INSTITUTO UNIVERSITÁRIO DE LISBOA

HONG KONG

PREVISÃO DO SCORE ATRIBUÍDO POR REVIEWERS

Mestrado em Ciência de Dados

Ano letivo 2021/2022 | MTCD-MP

Docentes:

- Anabela Costa
- Diana Mendes
- Sérgio Moro

5 de dezembro de 2021

Grupo 5:

Guilherme Mendonça	nº82575	email: gasvm@iscte-iul.pt
Marta Neves	nº 88660	email: msmns@iscte-iul.pt
René Porto	nº 101597	email: rapfr@iscte-iul.pt

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

1. SUMÁRIO EXECUTIVO

De acordo os Serviços de Turismo, 55.9 milhões de pessoas visitaram Hong-Kong, em 2019, sendo a cidade mais visitada do mundo. Uma vez que o turismo tem crescido de forma exponencial, é necessário que as plataformas digitais de arrendamento de alojamentos, apresentem uma oferta diversificada, tendo em conta o perfil do público-alvo. Cada vez mais, os consumidores apresentam uma escolha mais seletiva e é necessário que o *Airbnb* conheça o seu *target*, de modo a estar a par das suas preferências, com o intuito de aumentar a sua quota de mercado. Do ponto de vista estratégico, posicionar o *Airbnb* no mercado, significa diferenciação perante os concorrentes, e conquistar um lugar de destaque entre os seus consumidores.

Este projeto visa solucionar este problema, através da previsão do *score* atribuído por *reviewers*, em unidades de alojamento de Hong-Kong. Primeiramente, decidiu-se analisar quais os fatores que poderiam afetar a decisão dos consumidores na escolha das acomodações do *Airbnb*. Com esta análise, conseguir-se-á modificar a oferta tendo em conta a curva da procura e, consequentemente, conquistar tanto a fidelização dos clientes como um aumento de receitas.

Através do coeficiente de correlação de *Pearson*, excluíram-se as variáveis que apresentaram uma forte correlação linear, visto que estas poderiam contribuir com informação redundante. De seguida, para atender à necessidade do *Airbnb*, foram desenvolvidos diversos modelos de previsão, sendo que uns traduziram melhores resultados do que outros. O modelo que evidenciou ser o melhor, foi o da Rede Neuronal Artificial conseguindo obter um RMSE de 0.1178.

2. INTRODUÇÃO

O presente projeto foi proposto pelos docentes das Unidades Curriculares de Metodologias e Tecnologias para Ciência de Dados e Métodos de Previsão. O trabalho envolve o desenvolvimento de um modelo estatístico preditivo da avaliação (score) de unidades de alojamento local de Hong-Kong, atribuída por utilizadores da plataforma digital Airbnb.

O desenvolvimento deste projeto teve a Metodologia CRISP-DM como *guideline*, apresentada na página posterior, a qual servirá de apoio desde o conhecimento do problema até à identificação e descrição dos melhores resultados previstos obtidos.

No decorrer das páginas, para além de serem descritos todos os passos realizados, ir-se-á refletir criticamente sobre os mesmos, incluindo a extração dos dados, a sua preparação, o desenvolvimento e avaliação de três modelos estatísticos, capazes de refletir bons resultados. São eles a Regressão Linear Múltipla, Árvores de Decisão e Redes Neuronais. No final, ir-se-á fazer uma análise e comparação entre os três, para se concluir qual o modelo mais adequado à solução do problema.

Para auxiliar na conceção das etapas do trabalho, iremos recorrer ao uso da linguagem de programação R, que para além de nos permitir desenvolver o leque de modelos em estudo, também nos irá ajudar a tomar as decisões mediante a visualização dos resultados obtidos, seja através de gráficos ou tabelas.

Este relatório é uma compilação dos nossos conhecimentos adquiridos ao longo do primeiro semestre do Mestrado em Ciência de Dados, tendo sido utilizados na sua realização informações factuais que nos permitiram tratar a informação recolhida e fundamentar devidamente as nossas decisões.

3. METODOLOGIA CRISP - DM

A metodologia CRISP-DM ajuda as organizações a tomarem decisões mais assertivas, bem como a escolher os melhores investimentos. Esta metodologia é constituída por seis fases, cuja sequência se cruza com as necessidades do projeto e de quem o desenvolve. Estas etapas vão ser abordadas ao longo deste trabalho, bem como indicadas na figura abaixo indicada. É de salientar que, durante todas as fases do projeto, manteve-se todas as versões dos documentos e *scripts*, utilizando a ferramenta nativa do *Google Drive*.

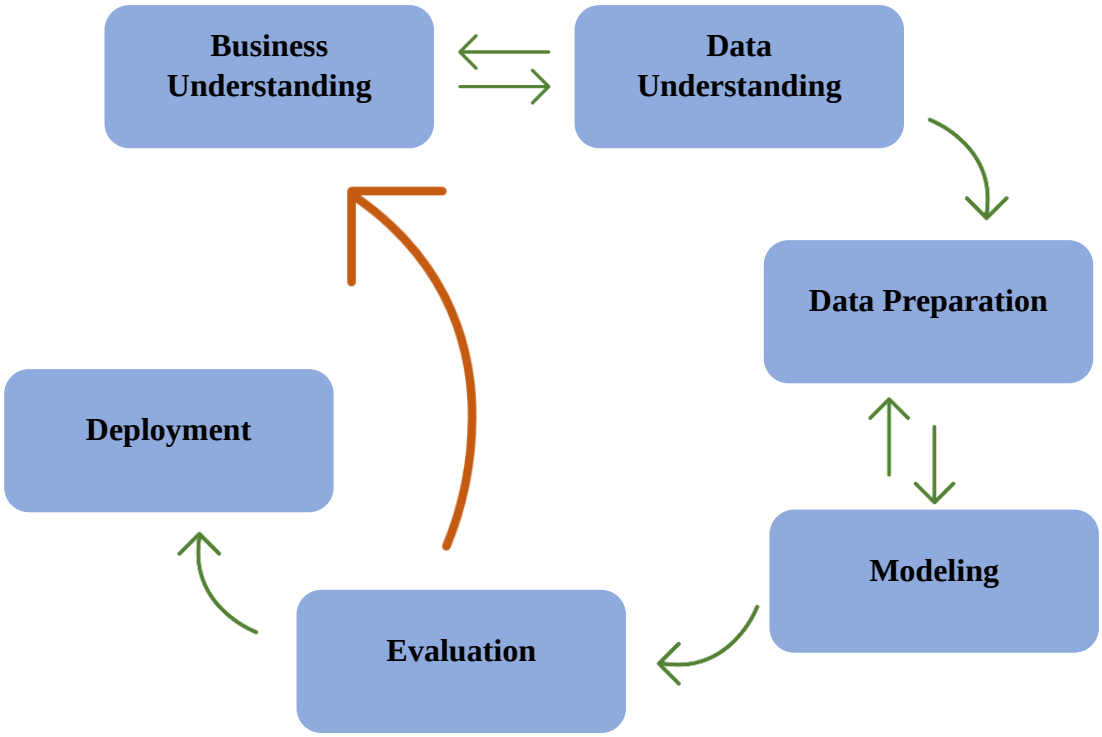

Figura 1 - Metodologia CRISP - DM.

3.4 Business Understanding

Como cientistas de dados, é fulcral perceber o tipo de negócio e setor que a empresa em estudo se encontra. É nesta fase, que são analisados tanto os objetivos, como os requisitos do projeto, tendo em conta as necessidades dos clientes.

3.4.1 Definição dos Objetivos do Negócio

O *Airbnb* consiste numa plataforma digital que apresenta, em diversos sítios do mundo, opções de estadias de curta-duração e experiências. O modelo de negócio, para além de diversificar a oferta do consumidor, obtém receita através de uma comissão dinâmica. Esta pode ser dividida, sendo cobrada ao anfitrião e ao hóspede ou, em alternativa, apenas ao anfitrião.

Quanto maior for o valor da reserva, mais apelativo se torna para o negócio, uma vez que a fatia total da comissão cobrada fica mais elevada, obtendo assim, uma percentagem mais eminente de receitas.

3.4.2 Avaliação Detalhada da Situação

O crescimento exponencial do *Airbnb* deveu-se principalmente a dois motivos: a crise financeira mundial de 2007, que levou a que turistas tivessem maior preocupação em encontrar alojamentos a um preço acessível, e a revolução digital, que permitiu responder a esta necessidade através da apresentação das várias opções disponíveis e comparação entre os preços das mesmas. O *Airbnb* apareceu e rapidamente cresceu, tendo-se não só impondo, como dominado o mercado onde se inseriu.

A plataforma permite, entre outros, fazer a comparação entre os preços dos diferentes alojamentos, efetuar a reserva, fazer o pagamento e, ainda, dispõe do serviço de classificação, tanto do anfitrião como do utilizador, algo que tem elevada importância na sociedade tecnologicamente industrializada, em que se vive hoje em dia.

Para que o negócio do *Airbnb* seja o mais lucrativo possível, a plataforma tem de garantir opções de qualidade aos consumidores, e por efeito de cadeia, isso irá garantir que mais consumidores optem por esta plataforma.

Quanto melhor for a avaliação realizada pelo consumidor, melhor será para o negócio, ajudando, assim, os demais a tomarem uma decisão de reserva, levando estes a recomendarem, ou não, o alojamento local. Por consequência, é necessário desenvolver um modelo o mais objetivo possível, que permita intervir em situações em que as acomodações, se encontrem abaixo do score médio atribuído por *reviewers* causando uma perda de receitas potenciais tanto para o anfitrião, como para o *Airbnb*.

3.4.3 Definição dos Objetivos Técnicos

O objetivo técnico deste trabalho, consiste em desenvolver um sistema de avaliação do score médio atribuído por *reviewers*, através das informações que se encontram no website do *Airbnb*. Para além de serem visíveis para todos os consumidores, são responsáveis pela classificação realizada pelos hóspedes.

Com a implementação deste modelo, conseguir-se-á proporcionar ao *Airbnb* uma oportunidade para estes adaptarem a sua oferta, aproximando-as daquilo que os clientes desejam encontrar na sua estadia, encontrando um equilíbrio entre a procura e a oferta.

3.4.4 Construção do Plano de Projeto

Para a construção do Plano de Projeto, é pretendido alcançar um modelo estatístico que aceite informações sobre as diversas variáveis em estudo, sendo disponibilizadas pela plataforma, com o objetivo final de prever o *score* atribuído por *reviewers*, em unidades de alojamento local.

Numa fase inicial, é crucial selecionar as variáveis a considerar para a estimação do modelo, sendo estas posteriormente tratadas para que o modelo obtido não capte resultados enviesados.

Para proporcionar ao *Airbnb* uma maior vantagem competitiva no mercado, ir-se-á testar vários modelos, com a finalidade de alcançar o melhor resultado possível, que permitirá conhecer o perfil do *target*.

3.5 Data Understanding

Para a realização desta fase, foi necessário recolher e analisar os dados tendo em vista, o alcance do objetivo final. Por conseguinte, foi realizado o *web scraping* do *Website* do *Airbnb*, na cidade de Hong -Kong, para o ano de 2022, extraindo um total de 17112 observações. Para elaborar esta fase do projeto, recorreu-se às bibliotecas *RSelenium*, *rvest* e *stringr*.

3.5.1 Criação dos URL's

Para a criação dos URL's de pesquisa, onde se irá concatenar as diferentes variáveis, tais como o *check-in*, o *check-out*, o preço e o número de hóspedes, sendo que cada uma delas irá gerar aproximadamente 300 acomodações.

Primeiramente, definiu-se que o primeiro *check-in* ocorreria no dia 1 de janeiro de 2022, e que o *check-out* aconteceria após 5 dias, e assim de forma sucessiva, num ciclo de 365 dias.

Em relação ao preço, para começar foi necessário estipular que o preço mínimo seria um euro, e para o máximo cem euros, formando um ciclo sendo que o próximo preço mínimo seria (i+1) e máximo (i+100€), conforme a Figura 2.

Preço Mínimo	Preço Máximo
1	100
101	200
201	300
301	400
401	500
501	600
601	

Tabela 1 - Preço mínimo e máximo.

No que se refere, ao número de hóspedes, limitou-se aos adultos, porque quando se incluía as crianças, gerava um número grande de valores duplicados, e demorava muito a correr.

3.5.2 Procura de Alojamentos

Para a procura de alojamentos, foi necessária a criação de um *script* que percorresse os URL's gerados, página a página, e extraísse todos os resultados existentes.

Após a extração de todos os alojamentos, existe uma elevada probabilidade de estes se repetirem, por inúmeras razões. A título exemplificativo, um alojamento que

apareça com o número de hóspedes selecionado, poderá encontrar-se, simultaneamente, na mesma gama de preços e, ainda, ter vagas disponíveis em épocas ou datas distintas.

Por este motivo, é fulcral eliminar todos os resultados duplicados para não ficarmos com informação redundante, pois poderá afetar em demasia o modelo que se pretende aplicar, dado existir um número elevado de repetições.

3.5.3 Extração de Variáveis

Após a compreensão da variável a prever, e durante a análise da página do *Airbnb*, foi possível avaliar os fatores que poderiam explicar o *score* atribuído por *reviewers*. Por essa razão, estudou-se tanto as variáveis impostas, como algumas adicionais com o objetivo de obter um modelo mais otimizado.

Variáveis obrigatórias:

- ***score:*** Score médio de *reviews* atribuído por hóspedes
- ***guest:*** Nº de hóspedes:
- ***property_type:*** Tipo de acomodação
  - *condo:* condomínio
  - *private room:* quarto privado
  -*serviced apartment:* Apartamento para curta-duração, incluindo serviços
  - *shared room*: quarto partlhado
  - *home:* casa
  - *loft:* sótão
- ***bed:*** Nº de camas
- ***bath_private:*** Casa de banho Variável Binária (Sim 1, Não 0)
- ***kitchen:*** Cozinha - Variável Binária (Sim 1, Não 0)
- ***nreviews:*** Nº de reviews
- ***superhost:*** Badge de "Super Anfitrião": Variável Binária (Sim 1, Não 0)

Variáveis Adicionais:

- ***wifi:*** Variável Binária (Sim 1, Não 0)
- ***elevator:*** Elevador - Variável Binária (Sim 1, Não 0)
- ***balcony:*** Varanda - Variável Binária (Sim 1, Não 0)

- ***pet:*** Animais - Variável Binária (Sim 1, Não 0)
- ***breakfast:*** Pequeno-almoço - Variável Binária (Sim 1, Não 0)
- ***free_cancel:*** Cancelamento gratuito - Variável Binária (Sim 1, Não 0)
- ***studio:*** T0 - Variável Binária (Sim 1, Não 0)
- ***smoking: Fumar*** - Variável Binária (Sim 1, Não 0)
- ***nrbath:*** Quantidade de casas de banho
- Entre outras.

Seguidamente, gerou-se diagramas de extremos e quartis para todas as variáveis, para perceber as posições dos quartis e verificar a existência de valores muito distantes – *outliers*. De forma complementar, decidiu-se não observar as variáveis binárias, uma vez que estas tornam os gráficos pouco úteis de analisar.

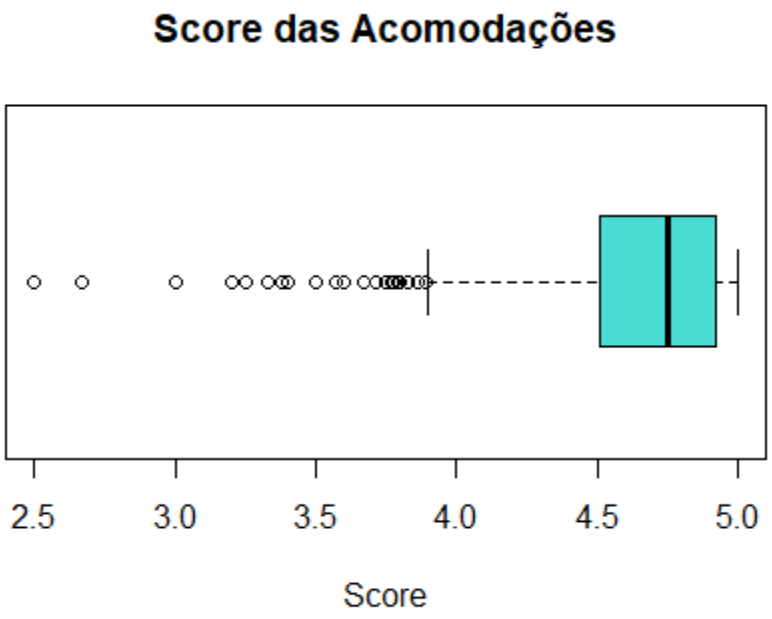

Figura 2 – Diagrama de extremos e quartis para a variável score.

Relativamente à variável *score*, percebe-se através do diagrama acima que existem muitos *outliers* inferiores, contudo, ao analisar os dados optou-se por não os excluir visto que, esta é a variável que se pretende prever. Neste caso, consegue-se perceber que existe um enviesamento à esquerda, isto é, os dados estão mais dispersos, ou seja, menos concentrados na parte inferior do que na parte superior.

