Propagation* [14].
As RN têm usualmente um custo computacional maior que LR ou as árvores de decisão. E os modelos não têm interpretabilidade, por esconderem a lógica na estrutura complexa das camadas intermédias, onde não é fácil entender a influência de cada preditor. Uma das vantagens apontadas às RN é a alta tolerância a ruído nos dados.
Os parâmetros no modelo de uma RN clássica são o *decay* para os pesos (regularização para contrariar o risco de sobreajustamento), o número de neurónios em cada camada escondida, e o número de camadas da rede. Numa rede neuronal ajustada com abordagens modernas, são comuns os hiperparâmetros:

- *learning rate*: indica o ritmo de aprendizagem, que influencia a velocidade até à convergência;
- *loss function*: é uma função de custo que compara a diferença entre a estimativa e o valor de referência, fornecido com a instância de treino, que dá uma distância com uma medida de avaliação de erro;
- épocas: cada época corresponde às iterações necessárias para percorrer todo o conjunto de treino, uma vez; uma época costuma envolver múltiplos *batches*;
- *batch size*: conjunto de instâncias de treino a considerar numa iteração de treino do modelo; útil por questões de eficiência, e fundamental quando o volume de dados total é superior à memória disponível;
- número de iterações: número de repetições de treino, com os diferentes *batches* até completar uma época.

No caso da classificação de Churn, existem também abordagens híbridas, que começam por dividir os clientes em grupos, com *clustering*. De seguida, é treinado um modelo de classificação de Churn para cada *cluster*.
À partida, não podemos apontar um dos algoritmos como o melhor, em abstrato. Dependendo do contexto, do volume de dados, do tipo de variáveis, e dispersão de valores em cada atributo, o mesmo algoritmo pode ter diferentes níveis de desempenho.

6 Algoritmos de Regressão

A regressão é usada quando procuramos estimar um valor numérico, contínuo, como a evolução do preço do imobiliário, por exemplo. Os modelos de **Regressão Linear** procuram prever o valor de saída y em função de uma combinação linear dos parâmetros de entrada x_j, ou variáveis independentes ou explicativas. Com uma só variável, temos uma regressão simples que podia ser representada por uma linha. Com vários atributos de entrada, a regressão corresponde a uma plano no espaço n-dimensional [4]. A diferença entre uma observação y e o valor previsto, $\hat{y}$, é chamado valor residual, representado por e na equação:

$$y(i) = \hat{y}(i) + e(i) = a_0 + \sum_{j=1}^{p} a_j x_j(i) + e(i),\ 1 <= i <= n. \quad (1)$$

Numa representação matricial, o valor de y corresponde a um produto entre a matriz dos atributos e a matriz dos coeficientes de regressão, acrescido ainda o vetor com os valores residuais. O objetivo é encontrar os coeficientes de regressão que minimizem os desvios. Uma funções mais usadas é a soma do quadrado dos erros (de *sum of squared errors* (SSE)), com o objetivo de minimizar a soma dos quadrados de $e(i)$. O ajuste ou treino do modelo de regressão linear corresponde a encontrar estes coeficientes de regressão, que podem ser usados depois, na equação anterior, para estimar novos casos.

Logistic Regression, ou regressão *Logit*, é outro tipo de regressão, usada quando a variável dependente é binária, e que estima a probabilidade de cada um dos dois resultados possíveis da predição. A relação entre variável dependente e variáveis independentes não tem de ser linear. É uma técnica usada também em tarefas de classificação. A ***Multinomial Logistic Regression*** é uma variante para problemas onde a variável dependente pode ter mais que duas classes.

Quando os variáveis independentes estão altamente correlacionadas, uma das técnicas aplicáveis é ***Ridge Regression***. Esta técnica usa um fator de redução do peso (*shrinkage*) dos coeficientes de regressão, com o objetivo de atenuar erros que resultam do efeito da variância no modelo. Usa uma regularização do tipo L2, que significa penalizar em função do quadrado da magnitude dos coeficientes. Na presença de grande correlação, os coeficientes são atenuados até valores próximos de zero, mas não nulos.
Lasso Regression, de *Least Absolute Shrinkage and Selection Operator* [23], vem na mesma linha de ponderação dos coeficientes a propósito da variância. Usa uma regularização do tipo L1, diferente de *Ridge*, penalizando em função da magnitude absoluta dos coeficientes (em vez do quadrado). Na presença de grande correlação, os coeficientes são atenuados até valores que podem chegar a zero. Numa situação limite, pode restar apenas um e os restantes serem anulados. ***ElasticNet Regression*** é um modelo de regressão híbrido, que combina as técnicas *Ridge* e *Lasso*.

