Exploração e comparação de dados numa Operadora Móvel

Luís Araújo
Dep. de Engenharia Informática
ISEP
Porto, Portugal
1190827@isep.ipp.pt

2nd Filipe Costa
Dep. de Engenharia Informática
ISEP
Porto, Portugal
1210098@isep.ipp.pt

3rd Jorge Gonçalves
Dep. de Engenharia Informática
ISEP
Porto, Portugal
1210107@isep.ipp.pt

***Abstract*—Neste trabalho irá ser analisado um dataset com dados reais sobre uma operadora móvel. Estes dados serão usados como dados de teste e treino para algoritmos de ˜ aprendizagem automática nos temas de Regressão e Classificação tais como: Regressão Linear; Árvores de Decisão; K-vizinhos-mais-próximos; Redes Neuronais.**

***Index Terms*—Machine learning, Algoritmos, Regressão e Classificação, Regressão Linear Simples, Regressão Linear Múltipla, RMSE, MAE, Árvores de Regressão, Árvores de Decisão, K-vizinhos-mais-próximos, Redes Neuronais.**

I. Introdução

Este documento foi realizado no âmbito da unidade curricular de Análise de Dados, do curso de Engenharia Informática do Instituto Superior de Engenharia do Porto.

O trabalho desenvolvido incidiu sobre um conjunto de dados de uma Operadora Móvel, mais concretamente sobre a situação anormal de perda de clientes.

Este mesmo trabalho teve como objetivo a Análise de Desempenho de Técnicas de Aprendizagem Automática e foram utilizados diversos algoritmos como: Regressão Linear, Árvores de Decisão, K-vizinhos-mais-próximos e Redes Neuronais. A ferramenta utilizada de forma a cumprir este objetivo foi R.

No próximo capítulo é apresentado o enquadramento teórico sobre os algoritmos utilizados no relatório.

A. Regressao Linear e Árvores de Regressão

A regressão linear, bastante utilizada para se estimar um valor esperado de uma certa variável, tem por base num conjunto de variáveis: dependentes e independentes. As variáveis independentes (ou preditoras) são importantes para o calculo da variável que se deseja prever, sendo que cada uma tem a sua influencia naquilo que e o valor esperado/previsto. Resumidamente, a regressão linear tem 2 objetivos: analisar um conjunto de variáveis preditoras e a forma como estas mesmas impactam a variável dependente. A regressão linear pode ser caraterizada por dois tipos: regressão linear simples e regressão linear múltipla. [1]

A regressao linear simples traduz a previsão de uma variável com a introdução de apenas uma variável preditora, como traduz a formula seguinte: [2]

$$Yi = \beta_0 + \beta_1 * Xi + \xi_i$$

A regressão linear múltipla traduz o cálculo da variável desejada introduzindo um conjunto de variáveis independentes:

$$Yi = \beta_0 + \beta_1 * X_1 i + \beta_2 * X_2 i + ... + \beta_k * \xi_i$$

Ambos os modelos tem por objetivos determinar como duas ou mais variáveis se relacionam, estimar uma função que permita determinar a relação entre variáveis e usar essa equação resultante para prever valores possíveis para a variável dependente. Por um lado, regressão linear não determina qualquer relação causal, pode apenas dar pistas que podem ser testadas estatisticamente. [3] Por outro lado, uma árvore de regressão é formada por nós de decisão e é uma ferramenta usada quando a variável dependente é quantitativa. A árvore de regressão é um exemplo de árvore de decisão e tem a função de calcular um valor médio de previsão para cada nó da árvore usando a soma de quadrados e analise de regressão. [4]

B. Árvores de Decisão

As Árvores de Decisão são uma ferramenta de suporte a decisões que mapeia todos os resultados possíveis a partir de uma série de escolhas. Estas consistem em um conjunto de nós de decisão, conectados por ramos, estendendo-se para baixo a partir do nó raiz até terminar em nós folha. Começando no nó raiz, que por convenção é colocado no topo do diagrama de árvore de decisão, as variáveis são testadas nos nós de decisão, com cada resultado possível resultando numa ramificação. Cada ramificação leva a outro nó de decisão ou a um nó folha final. As árvores também podem ser representadas por um conjunto de regras para melhorarem a legibilidade e a interpretabilidade humana. Estes métodos de aprendizagem estão entre os mais utilizados e têm sido aplicados a um grande número de campos (desde diagnóstico médico à gestão de risco de crédito na banca).

Um dos exemplos mais conhecidos de uma árvore de decisão é o exemplo do ténis como observamos na Fig 1. Neste exemplo conseguimos determinar se é um ”bom dia para jogar ténis” através da Árvore de Decisão. Primeiro testa-se

o nó de decisão ”Outlook”, gerando 3 ramos possíveis, em que dois deles seguem para outro nó de decisão, e o outro vai diretamente para uma folha com a classificação, respondendo a pergunta inicial. [5]

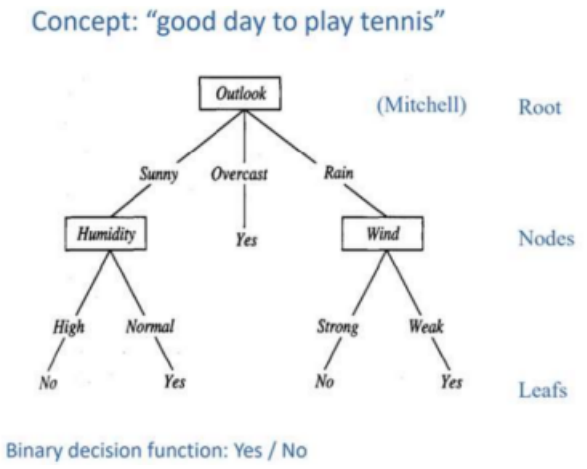

Fig. 1. Exemplo de uma arvore de decisão.