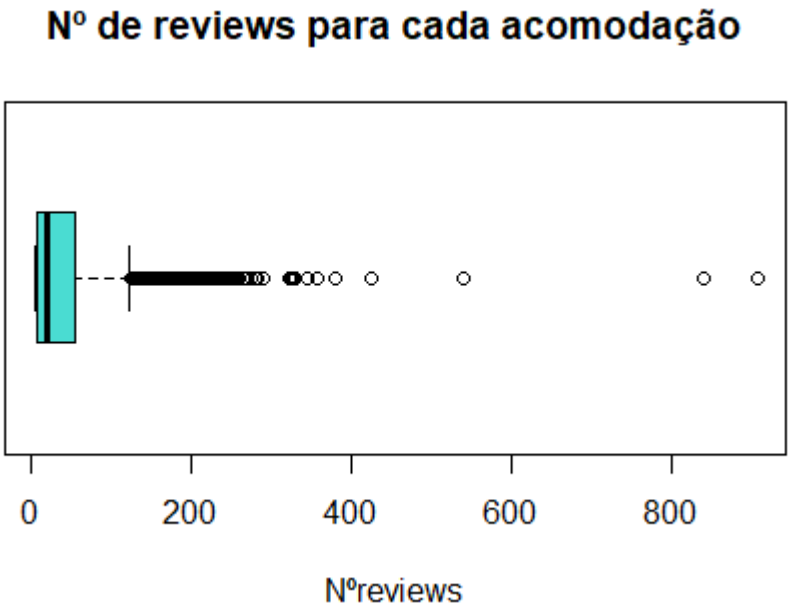

Figura 3 - Diagrama de extremos e quartis para a variável nreviews.

Relativamente à variável número de *reviews* escritas por usuários, observa-se um enviesamento à direita e muitos *outliers* superiores, nomeadamente valores acima de 150 que foram tratados, na fase de *Data Preparation.* Para além disso, verifica-se um enviesamento à direita, uma vez que a mediana se encontra mais próxima do primeiro quartil, do que do terceiro. Adicionalmente, o valor do terceiro quartil corresponde a 53, significando que o número de *reviews* de 75% das acomodações recolhidas é no máximo de 53 *reviews*, e o mínimo encontrado correspondia a apenas três.

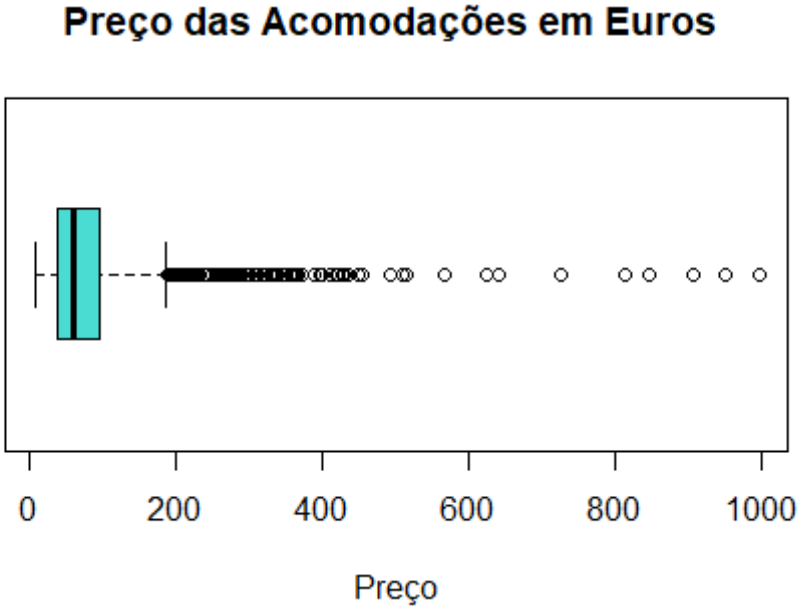

Figura 4 - Diagrama de extremos e quartis para a variável price.

De acordo com o diagrama da variável preço, verifica-se um enviesamento à direita, apresentado 199 *outliers* superiores. Adicionalmente, os resíduos desta variável aparecem quando as acomodações apresentam um valor superior a 200€, que se tratou na etapa de *Data Preparation*. De forma complementar, observou-se que o preço médio era de 83€.

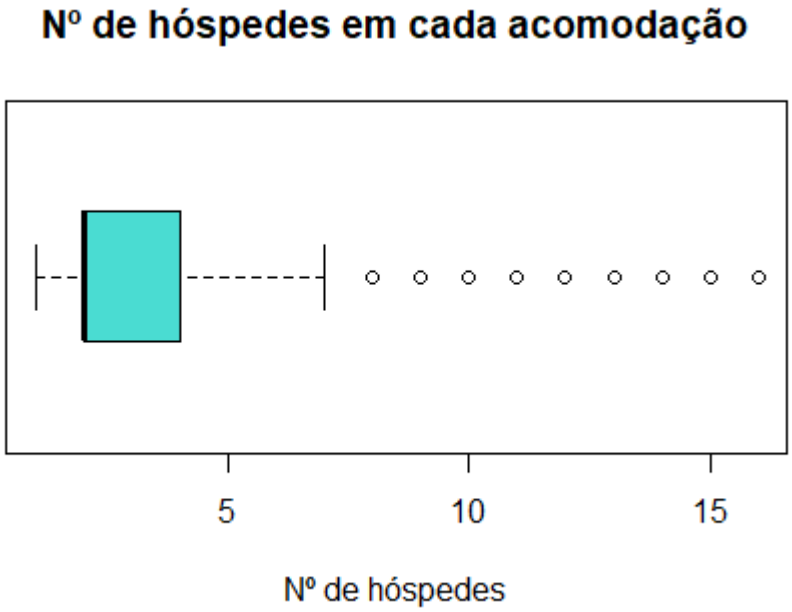

Figura 5 - Diagrama de extremos e quartis para a variável guest.

A variável número de hóspedes apenas assume valores entre um e quinze, inclusive, pelo que através da Figura 5, se percebe onde se enquadra a grande maioria das observações. Com este diagrama, verifica-se que o primeiro quartil e a mediana são iguais e correspondem a dois, isto significa que 50% do número de hóspedes das acomodações recolhidas, é no máximo para duas pessoas. Além disso, acomodações que permitem a partir de 7 pessoas, são consideradas isoladas, contudo não se retirou das observações por se tratar de uma variável discreta com valores baixos.

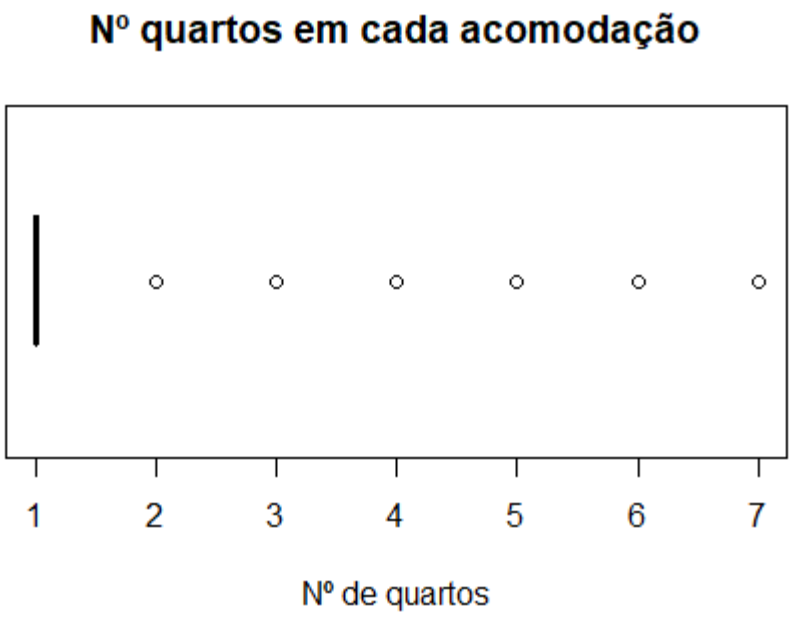

Figura 6 - Diagrama de extremos e quartis para a variável bedroom.

Ao observar o gráfico referente à variável número de quartos em cada acomodação, percebe-se que tudo o que seja superior a um é considerado *outlier* superior, isto porque a maior parte das observações se concentram neste valor. Por se tratar de uma variável discreta, compreendida entre um e sete, decidiu-se permanecer todas as observações.

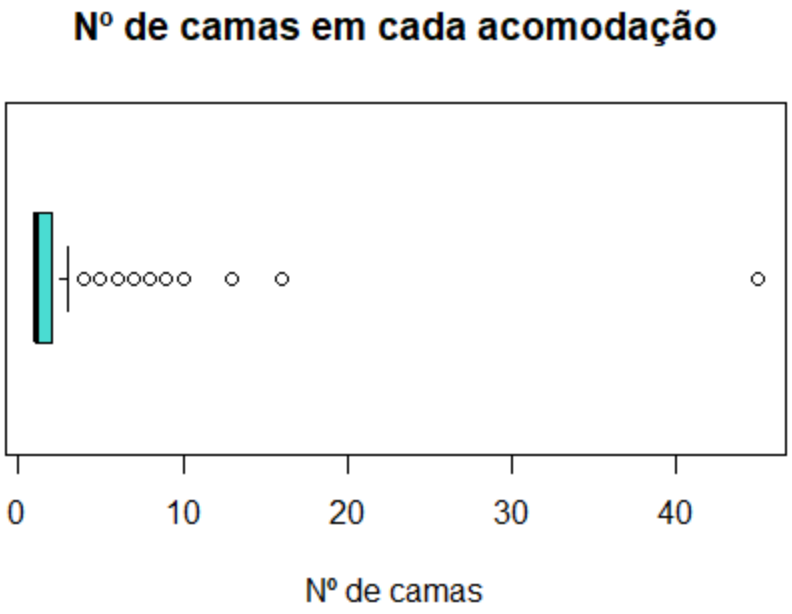

Figura 7 - Diagrama de extremos e quartis para a variável bed.

De acordo com a Figura 7, é possível analisar um enviesamento à direita, e *outliers* superiores. Além disso, o valor do primeiro quartil coincide com a mediana e, o terceiro quartil apresenta um valor de 2. Isto significa que o número de camas de 75% dos alojamentos considerados, apresenta no máximo duas. Por se tratar de uma variável discreta, decidiu-se manter todas as observações.

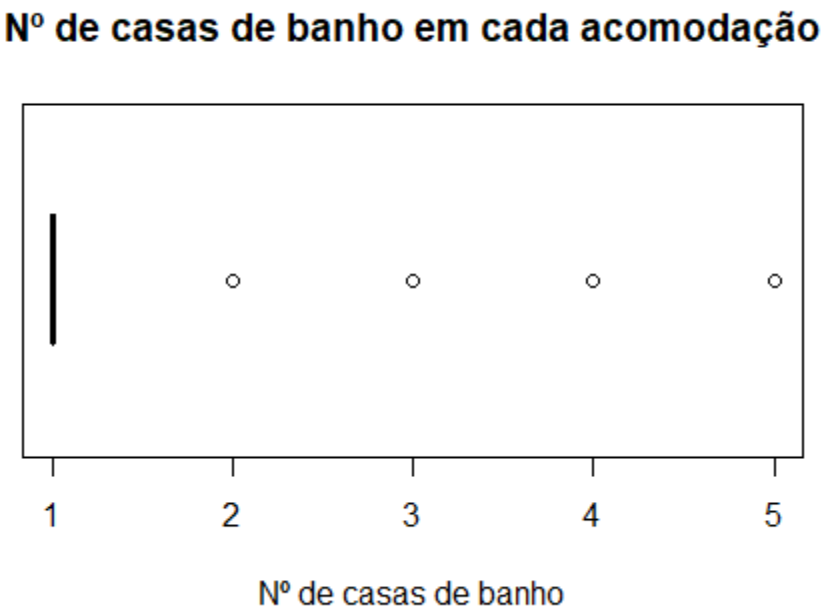

Figura 8 - Diagrama de extremos e quartis para a variável nrbath.

Relativamente à variável número de casas de banho em cada acomodação, é possível confirmar pela Figura 8, que primeiro quartil = mediana = terceiro quartil, significando que o número de casas de banho de 75% das acomodações consideradas é no máximo uma, sendo o resto considerado *outlier*. Por se tratar de uma variável discreta compreendida entre um e cinco, inclusive, decidiu-se manter os resíduos da forma original.

3.6 Data Preparation

É nesta fase que ocorre a preparação dos dados até obter o *dataset* final, que irá servir de base para a análise dos dados.

Partindo de uma base de dados com 17112 acomodações, procedeu-se à remoção das entradas duplicadas, uma vez que os alojamentos podem aparecer diversas vezes numa consulta no *Airbnb*, ficando com um total de 4366 observações.

Posteriormente, verificou-se que as variáveis ***price***, ***nreviews*** e ***score*** continham valores omissos. Como estes não nos forneciam informação relevante, decidiu-se eliminar.

O terceiro tratamento dos dados, ocorreu com a conclusão de que existiam 20 tipos de acomodações diferentes. Decidiu-se analisar e, numa fase preliminar, agrupou-se os tipos de acomodações que tinham um peso inferior a 5%, em relação ao *dataset*, e incluímos estes, na categoria "other", com o intuito de melhorar a visualização do gráfico abaixo ilustrado.

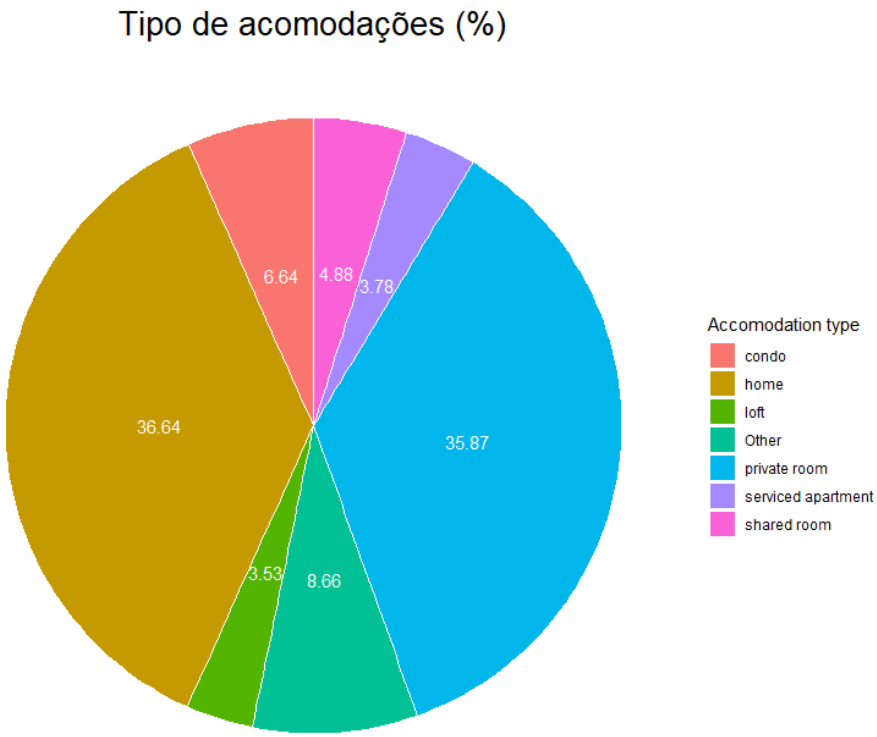

Figura 9 - Percentagem de tipo de acomodações.

Após a observação da Figura 9, conclui-se que tanto a casa, como o quarto privado são os alojamentos com mais volume, representando 36.64% e 35.97% dos dados, respetivamente. Além disso, observou-se que a percentagem de "other" era reduzida, e por esse motivo, decidimos excluir, finalizando assim com 2214 observações.

Através da fase de *Data Understanding*, conseguiu-se apurar que todas as variáveis continham *outliers*. É de salientar que, procurou-se tratar os outliers e não eliminar, uma vez que o número de observações já era reduzido.

Como não tratámos os outliers das variáveis ***score***, ***guest***, ***bedroom***, ***bed*** e ***nrbath***, esses diagramas mantiveram-se iguais, enquanto que nos restantes, após o tratamento, obteve-se os seguintes diagramas de extremos e quartis:

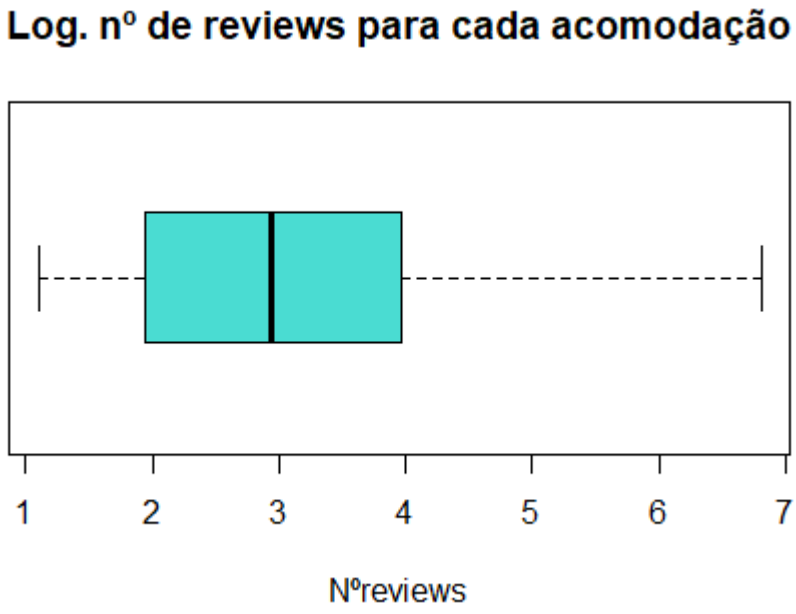

Figura 10 - Diagrama de extremos e quartis para a variável lognreviews.

