
Projeto

Data Warehouse, OLAP e Data Mining



Inteligência no Negócio

Da autoria de:

Ricardo Santiago / N°: 2020219352

Ricardo Silva / N°: 2020227184



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Índice

Introdução.....	4
Definição do modelo de dados para a DW	4
Seleção de tecnologia	5
Descrição das fontes dos dados	6
Plano de ETL da solução.....	6
Fase de preparação.....	7
Fases do ETL	7
Maiores desafios na implementação	9
Métricas	9
Automatização do processo ETL.....	9
Apresentação dos dados relativamente a OLAP	10
Descrição da análise realizada	10
Painel de Introdução	10
Painel Detalhado	11
Descobertas iniciais e a sua utilidade para suporte à decisão	13
Estratégias para otimizar a DW e as pesquisas OLAP	15
Técnicas para aceder à informação	15
Etapas Futuras	16
Conclusão – Parte 1	16
Parte 2 – Data Mining	17
Preparação dos dados.....	17
Fonte dos dados	17
Ferramentas utilizadas (software)	17
Alteração no Pentaho	17
Remoção de outliers	18
Adição da categoria de preço.....	18
Recomendação de preço (Estudo de classificação)	19
Objetivos	19
Processo de escolha do algoritmo	19
Escolha dos atributos	19

Hyperparameter search/choice	20
Testagem do modelo para 4 categorias (baseline test)	21
Testagem do modelo usando estratégia de regressão	23
Testagem do modelo para 5 categorias (melhoria contínua)	23
Testagem do modelo para 6 categorias (melhoria contínua)	24
Testagem do modelo para 10 categorias (melhor resultado)	25
Resultados obtidos com integração na interface GUI.....	27
Exemplificação com a cidade do Porto	27
Previsão de ocupação (Estudo de análise de séries temporais)	28
Objetivos do estudo	28
Gráficos temporais	29
Escolha do algoritmo - ARIMA	32
Processo de Data Mining	33
Processo de procura e escolha dos melhores parâmetros	34
Exemplificação usando a cidade do Porto	35
Comparação dos resultados com outras fontes	36
Resultados obtidos e Integração no produto final.....	37
Adição de pontos de interesse próximos.....	37
Conclusão – Parte 2.....	38

Índice de Figuras

Figura 1 - Modelo STAR desenvolvido	5
Figura 2 - Processo de ETL.....	7
Figura 3 - Visualização do Painel de Introdução	10
Figura 4 - Visualização do Painel Detalhado	11
Figura 5 - Alteração no Pentaho.....	18
Figura 6 - Correlação entre numericprice e atributos do dataset	20
Figura 7 - Hyperparameter choice/select with grid_search	21
Figura 8- 30-Run Average DT Confusion Matrix and Classification Report.....	22
Figura 9- 30-Run Average RandomForest Confusion Matrix and Classification Report	22
Figura 10- 30-Run Average DT Confusion Matrix and Classification Report.....	24
Figura 11- 30-Run Average DT Confusion Matrix and Classification Report.....	25
Figura 12- 30-Run Average DT Classification Report	26
Figura 13 - GUI Interface for Price Recommemdatation Lisbon	27
Figura 14-GUI Interface for Price Recommemdatation Porto.....	28
Figura 21 - Previsão da média de avaliações mensais por alojamento em Lisboa	33
Figura 26 - Obtenção do número de pontos de interesse próximos	37
Figura 27 - Correlação do atributo da contagem dos pontos de interesse	38

Introdução

A Airbnb começou em 2008, quando dois designers que tinham um espaço extra hospedaram três viajantes que procuravam um alojamento. Agora, milhões de anfitriões e hóspedes já criaram contas na Airbnb. É com esta ideia que decidimos desenvolver o projeto acerca desse mercado, mais propriamente, o seu mercado em Lisboa. Através da exploração dos dados disponibilizados pela empresa, é possível ganhar insights valiosos sobre o modo de operação das atividades da empresa.

Além disso, permite coletar conclusões sobre as preferências dos viajantes, comportamentos e tendências do mercado e desempenho dos anfitriões, permitindo assim que a empresa tome decisões informadas que impulsionam o seu crescimento e a excelência operacional.

O negócio consiste no arrendamento de alojamentos na cidade de Lisboa, apontando para dois segmentos de mercado: os donos dos alojamentos na cidade e pessoas que os queiram arrendar. Deste modo, a Airbnb pode oferecer recomendações personalizadas, melhorar a qualidade das listagens, ajustar preços de forma competitiva e antecipar demandas sazonais, tudo isso contribuindo para uma experiência mais satisfatória e lucrativa para todas as partes envolvidas. Também é importante realçar a proposta de valor para o anfitrião do serviço que fica a poder aceder a recomendações mais apropriadas para o tipo de alojamento que pretende listar.

Definição do modelo de dados para a DW

Optámos por um modelo de dados que permitisse otimizar o processo de análise posterior e que permitisse a integração de novos dados caso isso se verificasse necessário.

Como é possível observar na imagem na página seguinte, escolheu-se apenas usar um modelo STAR, sendo este composto pela tabela dos factos (*facts_table*) que contém os anúncios dos alojamentos. Esta relaciona-se com as diversas dimensões escolhidas. Cada uma destas representa um atributo diferente de cada anúncio na plataforma, tendo este uma série de dados que o caracterizam as dimensões escolhidas foram: uma para caracterizar o anfitrião do alojamento (*dim_hosts*), uma para caracterizar o tipo de combinação de características do alojamento (*dim_room_type_comb*) e uma para caracterizar a localização, mais propriamente o município do distrito de Lisboa em que se situa o alojamento (*dim_neighbourhood_group*). Assim, cada entidade representada nas diversas dimensões, relaciona-se com uma ou mais entidades representadas na tabela dos factos.

De notar relativamente à granularidade dos factos que cada *neighbourhood_group_cleansed* contém diversos *neighbourhood_cleansed*, porém escolhemos destacar em dimensão apenas os municípios.