7 Algoritmos de Clustering

O objetivo do *clustering* é determinar automaticamente os grupos homogéneos em que as instâncias podem ser classificadas, analisando semelhanças e diferenças nos dados.
Consideremos um espaço de dados com N pontos, que representam as instâncias de dados, e com X dimensões, que corresponde ao número de atributos. O algoritmo **K-Means** usa um processo iterativo para encontrar uma partição do espaço dos dados que otimize uma função para a qualidade daquela separação. A função pode ser a *sum of squared errors* (**SSE**), que terá o valor mínimo para a melhor solução de *clustering*.
Para K *clusters*, o processo inicia com a escolha ao acaso de K pontos do espaço de dados. Em cada iteração, faz-se a atribuição de *cluster*, e a atualização do centroide. Na atribuição de *cluster*, calcula-se a distância de cada ponto no conjunto de dados a cada um dos K pontos, sendo-lhe associado o que estiver mais perto. No final, o conjunto de associações define os *clusters*. A atualização de centroide envolve o recálculo do ponto médio de cada *cluster*, que corresponde à média para cada atributo, considerando os pontos do conjunto de dados que estão associados ao *cluster*. Se não houver alteração de centroide, ou se a diferença for muito pequena e abaixo de um limiar, então o processo termina. O algoritmo requer dados numéricos.
O algoritmo **Kernel K-means** é uma variante que permite fronteiras não lineares entre *clusters*, com uma função de *kernel*.
Se os dados forem categóricos, a técnica *Latent Class Analysis* (**LCA**) pode aplicar-se na identificação de *clusters* [6].
O método ***Expectation Maximization*** estima as probabilidades de cada ponto do conjunto de dados pertencer a cada *cluster*, em vez da atribuição rígida efetuada pelo algoritmo *K-Means*. O processo é iterativo, com ajustes nos parâmetros dos *clusters* e recálculo das probabilidades [15].
Clustering hierárquico tem como objetivo a definição de conjuntos de partições com diferentes níveis de profundidade, onde várias partições de um nível podem estar dentro de uma só partição do nível de cima, formando uma espécie de árvore, onde no final existem *clusters* terminais, com um só ponto do conjunto de dados. Na raiz existe o *cluster* de topo, com todos os pontos. Há dois tipos de abordagem, em função do sentido da evolução. O método aglomerativo inicia nos pontos e vai até ao topo. O método divisivo inicia no conjunto inteiro, e aplica progressivamente as divisões, em modo *top-down* [9].
Para *clusters* com formas menos convencionais, não convexas, existe ainda outra família de algoritmos que baseado na densidade dos conjuntos de pontos, como o DBSCAN.

8 Avaliação de Modelos

Saber se um modelo é confiável, ou se é minimamente adequado, requer uma avaliação. Desta forma, podemos determinar se uma alteração de parâmetros aumenta ou diminui o desempenho. No caso da classificação, é comum usar-se um conjunto de dados anotado, isto é, com a indicação da classe em cada caso. As instâncias neste conjunto serão divididas, por forma a usar-se um subconjunto para treino do modelo, com algoritmos e parametrização antes definidos, e outro subconjunto para avaliação do modelo resultante daquele treino com aprendizagem automática, e de onde será extraído um indicador quantitativo sobre o desempenho. A secção 9 descreve várias métricas usadas neste contexto.
Quando o número de instâncias é suficientemente grande, usa-se um subconjunto adicional, o conjunto de validação, que pode formar-se retirando uma fração ao conjunto de treino, e que