Em relação à variável original número de *reviews*, observou-se através da Figura 34, do Anexo 1, uma tendência exponencial das observações. Por esse motivo, decidimos realizar uma transformação logarítmica para normalizar os dados. Após esta modificação, a variável ficou sem *outliers*, como se pode comprovar pela Figura 10.

No diagrama referente à variável ***lognreviews***, o valor do terceiro quartil corresponde a aproximadamente 4 (3.970), significando que o número de *reviews* após a transformação logarítmica, de 75% dos alojamentos considerados é no máximo $e^{3.970}$ =52.98 *reviews*, e o mínimo encontrado foi de $e^{1.099}$ = 3. Isto indica que a alteração efetuada, não enviesou as observações.

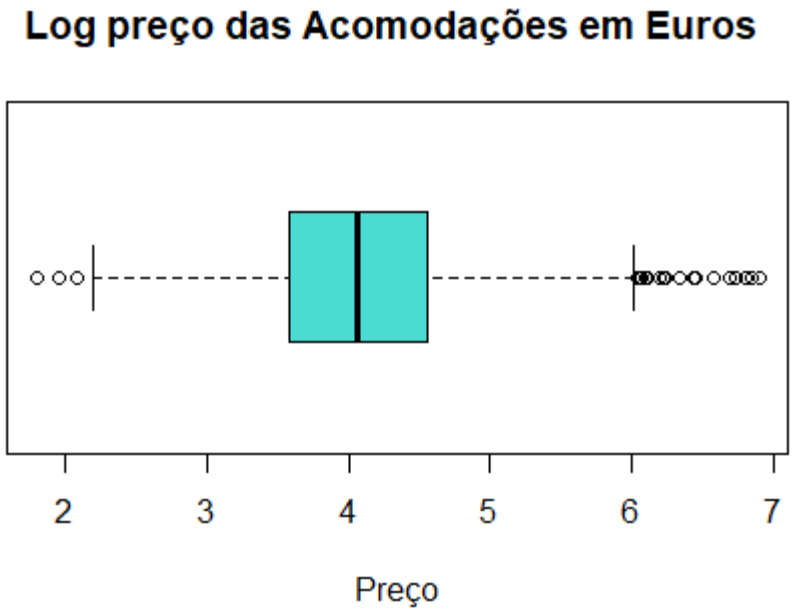

Figura 11 - Diagrama de extremos e quartis para a variável logprice.

Relativamente à variável original preço, seguiu-se o mesmo método que para a variável ***nreviews***, visto que o histograma ilustrado no Anexo III também apresentava uma distribuição exponencial dos dados.

Após a transformação logarítmica, surgiram novos *outliers*, tanto inferiores como superiores, por esse motivo, procedeu-se a uma substituição destes pelo valor do 5º e 95º percentil. Com esta modificação, como se pode averiguar pela Figura 12, a variável deixou de apresentar *outliers*.

Neste caso, após estas etapas, o preço médio de cada alojamento $e^{4.088}$ = 60€, sofrendo uma queda de 23€, relativamente à variável original.

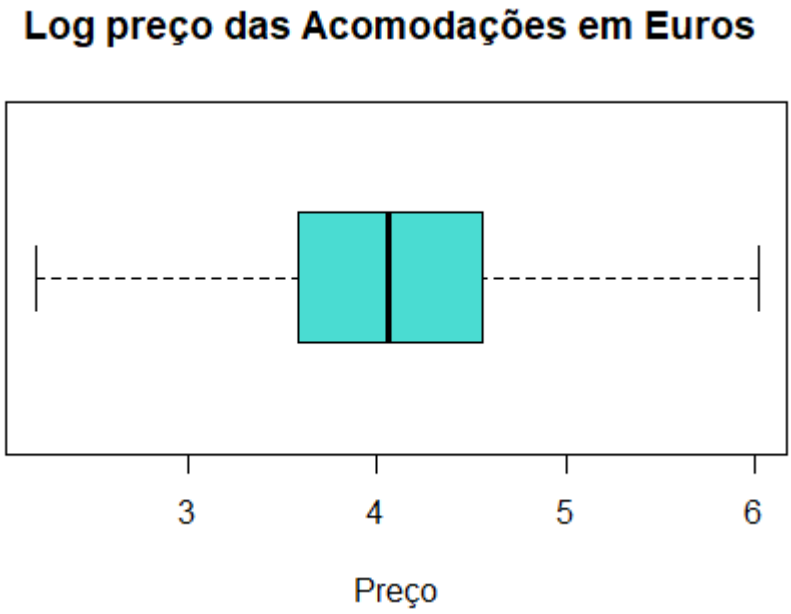

Figura 12 - Diagrama de extremos e quartis para a variável logprice, após substituição.

Feito o tratamento dos *outliers*, procurou-se perceber como é que as variáveis se relacionavam entre si, com especial atenção à relação da variável-alvo com as preditoras.

Para tal, foi utilizado a correlação de *Pearson*, que mede o grau de correlação entre duas variáveis. Os valores da correlação variam entre -1 e 1, com valores acima de zero a indicarem uma correlação positiva entre as variáveis e abaixo de zero uma correlação negativa.

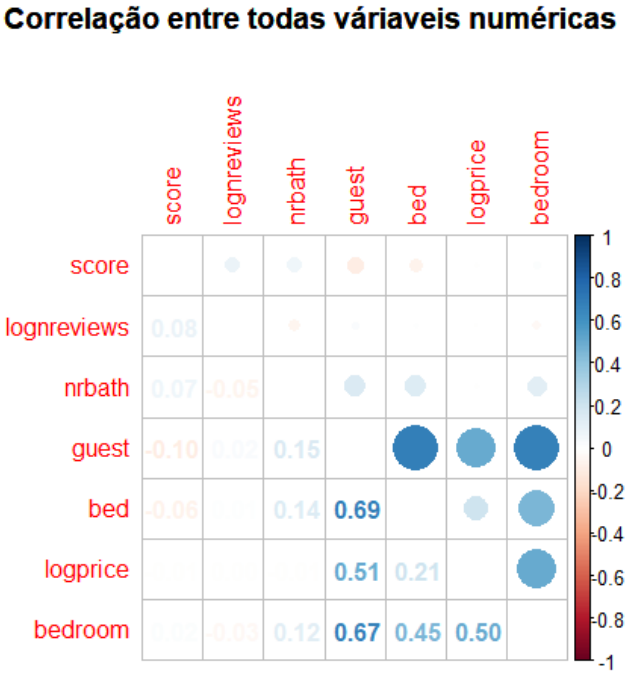

Figura 13 – Coeficiente de correlação de Pearson.

Como se pode observar através do gráfico acima ilustrado, as variáveis que apresentam maior poder de correlação são as seguintes combinações:

	guest
bed	0.69
logprice	0.51
bedroom	0.67

Tabela 2 - Variáveis correlacionadas.

A correlação entre as variáveis apresentadas na Tabela 2, são positivas e consideradas fortes entre ***bed – guest; bedroom – guest*** e moderada entre ***logprice-guest***. Deste modo, é possível concluir que há medida que o número de camas numa acomodação aumenta, o número máximo de hóspedes permitido aumenta. Esta é a correlação mais elevada no conjunto de dados, pelo que a variável ***bed*** foi retirada, uma vez que o modelo poderia conter informação redundante.

A segunda correlação mais elevada é entre a variável ***bedroom*** e ***guest***, com um valor de 0.67, indicando que um aumento do número de quartos dos alojamentos, provoca um aumento no número permitido de hóspedes. Em consequência da forte correlação, decidiu-se excluir do modelo a variável ***bedroom***, visto que poderia existir informação repetida.

Por último, apesar da correlação entre a variável ***logprice*** e ***guest,*** ser moderada também se eliminou, com o propósito de não obter informação redundante.

Também se analisou os gráficos de dispersão entre as variáveis, dos quais se retira algumas conclusões. Consegue-se perceber que alojamentos que permitem um maior número de hóspedes, têm tendência a ter *scores* mais baixos, pelo que é provável que exista uma maior procura por alojamentos menores.

Para além disto, é possível verificar que as acomodação com um número maior de *reviews*, apresentam um score mais elevado, pelo que as pessoas tendem a apresentar uma procura seletiva baseada nas *reviews* anteriores, de modo a poderem ter uma experiência melhor, isto porque muitas vezes as imagens fornecidas não correspondem à realidade.

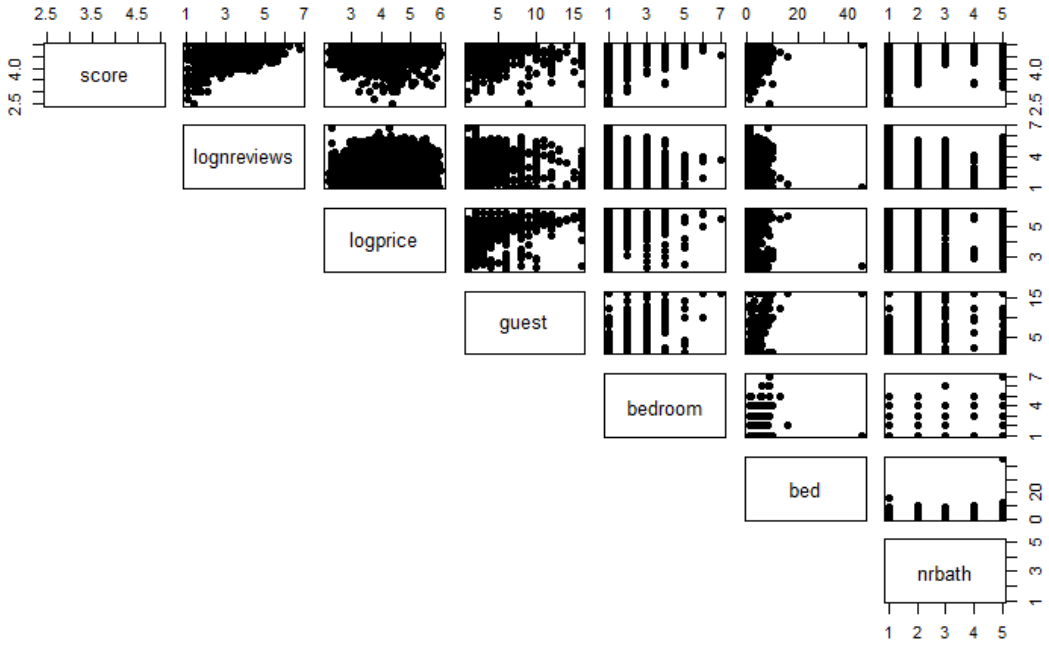

Figura 14 - Gráfico de dispersão.

A última preparação dos dados, foi realizada antes de criar o modelo das Redes Neuronais, uma vez que estas só funcionam com valores entre zero e um. Por esse motivo, procedeu-se à normalização dos dados através da técnica de *dummy* para a variável ***property_type*** e em relação às outras, usou-se a seguinte função, descrita no livro "Just Enough R! An Interactive Approach to Machine Learning and Analytics":

[1] $$New\ Value = \frac{OriginalValue - MinValue}{MaxValue - MinValue}$$

Equação 1 - Min-Max Normalization

Em relação às variáveis binárias, como estas já se encontravam no intervalo pretendido, não sofreram qualquer técnica de normalização.

3.7 Modeling

Nesta secção, será apresentado os modelos de previsão em prol de atingir o objetivo: prever o score atribuído por *reviewers* em unidades de alojamento local, comercializados na plataforma *Airbnb* para a cidade de Hong-Kong, durante o ano de 2022.

Para resolver este problema, foram desenvolvidos modelos que recorrem às metodologias da Regressão Linear Múltipla, Árvore de Decisão e Rede Neuronal Artificial.

3.7.1 Regressão Linear

Com base nas características das variáveis associadas a este problema, implementou-se a Regressão Linear Múltipla, sendo um dos modelos mais simples de analisar a previsão do score dos alojamentos em Hong-Kong.

Para métricas principais de comparação entre modelos, escolheu-se o *Trade-off* entre número de variáveis e *R-Adjusted*. O *R-Adjusted*, costuma ser utilizado para comparar a qualidade entre modelos e, ao contrário do coeficiente de determinação (R^2), não depende da quantidade de variáveis explicativas.

É de realçar que os resultados não irão ser surpreendentes, na medida em que a correlação entre a variável alvo – ***score*** – e as suas preditoras é fraca, contudo foi-se testando para encontrar o melhor Trade-Off.

Inicialmente, criou-se um *DataFrame (df2)* , ilustrado na Tabela 18 do Anexo IV, com as 24 variáveis explicativas do modelo, e aplicando a função *summary*, obteve-se o resultado apresentado na Tabela 3 . Como este resultado, não foi satisfatório visto que o *R-Adjusted* = 0.2619, continuou-se a procurar uma melhoria de resultado.

Dado que o *DataFrame* anterior, não era aceitável, optou-se por começar a reduzir o número de variáveis, com o intuito de alcançar um modelo mais simples. Assim, sendo, excluíram-se aquelas que apresentavam um p.value superior a 0.05, isto significa que estas não são estatisticamente significativas para explicar o *score* das acomodações de Hong-Kong, uma vez que a hipótese nula – ausência de relação entre as variáveis – não é rejeitada para um nível de significância estatística de 95%.

Preliminarmente, eliminou-se o ***nrbath*** e a ***tv***, obtendo o *DataFrame (df3)*, apresentado na Tabela 19 do Anexo V, com 22 variáveis e um *R-Adjusted* = 0.2622, alcançando assim, uma melhoria.

Através da Tabela 20 do Anexo VI, observa-se que se continuou a eliminar as variáveis que não eram estatisticamente significativas para prever o score, sendo que no DataFrame (df4), retirou-se o ***wi-fi*** e o ***elevator***, e, com esta alteração captou-se um melhor desempenho.

Para o DataFrame (df5), apresentado na Tabela 21 do Anexo VII, descartaram-se as variáveis preditoras ***dryver***, ***luggage_do***, uma vez que a hipótese nula era rejeitada, contudo, observou-se uma descida do *R-Adjusted*.

Por último, apesar do *DataFrame* anterior, apresentar uma descida do desempenho, decidiu-se eliminar ainda as variáveis ***pet***, ***fire_ext*** e ***first_aid***, designando este por *df6*, ilustrado na Figura 15. Realizada esta etapa, os resultados voltaram a piorar, neste que é o quinto modelo apresentado, ainda assim, decidiu-se optar por este, uma vez que é um modelo menos complexo de analisar e a diferença entre os desempenhos era reduzida.

	Q. Variáveis	R-Adjusted
df2	24	0.2619
df3	22	0.2622
df4	20	0.2626
df5	18	0.2625
df6	**15**	**0.2622**

Tabela 3 - Trade-Off entre a quantidade de variáveis preditoras e o R-Adjusted.

```
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.622320   0.051058  90.531  < 2e-16 ***
property_typehome              0.020211   0.023942   0.844 0.398676
property_typeprivate room     -0.077437   0.027369  -2.829 0.004706 **
property_typeserviced apartment  0.004264   0.038329   0.111 0.911429
property_typeshared room      -0.007591   0.043009  -0.176 0.859922
lognreviews                    0.023157   0.005055   4.581 4.89e-06 ***
guest                         -0.019504   0.002790  -6.990 3.63e-12 ***
superhost1                     0.228285   0.013502  16.908  < 2e-16 ***
studio1                       -0.071152   0.020618  -3.451 0.000569 ***
kitchen1                       0.084645   0.016484   5.135 3.07e-07 ***
washer1                        0.057463   0.016774   3.426 0.000624 ***
ac1                           -0.051421   0.029783  -1.727 0.084393 .
balcony1                       0.036678   0.016588   2.211 0.027125 *
hair_dryer1                    0.024467   0.014791   1.654 0.098244 .
breakfast1                     0.126738   0.050359   2.517 0.011917 *
parking1                       0.052141   0.026586   1.961 0.049980 *
smoking1                      -0.116514   0.023400  -4.979 6.88e-07 ***
bath_private1                 -0.097722   0.022303  -4.382 1.23e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2874 on 2196 degrees of freedom
Multiple R-squared:  0.2681,    Adjusted R-squared:  0.2624
F-statistic: 47.31 on 17 and 2196 DF,  p-value: < 2.2e-16
```

Figura 15 - DataFrame 6.

Com este modelo, conseguimos estimar que o *score* médio das acomodações de Hong-Kong aumenta em 0.1267 por cada unidade adicional da variável ***breakfast*** e, aumenta 0.2283 por cada unidade adicional da variável ***superhost***, *ceteris paribus*.