Figura 1 - Modelo STAR desenvolvido

Seleção de tecnologia

Relativamente às ferramentas a utilizar para desenvolver o projeto, procuramos priorizar as soluções recomendadas durante as aulas.

a) Para guardar os dados: *PostgreSQL* (pgAdmin 4)

Recorremos a um software que nos era familiar devido à sua utilização em outras disciplinas ao longo do curso, além de que o consideramos eficiente nos processos de armazenamento de dados para posterior utilização.

b) Para fazer o processo de ETL (Extraction-Transformation-Loading): *Pentaho*

Sendo um software específico para o processo de extração, transformação e carregamento de dados para a DW, fornecia todas as ferramentas necessárias para a realização dessa etapa do projeto. Permite uma modelação visual dos componentes do processo e uma definição do flow do mesmo. Comparando com outras soluções, esta realçou-se também devido ao facto de ter uma versão Community Edition.

c) Online analytical processing: *Tableau*

Um dos aspetos mais importantes foi a questão de estar disponibilizada uma licença para estudantes de modo a utilizar o software. Este permite criar visualizações que mostram os dados, assim como apresentar graficamente as descobertas acerca dos dados. Também é possível disponibilizar o trabalho desenvolvido, promovendo a partilha das conclusões para o público em geral.

Descrição das fontes dos dados

Usámos o dataset relacionado com o projeto “Inside Airbnb” que fornece dados sobre o impacto do Airbnb nas comunidades residenciais. Dos dados fornecidos pela plataforma, recorreremos ao ficheiro *listings.csv* referente à cidade de Lisboa no dia 17 de dezembro de 2023. Este contém um total de 22752 publicações referentes aos alojamentos que estavam listados na plataforma nesse dia, contando com 75 atributos acerca dos mesmos. Assim, cada anúncio contém uma série de informação sobre o mesmo, permitindo caracterizar as suas especificações e desempenho ao longo da permanência na plataforma. É possível aceder ao repositório do dataset [aqui](#).

Além disso, recorreremos ao API público disponibilizado pela KB Geo. Este permite obter a distância de um ponto até ao ponto mais próximo da linha costeira através da latitude e longitude desse ponto.

Plano de ETL da solução

Alguna parte dos dados que estamos a considerar já têm algum tratamento prévio no dataset disponibilizado (por exemplo a criação do atributo *neighbourhood_cleansed* que se refere ao conteúdo relativo à localização, mas já tratado para corresponder a uma certa lista de possibilidades).

Apesar disso, são necessárias algumas etapas de transformação dos dados de modo que se obtenham os atributos que pretendemos e que os dados se tornem realmente úteis para a fase de análise posterior.

Fase de preparação

Antes de os dados serem carregados para o software de ETL, tivemos a precaução de configurar uma base de dados local no pgAdmin de modo que esta receba os dados após sofrerem as transformações. De relembrar que também é necessário executar os comandos SQL relativos à criação das tabelas (comandos esses que são sugeridos pelo próprio Pentaho).

Em adição a isso, foi necessário obter uma chave API do serviço para que fosse possível fazer chamadas durante a execução do processo.

Fases do ETL

Segue-se a imagem representativa do processo de ETL desenvolvido.



Figura 2 - Processo de ETL

Os seguintes pontos descrevem com mais detalhe cada uma das etapas mostradas na imagem.

- 1) Leitura do ficheiro local listings.csv que contém informação sobre os anúncios, sendo possível escolher como é que cada atributo irá ser lido do ficheiro. Alguns tamanhos de atributos tiveram de ser aumentados de modo a ficarem corretamente formatados no processo de armazenamento na base de dados.
- 2) Seleção dos atributos a usar, visto que nem todos irão ser úteis para usar. Nesta etapa selecionou-se apenas 57 atributos dos 75 disponíveis inicialmente.
- 3) Filtragem de linhas para que apenas passem linhas que não contenham valor null em nenhum dos seguintes atributos fundamentais: *id*, *host_id*, *neighbourhood_group_cleansed*, *price*, *beds* e *room_type*.

-
- 4) Reposição de valores null pelo valor de string “N/A” nos atributos *neighborhood_overview* e *host_location*.
 - 5) Criação de um atributo extra para categorizar o *review_scores_rating* em apenas 5 valores (de 1 a 5).
 - 6) Conversão dos seguintes atributos para tipo boolean: *is_superhost*, *has_profile_pic*, *identity_verified*, *instant_bookable_bool* e *has_availability_bool*.
 - 7) Conversão dos seguintes atributos para tipo integer: *numericHost_Response_rate* e *numericHost_Acceptance_rate*.
 - 8) Obtenção do número de camas (atributo *beds*) através da informação contida no atributo *name*.
 - 9) Obtenção do número de casas de banho (atributo *bathrooms*) através da informação contida no atributo *bathrooms_text*.
 - 10) Conversão o preço (atributo *price*) para integer, criando um atributo (*numericPrice*) com essa informação.
 - 11) Carregamento dos dados para a tabela *dim_hosts*. Esta tem como atributo chave o *host_id* e contém todos os atributos relativamente exclusivos ao próprio anfitrião.
 - 12) Carregamento dos dados para a tabela *Dim_Room_Type_Comb*. Esta tem apenas atributos chave, sendo composta pelos campos que caracterizam um alojamento e pelos quais os clientes estão acostumados a fazer a pesquisa.
 - 13) Seleção dos atributos para a tabela dos factos, sendo esta composta por 39 atributos.
 - 14) Carregamento dos dados para a tabela *facts_table*.
 - 15) Agrupamento por *neighbourhood_group_cleansed* de modo a ser possível calcular a longitude e latitude (centroide) de cada município.
 - 16) Seleção dos atributos recebidos (*avg_lat* e *avg_long*).
 - 17) Ordenação dos dados por *neighbourhood_group_cleansed*.
 - 18) Obtenção da distância até à costa de cada município por meio de uma API.
 - 19) Converter a latitude e longitude médias calculadas para terem apenas 5 casas decimais.
 - 20) Carregamento dos dados para a tabela *dim_neighbourhood_group*. Esta tem como atributo chave o *neighbourhood_group_cleansed* e contém os centroides de cada município assim como a sua distância até à costa.