Entre as vantagens das Árvores de Decisão, temos a facilidade com que são compreendidas ou capacidade de se decidir as melhores opções. A grande desvantagem presente é o facto de que estas se podem tornar excessivamente complexas, dificultando o processo de tomada de decisão. [6]

C. Redes Neuronais

Redes neuronais artificiais são o coração dos algoritmos de aprendizagem profunda, sendo elas inspiradas pela estrutura do cérebro humano e da comunicação biológica neuronal do mesmo. Estas têm um número surpreendente de características observadas no processo cognitivo do ser humano tal como aprendizagem pela experiência. Para construir um sistema destes, é necessário definir o número e tipo de neurónios e como estes se conectam entre si, colocar um peso entre eles e perceber quais destes pesos são apropriados para o problema em questão, o que se converte numa nova aprendizagem. [7]

As neural networks por norma consistem em três tipos de camadas: uma camada de input, uma ou mais camadas escondidas e uma camada de output, todas elas ligadas por conexões com um determinado peso. A camada escondida é configurável pelo analista, quer em termos de número de camadas, mas também em quantidade de nós. Porém, A escolha da quantidade de nós pode por vezes ser complicada, porque apesar de uma grande quantidade de nós aumentar a flexibilidade e o poder da rede para identificar certos padrões, isto pode levar a overfitting o que afeta negativamente a qualidade dos resultados. No entanto, caso sejam escolhidos poucos nós para a camada escondida, podemos ter um problema na precisão em treinos, indicando que é necessário ter mais nós. [7]

Um neurónio artificial acaba por ter como função a realização de uma operação a partir de determinados inputs e calcular e enviar o seu resultado. Isto é feito em duas fases: primeiramente calcula-se a soma dos pesos dos valores de entrada e de seguida calcula-se o valor de ativação do neurónio através de uma função de ativação ou de transferência. A soma dos pesos é calculada pelos pesos associados a determinados inputs e com o peso do offset como e visível na seguinte formula. [7]

$$Xx1 * w1 + x2 * w2 + (\neg 1) * \theta$$

Posteriormente, a função de ativação irá verificar se o valor calculado no passo anterior é suficiente para ativar o neurónio, produzindo um output. Isto culmina num processo de aprendizagem através da backpropagation em que, na fase de treino, os valores do output são comparados com os valores reais calculando o valor do erro da previsão. De modo a minimizar os erros de previsão, a rede neuronal altera os pesos anteriores de acordo com o resultado obtido e volta a repetir o processo inicial. [7] Redes neuronais podem ter vários tipos de topologias, todas elas com características diferentes: [7]

- **feed-forward networks** que são redes que apenas têm conexões numa direção, não havendo recursividade;
- **recurrent networks** caso tenham feedback loops;
- **self-organized** caso não tenham um supervisor, tendo algoritmos de aprendizagem autónomos.

Estas estruturas podem ainda ser caracterizadas pelo seu número de camadas sendo que as monocamadas podem apenas resolver problemas de separação linear, quando as redes neuronais de multi-camadas, devido ao seu maior poder, conseguem resolver problemas mais complexos. [7]

D. Cross-Validation

Cross-validation é um método estatístico que é usado de modo a descobrir a eficiência de modelos de machine learning. Consiste em dividir os dados em conjuntos(partes), onde um conjunto é utilizado para treino e outro conjunto é utilizado para teste e avaliação do desempenho do modelo. A utilização do Cross-Validation tem altas chances de detetar se o modelo está em overfitting. [8] Existem diversas técnicas para se fazer a cross-validation, havendo 4 em destaque: [9]

- O metodo **Hold Out**, sendo o mais simples de todos em que se dividem os dados em 2 partes, sendo uma parte de treino e outra de teste. O nosso modelo é treinado com a parte de treino e posteriormente é pedido para prever os dados de teste. De seguida são comparados os resultados da previsão, com os dados verdadeiros.
- O metodo **K-fold cross-validation** que é um método modificado do método holdout, em que os dados são divididos em K subsets, em que K é um valor idealmente compreendido entre 5 e 10, sendo que quanto menor o K, mais semelhantes irão ser os resultados comparativamente ao método hold out. De seguida, treinamos o modelo usando k-1 ”folds”, validando e testando o modelo no K fold restante. Este processo é repetido até que cada K-fold seja usado como teste no modelo.

- O metodo **Repeated K-Fold cross-validation** que consiste em aplicar o método anterior, K-Fold cross-validation mais do que uma vez
- O método **Leave one out cross-validation** em que se maximiza o K-Fold cross-validation, igualando o K ao (N) número de entradas nos nossos dados. Este método é aplicado N vezes, onde se treina o modelo com todos os dados exceto 1, sendo feita uma previsão para o mesmo e testando o modelo. O erro de previsão irá ser a média de todas as estimativas de erro obtidas anteriormente.

Os modelos de regressão também têm a sua performance medida, sendo isto feito através da determinação da precisão do modelo na previsão de dados que não são usados na construção do modelo. Algumas das métricas usadas para quantificar a performance dos modelos de regressão são: [9]

- **R-squared** que representa a correlação quadrada entre os valores previstos e os obtidos, em que quanto maior o R2, melhor o modelo. Regra geral, este é o critério usado em regressão linear.
- **Root Mean Squared Error** que mede o valor médio do erro de previsão do modelo na previsão dos valores, ou seja, a diferença média entre os valores previstos e os valores obtidos. Quanto menor o RMSE, melhor é a performance do modelo.
- **Mean Absolute Error** que é uma alternativa ao Root Mean Squared Error que apresenta a diferença de ser menos sensível a outliers. Corresponde à diferença média absoluta entre os valores previstos e os obtidos, e quanto menor o MAE, melhor a performance do modelo.