serve para controlar o processo de aprendizagem, para ajustar parâmetros.
Independentemente do tamanho relativo destes subconjuntos, existem várias abordagens para realizar a divisão das instâncias, entre treino e teste. Por um lado, pode fazer-se uma escolha aleatória, definindo o tamanho do conjunto de teste em termos relativos, com 10 a 30% do tamanho do conjunto total. Este processo é designado ***Hold-Out Split Method***, em alguma bibliografia. Mas existe sempre o risco das instâncias pertencentes a qualquer um dos conjuntos não serem as mais adequadas. E nesse caso o critério de divisão das instâncias iria afetar o desempenho.
O método ***Bootstrap*** vem da estatística, onde é usado para estimativas relativamente a uma população a partir de médias das estimativas obtidas para diversas amostras, mais pequenas, dessa população. A mesma estratégia pode usar-se na avaliação de um modelo. O processo é parametrizado com o tamanho da amostra e o número de repetições a efetuar. Em cada repetição define-se a amostra, escolhendo ao acaso as instâncias que dela fazem parte. De seguida faz-se o ajuste, ou treino, do modelo usando a amostra como conjunto de treino. Para conjunto de teste pode escolher-se o resto da população (tudo excepto a amostra). Daí extrai-se o indicador de desempenho. No final das repetições, calcula-se o valor médio do indicador. Continuará a existir o risco de uma amostra ser usada várias vezes.
Se usarmos ***K-fold cross-validation***, os dados são divididos em K subconjuntos de tamanho igual, sendo alternadamente usado cada um como validação de um modelo treinado com os restantes K-1 subconjuntos. Desta forma, garante-se que todas as instâncias terão sido consideradas na avaliação. É comum usar-se 5 ou 10 para o valor de K.
Um dos problemas com modelos de aprendizagem é o risco de **sobreajustamento** (ou *overfitting*, na bibliografia em Inglês), e que consiste numa adaptação ao conjunto de treino demasiado forte. Daqui resulta que novos casos que sejam iguais ou muito parecidos a casos treinados terão resultados certos, mas para casos com características não vistas o risco de erro é grande [4].
Quando o número de instâncias de cada classe é substancialmente diferente, diz-se que não existe balanceamento entre as classes. Este desequilíbrio pode ter impacto na avaliação, mas também na qualidade do treino, se a classe minoritária não for representada adequadamente. Por vezes, usa-se subamostragem (em relação à classe com mais instâncias), e/ou sobreamostragem (em relação à classe com menos instâncias), para remover ou adicionar instâncias, procurando colmatar o desequilíbrio [1].

9 Métricas de desempenho

Com o conjunto de teste preparado para avaliar um modelo, é preciso quantificar a adequação das respostas para todas as instâncias testadas. Em geral, um sistema preditivo é complexo, e há várias métricas de desempenho possíveis.
Pensando no caso da classificação, *accuracy*, ou **exatidão**, é a taxa de acerto, que corresponde ao número de classificações corretas sobre o total de casos a classificar. Apesar de ser simples, é uma medida onde um valor elevado nem sempre corresponde a bom desempenho do sistema, especialmente quando não há equilíbrio entre as classes.
A **matriz de confusão** é uma grelha com indicadores numéricos, como indicado na Figura 1, usada em Recuperação de Informação e também em classificação. Considerando uma classificação em duas classes, por exemplo positivo e negativo, os valores da tabela são:

- VP: verdadeiros positivos, ou o número de casos corretamente classificados como positivos;

- VN: verdadeiros negativos, ou o número de casos corretamente classificados como negativos;
- FP: falsos positivos, ou o número de casos classificados como positivos mas que eram negativos;
- FN: falsos negativos, ou o número de casos classificados como negativos mas que eram positivos.

		valor devolvido	
		Pos	**Neg**
valor real	**Pos**	VP	FN
	Neg	FP	VN

Figura 1: Matriz de confusão

Com estes elementos podem calcular-se as seguintes medidas: **precisão** (ou *positive predictive value*, PPV), sensibilidade (***recall*** ou *true positive rate*, na bibliografia em Inglês), **especificidade** (ou *true negative rate*), e *fallout* (ou taxa de falsos positivos), de acordo com as fórmulas seguintes.