De realçar também a necessidade de executar certos comandos SQL referentes à indicação das chaves (primárias e estrangeiras) e à eliminação de uma linha a null gerada pelo Pentaho em cada tabela no final do processo. Isto poderia ser automatizado criando uma nova etapa no Pentaho após o final do processo de ETL referido anteriormente.

Maiores desafios na implementação

Uma das características desta implementação é o facto das chamadas API demorarem alguns segundos a serem executadas, o que aumenta o tempo de duração do processo. Isto podia ser melhorado através de um armazenamento temporário de resultados para um documento externo que contivesse os resultados da primeira execução, ao qual as seguintes recorreriam a esse documento.

Visto que foi a nossa primeira utilização deste programa, não foi um processo intuitivo de aprendizagem pois consideramos que este não é muito intuitivo para se perceber de forma simples como proceder quando se pretende certo resultado. Ainda sim, consideramos que a implementação dos diversos scripts em JavaScript foi a etapa mais desafiadora.

Métricas

Tamanho original dos dados: 30 MB (Lisboa).

Tamanho dos dados no primeiro carregamento: 26MB (Lisboa).

Tamanho dos dados nos carregamentos seguintes: 46MB (caso se adicionasse informação referente ao Porto).

Tempo decorrido no primeiro carregamento: 23 segundos (sendo que 17 segundos são referentes ao API).

Tempo usado para cada update: 22 segundos (considerando caso de adição de informação referente ao Porto).

Automatização do processo ETL

O processo está todo automatizado, desde a leitura do ficheiro de entrada de dados até à inserção dessa informação nas tabelas referentes à data warehouse. Assim, seria apenas necessário atualizar o ficheiro de entrada com as atualizações necessárias ou alterar o próprio ficheiro de entrada. De realçar que é necessário manter a formatação presente no ficheiro original usado, de modo que não exista atributos novos ou conflito entre atributos.

Apesar de estarmos a considerar a cidade de Lisboa e que um anúncio de um novo alojamento não é algo que ocorre com elevada frequência ou que suscite grande interesse de atualização imediata da DW, o sistema poderia dar suporte a atualizações com qualquer frequência pretendida pelos administradores do sistema, visto que todas as tabelas estão preparadas para receber novos dados e integrá-los com os existentes no momento.

De realçar que o sistema está preparado para acrescentar dados acerca de outras cidades, desde que estes sigam o formato tal como referido anteriormente. Para isso basta obter o ficheiro *listings.csv* referente a outras cidades disponíveis no repositório e selecionar como input do processo. O teste que realizámos foi com a possível adição da cidade do Porto e foi bem-sucedido.

Apresentação dos dados relativamente a OLAP

Os dados são apresentados através de dois painéis do *Tableau*. Um tem diversas informações acerca dos alojamentos, podendo estas ser filtradas através de intervalos referentes às características inerentes aos mesmos. O outro painel contém uma visualização de um mapa dos alojamentos e alguma informação que pode ser filtrada pelo município pretendido.

Deste modo, o utilizador final tem dois quadros com que pode interagir de modo a obter a informação pretendida através de diversas perspetivas.

Descrição da análise realizada

Painel de Introdução

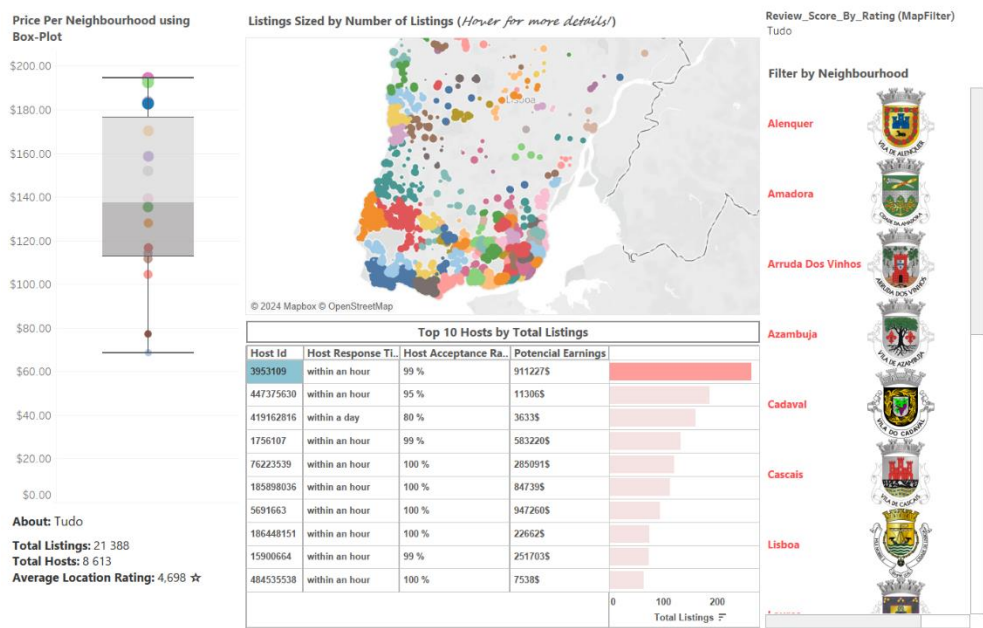


Figura 3 - Visualização do Painel de Introdução

Este painel serve como um painel introdutório que permite ter uma visualização mais geral dos diferentes municípios de Lisboa, a sua localização num mapa, o preço médio por noite praticado em cada um e outros fatores como o número de listings, hosts e ainda o rating associado à localização. Além disso, mostra o top 10 de hosts com mais listings em cada município.