E. K-vizinhos-mais-próximos

No ramo da inteligência artificial, alguns dos métodos de classificação mais usados é o dos Nearest Neighbours (NN) e o dos K-vizinhos-mais-próximos (KNN). Estes algoritmos têm como objetivo classificar os dados de uma dada instância com base nos seus vizinhos mais próximos sendo que o algoritmo dos vizinhos mais próximos o faz para K = 1, enquanto que o KNN o faz para os K vizinhos mais próximos. Estes algoritmos são dos mais simples de supervisionar e são dos mais estudados na inteligência artificial. [10]

Os algoritmos de K vizinhos mais próximos acabam por ser algoritmos de aprendizagem preguiçosa, pois estes guardam os dados de exemplo de treino deixando o processamento dos mesmos em ”pausa” até a criação de novas previsões. De modo a fazer uma previsão, o KNN encontra os K vizinhos mais próximos de um determinado ponto e calcula a sua classificação ou a sua regressão com base nos vizinhos. Para executar estes algoritmos é necessário determinar o K, o que por vezes pode não ser fácil, pois caso o seu valor seja demasiado baixo, os resultados podem ser sensíveis a ”barulho” levando a resultados mais instáveis. Pelo outro lado, valores altos de K podem levar a que os vizinhos mais próximos incluam pontos de outras classes ou que a carga computacional aumente. [10] Os algoritmos de vizinhos mais próximos para além de serem preguiçosos, podem ser postos noutras categorias, sendo também não paramétricos e discriminativos. [10] Estes algoritmos são bastante importantes sendo usados em diversas áreas, nomeadamente na interseção entre a classificação de padrões, a visão computacional, reconhecimento de imagem e vídeos e a biometria e em sistemas de recomendação. [10]

II. Regressão

A. Exercício 1

De forma a dar inicio ao estudo dos dados, é primeiramente nos pedido para que carreguemos o ficheiro **Clientes_DataSet.csv** no ambiente R, verifiquemos a sua dimensão e obtenhamos um sumário dos dados. Recorrendo à função **csv** da biblioteca **read**, indicamos a localização do ficheiros e especificamos na variável opcional **head** que a primeira linha contém o nome das colunas. Por fim, armazenamos os dados na variável *Clientes_DataSet*, sobe a forma de um data frame. De seguida, obtemos a dimensão com a função **dim**, onde usamos *Clientes_DataSet* como parâmetro e obtemos os valores de *7043* linhas e *21* colunas. No contexto do sumário, recorrendo à função **summary** obtemos uma analise sobre cada coluna, sendo de destacar que as colunas em que os dados são números, é nos dado o detalhe sobre algumas medidas de localização, como podemos verificar para Fig. 2.

```
> summary(Clientes_DataSet)
  ClienteID          Genero             Maior65       Colaborador
Length:7043        Length:7043        Min.   :0.0000   Length:7043
Class :character   Class :character   1st Qu.:0.0000   Class :character
Mode  :character   Mode  :character   Median :0.0000   Mode  :character
                                      Mean   :0.1621
                                      3rd Qu.:0.0000
                                      Max.   :1.0000

 Dependentes         Fidelização     TipoServiço        LinhasMultiplas
Length:7043        Min.   : 0.00   Length:7043        Length:7043
Class :character   1st Qu.: 9.00   Class :character   Class :character
Mode  :character   Median :29.00   Mode  :character   Mode  :character
                   Mean   :32.37
                   3rd Qu.:55.00
                   Max.   :72.00

 ServiçoInternet    SegurançaOnline    CópiadeSegurançaOnline  ProteçãoTM
Length:7043        Length:7043        Length:7043             Length:7043
Class :character   Class :character   Class :character        Class :character
Mode  :character   Mode  :character   Mode  :character        Mode  :character

 SuporteTécnico     ServiçoStreamingTV ServiçoStreamingFilmes TipodeContrato
Length:7043        Length:7043        Length:7043             Length:7043
Class :character   Class :character   Class :character        Class :character
Mode  :character   Mode  :character   Mode  :character        Mode  :character

 FaturaEletronica   MétododePagamento   TarifaMensal     TotalTarifas       Ativo
Length:7043        Length:7043        Min.   : 18.25   Min.   :  18.8   Length:7043
Class :character   Class :character   1st Qu.: 35.50   1st Qu.: 401.4   Class :character
Mode  :character   Mode  :character   Median : 70.35   Median :1397.5   Mode  :character
                                      Mean   : 64.76   Mean   :2283.3
                                      3rd Qu.: 89.85   3rd Qu.:3794.7
                                      Max.   :118.75   Max.   :8684.8
                                                       NA's   :11
```

Fig. 2. Sumario do ficheiro Clientes_DataSet.csv

B. Exercício 2

Face à dimensão e quantidade de dados que temos, é comum encontrarmos dados com valores que não nos permitam estudá-los, sendo necessária a sua remoção do dataset. O pressuposto do exercício 2 é a execução do pré-processamento dos dados. Para dar início a este exercício, recorremos à biblioteca **na**, mais concretamente à função **omit**, para agilizar a remoção de dados não validos - **NA** - para estudo do nosso data frame *Clientes_DataSet.csv*. Após análise do estado corrente do data frame, identificamos as colunas com valores numéricos que

não fazem sentido semanticamente serem iguais ou inferiores a zero. As colunas identificadas foram: Fidelização, TarifaMensal, TotalTarifas. Relativamente aos outliers, fizemos a sua identificação através da função **boxplot**, recorrendo mais precisamente à propriedade **out** para obter o números de outliers. Nas três colunas não foram identificados outliers. Para concluir o exercício, efetuamos a remoção do *ClienteID*, uma vez que a identificação do cliente não é relevante para o estudo em questão.