De forma complementar, a equação estimada consegue explicar 26.81% da variabilidade da variável ***score***.

Feita a limpeza dos dados, dividiu-se as 2214 acomodações, num conjunto de teste e treino, para posteriormente avaliar os resultados. Decidiu-se fixar o conjunto de treino em 70% e o de teste em 30%, para não correr o risco de *Overfitting* e, alcançou-se os seguintes resultados:

Treino	
RMSE	0.2893
MSE	0.0837
MAE	0.2054
Teste	
RMSE	0.2860
MSE	0.0818
MAE	0.2116

Tabela 4 - Resultado da divisão em conjunto de treino (70%) e teste (30%).

Em cada uma das métricas apresentadas na Tabela 4, o objetivo é minimizar o respetivo valor. Assim sendo, os valores das medidas de erros, para os dados vistos e não vistos, não são considerados elevados. No entanto, antes de concluir se o modelo de regressão linear é aceitável (ou não), para estimar o ***score*** atribuído por *reviewers* em unidades de alojamento de Hong-Kong, irá aplicar-se o método de reamostragem Validação Cruzada.

Após a implementação deste método, para k=10, os resultados não se alteraram de RMSE, MSE e MAE não se alteraram, pelo que é expectável uma vez que as variáveis preditoras não são correlacionadas com a variável ***score***.

Teste	
RMSE	0.2860
MSE	0.0818
MAE	0.2116

Tabela 5 - Resultado da Validação Cruzada com K=10.

Com um Erro Absoluto Médio de 0.2116 (isto é, dois décimos de *score*) não é possível ter certeza de uma determinação precisa do *score* médio atribuído por *reviewers*. Deste modo, é de considerar uma abordagem não linear para determinar o ***score.***

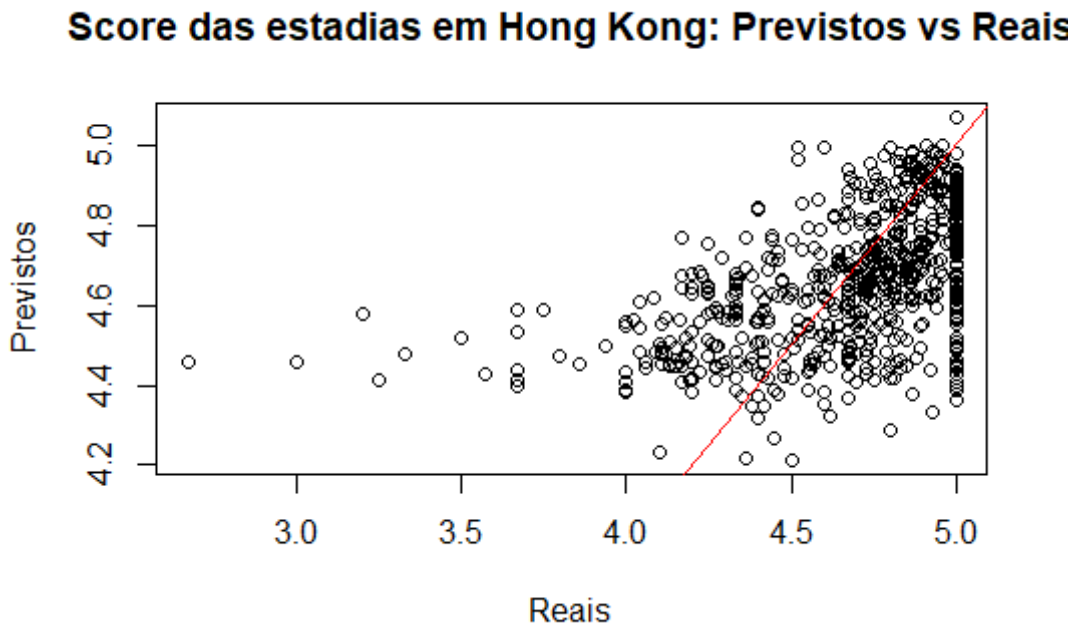

Figura 16 - Modelo Final - Score dos alojamentos em Hong-Kong.

O gráfico acima compara as previsões do modelo final da Regressão Linear com os dados do conjunto de teste. É possível averiguar uma concentração maior junto da reta da regressão para *scores* acima de 4.5, nos dados reais. Abaixo desse valor, os valores começam a dispersar consideravelmente, inclusive com vários casos cuja previsão foi mais elevada tendo em conta o *score* real atribuído por *reviewers*.

3.7.2 Árvore de Decisão

As árvores de decisão são métodos supervisionados e não paramétricos de *Machine Learning*. Estes métodos tanto podem ser utilizados em problemas de classificação como em problemas de regressão. As árvores de decisão dividem, de forma reiterada, um problema em grupos mais simples, de forma a que se possa tirar ilações, tendo em conta os valores que cada variável ou atributo possam assumir.

As árvores de decisão apresentam algumas vantagens comparativamente com outros modelos, nomeadamente:

- Fácil interpretação e compreensibilidade;

- Pode ser usado para identificar as variáveis mais significativas no conjuntos de dados;
- Facilidade em lidar com diversos tipos de informação (variáveis nominais e ordinais);
- Trata-se de um método não paramétrico e por isso, não assume nenhuma distribuição dos dados;
- Apresenta robustez à presença de *outliers*, bem como de variáveis redundantes e irrelevantes devido à sua capacidade de escolher os atributos a serem utilizados.

Contudo, estas também apresentam algumas desvantagens, entre as quais:

- Instabilidade – quando pequenas perturbações no conjunto de treino originam árvores bastante diferentes;
- Preferência por árvores mais pequenas, o que geralmente leva a um sobre ajustamento o modelo.

Metodologia:

A metodologia utilizada para resolver o problema de regressão pode ser esquematizada da seguinte forma:

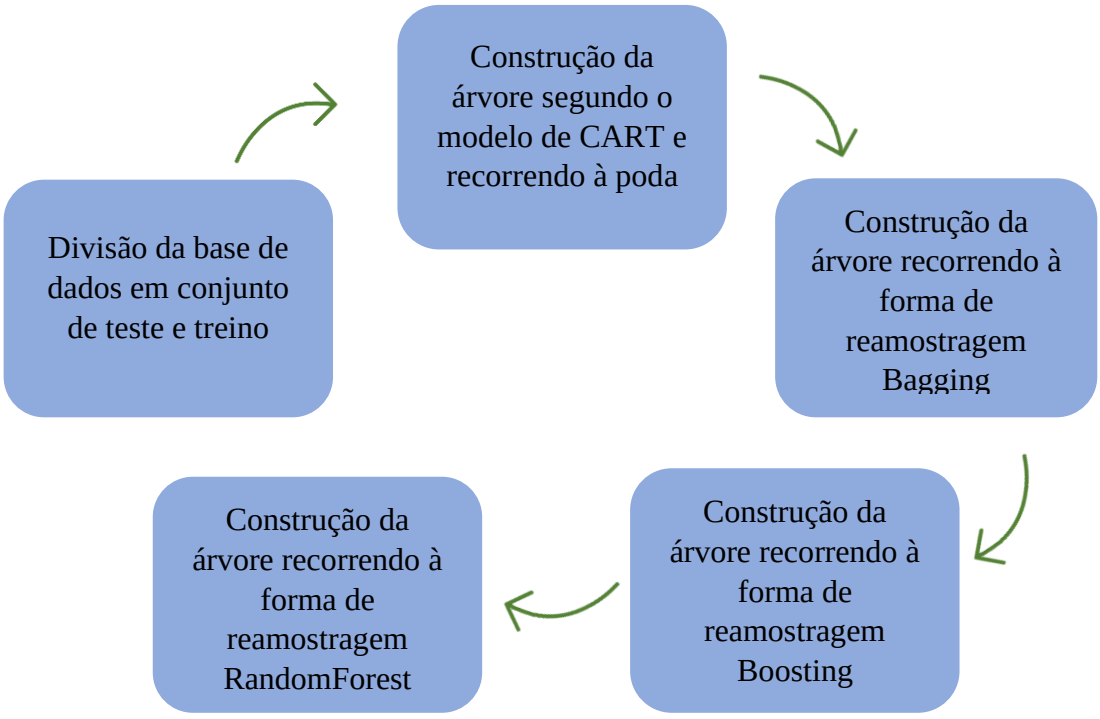

Figura 17 - Metodologia da construção da Árvore de Decisão.

A escolha da divisão do conjunto de treino e teste para os modelos fez-se com base na prevenção do *Overfitting*, assim sendo optou-se por fixar 70% para o conjunto de treino e 30% para o teste.

Modelo da Árvore com poda (Modelo CART) com os parâmetros padrão do da função rpart():

O modelo CART trata-se de um modelo recursivo binário, isto é, o modelo escolhe o melhor atributo para dividir em dois grupos de forma a diminuir o erro geral resposta real e constante prevista seja minimizado (SSE).

Pressupostos da função:

- A árvore de decisão vai crescer até que o CP – Parâmetro de Complexidade – atinja um valor limite, de 0.01, ou que a regra de divisão em cada sub-nó origine folhas puras, ou seja, até que a árvore se torne demasiado complexa ou que a divisão não traduza nenhum ganho.

Modelo da Árvore antes da poda:

A árvore criada (Figura 18), apresenta uma profundidade de cinco, sendo que deu origem a seis nós internos e oito nós folha.

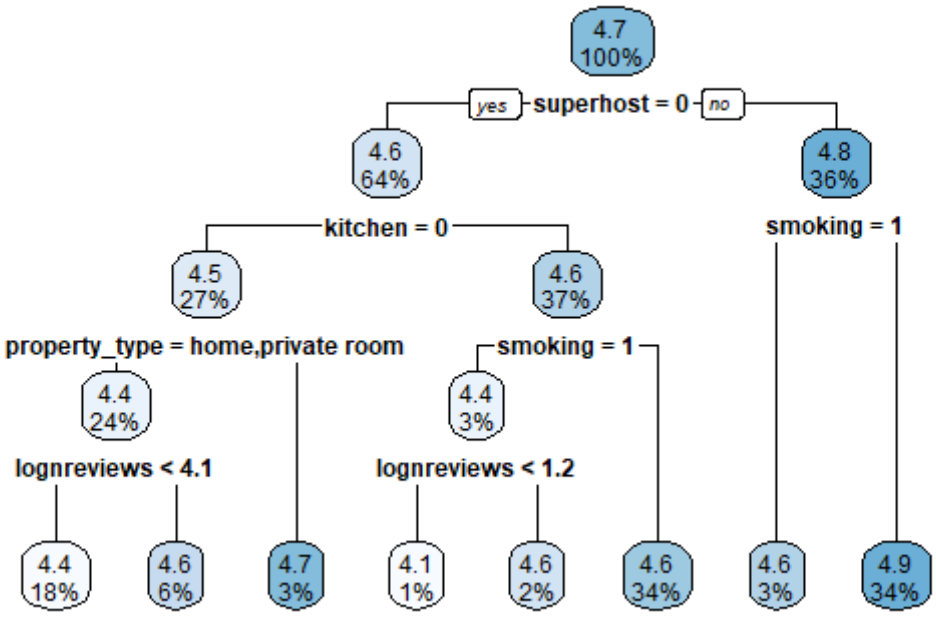

Figura 18 – Modelo da Árvore de Decisão antes da Poda.

Na Tabela 6, apresentada em baixo, encontra-se a importância de cada uma das variáveis, podendo assim concluir que o ***superhost*** e a ***kitchen*** têm o maior impacto nas divisões da árvore.

```
model_tree$variable.importance
  superhost       kitchen property_type   lognreviews       smoking        washer       balcony
29.1086424     6.2769270     5.6353917     4.9963727     4.0523620     2.6204647     1.9647041
      guest     breakfast        studio       parking
  0.4570578     0.1517145     0.1066468     0.1034055
```

Tabela 6 - Importância de cada uma das variáveis.

Na Tabela 7, abaixo indicada encontra-se os valores do Parâmetro de Complexidade (CP), Erro Relativo (rel. Error), Erro da Validação Cruzada (xerror) e Desvio Padrão da Validação Cruzada (xstd), da árvore gerada. Podemos observar que há medida que o CP diminui, o erro relativo também, mas chega a um ponto que a redução não é significativa, e é aí que se poda, como irá ser realizado mais à frente.

	CP	nsplit	rel error	xerror	xstd
1	0.164775	0	1.00000	1.00043	0.065175
2	0.035532	1	0.83523	0.83649	0.056884
3	0.017414	2	0.79969	0.80286	0.057222
4	0.015239	3	0.78228	0.79321	0.056532
5	0.011828	4	0.76704	0.78667	0.055924
6	0.011138	5	0.75521	0.78872	0.056504
7	0.010000	7	0.73294	0.78314	0.056491

Tabela 7 - CP, Erro Relativo, Erro de validação Cruzada e Desvio Padrão da Validação Cruzada do Modelo de Árvore de Decisão Antes da Poda.

Na avaliação do modelo da árvore de decisão antes da poda, foi obtido o resultado apresentado na Tabela 8, e podemos retificar que os valores pioraram ligeiramente, em relação ao modelo de Regressão Linear.

Antes da poda	
RMSE	0.2956
MAE	0.2151

Tabela 8 - RMSE e MAE antes da poda.

Modelo da Árvore depois da poda:

A poda foi realizada através da teoria, de que o corte na árvore deve ser feito onde o Erro Relativo (xerror) é menor e, onde na Figura, que representa a evolução deste em função do CP, começa a existir uma estabilização na curva.

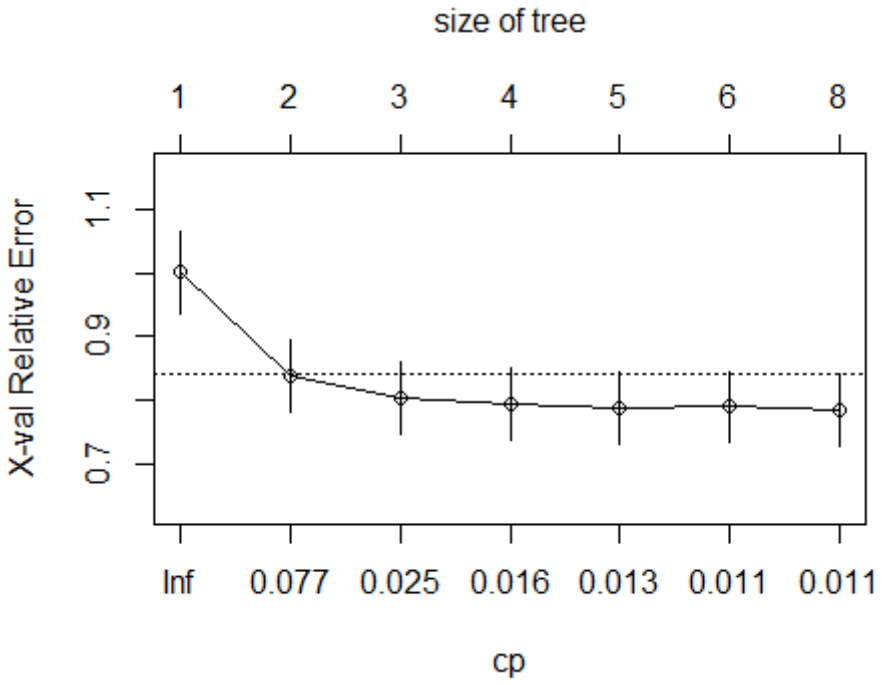

Figura 19 - Evolução do erro relativo da validação cruzada em função do CP – Modelo da Árvore de Decisão depois da Poda.

Tendo em conta o que foi mencionado anteriormente, o corte foi feito na terceira divisão (nsplit=3). A árvore criada com recurso à poda (Figura 20), deu origem a um nó interno, e a três nós terminais ou folha. Adicionalmente, esta árvore apresenta apenas um nível de profundidade igual a três. Em baixo, encontra-se também a importância de cada uma das variáveis (Tabela 9), bem como a Tabela 10, que mostra o CP, rel.error, xerror e xsts da árvore gerada.

Através da Tabela 9, é possível afirmar que o ***superhost*** e a ***kitchen***, continuam a ser as variáveis com maior impacto nas divisões das árvores.