Como podemos observar acima, os dados mostrados podem ser filtrados por município, sendo que o mapa pode ainda ser filtrado para mostrar apenas alojamentos pertencentes a uma certa categoria de avaliações médias. Assim, estes filtros permitem alterar a visualização disponível nas seguintes planilhas:

➤ Price Per Neighbourhood using Box-Plot

Mostra o preço médio por município, sendo que quando estão todos selecionados, é possível ter ideia de comparação entre estes.

➤ About: <Neighbourhood Group Cleansed>

Informações gerais para cada município selecionado como o número total de anúncios, o número total de anunciantes e a classificação média da localização.

➤ Listings Sized by Number of Listings (Hover for more details!)

Mapa contendo os diversos alojamentos anunciados, sendo que cada cor representa localizações mais específicas que os municípios e o tamanho destas cores é proporcional ao número de listings associados a cada localização. Ao interagir com o mapa é possível ver os detalhes de cada um dos alojamentos e do seu anfitrião.

➤ Top 10 Hosts by Total Listings

Lista dos anfitriões com mais alojamentos anunciados naquele município. Além disso, mostra uma estimativa de quando é que já podem ter recebido através da multiplicação da média dos preços praticados nos diversos alojamentos com a soma do número de reviews de todos os alojamentos publicados. Mostra também dados suplementares como o tempo de resposta associado a esse host e a taxa de aceitação do mesmo.

Painel Detalhado

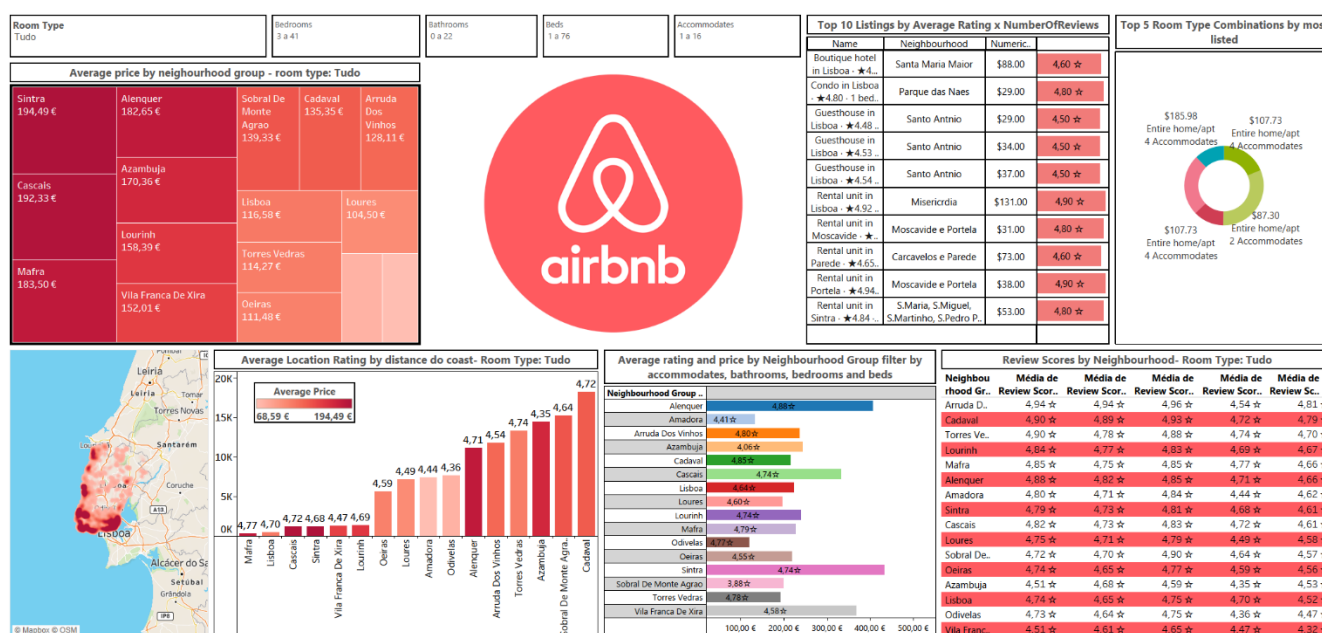


Figura 4 - Visualização do Painel Detalhado

Este painel serve para dar aos nossos utilizadores uma visão mais específica das diferentes dimensões que estão associadas às listings e que podem afetar tanto o preço como o review score associado a estas.

Como podemos observar acima, este painel apresenta a possibilidade de filtragem por tipo de alojamento, número de quartos, número de casas de banho, número de camas e capacidade do alojamento (número de hóspedes). No entanto, deve-se realçar que estes filtros não se aplicam a todas as planilhas presentes no painel. Deste modo apresentamos as seguintes planilhas:

➤ Average price by neighbourhood group - room type: <Room Type>

Heatmap que mostra o preço médio de cada município para o tipo de alojamento filtrado, onde a cor mais escura indica um maior preço médio.

➤ Mapa de densidade para o preço médio por neighbourhood

Serve para auxiliar de forma visual a planilha anterior, sendo que a intensidade da cor associada ao preço médio de cada município, aumenta de forma proporcional com o preço.

➤ Top 10 Listings by Average Rating x NumberOfReviews

Representação estática dos 10 alojamentos melhor avaliados, usando a fórmula $[\text{Review Scores Rating}] * [\text{Reviews Per Month}]$. Além disso, mostra qual o nome da listing, qual o seu município, o seu preço por noite e ainda o review score rating.

➤ Top 5 Room Type Combinations by most listed

Representação estática das 5 combinações de características de alojamentos mais disponíveis nos anúncios, contendo informação do preço médio e de cada um dos elementos referentes a essa combinação como número de camas, casas de banho, quartos e capacidade de alojamento.

➤ Average Location Rating by distance do coast- room type: <room type>

Representação por gráficos de barras em que é possível perceber como é que a distância a costa de um município faz variar o review score atribuído a localização.