C. Exercício 3

Relativamente ao exercício 3, era pedido que se criasse um diagrama de correlação entre todos os atributos.

De facto, este diagrama de correlação é um tipo de exibição de dados que mostra a relação entre duas variáveis numéricas. Assim, e como estes dados têm de ser numéricos, passou-se à observação em detalhe dos dados importados.

Em primeiro lugar, passou-se à atribuição de 0 e 1 aos atributos não numéricos que apenas tinham 2 opções (no caso variavam entre "Sim" e "Não" e "Feminino" e "Masculino"). Em segundo lugar, para os atributos não numéricos com mais de 2 opções, foi utilizada a livraria "fastDummies" com a função "dummy cols" aplicando a técnica "One-Hot Encoding". Assim, para além de variáveis numéricas já existentes, todos os restantes atributos ficaram nesse mesmo tipo. Por fim, alterou-se o nome das colunas dos dados e realizou-se o diagrama de correlação como se verifica na Fig. 3. Devido à enorme quantidade de atributos, o diagrama de correlação segue junto com os scripts em formato pdf de forma a se poder observar melhor.

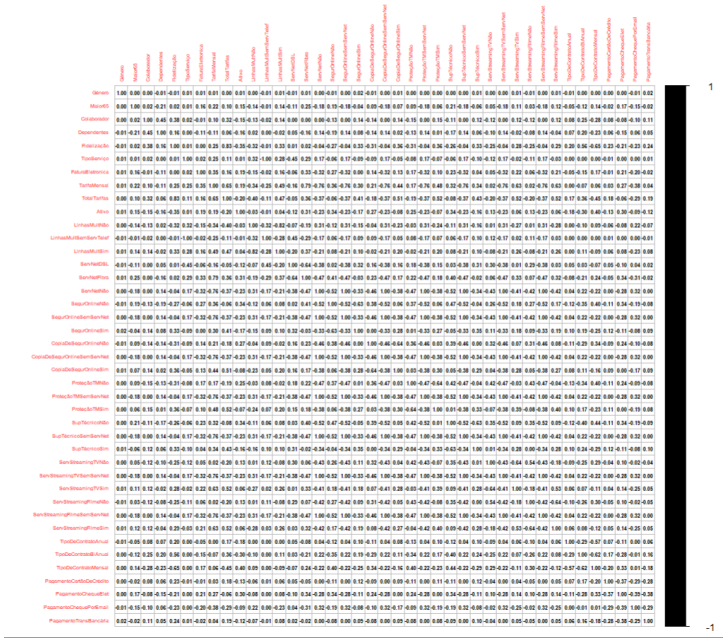

Fig. 3. Diagrama de correlação entre todos os atributos.

D. Exercício 4

No contexto do exercício 4 era necessário criar um modelo de regressão linear simples com o objetivo de determinar o período de "Fidelização" com base na "TarifaMensal" do nosso data frame. Para a criação do modelo, utilizamos a técnica **hold out** para criação de amostras de treino e de teste. As amostras foram criadas a partir do data frame *Clientes_DataSet* com proporções de 70% para treino e 30% para teste. Uma vez obtidas as amostras de treino e de teste, utilizamos a função **lm**, em que especificamos que a "Fidelização" depende da "TarifaMensal" através da sintaxe "Fidelização TarifaMensal" e utilizamos os dados de treino para criação do modelo, dando assim por concluída a alínea a). Para a resolução da alínea b), utilizamos a função **plot** e **abline** para criar o respetivo diagrama de dispersão, assim como a sua reta de regressão linear simples, como podemos ver na Fig. 4. Para finalizar o exercício 4, na alínea c) era pedido o

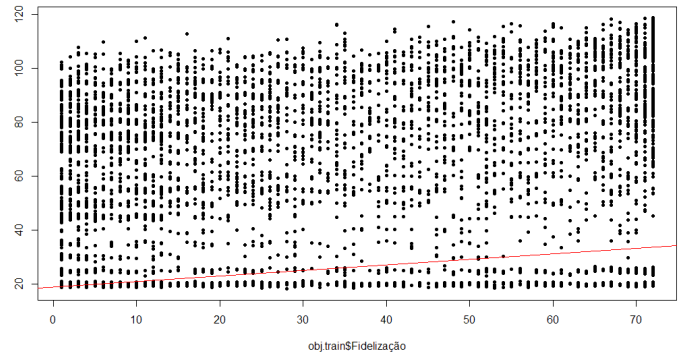

Fig. 4. Visualização do diagrama de dispersão e reta de regressão

calculo do erro médio absoluto (MAE) e da raiz quadrada do erro médio (RMSE). Para tal, determinamos a previsão com o modelo de regressão linear previamente criado, utilizando os dados de testes e obtivemos a diferença entre o resultado do nosso modelo, fase ao valor esperado dos dados de teste.Os resultados obtidos foram: 20.48227 para o MAE e 23.48042 para o RMSE.