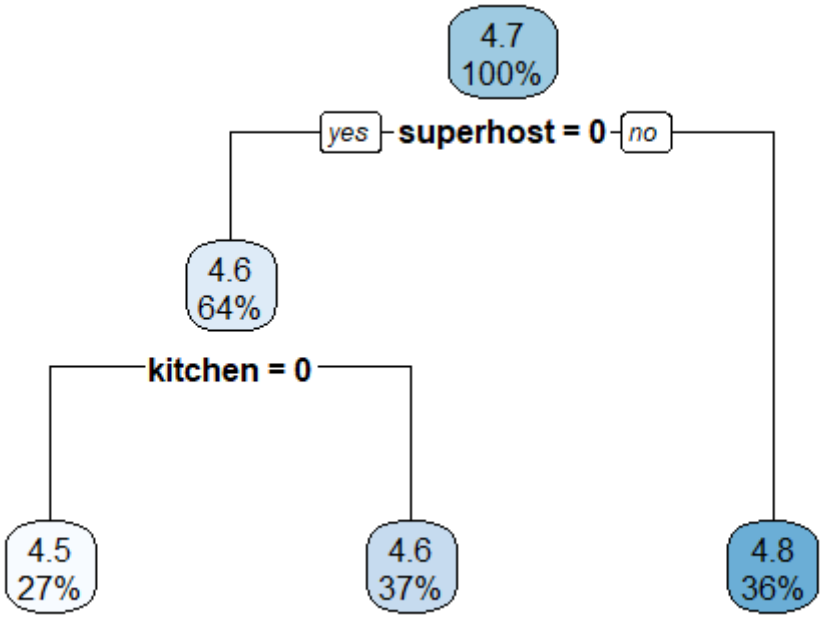

Figura 20 - Modelo de Árvore de Decisão Com Recurso à Poda.

```
model_tree_prune$variable.importance
  superhost       kitchen        washer property_type       balcony         guest   lognreviews
 29.1086424     6.2769270     2.6204647     2.5590379     1.9647041     0.4570578     0.3319344
     studio       parking
  0.1066468     0.1034055
```

Tabela 9 - Importância de Cada Uma das Variáveis no Modelo de Árvore de Decisão Com Recurso à Poda.

```
         CP nsplit rel error  xerror     xstd
1 0.164775      0   1.00000 1.00043 0.065175
2 0.035532      1   0.83523 0.83649 0.056884
3 0.025000      2   0.79969 0.80286 0.057222
```

Tabela 10 - CP, Erro Relativo, Erro de validação Cruzada e Desvio Padrão da Validação Cruzada no Modelo de Árvore de Decisão Com Recurso à Poda.

Através da Tabela 11, e comparativamente com a Tabela 8, percebe-se que o RMSE obteve uma melhoria, contudo o MAE subiu ligeiramente.

Depois da poda	
RMSE	0.2904
MAE	0.2168

Tabela 11 - RMSE e MAE depois da poda.

Modelo da Árvore com *Bagging*:

Bagging é uma técnica que permite reamostrar múltiplos conjuntos de treino com reposição e escolhidos aleatoriamente e, por consequência, gerar múltiplas árvores. Em cada um dos nós é feita a escolha de qual é o melhor atributo para dividir aquele nó (atributo que permite obter um subconjunto mais homogéneo ou que gera um maior ganho de informação). As árvores geradas são combinadas para gerar apenas um modelo. Nos modelos de regressão, o modelo final vai ser a média das árvores geradas (os valores médios da variável resposta para todas as instâncias da Folha), o que permite ter uma melhor generalização do modelo devido à diminuição da variância do modelo.

Para o modelo da árvore com *Bagging* o único parâmetro que foi usado, para melhorar os resultados foi o número de reamostragens de *bootstrapping* (*nbagg*). Foram usados os valores de 100, 200, 300 e 500, e foi selecionado o modelo que minimiza-se o RMSE e maximiza-se o *Rsquare*.

A Tabela 12, mostra os valores obtidos para os diferentes valores de *nbagg*, e a Figura 21 , apresenta de seguida a importância de cada uma das variáveis para a construção do modelo que apresentou melhores resultados (*nbagg* = 100).

	RMSE	Rsquare
100	**0.2932**	**0.2526**
200	0.2938	0.2450
300	0.2937	0.2503
500	0.2932	0.2507

Tabela 12 - Valores obtidos para os diferentes valores de nbagg.

Através da Figura 21, pode concluir-se que as variáveis que têm maior impacto, com base na soma da redução da função perda atribuída à variável em cada divisão da árvore, são: o número de *reviews* feita por hóspedes na cidade de Hong-Kong, o número de hóspedes permitido, a máquina de lavar roupa e se é permitido ou não fumar dentro do alojamento.

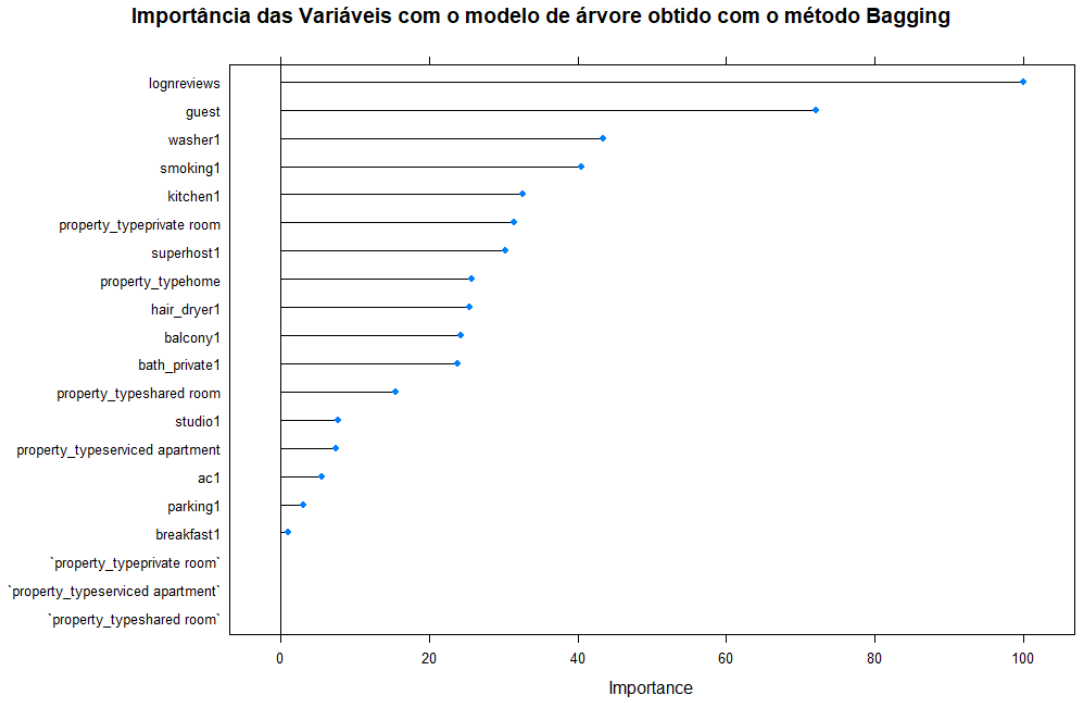

Figura 21 - Importância das Variáveis com o modelo de árvore com bagging.

Modelo da Árvore com *Boosting*:

Boosting ou *Gradient Boosting* é uma tentativa de melhorar a previsão diminuindo tanto a variância como o *bias*. Este modelo para diminuir o *bias* foca-se em previsões fracas e tenta melhorar o modelo em cada iteração, usando os pesos médios de modelos fracos e desta forma a diminuir a variância. Estas árvores crescem de forma sequencial e dependem fortemente da árvore gerada anteriormente. O erro de classificação é melhorado pela árvore que será gerada a seguir. Uma das vantagens destes modelos é a capacidade de suportarem diferentes funções de perda, porém estes têm tendência a fazer *overfitting*.

Para a construção deste modelo a profundidade máxima que foi testada foi três para evitar que o modelo sofresse de *overfitting*. Para o número mínimo de observações em cada nó terminal (*n.minobsinnode)* foram usados todos os números inteiros de cinco a vinte, para avaliar como estes afetam o modelo. Quanto à taxa de aprendizagem (ou *shrinkage*), usou-se 0.002 e 0.005, e, em relação ao número de árvores (ou iterações de *boosting*) variou- se entre 300 e 1000, com o objetivo de tentar obter melhores resultados

de RMSE. A Tabela mostra o valor de RMSE e *R-squared*, tendo em conta as alterações efetuadas.

	RMSE	R-square
1º		
Depth=3		
n.trees=1000	0.2927	0.2535
Shrinkage=0.02		
n.minobsinnode= 20		
2º		
Depth=2		
n.trees=500	0.2919	0.2564
Shrinkage=0.05		
n.minobsinnode= 15		
3º		
Depth=2		
n.trees=500	0.2923	0.2529
Shrinkage=0.05		
n.minobsinnode= 5		
4º		
Depth=2		
n.trees=300	0.2894	0.2707
Shrinkage=0.05		
n.minobsinnode= 5		
5º		
Depth=2		
n.trees=1000	**0.2874**	**0.2749**
Shrinkage=0.02		
n.minobsinnode= 20		

Tabela 13 - RMSE e R-square para o Modelo Boosting.

Através da Tabela 13, verifica-se uma evolução no sentido do aumento do RMSE, em relação à profundidade da árvore e ao número de árvores, demonstrando que não foi uma boa decisão. Além disso, é possível visualizar que há medida que o número mínimo de observações nos nós terminais aumenta e a taxa de aprendizagem diminui, o RMSE também diminui. Deste modo, o modelo escolhido foi o quinto, uma vez que apresenta o RMSE mais reduzido e o *R-square* mais elevado.

Com base na Figura 22, retém-se que as variáveis ***superhost***, e número de *reviews* são as que apresentam ser mais importantes para a divisão da árvore.

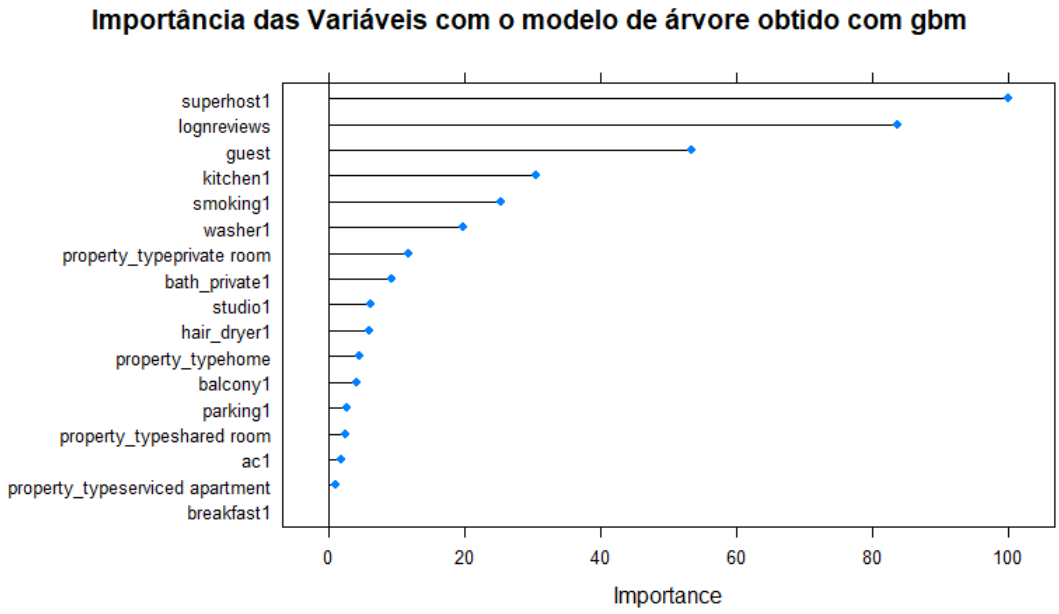

Figura 22 - Importância das variáveis com o Modelo de Árvore Obtido com Boosting (GBM).

Modelo da Árvore com *Random Forest*:

Tal como no *Bagging*, o *Random Forest* usa amostras *bootstrap*, mas para cada árvore ajustada uma subamostra aleatória de atributos é usado no processo de *fitting*. Aqui, o algoritmo de Árvores de Classificação e Regressão (CART) cria árvores de decisão aleatoriamente e, em seguida, calcula a média dos resultados - não há otimização em etapas. Se todos os recursos forem usados no processo de *fitting*. O *Random Forest* tem como vantagens a capacidade de lidar bem com dados de alta dimensionalidade e de se dar bem com a ausência de valores.

Para a criação deste modelo, considerou-se um número mínimo de observações nos nós terminais de cinco, fazendo apenas variar o número de variáveis preditoras, e através da Tabela 14, confirma-se que o RMSE aumenta, há medida que o número de variáveis aumenta. Assim sendo, o melhor modelo é aquele em que o número de variáveis é igual a dois, uma vez que RMSE é igual a 0.2882.

mtry	splitrule	RMSE	Rsquared	MAE
2	variance	0.2882932	0.2886527	0.2042981
2	extratrees	0.2905099	0.2788480	0.2057411
5	variance	0.2877067	0.2753290	0.2019657
5	extratrees	0.2877417	0.2747828	0.2016227
9	variance	0.2966053	0.2445166	0.2076130
9	extratrees	0.2940619	0.2549461	0.2051103
13	variance	0.3017568	0.2290373	0.2106326
13	extratrees	0.2984459	0.2431941	0.2078306
17	variance	0.3044577	0.2220346	0.2126835
17	extratrees	0.3016974	0.2341544	0.2099348

Tabela 14 - Relação entre mtry e RMSE, Rsquare e MAE.

Visualizando a Figura 23, averigua-se que a variável ***superhost***, continua a ser a variável mais importante, seguindo o número de *reviews*.

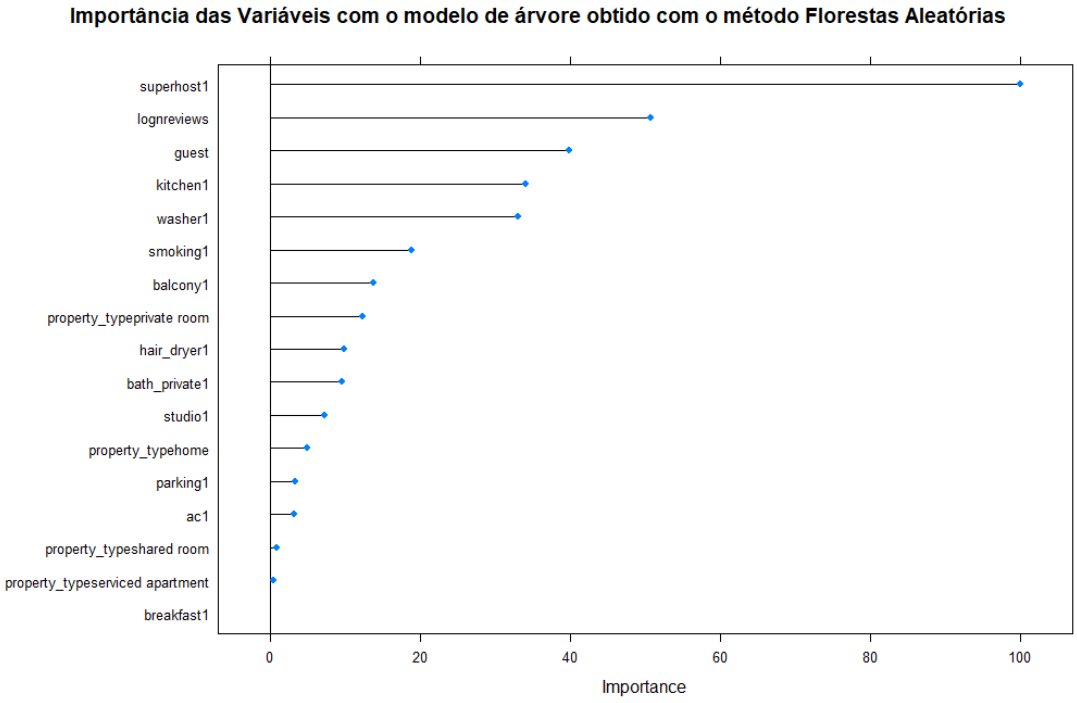

Figura 23 - Importância das variáveis com o Modelo de Árvore Obtido com Random Forest).

3.7.3 Rede Neuronal

As Redes Neuronais Artificiais (RNA) são modelos que se fundamentam na observação de dados para treinar uma rede artificial em que o próprio sistema, através da experiência de observação dos dados, aprende uma aproximação dos relacionamentos ao adaptar os seus parâmetros, sendo capaz de reconhecer padrões e implementar a aprendizagem de máquina.

Metodologia:

As configurações ideais dos parâmetros de uma Rede Neuronal Artificial para alcançar o melhor resultado são questões por responder, uma vez que não é possível fundamentar os resultados ocultos gerados por este tipo de rede. Assim sendo, mesmo que exista interesse em explicar quais são os parâmetros ótimos durante a construção de uma Rede Neuronal Artificial, as bases existentes são insuficientes para determinar a metodologia da seleção dos parâmetros de uma RNA. Isto é, não é possível definir quantos neurónios ou camadas uma determinada RNA deve ter para obter o melhor resultado possível, por exemplo. É com base neste facto que, para o desenvolvimento deste trabalho, se utilizou o método de tentativa em erro com o intuito de perceber como tentar minimizar o RMSE (Raiz quadrada do erro-médio) do modelo de previsão em Rede Neuronal Artificial.