➤ Average rating and price by Neighbourhood Group filter by accommodates, bathrooms, bedrooms and beds

Representação do preço e classificação geral média de cada município, sendo possível filtrar por diversos atributos referentes ao alojamento como número de casas de banho, quartos, camas e capacidade de alojamento.

➤ Review Scores by Neighbourhood- Room Type: <Room Type>

Classificações médias das várias categorias de avaliações para cada um dos municípios, permitindo filtrar também por tipo de alojamento.

Descobertas iniciais e a sua utilidade para suporte à decisão

Nesta secção do relatório estão descritas as principais descobertas obtidas através da análise e representação dos dados, assim como a utilidade dessas descobertas para o negócio em questão. Estas são reportadas no formato de Questão – Resposta (R) – Planilha (P) - Utilidade_para_Utilizador (U).

1. Qual é o município com o preço médio dos alojamentos mais baixo? E mais elevado?

R: O município lisboeta com o preço médio mais baixo é Amadora e com o preço mais elevado é Sintra. O preço médio nestes municípios distancia-se por 80,9\$.

P: Average price by neighbourhood group - room type: <Room Type>

U: Com esta informação, um utilizador anfitrião que use a plataforma consegue ter noção do preço médio praticado em cada zona habitacional. Com isso, pode praticar preços mais adequados e mais atrativos para os potenciais hóspedes. Além disso, possibilita que, no caso de pretender investir numa compra de uma habitação para posterior alojamento, perceba a rentabilidade do investimento e o tempo de retorno previsto.

2. Em que zonas é que situa a maioria dos anúncios da plataforma?

R: Na zona costeira da cidade, mais propriamente na zona sudoeste da cidade.

P: Mapa de densidade para o preço médio por neighbourhood

U: Com esta informação, o anfitrião pode ter ideia da concentração de anúncios, podendo ter noção da concorrência que poderá vir a ter. Por outro lado, a plataforma pode perceber em que áreas residenciais têm maior influência. Assim, no caso de optar por dinamizar um evento de promoção do serviço, já se tem ideia de onde é que este tipo de eventos pode ser mais eficaz. Também pode vir a focar uma maior porção dos seus produtos publicitários acerca das zonas em que a oferta é maior, visto que essa zona é mais provável satisfazer a necessidade de disponibilidade um maior número de hóspedes.

3. Quais os anúncios da cidade mais bem-sucedidos na plataforma?

R: Top 10 alojamentos tendo em conta as avaliações dadas pelos utilizadores e a contagem de avaliações dos mesmos

P: Top 10 Listings by Average Rating x NumberOfReviews

U: Com esta informação, a plataforma pode analisar as características inerentes a estes alojamentos e ao serviço providenciado pelo anfitrião de modo a descobrir o que é que se destaca para estes serem mais bem-

sucedidos que os outros. Com isto, pode detetar padrões nos serviços que agradam aos hóspedes (por exemplo se o anfitrião deixa uma mensagem no alojamento para ser vista à chegada dos hóspedes ou a disponibilidade de resposta do anfitrião), apresentando assim sugestões aos anfitriões que pretendam anunciar os seus alojamentos de modo que ofereçam melhores serviços e aumentem a rentabilidade do seu negócio na plataforma.

4. Como é que a distância a costa de um município afeta o seu preço e a avaliação associada à sua localização?

R: Embora não exista uma ligação direta entre estes fatores, denota-se que zonas como Mafra, Cascais e Sintra, cuja distância à costa é muito baixa, tem um preço médio por noite superior às restantes localizações e uma classificação de localização alta (superior a 4.6).

P: Average Location Rating by distance do coast- Room Type: <Room Type>

U: Com estas conclusões, os hosts conseguem perceber melhor, com base na distância das suas listings à costa, qual é o preço mais adequado tendo em conta o mercado atual. Além disso, essa informação reforça a perceção de que os usuários tendem a atribuir classificações mais elevadas a propriedades localizadas próximas à costa. Isto pode ser ainda usado por utilizadores que estejam à procura de adquirir propriedade junto à costa, para perceberem se o investimento desta compra seria rentável.

5. Quais são as principais características dos alojamentos listados no Airbnb?

R: As principais características são: tipo de quarto como casa inteira/apartamento, capacidade de alojamento para duas pessoas, 1 cama, 1 quarta e 1 casa de banho. Havendo para este caso, um total de 2420 listings com um preço médio de 87.30\$ por noite.

P: Top 5 Room Type Combinations by most listed

U: Esta informação, permite aos utilizadores da plataforma, ter uma visão mais detalhada acerca das características mais comuns nos alojamentos listados. Além disso, para cada elemento do Top 5, os utilizadores podem perceber como varia o preço em conformidade com as comodidades presentes.

6. Como é que são classificados os diferentes municípios nas diferentes categorias de rating?

R: O município com melhor avaliação de check-in é Arruda dos Vinhos (4.94), o com melhor avaliação de limpeza é Arruda dos Vinhos (4.94), com melhor avaliação referente a comunicação é Arruda dos Vinhos (4.96), com melhor avaliação de localização é Mafra (4.77) e por fim, melhor avaliação de valor (na relação preço/qualidade) é Arruda dos Vinhos (4.81).

P: Review Scores by Neighbourhood- Room Type: <Room Type>

U: Com estas conclusões, a Airbnb, conforme a localização das diferentes listings dos hosts, pode dar-lhe sugestões de diferentes áreas em que pode melhorar. Por exemplo, se o utilizador viver numa zona onde a avaliação referente a limpeza seja muito baixa, a plataforma pode sugerir a este para focar-se mais neste aspeto de forma a destacar-se das propriedades listadas no mesmo município.

7. Tendo as características do alojamento, qual o preço médio praticado em cada município de Lisboa?

R: Para um certo número de quartos, casas de banho, camas e capacidade de alojamento, observar o preço médio para o município do seu alojamento.