E. Exercício 5

Tendo em conta o conjunto de dados em análise, pretendemos prever o Total Tarifas. Para isso, aplicamos as seguintes técnicas:

- Regressão linear múltipla
- Árvore de regressão
- Rede neuronal

Primeiramente aplicamos o método **hold out** para dividir os dados onde 70% são dados de treino e 30% são dados de teste. Na alínea a), utilizamos a função **lm** onde especificamos que o atributo "TotalTarifas" depende de todos os outros atributos e utilizamos os dados de treino para criação do modelo. Posto isto, obtemos os seguintes coeficientes resultantes do modelo:

```
> summary(lmTotalTarifas.5a)$coefficients
                              Estimate  Std. Error      t value      Pr(>|t|)
(Intercept)              -1823.109369  56.8167630 -32.08752616 3.239743e-20
Género                      -8.646159  19.8530930  -0.43550692  6.632137e-0
Maior65                     -3.522464  28.7953670  -0.12232744  9.026447e-0
Colaborador                 24.664138  23.9724096   1.02885518  3.035985e-0
Dependentes                -54.707619  25.4252958  -2.15170041  3.146970e-0
Fidelização                 60.570514   0.6895048  87.84640365  0.000000e+0
TipoServiço               -269.308667 245.7128104  -1.09603023  2.731194e-0
FaturaEletronica            -9.986895  22.2884679  -0.44807454  6.541192e-0
TarifaMensal                45.308768   9.6724481   4.68431233  2.885026e-0
Ativo                     -144.878549  26.2792058  -5.51304898  3.707553e-0
LinhasMultNão              -46.467854  53.8773539  -0.86247470  3.884686e-0
ServNetDSL                 -65.238433 630.7132479  -0.10343596  9.176212e-0
ServNetFibra              -318.300344 872.0743284  -0.36499222  7.151330e-0
SegurOnlineNão            -147.806640  55.3105836  -2.67230302  7.558208e-0
CopiaDeSegurOnlineNão     -226.798832  54.0527283  -4.19588130  2.766227e-0
ProteçãoTMNão             -151.185387  54.4532301  -2.77642643  5.516890e-0
SupTécnicoNão             -121.692807  55.4678498  -2.19393410  2.828696e-0
ServStreamingTVNão           3.381409  99.9551105   0.03382928  9.730147e-0
ServStreamingFilmeNão       11.586799 100.2777443   0.11554707  9.080163e-0
TipoDeContratoAnual        -19.124385  31.3578137  -0.60987624  5.419721e-0
TipoDeContratoBiAnual     -223.064589  37.8416328  -5.89468722  4.005343e-0
PagamentoCartãoDeCrédito    37.895908  29.9385715   1.26578878  2.056489e-0
PagamentoChequeElet        -52.568948  29.4111288  -1.78738287  7.393744e-0
PagamentoChequePorEmail    220.547049  32.0527296   6.88075718  6.702151e-1
```

Fig. 5. Resumo de coeficientes do modelo de regressão múltipla.

Na alínea b), utilizamos a função **rpart** com o método "Anova" para obter a árvore de regressão utilizando o mesmo dataset.

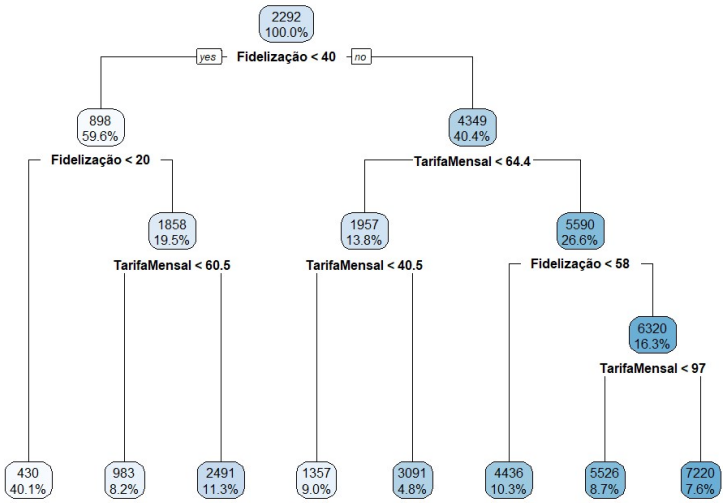

Fig. 6. Árvore de regressão.

Na alínea c), utilizamos a função **neuralnet** e o dataset das alíneas anteriores, para gerar uma rede neuronal. Uma vez que pretendemos relacionar um atributo com todos os outros, optamos por colocar o número de nodes igual a 1. Foi testado pela equipa colocar o número de nodes como c(12, 6, 3). Verificou-se que, apesar do valor do mae nao ter diferido muito para apenas 1 nó, o valor do rmse baixou significativamente. Contudo, o grupo optou por utilizar 1 nó. Assim sendo, obtemos a seguinte rede neuronal:

F. Exercício 6

Após obtermos os modelos no exercicio 5, pretendemos efetuar uma comparação de resultados onde iremos analisar o erro médio absoluto MAE e a raiz quadrada do erro médio (RMSE) para cada um dos modelos. Recorremos ao método k-fold cross validation, onde analisamos 10 iterações diferentes

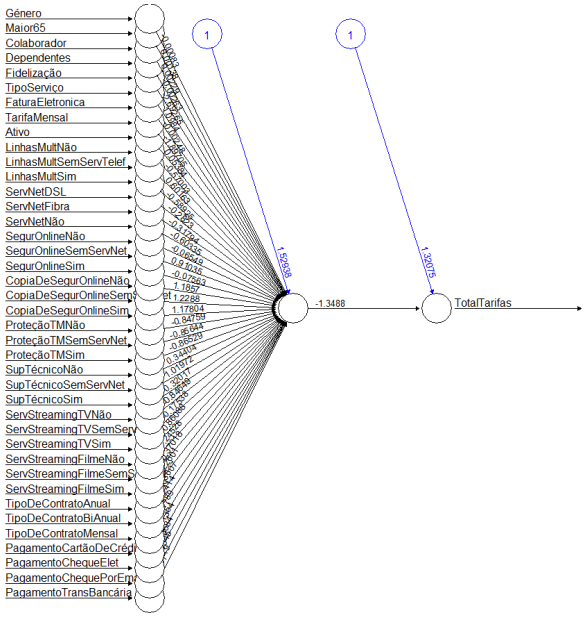

Fig. 7. Rede Neuronal

dos 3 modelos e calculamos o MAE e o RMSE obtendo os valores da seguinte tabela:

TABLE I
VALORES MAE E RMSE PARA CADA MODELO

	mean	**sd**
Regressão Linear Múltipla	689.7435	14.988614
Árvore de regressão	577.3373	23.622812
Rede Neuronal	274.6485	6.292916

Analisando os dados obtidos, verificamos que a árvore de regressão e a rede neuronal são os dois melhores modelos.