$$precisão = \frac{VP}{VP + FP} \tag{2}$$

$$recall\ ou\ TVP = \frac{VP}{VP + FN} \tag{3}$$

$$especificidade = \frac{VN}{VN + FP} \tag{4}$$

$$fallout\ ou\ TFP = \frac{FP}{FP + VN} \tag{5}$$

$$F1 = \frac{2 \times precisão \times recall}{precisão + recall} \tag{6}$$

A **Medida F** combina precisão e *recall* num só indicador sobre a qualidade do modelo. Tem variantes relativamente ao peso a dar a cada uma, mas uma opção comum é **F1**.
Quando existe interesse especial na classe menos frequente, é possível usar-se a ***macro average* F1**, que prevê o cálculo de F1 de forma independente para cada classe, apurando-se depois o valor médio entre as classes. Desta forma temos um tratamento igual entre classes, independentemente o número de casos de cada uma.
Estas medidas têm um valor entre 0 e 1, onde 1 seria o desempenho perfeito. Podem também ser apresentadas na forma de percentagem, com o resultado multiplicado por 100.
Quando a classificação, por exemplo de Churn, é baseada numa probabilidade, é comum usar-se um valor de referência, ou limiar Tr. Quando a probabilidade apurada é acima de Tr, considera-se Churn, e se for abaixo de Tr, considera-se Não-Churn. A curva *Receiver Operating Characteristic*, ou **curva ROC**, é uma representação gráfica que relaciona sensibilidade (nas ordenadas) com 1-especificidade, ou taxa de falsos positivos, (abcissas no eixo X), como ilustrado na Figura 2, para cada valor do limiar Tr, entre 0 e 1. A área sob a curva ROC é um indicador de desempenho designado **AUC** (do Inglês: *area under the ROC curve*), muito usado para tarefas de classificação, e em particular onde a distribuição dos casos pelas classes não é equilibrada. É também usada em métodos estatísticos convencionais, incluindo na

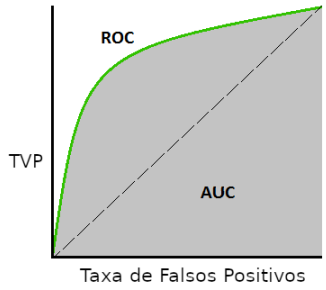

Figura 2: Exemplo de Curva ROC e AUC

área médica[1]. Quanto maior o valor da área de AUC, melhor é o desempenho. Esta medida é independente de um limiar em particular, dando um resultado global, associado a toda a gama de valores que o limiar pode tomar, refletindo a qualidade do modelo e não da escolha realizada para o limiar. Um classificador aleatório teria a função identidade como curva ROC e 0,5 como AUC. Um bom classificador deve ter um valor de AUC superior.
Note-se que no caso da classificação de SPAM em correio eletrónico, a AUC não seria uma boa medida, pois nesse cenário a prioridade é minimizar os falsos positivos. Por outro lado, se pretendermos o mesmo tipo de indicador, mas considerando ao mesmo tempo a distribuição entre classes, existe uma variante designada *Area Under the Kappa curve* (**AUK**) [12]. Se o classificador trabalhar com mais que duas classes, por exemplo A, B e C, poderia calcular-se o valor de AUC três vezes, numa abordagem de cada classe contra as restantes (A *vs* B e C; B *vs* A e C; C *vs* A e B).
O *Top decile lift* (**TDL**) é uma medida de desempenho pensada para dar mais peso ao decil (ou analogamente para o n-ésimo percentil) dos casos com maior probabilidade de pertencerem à classe de interesse (por exemplo serem clientes em risco de Churn). O valor TDL pode calcular-se ordenando os casos pela probabilidade apurada, e depois dividir a proporção de casos positivos no decil de topo pela proporção de casos positivos em todo o conjunto. Um modelo aleatório teria um TDL a 1. Com um valor TDL de 3, temos a indicação de que há 3 vezes mais casos positivos (por exemplo casos de Churn) no decil de topo (nos 10% de cima) que no conjunto inteiro. Se o sistema em causa debitar probabilidade de Churn, este tipo de métrica pode ser muito relevante. Veja-se uma situação em que a intervenção junto de clientes é feita apenas para os casos de maior probabilidade de desistência, sendo particularmente importante um bom desempenho nessa franja de casos.
É possível ainda adaptar a medida de desempenho à importância que terá um caso mal classificado, para determinado contexto. Para tal, considerar-se-ia uma ***Loss function***, ou função que determina o custo a considerar. Na área que interessa a este estudo, o custo podia levar em consideração o montante gasto por um cliente. Existem trabalhos com esta estratégia por exemplo na análise de Churn em cartões de crédito [19].

Uma das possibilidades para avaliar uma modelo de regressão é deixar, na fase de treino, algumas instâncias de fora, ao estilo *Hold-out*. Sobre os valores estimados para essas instâncias, extrai-se uma medida, como o *mean squared error* (**MSE**, ou erro quadrático médio), e compara-se com o valor obtido para essa medida, relativamente às instâncias que estiveram na base do modelo. Se houver uma diferença significativa, então o modelo não está bem ajustado, não é generalizável para dados não treinados.