P: Average rating and price by Neighbourhood Group filter by accommodates, bathrooms, bedrooms and beds

U: Esta informação permite aos utilizadores da plataforma terem uma ideia do preço médio praticado em alojamentos com características iguais ou semelhantes ao seu. Assim, estes podem evitar anúncios com preços demasiado fora da realidade do mercado atual. Isto faz com que anúncios com características semelhantes, tenham preços com menor variância, permitindo maior rentabilidade para o anfitrião e para a plataforma.

8. Quais são os hosts com maior número de anúncios e quais são os seus potenciais ganhos?

R: Top10 hosts tendo em conta a contagem de listagens de cada um e usando o preço médio por noite de cada listagem vezes o número de reviews desde que foi publicada, para calcular os potenciais ganhos.

P: Top 10 Hosts by Total Listings

U: Com esta informação, os utilizadores da plataforma conseguem ter uma melhor noção da relação custo/benefício entre o número de listagens dos hosts e os potenciais ganhos monetários.

Estratégias para otimizar a DW e as pesquisas OLAP

Relativamente à otimização da DW, apenas os atributos que achamos necessários para a realização do projeto foram importados para a base de dados.

Para otimizar o desempenho das pesquisas realizadas, designou-se um conjunto de primary e foreign keys que permitem à base de dados criar automaticamente índices que vão fazer com que a obtenção dos dados seja mais eficiente. Estas chaves designadas estão representadas anteriormente no relatório na imagem relativa ao modelo de dados desenvolvido, sendo necessário executar alguns comandos SQL para as definir após a criação das tabelas.

Técnicas para aceder à informação

Ao longo do desenvolvimento do projeto, considerou-se a necessidade de ter técnicas de manipular o acesso à informação, desde o desenho do modelo de dados até às pesquisas efetuadas à base de dados.

A título de exemplo é possível referir que os conceitos de “drill-down” e “roll-up” estão presentes no facto de termos acesso ao município e ao nome da localização específica do local dentro desse município. Assim, foi possível analisar os dados relativos aos municípios, mas também a localizações mais específicas (como quando no mapa os alojamentos aparecem com cores diferentes referentes ao local em que estão).

O conceito de “slice and dice” está muito presente desde o momento da conceção das dimensões, estando relacionado com diversas técnicas de agrupamento de dados usados. Alguns exemplos aplicados podem ser relativos aos agrupamentos e/ou ordenamento de localizações, anfitriões e tipos de alojamento.

Etapas Futuras

Numa fase futura do projeto pretende adicionar-se uma dimensão de tempo, permitindo à plataforma filtrar os dados desde 2019 até 2023, de modo a ter uma melhor noção da tendência do mercado de alojamento. Mais especificamente, ter uma ideia de como a aderência aos diferentes municípios têm vindo a mudar ao longo dos anos, em conjunto, com outras dimensões como o custo por noite.

A plataforma também está configurada de modo a ser possível introduzir novas cidades como o Porto, o que nos vai permitir numa próxima etapa, fazer comparações entre as diferentes cidades tendo em conta fatores como preço e classificações.

Por fim, poderá ser possível, adicionar outras dimensões como a distância entre os diferentes municípios aos monumentos nacionais.

Conclusão – Parte 1

Ao analisar e posteriormente dispor de forma visual e interativa dados e conclusões referentes às diferentes dimensões que afetam as listagens, conclui-se que a solução desenvolvida pode melhorar o modo de funcionamento da Airbnb no processo de fornecer recomendações aos seus utilizadores de forma a criar valor conjunto para a empresa e para os hosts que colocam os seus anúncios na plataforma.

Além disso, a nossa solução, apresenta um grande potencial de escalabilidade para o Airbnb, pois está configurada para carregar, analisar e extrair conclusões de dados referentes a várias cidades, não estando apenas limitado à cidade de Lisboa.

Parte 2 – Data Mining

Nesta etapa do trabalho, foi proposto o uso de ferramentas de data mining de modo a complementar a análise de dados desenvolvida anteriormente com capacidades descritivas e preditivas. Assim, esta fase é referente ao processo de preparação dos dados e à exploração de duas técnicas de análise que se consideraram mais apropriadas tendo em conta a natureza dos dados e os objetivos deste projeto. Na parte final é referido o valor que este produto pode ter para os utilizadores finais tais como os anfitriões dos alojamentos e a própria plataforma.

Tal como referido acima, os objetivos que determinámos para esta fase do projeto foram determinantes para a escolha das duas técnicas de análise de dados que se aplicaram. Dentro de uma vasta gama de possíveis metodologias referentes esta etapa, optámos por duas que tinham capacidade de obter formas de dados mais adaptadas às preferências previstas dos utilizadores.

Preparação dos dados

Fonte dos dados

Os dados que foram usados nesta parte são referentes aos diversos ficheiros disponibilizados pelo projeto “Inside Airbnb”. De forma a termos uma maior quantidade de dados para analisar, decidiu-se usar os dados referentes à lista de alojamentos de datas anteriores para complementar a mais recente já implementada.

Assim, considerando a cidade de Lisboa, foram usados quatro ficheiros referentes às quatro datas disponíveis na plataforma que foram importados para a base de dados para a tabela *facts_table*. Além destes, para a etapa de previsão temporal referida mais à frente neste relatório, foi também usado os ficheiros referentes às avaliações e ao calendário de disponibilidade de alojamentos mais recente.

Ferramentas utilizadas (software)

Seguindo algumas metodologias adotadas na primeira fase do projeto, foi utilizado o *Pentaho* para realizar o processo de ETL e o *PostgreSQL* (pgAdmin 4) para armazenar a informação numa estrutura adequada de base de dados. Para realizar todo o processo de preparação dos dados (complementar ao realizado na fase anterior) e de data mining, foi utilizado *Python* devido à familiaridade com a linguagem e à sua variedade de bibliotecas relativamente a este tipo de exploração de dados.

Alteração no Pentaho

Primeiramente, de forma a termos uma maior quantidade de dados para analisar, decidimos usar os dados do projeto “Inside Airbnb” referentes à lista de alojamentos de datas anteriores para complementar a mais recente já implementada. Para isso, alterou-se a configuração do *Pentaho* para ter acesso a esses ficheiros de entrada.