G. Exercício 7

Observamos pelo exercício 6, através da média, que os melhores modelos são a árvore de regressão e a rede neuronal (1). Contudo, iremos comprovar estatisticamente através dos testes: t.test e wilcox.

Após obtermos os dois melhores modelos, pretendemos verificar se são estatisticamente significativos e para isso recorremos a um **t.test** e um **wilcox.test**. Utilizando um nível de significância de 5% e analisando o p-value obtido em cada um dos testes, concluímos que existe uma diferença significativa entre os dois melhores modelos para ambos os testes.

TABLE II
TABELA COM VALORES P-VALUE PARA O T.TEST E WILCOX.TEST

	p-value
t.test	1.613e-12
wilcox.test	1.083e-05

Verificamos que o modelo com melhor desempenho é a rede neuronal de acordo com os valores MAE e RMSE e, através

deste exercício, que esta rede é significativamente diferente da árvore de regressão.

III. Classificação

A. Exercício 8

Relativamente ao exercício 8, era pedido que se estudasse a capacidade preditiva relativamente ao atributo Ativo usando os métodos:

- Árvore de Decisão
- Rede Neuronal
- K-vizinhos-mais-próximos

Relativamente à árvore de decisão, para a criação do modelo, utilizou-se a técnica **hold out** para criação de amostras de treino e de teste. As amostras foram criadas a partir do data frame Clientes DataSet com proporções de 70% para treino e 30% para teste. Após isso utilizou-se a função **rpart** para se fazer a árvore utilizando o atributo "Ativo" com a sintaxe "Ativo ." uma vez que o ponto final representa todos os atributos. De seguida fez-se o plot desta mesma árvore como se observa na Fig. 8. Por fim, fez-se a previsão com a árvore e os dados de teste.

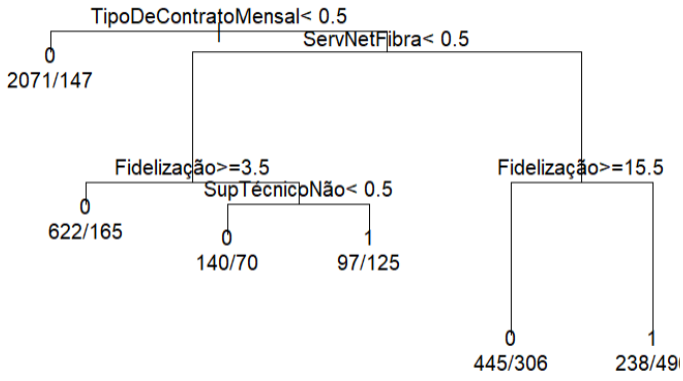

Fig. 8. Árvore de decisão relativamente ao atributo Ativo.

Relativamente à rede neuronal, utilizamos a técnica **hold out** para criação de amostras de treino e de teste através dos dados normalizados da mesma forma que na alínea acima descrita. Esta rede neuronal foi testada com números nós diferentes, estando atualmente a ser utilizado com c(12, 6, 3). Testou-se apenas com 1 nó e verificou-se que a accuracy era bastante menor por isso optou-se pelo número de nós referido anteriormente. De seguida utilizou-se a função **neuralnet** com o atributo "Ativo" com a sintaxe "Ativo ." e fez-se o sumário dos resultados ao invocar a "result_matrix". Por fim, criou-se um data.frame com os dados de teste excluindo o atributo a ser estudado ("Ativo") e fez-se a previsão utilizando a função **compute**. Ainda assim, para se avaliar o modelo desnormalizaram-se os dados e calculou-se a raiz quadrada do erro médio (RMSE) obtendo o resultado **0.5027684**.

Por fim, e quanto aos K-vizinhos-mais-próximos, realizou-se o **hold out** e criaram-se matrizes de exemplos de teste e treino juntamente com um vetor com as respostas de cada observação do conjunto de treino. Após isso realizou-se a previsão do atributo "Ativo" utilizando-se o **KNN para a regressão** que retorna o valor médio dos k vizinhos mais próximos. No final retirou-se o k mínimo e máximo que deram **45** e **47** respetivamente e observaram-se os seguintes plots nas Fig. 9 e 10.

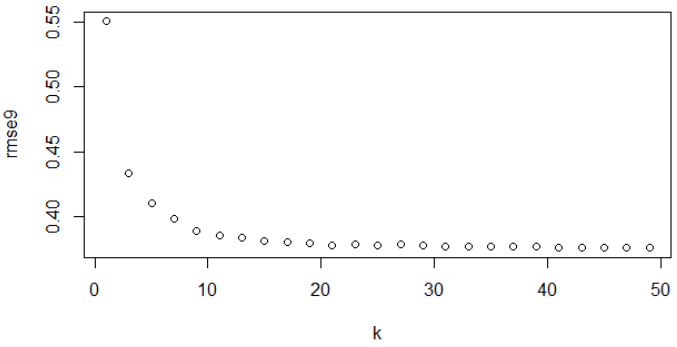

Fig. 9. Plot do atributo Ativo através do método K-vizinhos-mais-próximos para o valor mínimo