[1]https://www.medcalc.org/manual/roc-curves.php

A avaliação de uma solução de *clustering* pode ser feita de modo interno, externo ou relativo. Em modo interno, usam-se critérios retirados dos próprios dados, numa avaliação dentro de cada *cluster* ou entre *clusters* [18]. Há medidas para avaliar a compacidade, que corresponde ao grau de semelhança entre os elementos do mesmo *cluster*. Por outro lado, podemos também estimar a distância entre os elementos de *clusters* diferentes. Estabelecendo um compromisso entre essas medidas, de compacidade e de separação dos *clusters*, temos uma noção da qualidade do processo de *clustering*.
Em avaliação com modo externo, usa-se informação adicional, que não provém dos dados, como por exemplo o conhecimento prévio sobre os *clusters* reais, fornecido por especialistas do domínio. Numa avaliação em modo relativo, procura-se estabelecer uma comparação entre diversos processos de *clustering*.

10 Sistemas de Recomendação

Um sistema de recomendação é uma solução de software que procura pistas ou sugestões de itens ou serviços relevantes para um utilizador [17], usualmente suportado por técnicas de *Machine Learning*. Tais sistemas têm sido amplamente usados em sites de comércio eletrónico, para encontrarem as melhores sugestões de compra, nomeadamente a partir de dados com as últimas aquisições dos utilizadores e os termos de pesquisa usados em motores de busca. Outras aplicações comuns são a recomendação de livros e a sugestão de vídeos online [20], para referir algumas das mais populares. As principais abordagens num motor de recomendação são:

- ***Collaborative filtering***: técnica onde as recomendações para um utilizador decorrem de um modelo do comportamento anterior de um grupo de utilizadores com perfil semelhante [25]. É comum o recurso a mecanismos de recolha de feedback, ou avaliação, de itens, ou serviços, por parte dos utilizadores. A partir das semelhanças na avaliação, faz-se uma extrapolação da avaliação num novo caso (com base na avaliação desse caso por utilizadores de perfil semelhante).

- ***Content-based filtering***: a recomendação assenta na análise de semelhança entre os itens ou conteúdos escolhidos pelo utilizador. A recomendação para um utilizador não depende de outros utilizadores, mas antes do seu histórico e análise de semelhança da caracterização dos itens que escolheu antes. Esta técnica pode usar-se sobre itens novos, para os quais não existem avaliações expressas por utilizadores (e onde a abordagem *Collaborative* não funcionaria). Por outro lado, o processo depende da existência de metadados com propriedades dos itens [8].

- **Híbridos**: combinam ambas as estratégias acima, aplicando possíveis fatores de ponderação.

Os motores de recomendação realizam análise prescritiva, isto é, que mediante um processo análise emitem determinada sugestão. Alguns desafios que se colocam nestes sistemas são a dispersão de dados e o *cold-start* [21]. O primeiro aspeto tem a ver com o facto de que a existência de muitos dados nem sempre significa utilidade imediata desses dados. Muitas vezes, os dados disponíveis podem corresponder a itens diferentes. Se utilizadores diferentes avaliam itens diferentes, será difícil encontrar os padrões de semelhança ideais para suportar uma recomendação. O problema de *cold-start* tem a ver com a fase inicial de um sistema, ou da operação relativamente a um novo utilizador. Alguns modelos dependem da existência

prévia de um histórico de dados, por exemplo sobre escolhas anteriores. Novos utilizadores, que podem surgir ao longo de todo o período de funcionamento do sistema, podem não ser tratáveis, ou não receber a melhor resposta do modelo, que foi ajustado sobre dados potencialmente não representativos.

11 Conclusão

No âmbito do projeto APRA-CP.v2, este documento faz um levantamento de conceitos relacionados com *Machine Learning*, muito em voga atualmente, como parte de um trabalho introdutório de contextualização para apoio ao desenho de um sistema preditivo, e à escolha fundamentada das abordagens a seguir para o implementar.
Foram descritos vários tipos de dados e o típico pré-processamento que se lhes aplica em processos desta natureza. Enumeraram-se várias abordagens para a seleção de características, na secção 4. Distinguiram-se várias atividades de *Data Mining* que envolvem aprendizagem automática, designadamente Classificação, Regressão e *Clustering*, e foram caracterizados alguns métodos ou algoritmos para cada uma delas. Nas secções 8 e 9 apresentámos conceitos e procedimentos usualmente empregues na avaliação dos modelos, bem como as métricas mais usadas nessa avaliação, respetivamente. Foi ainda apresentado o conceito de Sistema de Recomendação, e descritas sumariamente as principais abordagens para formar o modelo de recomendação.