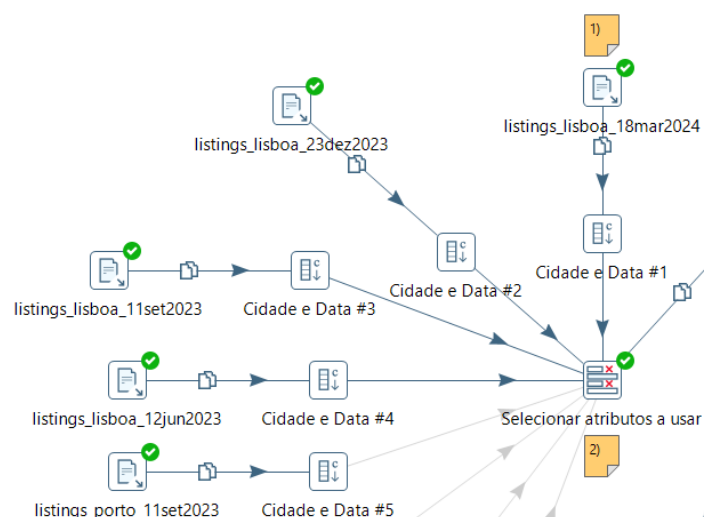


Figura 5 - Alteração no Pentaho

Assim, cada linha final na base de dados terá duas novas features: a cidade (*source_city*) e a data do ficheiro de entrada de origem do alojamento (*source_date*). Isto faz com que os atributos-chave da tabela *facts_table* alterem para uma combinação do id do alojamento e da data do ficheiro, permitindo assim que esta tenha diferentes linhas referentes ao mesmo alojamento (em datas diferentes). Além desses atributos, foram também acrescentados os referentes à combinação do tipo de alojamento de modo a facilitar a importação dos dados para a ferramenta seguinte: *room_type*, *accommodates*, *bathrooms*, *bedrooms*, *beds*.

Remoção de outliers

De seguida procedeu-se à importação dos dados da base de dados resultante do processo anterior para *Python* e à remoção das linhas consideradas outliers. Para isso, considerou-se o método IQR (Intervalo Interquartil). Este consiste na diferença entre o terceiro quartil (Q3) e o primeiro quartil (Q1) de um certo conjunto de dados, definindo-se outliers como valores que estão acima de $Q3 + 1,5 * IQR$ ou abaixo de $Q1 - 1,5 * IQR$.

Assim, determinou-se quais linhas seriam consideradas como outliers tendo em conta a união dos diversos conjuntos de outliers das seguintes features de forma individual: 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'numericprice', 'minimum_nights' (futuramente usadas para treinar os modelos classificadores). Após esta remoção, a tabela ficou com 67452 entradas.

Adição da categoria de preço

A próxima etapa consistiu na adição de uma feature que representa a categoria de preço em que o alojamento está inserido (*price_cat*). Considerou-se que as diversas categorias seriam determinadas tendo em conta a diversidade presente nos dados e, deste modo, cada categoria deveria representar uma quantidade semelhante de informação da base de dados.

Para isso, recorreu-se ao cálculo dos diferentes quartis da feature *numericprice* e, seguidamente, da categoria de preço a que cada alojamento pertence. Apesar da noção de que o número de categorias não poderia ser muito elevado (devido à possível imprecisão dos algoritmos futuramente usados), é de realçar que este não foi definido desde o início, sendo que ia sofrendo alterações consoante os resultados obtidos nos algoritmos relatados mais à frente.

Recomendação de preço (Estudo de classificação)

Objetivos

Logo desde o começo do desenvolvimento deste trabalho que considerámos que um dos objetivos do seu desenvolvimento, que seria mais valorizado por potenciais futuros utilizadores da ferramenta, seria uma forma de conseguir dar uma recomendação do preço para um alojamento tendo em conta todas as suas características iniciais de colocação do anúncio do imóvel na plataforma.

Processo de escolha do algoritmo

Para isso, a ferramenta que proporcionasse esse serviço teria de ter em conta os alojamentos já existentes e relacionar as suas características com os preços praticados na plataforma. Isso levou a que adotássemos uma metodologia de aprendizagem computacional supervisionada, em que o algoritmo era alimentado por um conjunto de features (referentes às características do alojamento) e as associava à classe de preço a que esse alojamento tivera sido colocado.

Assim, decidimos primeiramente aplicar diversos algoritmos classificadores de modo a perceber qual obteria o melhor resultado em prever corretamente a categoria de preço a que o alojamento mais se adequava. Importante notar que para um alojamento com diferentes `source_dates` consideramos a data mais recente para definir a sua categoria de preço.

Escolha dos atributos

Para treinar o nosso modelo, tivemos de ter em conta quais serão os atributos mais adequados para classificar corretamente os *listings* tendo em conta as diferentes categorias de preço.

Numa abordagem inicial, fizemos uma matriz de correlação entre todos os atributos presentes no *dataset* e depois fizemos uma *view* desta matriz que mostrava, por ordem decrescente de correlação, a correlação entre o atributo *numericprice* (preço por noite dos *listings*) com os outros atributos. O objetivo era perceber se haveria algum conjunto de atributos com alta correlação que permitisse facilmente identificar a categoria de preço.

numericprice	1.000000
accommodates	0.481113
beds	0.388108
bedrooms	0.324263
bathrooms	0.177926
review_scores_location	0.151078
review_scores_cleanliness	0.119513
review_scores_rating	0.108730
review_scores_accuracy	0.100078
minimum_nights	0.083868
availability_365	0.079800
instant_bookable_bool	0.068612
review_scores_value	0.068449
review_scores_communication	0.059890
review_scores_checkin	0.054584
maximum_nights	0.045426
minimum_nights_avg_ntm	0.028479
maximum_maximum_nights	0.021929
availability_90	0.018740
minimum_minimum_nights	0.017054
maximum_nights_avg_ntm	0.009990
has_availability_bool	0.007030
maximum_minimum_nights	0.003512
minimum_maximum_nights	-0.002121

Figura 6 - Correlação entre numericprice e atributos do dataset

No entanto, observou-se que não existia uma grande correlação entre os atributos, sendo os 4 atributos com maior correlação: *accommodates*, *beds*, *bedrooms* e *bathrooms*. Os restantes atributos apresentavam uma correlação de 0.151 ou menos.