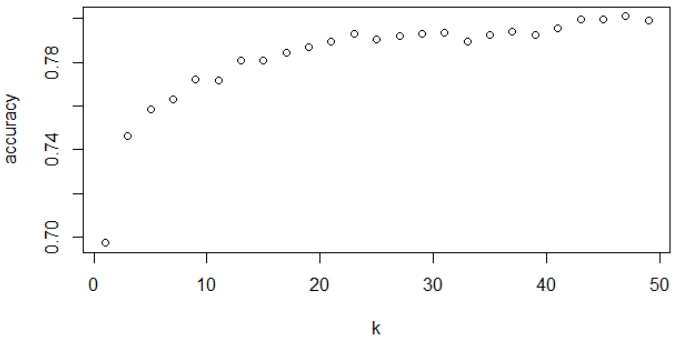

Fig. 10. Plot do atributo Ativo através do método K-vizinhos-mais-próximos para o valor máximo

B. Exercício 9

Utilizando o método k-fold, foi-nos solicitado obter a média e o desvio padrão da taxa de acerto da previsão do atributo **Ativo** com os dois melhores modelos obtidos na alínea anterior. Para obter os 2 melhores modelos, verificamos a **Accuracy** de cada modelo obtendo os seguintes valores:

Analisando os valores obtidos, consideramos que o modelo **Árvore de Decisão** e o modelo **K-vizinhos mais próximos** são os melhores porque apresentam uma **accuracy** mais perto de 100%. Realizamos 10 iterações diferentes para os dois modelos e calculamos a média e o desvio-padrão obtendo os valores da seguinte tabela:

TABLE III
COMPARAÇÃO DE ACCURACY PARA OS 3 MODELOS

	Accuracy
Árvore de Decisão	80.66%
Rede Neuronal	64%
K-vizinhos mais próximos	80.09%

TABLE IV
10 ITERAÇÕES PARA OBTER ACCURACY DOS DOIS MELHORES MODELOS

	KNN	**Árvore de Decisão**
1	0.7718579	0.7950820
2	0.7938719	0.8105850
3	0.7786999	0.7883817
4	0.8029412	0.7882353
5	0.7701149	0.7701149
6	0.7851740	0.7776097
7	0.7747489	0.7718795
8	0.7896213	0.7938289
9	0.7971223	0.8000000
[0	0.7852162	0.7880056

C. Exercício 10

Observamos pelo exercício anterior são a árvore de regressão e os K-vizinhos-mais-próximos. Contudo, iremos comprovar estatisticamente através dos testes: t.test e wilcox

Após obtermos os dois melhores modelos, pretendemos verificar se são estatisticamente significativos e para isso recorremos a um **t.test** e um **wilcox.test**. Utilizando um nível de significância de 5% e analisando o p-value obtido em cada um dos testes, concluímos que não existe uma diferença significativa entre os dois melhores modelos para ambos os testes.

Concluímos que a árvore de regressão e os K-vizinhos-mais-próximos são significativamente iguais. Para isso, utlizamos um boxplot e verificamos que a técnica da árvore de decisão tem, ligeiramente, um melhor desempenho como se pode ver na Fig 11.

D. Exercício 11

Para finalizar o trabalho, no exercício 11 é nos pedido para efetuar uma comparação dos modelos, árvore de decisão, rede neuronal e K-vizinhos-mais-próximos. A comparação é feita utilizando os critérios de *Accuracy*, *Sensitivity*, *Specificity* e *F1*. Nesse sentido é necessário a obtenção de uma matriz de confusão para cada um dos modelos. A matriz de confusão tem como objetivo permitir quantificar quantos verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos um determinado modelo gerou quando comparado com os dados de teste. A nossa abordagem para a obtenção de um resultado mais preciso foi a de fazer dez iterações, onde em cada iteração utilizamos a função **sample** para recolher uma amostra da nossa população e sobre essa amostra utilizamos a técnica **hold out** para criar os nossos dados de treino e de testes. De seguida, criamos cada um dos três modelos com os dados de treino, comparamos com os dados de testes, obtemos a matriz de confusão e utilização a matriz de confusão calculamos o resultado para cada medida de desempenho. Os resultados das medidas de desempenho são armazenados numa lista, para no fim fazermos uma média de todos os valores obtidos. Os resultados das médias das medidas de desempenho podem ser observados na tabela VII.

TABLE V
VALORES MÉDIA E DESVIO PADRÃO PARA CADA MODELO

	Média	**Desvio-Padrão**
Árvore de Decisão	0.7884	0.0126
K-vizinhos mais próximos	0.7849	0.0111

TABLE VI
TABELA COM VALORES P-VALUE PARA O T.TEST E WILCOX.TEST

	p-value
t.test	0.5255
wilcox.test	0.5966

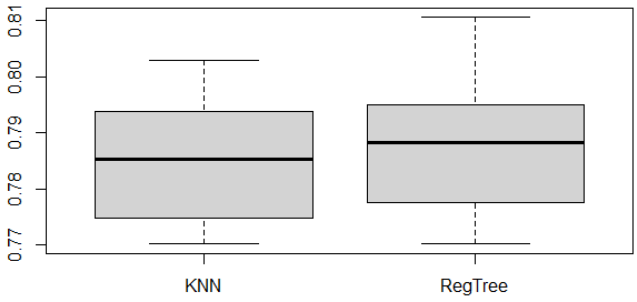

Fig. 11. Boxplot com o desempenho da técnica KNN e Árvore de Regressão

TABLE VII
RESULTADO DAS MEDIDAS DE DESEMPENHO DOS MODELOS.