Para a criação da RNA que consiga prever o *score* das acomodações na cidade de Hong-Kong, anunciados na plataforma *Airbnb*, seguiram-se cinco passos, conforme figura abaixo:

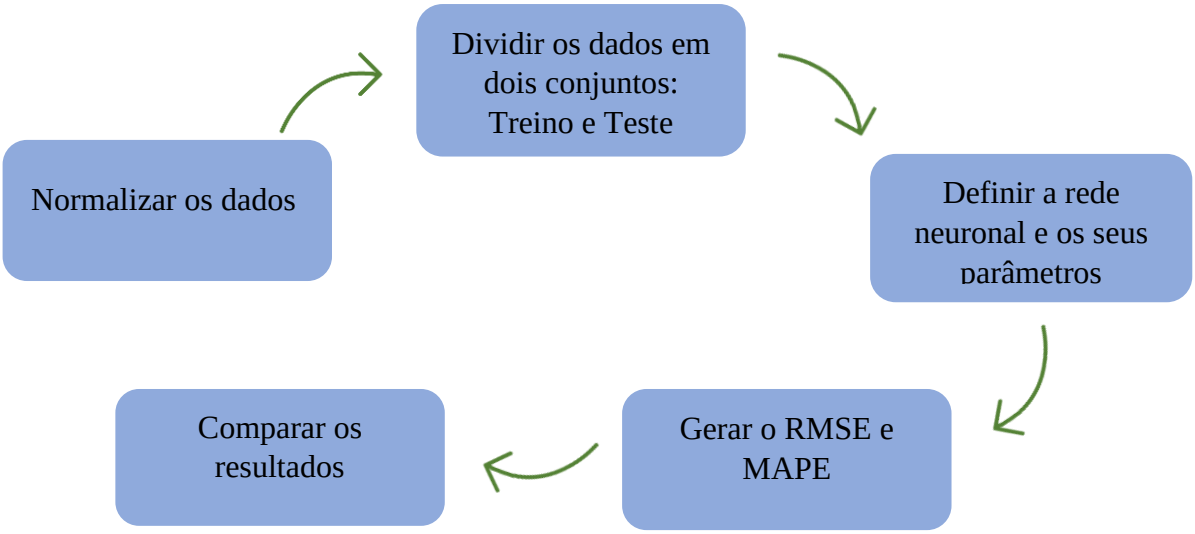

Figura 24 - Etapas da Elaboração da Rede Neuronal.

Inicialmente, decidiu-se construir o modelo com as 19 variáveis selecionadas, e tendo em conta essa decisão, procedeu-se às etapas seguintes. Primeiramente, usou-se o algoritmo *backpropagation*, e de forma simplificada é um método para treinar uma rede neuronal em que, durante o processo a rede faz uma previsão e calcula a perda de forma a se ajustar, utilizando a retropropagação, para atualizar os neurónios individuais na rede a fim de fazer uma melhor previsão.

Com este método, a definição da taxa de aprendizagem passa a ser um parâmetro importante para a construção da RNA. A taxa de aprendizagem é o hiperparâmetro que controla o quanto estamos ajustando os pesos da nossa RNA em relação ao gradiente de perda. Entretanto, quanto mais baixa for a taxa de aprendizagem, mais devagar se passa pela inclinação descendente, pelo que ao usar uma taxa muito baixa com a intenção de analisar toda a rede neuronal de forma minuciosa, o esforço computacional e o tempo de análise serão elevados.

Adicionalmente, definiu-se *stepmax* de $1e7$, sendo que quando atingir este valor irá levar à interrupção do processo de treinar a RNA.

Neste sentido, desenvolveram-se cinco tentativas, com duas camadas e ajustes nos neurónios. Destas cinco alternativas, a melhor foi a segunda, configurada em duas camadas, com um e trinta neurónios, respetivamente; algoritmo *backpropagation*, com a taxa de aprendizagem de 0.001, o que resultou num RMSE 0.1166, conforme a tabela abaixo.

Tentativa	%Treino	%Teste	Hidden	Algoritmo	Tx.Aprendizagem	RMSE	MAPE
1	70	30	1,20	backprop	0.001	0.1167	0.1219
2	**70**	**30**	**1,30**	**backprop**	**0.001**	**0.1166**	**0.1218**
3	70	30	1,50	backprop	0.001	0.1340	0.1476
4	70	30	10,10	backprop	0.001	0.1284	0.1284
5	70	30	20,20	backprop	0.001	0.1354	0.1282

Tabela 15 - As cinco tentativas de RNA com ajuste nos neurónios

Com esta configuração, esta primeira Rede Neuronal Artificial é representada pelo seguinte gráfico:

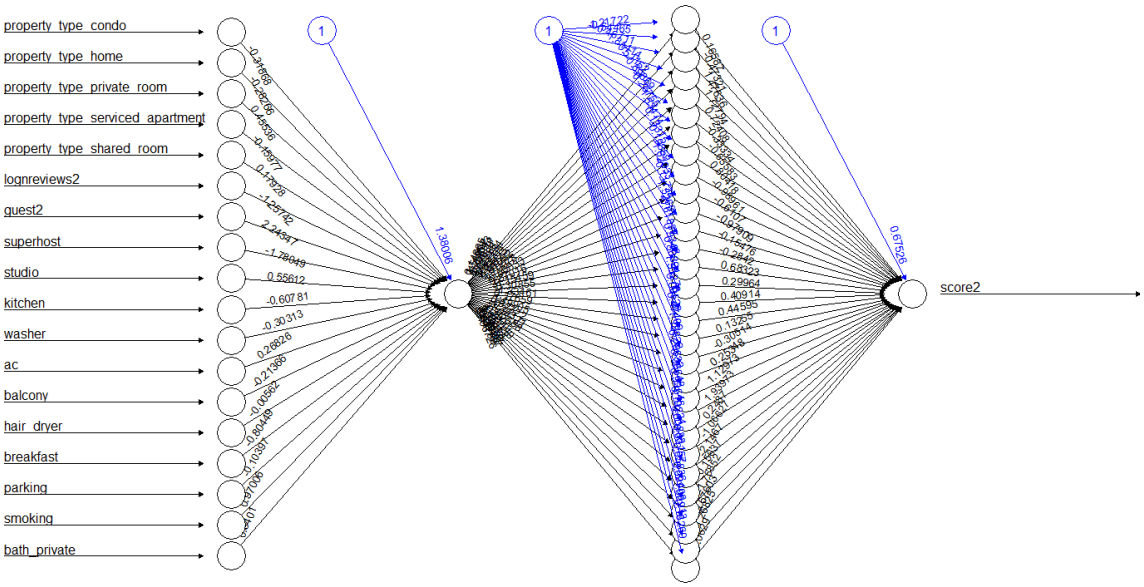

Figura 25 - Rede Neuronal Artificial com 18 entradas, duas camadas com um e trinta neurónios e uma variável de resposta.

Observando a Figura 26, concluiu-se que para *scores* reais entre 0.2 e 0.6, previu-se um score mais elevado, confirmando-se com a dispersão existente entre esses valores. Contudo, para *scores* com valores mais elevados, o erro de previsão é menor, uma vez que estes encontram-se mais próximos da reta.

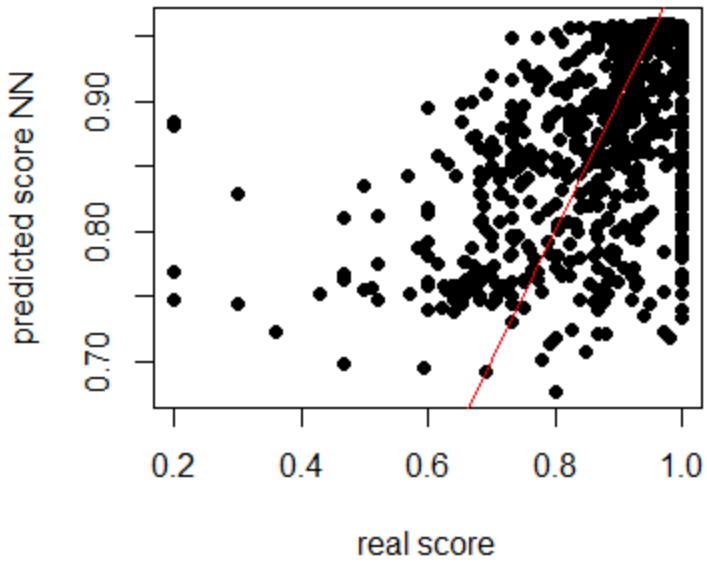

Figura 26 - Representação gráfica dos scores previstos e os reais.

Com o objetivo de simplificar o modelo e estudar o comportamento da RNA, excluiu-se algumas variáveis que se determinou não serem relevantes, como por exemplo, o ***parking***, o ***studio***, ***balcony***, ficando no total com 12.

Excluídas as variáveis, repetiu-se o procedimento anterior, e comparando os resultados, analisou-se que a diferença entre eles era pouco significativa, e por esse motivo, optou-se pelo modelo mais simples – tentativa 2, da Tabela abaixo indicada.

Tentativa	%Treino	%Teste	Hidden	Algoritmo	Tx.Aprendizagem	RMSE	MAPE
1	70	30	1,20	backprop	0.001	0.1179	0.1245
2	**70**	**30**	**1,30**	**backprop**	**0.001**	**0.1178**	**0.1245**
3	70	30	1,50	backprop	0.001	0.1179	0.1245
4	70	30	10,10	backprop	0.001	0.1331	0.1286
5	70	30	20,20	backprop	0.001	0.1362	0.1306

Tabela 16 - As cinco tentativa da RNA, com o modelo mais simplificado.

Dada a preferência pelo modelo mais simples, e uma vez que a diferença de desempenhos era pequena, construiu-se a Rede Neuronal Artificial com 70% dos dados no conjunto treino e 30% no conjunto teste, com duas camadas ocultas, com um e trinta neurónios cada, respetivamente, com o algoritmo de retropagação e uma taxa de aprendizagem de 0.001, RNA esta que pode ser representada graficamente da seguinte forma:

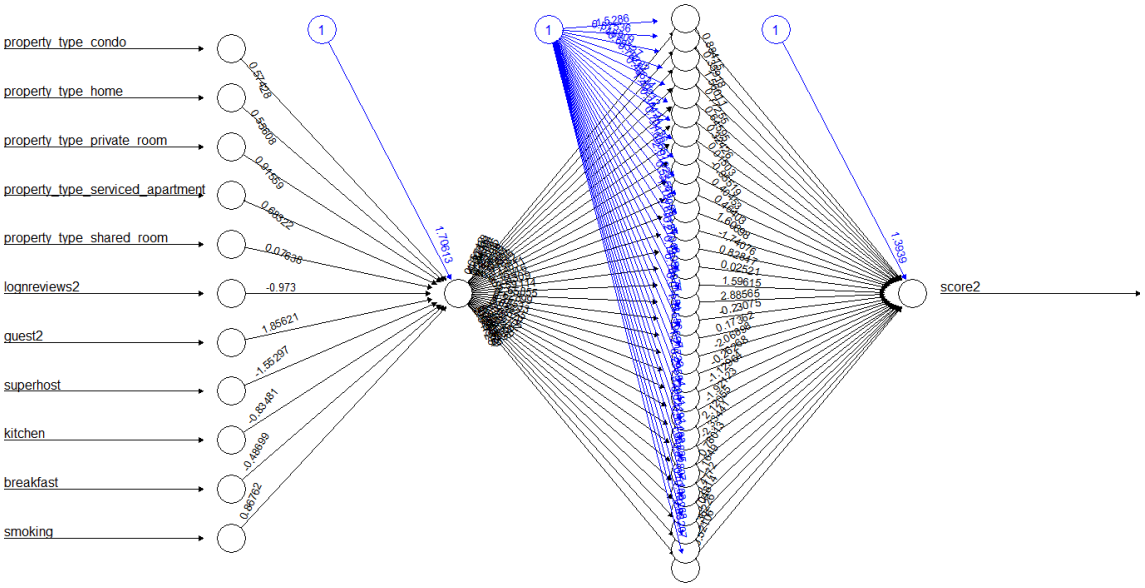

Figura 27 - Rede Neuronal Artificial com 11 entradas, duas camadas com um e trinta neurónios e uma variável de resposta.

Assim, tendo por base o segundo modelo de RNA apresentado neste relatório, é importante perceber como as variáveis influenciam o resultado do modelo ao avaliar os seus pesos generalizados.

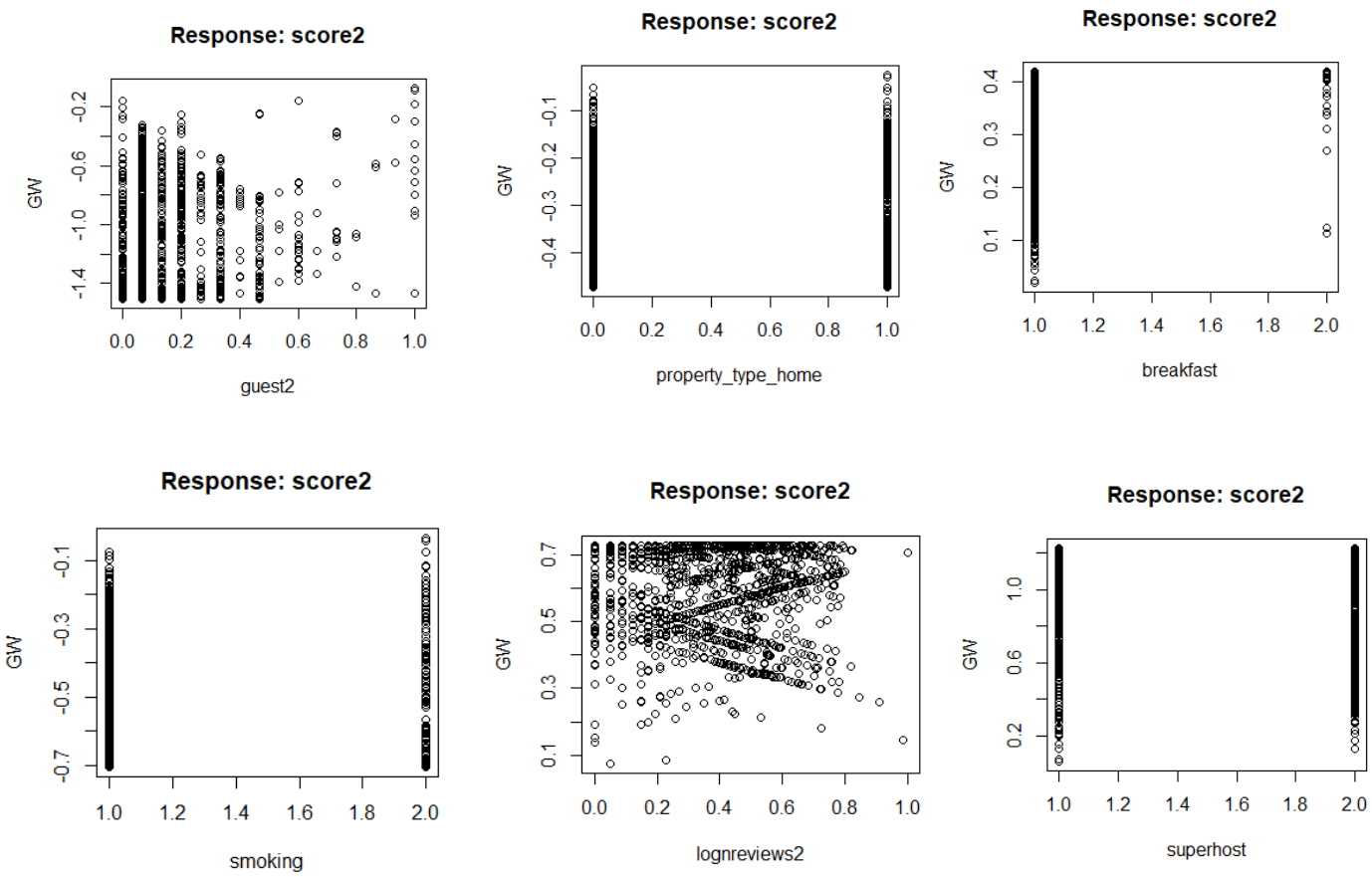

Figura 28 - Pesos generalizados da Rede Neuronal Artificial.

Depois de analisada a importância das variáveis para o modelo, chegou-se à conclusão de que as variáveis ***property_type*** são as que apresentam alguma concentração perto do zero, o que indica que em alguns pontos a influência foi nula. No entanto, é igualmente importante pontuar que a variável ***guest2*** desempenhou influência negativa no modelo, enquanto que as variáveis ***breakfast***, ***smoking*** e ***superhost*** foram as que apresentaram maior peso positivo no modelo.

3.8 Evaluation

Nesta etapa, realiza-se uma comparação e avalia-se os modelos de acordo com os resultados obtidos, com o propósito de escolher aquele que melhor se adequa ao objetivo final. Se não existirem melhorias a fazer, segue-se na fase seguinte o *Deployment*.

Ao longo deste trabalho, foram desenvolvidos sete modelos de previsão, e à luz dos resultados obtidos em termos de RMSE, realizou-se a avaliação. Uma vez que o objetivo é fazer uma previsão, utilizar-se-á o RMSE (Raiz quadrada do erro-médio), para fazer a comparação entre modelos, visto que é uma boa medida de quão preciso o modelo é para prever a resposta, sendo que, em linhas gerais, os valores mais baixos de RMSE indicam o melhor ajuste.