Agradecimentos

Este artigo relata trabalho desenvolvido no âmbito do projeto "*Admin Portal & Reporting Analytics for Cloud Providers resellers - V2*" (APRA-CP.v2), com a referência ALT20-03-0247-FEDER-038500, apoiado pelo Programa Operacional Regional do Alentejo 2014/2020.

Referências

[1] Burez, J., Van den Poel, D. (2009). *Handling class imbalance in churn prediction problem.* Expert Systems with Applications, 36. 4626-2636

[2] Chen, T.Q. and Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* arXiv:1603.02754v3

[3] Cortes C, Vapnik V (1995). *Support-vector networks.* Machine Learning, 20(3):273–297

[4] David Hand, Heikki Mannila and Padhraic Smyth (2001). *Principles of Data Mining.* MIT Press

[5] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). *From data mining to knowledge discovery in databases.* AI Magazine, 17(3), 37–53. doi:10.1609/aimag.v17i3.1230

[6] Formann, A. K. (1984). *Latent class analysis: Introduction to theory and application.* Weinheim: Beltz.

[7] García, D.L., Nebot, À. & Vellido, A. (2017). *Intelligent Data Analysis Approaches to Churn as a Business Problem: a Survey.* Knowledge and Information Systems 51: 719 https://link.springer.com/article/10.1007/s10115-016-0995-z

[8] Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A. (2015). *Recommendation systems: Principles, methods and evaluation*. Egyptian Informatics Journal, Volume 16, Issue 3, November 2015, Pages 261-273, ISSN 1110-8665

[9] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

[10] Jennifer Dy and Carla Brodley (2004). *Feature selection for unsupervised learning*. Journal of Machine Learning Research, 5(1):845–889

[11] Karegowda, A. G., Jayaram, M. A., & Manjunath, A. S.(2011). *Feature Subset Selection using Cascaded GA & CFS: A Filter Approach in Supervised Learning*. International Journal of Computer Applications. 975–8887.

[12] Kaymak, U., Ben-David, A. and Potharst, R. (2012). *The AUK: A Simple Alternative to the AUC*. Engineering Applications of Artificial Intelligence, Volume 25, 5, 1082-1089.

[13] Kohavi, R. (1995). *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. Doctoral dissertation, Stanford University

[14] Martin Riedmiller and Heinrich Braun (1993). *A direct adaptive method for faster backpropagation learning: the RPROP algorithm*. IEEE International Conference on Neural Networks, 16:586–591

[15] McLachlan, G. and Krishnan, T. (2008). *The EM algorithm and extensions, 2nd Edition*, New Jersey: John Wiley and Sons

[16] Michalski, R., Carbonell, J., & Mitchell, T.(1986). *Machine Learning. An Artificial Intelligence Approach*. Morgan Kaufmann Publishers

[17] Mohammad Aamir and Mamta Bhusry (2015). *Recommendation System: State of the Art Approach*. International Journal of Computer Applications 120(12):25-32

[18] Mohammed J. Zaki and Wagner Meira (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press

[19] Nie, G., Rowe, W., Zhang, L., Tian, Y., & Shi, Y. (2011). *Credit card churn forecasting by logistic regression and decision tree*. Expert Syst. Appl., 38, 15273-15285.

[20] *Paul Covington, Jay Adams and Emre Sargin (2016). Deep Neural Networks for YouTube Recommendations*. In Proceedings of the 10th ACM Conference on Recommender Systems, USA 2016

[21] Prem Melville and Vikas Sindhwani (2010). *Recommender Systems*. In Encyclopedia of Machine Learning, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010

[22] Quinlan, JR. (1993). *C4.5: Programs for Machine Learning*. Machine Learning 16, 235-240. Morgan Kaufmann Publishers, Inc.

[23] Tibshirani, Robert (1996). *Regression Shrinkage and Selection via the lasso*. Journal of the Royal Statistical Society. Series B (methodological). Wiley. 58 (1): 267–88

[24] Xiao J, Xiao Y, Huang A, Liu D, Wang S (2015). *Feature-selection-based dynamic transfer ensemble model for customer churn prediction*. Knowledge and Information Systems, 43(1):29–51

[25] Xiaoyuan Su and Taghi M. Khoshgoftaar (2009). *A Survey of Collaborative Filtering Techniques*. In Advances in Artificial Intelligence 2009