Medida	***Árvore de decisão***	***Rede neuronal***	***K-vizinhos***
Accuracy	78.72%	63.19%	88.28%
Specificity	0.438%	0.786%	0.958%
Sensitivity	0.915%	0.209%	0.605%
F1	0.0371	0.0119	0.0468

Com os resultados obtidos, podemos determinar que o modelo mais preciso é o K-vizinhos-mais-próximos, seguido pela Árvore de decisão, ficando a Rede neuronal por ultimo no que diz respeito a Accuracy. No que diz respeito a verdadeiros negativos - Specificity, o K-vizinhos-mais-próximos mostrou melhor resultado, seguido da rede neuronal, ficando a Árvore de decisão com a pior previsão. Já para a Sensitivity - identificação dos falso negativos, a Árvore de decisão teve mais sucesso na sua identificação, seguido pelos K-vizinhos-mais-próximos, sendo a rede neuronal a que pior resultado

teve. Por ultimo, a medida F1, ou a harmonia entre os erros cometidos, sejam eles negativos ou positivos, a Rede neuronal mostrou obter um melhor resultado, seguido pela Árvore de decisão e pelos K-vizinhos-mais-próximos.

IV. Conclusões

Recorrendo aos dados fornecidos no contexto de uma operadora telefónica, mais concretamente o registo de serviços ativos, fidelizações, tarifas etc, foi-nos possível desenvolver uma análise dos dados, tendo como base os conceitos teóricos lecionados na unidade curricular. Após a análise, concluímos que:

- Recorrendo aos dados fornecidos no contexto de uma operadora telefónica, mais concretamente o registo de serviços ativos, fidelizações, tarifas etc, foi-nos possível desenvolver uma análise dos dados, tendo como base os conceitos teóricos lecionados na unidade curricular. A utilização destes dados permitiu-nos então o estudo de algoritmos mais avançados, nomeadamente algoritmos de regressão linear simples, regressão linear múltipla, árvores de decisão, k vizinhos mais próximos e redes neuronais, sendo este algoritmos divididos em duas grandes áreas, Regressão e Classificação.
- Na primeira parte (Regressão) do trabalho foi-nos possível perceber de maneira concreta a correlação que existe entre variáveis, nomeadamente no exercício 3 e 4, onde neste ultimo conseguimos observar a reta de regressão da tarifa fase à fidelização.
- Do exercício 5 a 7 aplicamos algoritmos de regressão linear múltipla, árvore de regressão e de rede neuronal, onde utilizamos a técnica de hold out para criar os nossos modelos com dados de treino e validamos o erro nos nossos modelos com dados de testes, percebendo assim o quão eficaz os modelos são. Concluímos que as técnicas da árvore de regressão e rede neuronal são as mais eficazes. Por fim, recorrendo à média, conseguimos então distinguir os dois melhores modelos, validando no final com testes de hipóteses (t.test e wilcox.test). Reparamos que existe uma diferença significativa entre a árvore de regressão e a rede neuronal, sendo esta útlima a técnica mais eficaz como verificamos através dos valores MAE e RMSE. Também concluímos que, uma rede neuronal com mais nós, apesar do tempo de criação, tem valores de MAE e RMSE menores. Contudo, o grupo optou por utilizar apenas 1 nó.
- Para finalizar, na última parte do trabalho (Classificação), o estudo foi efetuado ainda com a técnica de hold out, mas com o foco desta vez em árvore de decisão, rede neuronal e k vizinhos mais próximos, onde procurávamos avaliar as medidas de desempenho dos diferentes modelos. Concluímos que as técnicas da árvore de decisão e K-vizinhos-mais-próximos tiveram uma maior percentagem de accuracy. Novamente recorrendo à média, distinguimos os dois melhores modelos com testes de hipóteses (t.test e wilcox.test). Contudo, ao contrário do exercício 7, estes 2 modelos não tinham uma diferença significativa. Para uma melhor observação, fizemos um boxplot e concluímos que a técnica da árvore de decisão teve um desempenho ligeiramente superior aos K-vizinhos-mais-próximos.
- Por ultimo, no exercício 11, fizemos uma avaliação das medidas de desempenho, a qual é bastante demorada dado a dimensão dos dados e a complexidade algorítmica, sobre a Accuracy, Sensitivity, Specificity e F1 a todos os modelos, onde iterando dez vezes com amostras diferentes da mesmo população, criando assim vetores de que nos permitiram no fiz fazer uma média de todos os resultados para as medidas de desempenho. Com estas medidas é nos possível no fim fazer uma escolha do melhor algoritmo mediamente o que seja mais crítico para nós.

Conclusão sobre a perda de clientes:

- O grupo considerou que a perda abrupta de clientes se deveu à qualidade de serviço não ser a desejada.
- Como se observa no diagrama de correlação, os serviços prestados pela operadora estão relacionados com a fidelização e com o total de tarifas. Por sua vez, o total de tarifas também está relacionado com a fidelização.
- Assim, o grupo supõe que a queda abrupta de clientes se deve à não renovação da fidelização.

References

[1] https://pt.wikipedia.org/wiki/Regress
[2] "Analise de Regressão Linear Múltipla I Introdução," 2011.
[3] Vitor Vieira Vasconcelos Professor Seguir, "Regressao Linear I," SlideShare. https://pt.slideshare.net/vitor_vasconcelos/regresso-linear-i
[4] "Arvore de regressão," IBM. https://www.ibm.com/docs/pt-br/cognos-analytics/11.1.0?topic=tests-regression-tree
[5] https://moodle.isep.ipp.pt/pluginfile.php/194083/mod_resource/content/6/Decision
[6] "What is a Decision Tree Diagram," Lucidchart. https://www.lucidchart.com/pages/decision-tree.
[7] https://moodle.isep.ipp.pt/pluginfile.php/194084/mod_resource/content/6/Neural
[8] https://medium.com/@edubrazrabello/cross-validation-avaliando-seu-modelo-de-machine-learning-1fb70df15b78
[9] https://moodle.isep.ipp.pt/pluginfile.php/194086/mod_resource/content/3/Cross-validation.pdf
[10] https://moodle.isep.ipp.pt/pluginfile.php/194087/mod_resource/content/6/kNN