Neste sentido, como é possível observar, o melhor modelo, em termos de RMSE, é a Rede Neuronal Artificial, seguida da Regressão Linear Múltipla, com e sem Validação Cruzada. Estes resultados podem ser explicados devido ao facto, de as redes conseguirem aprender por si mesmas e serem capazes de produzir resultados que não se limitam aos valores fornecidos pelas variáveis de entrada. No entanto, a razão pela qual a Regressão Linear Múltipla, deu melhores resultados que as árvores de decisão, pode justificar-se pelo motivo de que esta consegue avaliar a capacidade explicativa de cada variável preditora.

	Model	RMSE
1	Regressão Linear - Multipla	0.286028261123184
2	Regressão Linear - Multipla (Validação Cruzada)	0.286028261123184
3	Árvore de Decisão	0.290452916909531
4	Árvore de Decisão - Bagging	0.288735645777365
5	Árvore de Decisão - Boosting	0.287419100188278
6	Árvore de Decisão - Florestas Aleatórias	0.288836492255406
7	Rede Neuronal	0.117967482597855

Tabela 17 - Comparação dos Modelo de Previsão ao longo do estudo.

3.9 Deployment

O mercado das plataformas digitais que permitem arrendar alojamentos, está cada vez mais competitivo e por esse motivo, exige reestruturações rápidas dentro das empresas. Afinal de contas, quem encontra um diferencial consegue tirar o melhor proveito, sendo que facilmente aumenta a sua quota de mercado.

Mas, para isso, é preciso criar que o *Airbnb* adote este projeto, uma vez que irá permitir otimizar os processos, em busca de eficiência, lucratividade e fidelização dos clientes.

Através deste link consegue-se ver em detalhe um protótipo iterativo do projeto desenvolvido através do *software* Figma: "https://www.figma.com/proto/xw8F2EzJ16zOlS3J4m8xyo/Airbnb-(Rene_Project)?page-id=0%3A1&node-id=202%3A3898&viewport=241%2C48%2C0.36&scaling=scale-down&starting-point-node-id=202%3A3898".

O objetivo deste estudo, é prever o *score* atribuído por *reviewers*, em unidades de alojamento em Hong-Kong, pelo que através disto o *Airbnb* poderá tirar o maior proveito desta informação. Nomeadamente, através do valor previsto poderá identificar quais os fatores preferências na escolha do consumidor e, com isto, apresentar uma oferta mais apelativa tendo em conta o perfil do *target*.

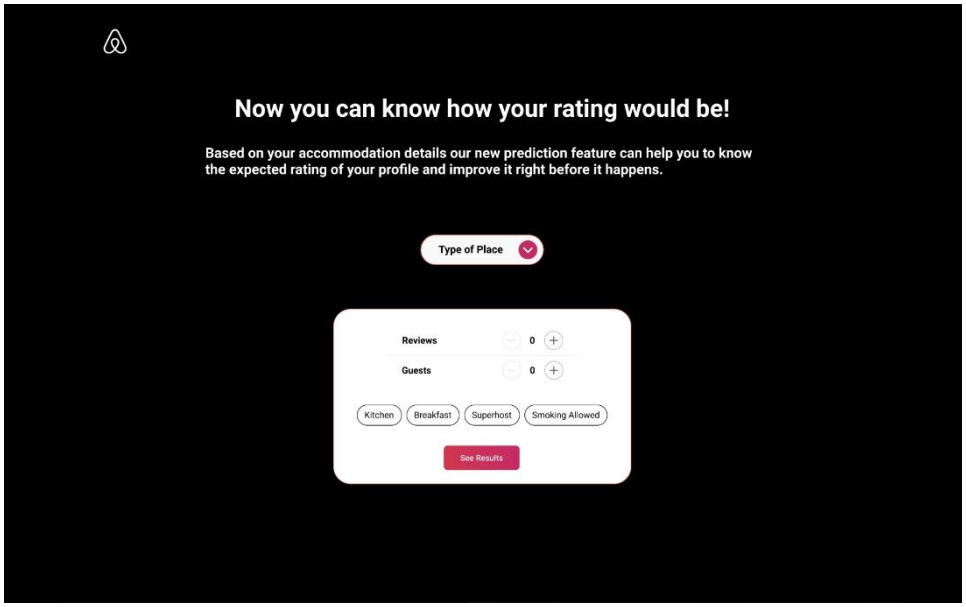

Figura 29 - Mockup do projeto (Parametrização).

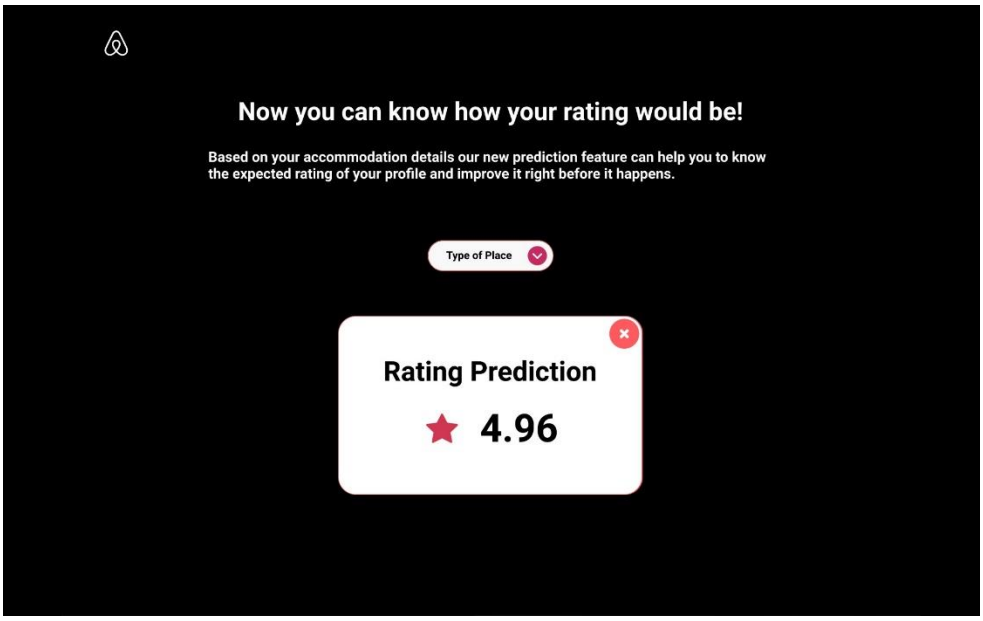

Figura 30 - Mockup do projeto (Resultado).

4. QUESTÃO 2 – O que fazer com dez linhas de código?

Para o desenvolvimento da Questão 2, foi escolhido o *dataset* "Rain in Australia", que contém informação ao longo de 10 anos sobre a temperatura, a velocidade do vento, a humidade, a pressão atmosférica, as nuvens da cidade de Cobar, na Austrália, e uma última categoria (variável *RainTomorrow*) com a indicação de se choveu, ou não, no dia a seguir. O *dataset*, ilustrado no Anexo IX, apresenta cerca de 2500 registos e o objetivo pretendido através do mesmo é prever a variável referida anteriormente.

Primeiramente, importou-se do ficheiro CSV, criou-se o *index* e dividiu-se os dados em conjunto de treino e do conjunto de teste. Para o desenvolvimento da questão, o modelo escolhido foi o de Regressão Logística, uma vez que esta pode ser aplicada para quando a variável target é binária, isto é, Sim/Não ou 1/0.

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  20.644429  22.839381   0.904  0.36605
MinTemp       0.094033   0.068202   1.379  0.16797
MaxTemp      -0.089480   0.098145  -0.912  0.36192
Rainfall      0.019387   0.017880   1.084  0.27823
WindGustSpeed 0.061897   0.011582   5.344 9.07e-08 ***
WindSpeed9am  0.005060   0.018413   0.275  0.78349
WindSpeed3pm -0.014590   0.019368  -0.753  0.45127
Humidity9am   0.008066   0.010708   0.753  0.45131
Humidity3pm   0.063278   0.011957   5.292 1.21e-07 ***
Pressure9am   0.605935   0.079509   7.621 2.52e-14 ***
Pressure3pm  -0.636134   0.081407  -7.814 5.53e-15 ***
Cloud9am      0.068921   0.046149   1.493  0.13532
Cloud3pm      0.133268   0.049551   2.690  0.00716 **
Temp9am       0.160283   0.088546   1.810  0.07027 .
Temp3pm      -0.080988   0.096519  -0.839  0.40142
```

Figura 31 - Summary da base de dados.

Quando o *p-value* é inferior 0,05 rejeita-se a hipótese nula, o que significa que são variáveis estatisticamente significativas. Através da Figura 30, observa-se que, as variáveis que têm maior poder explicativo e relevância para a criação do são a *WindGustSpeed*, a *Humidity3pm*, a *Pressure9am*, a *Pressure3pm* e a *Cloud3pm*.

Através de uma análise prévia da variável target (Figura 31) pode-se observar que os dados são não balanceados, pois em cerca de 80% da amostra, não choveu no dia seguinte.

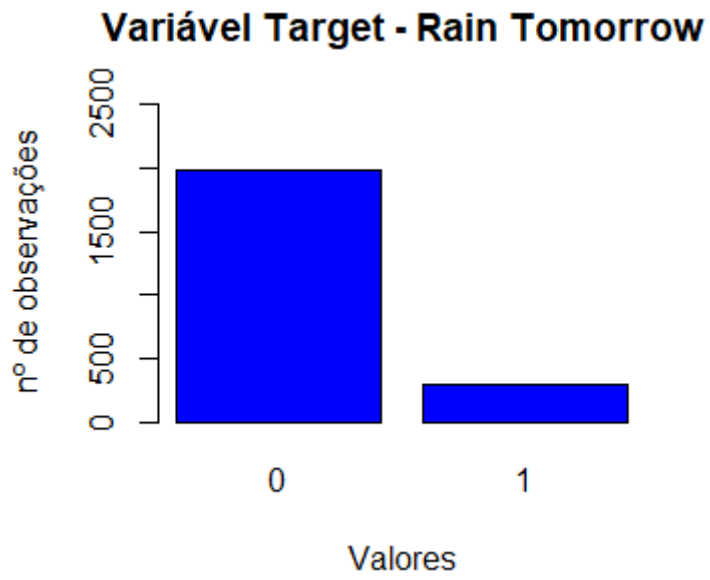

Figura 32 - Balanceamento dos dados.

Assim sendo, utilizou-se a Curva ROC para a análise da taxa dos *True* e *False Positives*:

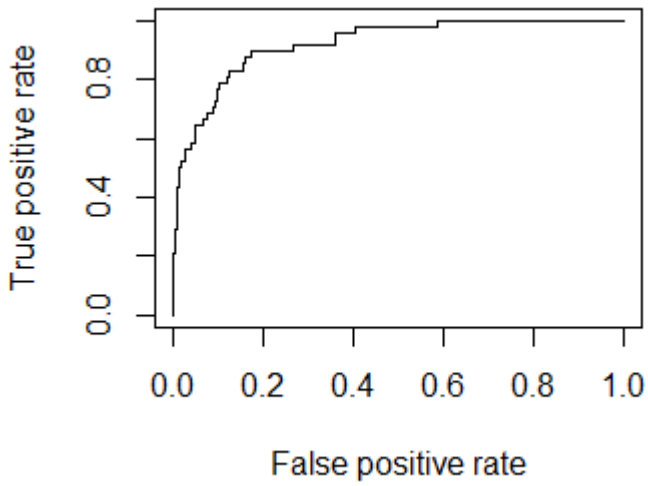

Figura 33 - Curva ROC.

Esta curva representa o *trade-off* entre a *Sensitivity (True Positive Rate)* e a *Specificity (1 – False Positive Rate)*. Quanto mais aproximada ao canto superior esquerdo a Curva ROC se encontra, melhor é o desempenho do modelo. Pode-se, então, afirmar que o modelo tem uma elevada taxa de desempenho, visto apresentar uma Curva ROC que se aproxima bastante do ponto ideal.

Para terminar, recorreu-se à métrica AUC (*Area Under the Curve*) para calcular a área por baixo da curva ROC. O AUC é um indicador que é útil para sumarizar o desempenho de dois componentes em apenas um. Em geral, quanto maior o AUC, melhor a precisão da previsão. Neste caso, o AUC foi de 0.9233 o que indica, uma vez mais, a elevada taxa de desempenho do modelo.

5. CONCLUSÃO

A realização deste trabalho, teve por base a metodologia CRISP-DM, sendo que as etapas foram seguidas da seguinte maneira: começou-se pela fase de *Business Understanding*, passou-se para a fase de *Data Understanding*, depois *Data Preparation*, seguindo para o *Modeling* e terminou-se na etapa *Evaluation*.

Ao longo do desenvolvimento do projeto, conseguiu-se adquirir e aprofundar alguns conhecimentos, nomeadamente os modelos de previsão lecionados em aula, sendo eles a Regressão Linear, as Árvores de Decisão e as Redes Neuronais Artificiais.

Para além dos conhecimentos teóricos adquiridos também conseguiu-se aprender alguns conhecimentos práticos, como a realização de *Web Scrapping*, a análise de variáveis explicativas e a aplicação dos modelos referidos anteriormente.

Ao longo de toda a execução do trabalho, foram surgindo algumas barreiras, as quais conseguiu-se ultrapassar, sendo que a maior limitação que se encontrou foi durante a obtenção de alojamentos, devido à velocidade da internet de cada elemento do grupo. Por este motivo, foi necessário colocar Sys.sleep(X) em diferentes partes do *script*, de forma a que as máquinas pessoais pudessem efetuar a pesquisa, depois de todos os elementos da página *web* terem sido devidamente carregados. A não colocação desta linha de código, gerava o seguinte erro “ `Error in webElem[[j]] : subscript out of bounds`”.

6. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Data Science Project Management. 2020. *CRISP-DM - Data Science Project Management.* [online]. Disponível em: <https://www.datascience-pm.com/crisp-dm-2/>[Acesso em 30 de novembro de 2021].

[2] Harrison, J . 2020. R Bindings for 'Selenium WebDriver . Package 'RSelenium' . . Disponível em: <https://cran.r-project.org/web/packages/RSelenium/RSelenium.pdf> . [Acesso em 10 de novembro de 2021].

[3] Lusa, D. 2020. Diário de Notícias. Número de visitantes em Hong Kong cai 14% em meio ano de protestos. [online] Disponível em: <https://www.dn.pt/mundo/numero-de-visitantes-em-hong-kong-cai-14-em-meio-ano-de-protestos-11712676.html>[Acesso em 26 de novembro de 2021].

[4] Mccrain, J . 2020. Tutorial RSelenium . Tutoriais / web_scraping_R_selenium . [online]. Disponível em: <http://joshuamccrain.com/tutorials/web_scraping_R_selenium.html> [Acesso em 15 de novembro de 2021].

[5] Medium. 2020. *Bagging And Boosting Method*. [online] Disponível em: <https://medium.com/@ruhi3929/bagging-and-boosting-method-c036236376eb> [Acesso em 25 de novembro de 2021].

[6] Kim, J .2020. RSelenium . RDocumentação . Kim, J . (2020) . RSelenium . RDocumentação . [online] Disponível em: <https://www.rdocumentation.org/packages/RSelenium/versions/1.7.7>[Acesso em 10 de novembro de 2021].

7. ANEXOS

Anexo I:

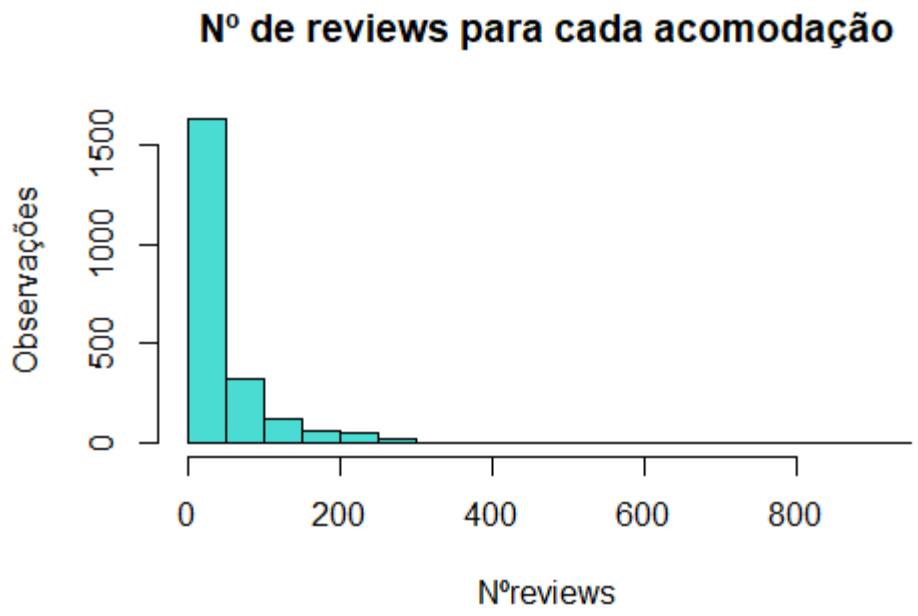

Figura 34 - Nº de reviews para cada acomodação.

Anexo II:

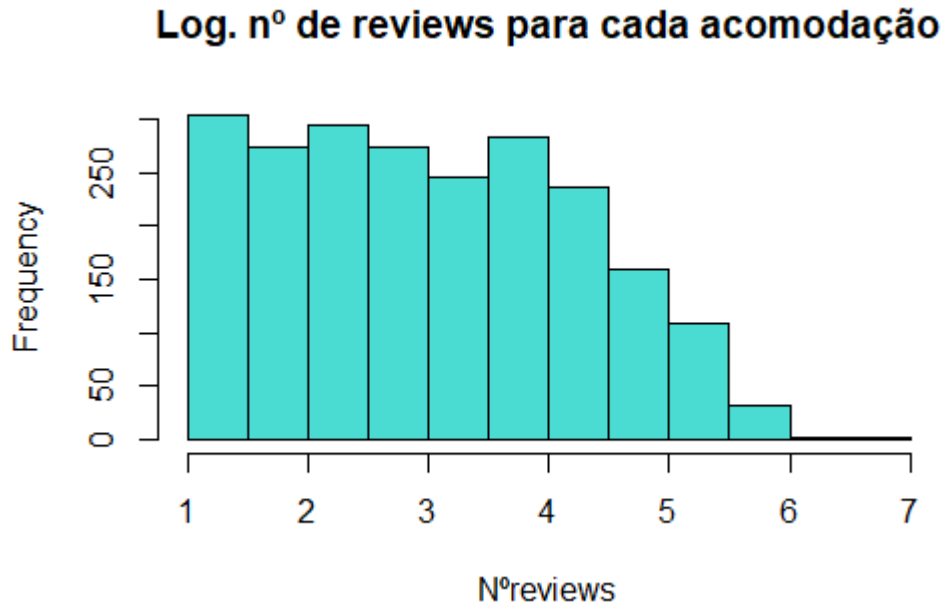

Figura 35 - Logarítmo do nº de reviews para cada acomodação.

Anexo III:

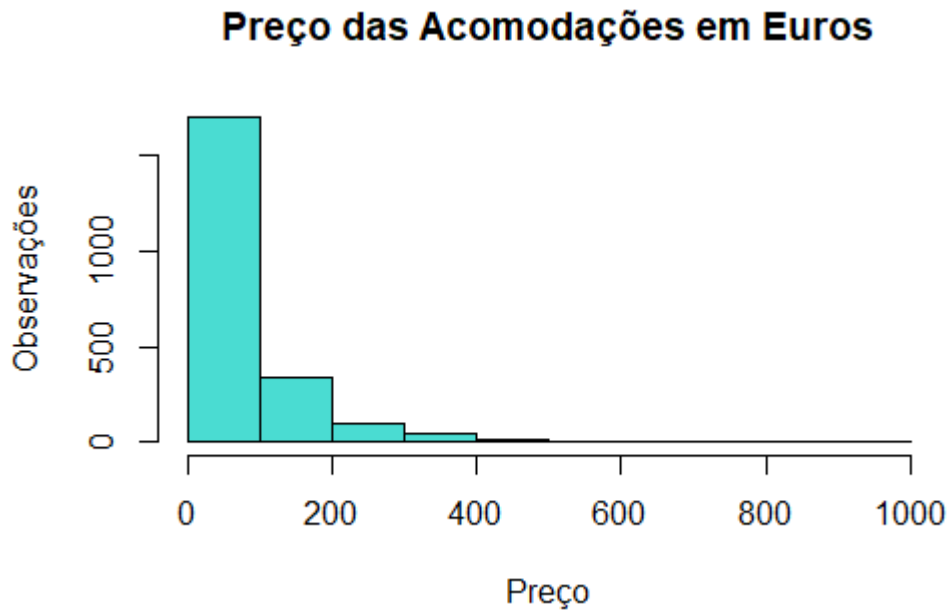

Figura 36 - Preço das acomodações em Euros.

Anexo IV:

```
Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.645122   0.062811  73.953  < 2e-16 ***
property_typehome              0.022767   0.024019   0.948 0.343306
property_typeprivate room     -0.077101   0.027707  -2.783 0.005436 **
property_typeserviced apartment  0.004542   0.038949   0.117 0.907173
property_typeshared room      -0.009179   0.043289  -0.212 0.832090
lognreviews                    0.023076   0.005107   4.519 6.55e-06 ***
guest                         -0.019455   0.002853  -6.818 1.19e-11 ***
nrbath                         0.001613   0.004793   0.337 0.736456
superhost1                     0.225813   0.013695  16.489  < 2e-16 ***
studio1                       -0.071464   0.020802  -3.435 0.000603 ***
kitchen1                       0.083260   0.016711   4.982 6.77e-07 ***
wifi1                         -0.023122   0.042841  -0.540 0.589438
tv1                           -0.015841   0.015177  -1.044 0.296729
elevator1                     -0.006128   0.015190  -0.403 0.686670
washer1                        0.053303   0.018423   2.893 0.003849 **
dryer1                         0.007916   0.015978   0.495 0.620350
ac1                           -0.046694   0.030234  -1.544 0.122633
balcony1                       0.034159   0.016858   2.026 0.042855 *
luggage_do1                    0.020192   0.014251   1.417 0.156662
hair_dryer1                    0.024446   0.015290   1.599 0.110002
pet1                           0.027721   0.020317   1.364 0.172569
fire_ext1                     -0.097861   0.104701  -0.935 0.350059
first_aid1                     0.014357   0.170006   0.084 0.932705
breakfast1                     0.128984   0.050739   2.542 0.011087 *
parking1                       0.049686   0.026758   1.857 0.063467 .
smoking1                      -0.118808   0.023834  -4.985 6.69e-07 ***
bath_private1                 -0.094036   0.022712  -4.140 3.60e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2875 on 2187 degrees of freedom
Multiple R-squared:  0.2706,    Adjusted R-squared:  0.2619
F-statistic: 31.21 on 26 and 2187 DF,  p-value: < 2.2e-16
```

Tabela 18 - Dataframe2.

Anexo V:

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.646624   0.062323  74.558  < 2e-16 ***
property_typehome             0.021265   0.023968   0.887 0.375052
property_typeprivate room    -0.078627   0.027666  -2.842 0.004525 **
property_typeserviced apartment  0.001588   0.038828   0.041 0.967387
property_typeshared room     -0.007290   0.043246  -0.169 0.866150
lognreviews                   0.023095   0.005102   4.526 6.32e-06 ***
guest                        -0.019643   0.002802  -7.011 3.13e-12 ***
superhost1                    0.227527   0.013594  16.738  < 2e-16 ***
studio1                      -0.072912   0.020743  -3.515 0.000449 ***
kitchen1                      0.083986   0.016694   5.031 5.28e-07 ***
wifi1                        -0.030299   0.042317  -0.716 0.474071
elevator1                    -0.006717   0.015154  -0.443 0.657615
washer1                       0.053112   0.018361   2.893 0.003857 **
dryer1                        0.007824   0.015968   0.490 0.624184
ac1                          -0.047537   0.030216  -1.573 0.115812
balcony1                      0.034959   0.016838   2.076 0.037992 *
luggage_do1                   0.020604   0.014197   1.451 0.146844
hair_dryer1                   0.026521   0.015139   1.752 0.079942 .
pet1                          0.028554   0.020283   1.408 0.159337
fire_ext1                    -0.099251   0.104673  -0.948 0.343132
first_aid1                    0.016348   0.169957   0.096 0.923381
breakfast1                    0.126972   0.050696   2.505 0.012331 *
parking1                      0.049606   0.026728   1.856 0.063596 .
smoking1                     -0.118649   0.023826  -4.980 6.87e-07 ***
bath_private1                -0.097903   0.022422  -4.366 1.32e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2875 on 2189 degrees of freedom
Multiple R-squared:  0.2702,    Adjusted R-squared:  0.2622
F-statistic: 33.77 on 24 and 2189 DF,  p-value: < 2.2e-16
```

Tabela 19 - Dataframe3.

Anexo VI:

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  4.6174172  0.0514887  89.678  < 2e-16 ***
property_typehome            0.0215694  0.0239546   0.900 0.367991
property_typeprivate room   -0.0788169  0.0276522  -2.850 0.004409 **
property_typeserviced apartment  0.0008393  0.0388008   0.022 0.982744
property_typeshared room    -0.0073780  0.0432261  -0.171 0.864487
lognreviews                  0.0229083  0.0050961   4.495 7.31e-06 ***
guest                       -0.0195741  0.0027997  -6.992 3.60e-12 ***
superhost1                   0.2268495  0.0135685  16.719  < 2e-16 ***
studio1                     -0.0717741  0.0206903  -3.469 0.000533 ***
kitchen1                     0.0839869  0.0166074   5.057 4.61e-07 ***
washer1                      0.0516572  0.0182804   2.826 0.004759 **
dryer1                       0.0078971  0.0159624   0.495 0.620841
ac1                         -0.0511079  0.0299387  -1.707 0.087948 .
balcony1                     0.0354713  0.0168204   2.109 0.035073 *
luggage_do1                  0.0202308  0.0141828   1.426 0.153886
hair_dryer1                  0.0263526  0.0150790   1.748 0.080668 .
pet1                         0.0297869  0.0202275   1.473 0.141003
fire_ext1                   -0.1005901  0.1046327  -0.961 0.336475
first_aid1                   0.0147951  0.1698871   0.087 0.930610
breakfast1                   0.1305719  0.0504984   2.586 0.009783 **
parking1                     0.0501742  0.0266575   1.882 0.059944 .
smoking1                    -0.1168297  0.0237241  -4.925 9.09e-07 ***
bath_private1               -0.0991308  0.0223700  -4.431 9.82e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2874 on 2191 degrees of freedom
Multiple R-squared:  0.2699,    Adjusted R-squared:  0.2626
F-statistic: 36.82 on 22 and 2191 DF,  p-value: < 2.2e-16
```

Tabela 20 - Dataframe4.

Anexo VII:

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.613005   0.051402  89.743  < 2e-16 ***
property_typehome             0.021307   0.023955   0.889 0.373846
property_typeprivate room    -0.073974   0.027438  -2.696 0.007070 **
property_typeserviced apartment  0.008245   0.038418   0.215 0.830094
property_typeshared room     -0.002798   0.043103  -0.065 0.948243
lognreviews                   0.023639   0.005069   4.663 3.30e-06 ***
guest                        -0.019362   0.002792  -6.935 5.31e-12 ***
superhost1                    0.227875   0.013503  16.875  < 2e-16 ***
studio1                      -0.071218   0.020619  -3.454 0.000563 ***
kitchen1                      0.082064   0.016559   4.956 7.76e-07 ***
washer1                       0.057158   0.016795   3.403 0.000678 ***
ac1                          -0.049018   0.029811  -1.644 0.100260
balcony1                      0.037387   0.016593   2.253 0.024345 *
hair_dryer1                   0.026978   0.014881   1.813 0.069974 .
pet1                          0.030101   0.020175   1.492 0.135851
fire_ext1                    -0.107802   0.104529  -1.031 0.302507
first_aid1                    0.008435   0.169843   0.050 0.960395
breakfast1                    0.125367   0.050361   2.489 0.012872 *
parking1                      0.050217   0.026615   1.887 0.059320 .
smoking1                     -0.120322   0.023512  -5.117 3.37e-07 ***
bath_private1                -0.097296   0.022321  -4.359 1.37e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2874 on 2193 degrees of freedom
Multiple R-squared:  0.2692,    Adjusted R-squared:  0.2625
F-statistic: 40.39 on 20 and 2193 DF,  p-value: < 2.2e-16
```

Tabela 21 - Dataframe5.

Anexo VIII:

```
Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.622320   0.051058  90.531  < 2e-16 ***
property_typehome             0.020211   0.023942   0.844 0.398676
property_typeprivate room    -0.077437   0.027369  -2.829 0.004706 **
property_typeserviced apartment  0.004264   0.038329   0.111 0.911429
property_typeshared room     -0.007591   0.043009  -0.176 0.859922
lognreviews                   0.023157   0.005055   4.581 4.89e-06 ***
guest                        -0.019504   0.002790  -6.990 3.63e-12 ***
superhost1                    0.228285   0.013502  16.908  < 2e-16 ***
studio1                      -0.071152   0.020618  -3.451 0.000569 ***
kitchen1                      0.084645   0.016484   5.135 3.07e-07 ***
washer1                       0.057463   0.016774   3.426 0.000624 ***
ac1                          -0.051421   0.029783  -1.727 0.084393 .
balcony1                      0.036678   0.016588   2.211 0.027125 *
hair_dryer1                   0.024467   0.014791   1.654 0.098244 .
breakfast1                    0.126738   0.050359   2.517 0.011917 *
parking1                      0.052141   0.026586   1.961 0.049980 *
smoking1                     -0.116514   0.023400  -4.979 6.88e-07 ***
bath_private1                -0.097722   0.022303  -4.382 1.23e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2874 on 2196 degrees of freedom
Multiple R-squared:  0.2681,    Adjusted R-squared:  0.2624
F-statistic: 47.31 on 17 and 2196 DF,  p-value: < 2.2e-16
```

Tabela 22 - Dataframe6.

Anexo IX:

	MinTemp	MaxTemp	Rainfall	WindGustSpeed	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainTomorrow
1	17.9	35.2	0.0	48.0	6.0	20.0	20.0	13.0	1006.3	1004.4	2.0	5.0	26.6	33.4	0
2	18.4	28.9	0.0	37.0	19.0	19.0	30.0	8.0	1012.9	1012.1	1.0	1.0	20.3	27.0	0
3	15.5	34.1	0.0	30.0	16.861751152073733	7.0	43.575268817204304	7.0	1012.4350230414742	1011.6	0.0	1.0	24.347926267281107	32.7	0
4	19.4	37.6	0.0	46.0	30.0	15.0	42.0	22.0	1012.3	1009.2	1.0	6.0	28.7	34.9	0
5	21.9	38.4	0.0	31.0	6.0	6.0	37.0	22.0	1012.7	1009.1	1.0	5.0	29.1	35.6	0
6	24.2	41.0	0.0	35.0	17.0	13.0	19.0	15.0	1010.7	1007.4	1.0	6.0	33.6	37.6	0
7	27.1	36.1	0.0	43.0	7.0	20.0	26.0	19.0	1007.7	1007.4	8.0	8.0	30.7	34.3	0
8	23.3	34.0	0.0	41.0	17.0	19.0	33.0	15.0	1011.3	1009.9	3.0	1.0	25.0	31.5	0
9	16.1	34.2	0.0	37.0	15.0	6.0	25.0	9.0	1013.3	1009.2	1.0	1.0	20.7	32.8	0
10	19.0	35.5	0.0	48.0	30.0	9.0	46.0	28.0	1008.3	1004.0	1.0	5.0	23.4	33.3	0
11	19.7	35.5	0.0	41.0	15.0	17.0	61.0	14.0	1007.9	1005.8	1.0	5.0	24.0	33.6	0
12	20.9	37.8	0.0	30.0	11.0	7.0	27.0	9.0	1012.6	1010.1	0.0	1.0	29.8	36.4	0
13	23.9	39.1	0.0	39.0	24.0	9.0	40.0	15.0	1013.6	1010.4	0.0	2.0	29.1	37.0	0
14	24.9	41.2	0.0	43.0	17.0	11.0	25.0	15.0	1012.9	1010.1	1.0	3.0	31.5	38.1	0
15	25.2	40.5	0.0	44.0	13.0	22.0	24.0	15.0	1012.4	1009.0	4.0	6.0	31.4	37.8	0
16	21.6	34.2	0.0	44.0	17.0	19.0	19.0	8.0	1014.1	1012.3	0.0	0.0	25.0	32.2	0
17	18.4	31.8	0.0	33.0	17.0	15.0	25.0	5.0	1016.3	1013.8	0.0	1.0	19.9	30.3	0
18	17.9	34.2	0.0	61.0	22.0	17.0	46.0	19.0	1016.4	1013.5	1.0	2.0	21.6	32.2	0
19	21.4	37.5	0.0	43.0	26.0	9.0	34.0	29.0	1013.1	1009.6	7.0	6.0	26.2	34.1	1
20	23.3	39.4	4.8	59.0	19.0	17.0	54.0	14.0	1011.1	1008.5	1.0	7.0	27.0	37.0	0
21	25.4	33.5	0.0	46.0	9.0	28.0	46.0	52.0	1012.0	1009.8	4.0	7.0	28.9	29.7	0
22	21.8	30.7	0.0	56.0	24.0	19.0	71.0	63.0	1008.6	1006.2	7.0	7.0	24.4	27.3	1
23	20.3	36.0	18.0	94.0	13.0	7.0	89.0	50.0	1008.6	1006.7	7.0	4.0	24.7	33.4	1
24	22.1	34.7	8.6	50.0	11.0	15.0	46.0	23.0	1008.6	1008.3	2.0	6.0	28.1	33.2	0
25	19.7	37.3	0.0	28.0	13.0	6.0	19.0	10.0	1013.1	1011.8	1.0	1.0	26.4	35.0	0
26	23.8	39.9	0.0	31.0	17.0	7.0	50.0	16.0	1014.6	1012.1	0.0	1.0	28.7	38.7	0

Tabela 23 - Dataset "Rain in Australia".