Por esses motivos, optamos por selecionar todos os atributos que podem ser convertidos para *float* e depois dentro destes, selecionar apenas aqueles que os *hosts* colocam nos seus *listings*. Usamos então os seguintes atributos: *latitude*, *longitude*, *minimum_nights*, *maximum_nights*, *has_availability_bool*, *instant_bookable_bool*, *accommodates*, *bathrooms*, *bedrooms* e *beds*. Ao introduzir estes atributos, e não apenas os com maior correlação, permitimos ao modelo capturar nuances nos dados e não se perde tanta informação.

Hyperparameter search/choice

Nesta etapa do projeto, efetuamos a busca e seleção dos valores ideais para os Hiper parâmetros do nosso modelo. Os hiperparâmetros são as configurações que não são aprendidas diretamente durante o treinamento do modelo, mas afetam o seu comportamento e desempenho. Para definir os valores ideais, usamos uma estratégia de *grid_search*, onde avaliamos sistematicamente todas as combinações possíveis de valores para os hiperparâmetros numa grade pré-definida. Usamos como critério de avaliação do desempenho a *accuracy* do modelo para os diferentes hiperparâmetros. Assim, vamos usar este processo em todos os algoritmos que pretendemos testar.

Tomando como exemplo, o caso do algoritmo *Decision Tree*. Onde definimos uma grade com todos os parâmetros que pretendemos testar, como a profundidade máxima da árvore e o número mínimo de amostras para ser um nó folha. De seguida, com auxílio da biblioteca *sklearn.model_selection*, aplicamos a *grid_search* usando um valor de validação cruzada de 5. Este valor significa que o conjunto de dados será dividido em 5 partes iguais, e o modelo será treinado e testado 5 vezes, alternando entre os conjuntos de treinamento e teste em cada iteração. No final, guardamos os valores ideais para usarmos depois na próxima etapa de testagem do modelo.


```
# Define the parameters grid
param_grid = {
    'criterion': ['entropy'],
    'max_depth': list(range(20, 35, 2)),
    'min_samples_split': list(range(1, 10, 1)),
    'min_samples_leaf': list(range(1, 5, 1))
}

# Initialize the GridSearchCV object
grid_search = GridSearchCV(DecisionTreeClassifier(), param_grid, cv=5, scoring='accuracy')
# Perform grid search
grid_search.fit(X_train, y_train)
# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)
```

Figura 7 - Hyperparameter choice/select with grid_search

Testagem do modelo para 4 categorias (baseline test)

O processo de testagem do modelo baseia-se em realizar 30 execuções deste, para diferentes algoritmos de classificação, e avaliar a sua performance, fazendo uma média das 30 execuções, para as seguintes métricas: *Train Accuracy*, *Test Accuracy*, *Precision*, *Recall* e *F1-Score*. Sendo as últimas três métricas extraídas do *classification_report* e usando *weighted avg* para ter em conta o número de ocorrências de cada classe.

A *accuracy* de teste e treino, mede o quanto bem um modelo prevê corretamente a categoria dos *datasets* de teste e treino, respetivamente. A *precision* mede a proporção de exemplos classificados corretamente como pertencentes a uma categoria específica em relação ao total de exemplos classificados nessas categorias. O *Recall* mede a proporção de exemplos verdadeiramente pertencentes a uma categoria específica que foram corretamente identificados, em relação ao, total de exemplos verdadeiramente pertencentes a essa categoria. O *F1-Score*, é uma medida única que tem em consideração tanto a *precision* como o *recall*, dando uma ideia geral do desempenho do modelo tendo em conta um balanceamento entre essas duas métricas.

A tabela seguinte representa os resultados obtidos para cada um dos algoritmos testados, considerando inicialmente apenas 4 categorias de preço (10.0–65.0 €; 65.0–91.0 €; 91.0–130.0 €; 130.0–274 €)

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Simple Classifier	Decision Trees	0.983	0.845	0.843	0.842	0.842
	KNN	0.980	0.756	0.755	0.754	0.755
Ensembles	RandomForests	0.983	0.840	0.838	0.837	0.837
	AdaBoost	0.493	0.497	0.504	0.486	0.492

Analisando os resultados obtidos, destaca-se que os dois algoritmos com pior performance foram o *KNN* e o *AdaBoost*.

Relativo ao *AdaBoost*, este apresenta os valores de *accuracy*, tanto nos casos de treino como de teste, *precision*, *recall* e *f1-score*, mais baixos de entre todos os algoritmos. No caso do *KNN*, embora apresente valores de *accuracy* superiores a 0.50, ou seja, classifica pelo menos metade das amostras dos conjuntos de treino e teste corretamente, tanto o *RandomForest* como as *Decision Tress* apresentam melhor performance. Além disso, o algoritmo apresenta um tempo de execução muito superior aos outros.

O algoritmo com melhor performance foi o *Decistion Tress (DT)*, no entanto a diferença entre este e o *Random Forest*, é na ordem das centíssimas. Por este motivo, foi necessário fazer uma análise mais detalhada analisando outros fatores como a *confusion_matrix* e uma visão mais detalhada do *classification_report*. A *confusion_matrix* mostra a frequência com que cada classe é prevista corretamente. Enquanto o *classification_report* permite ter uma noção da *precision* e *recall* de cada classe. Obteve-se os seguintes resultados:

30-Run Average DT Confusion Matrix:					30-Run Average DT Classification Report:				
[[4475.867 57.7 272.433 154.]						precision	recall	f1-score	support
[64.267 2766.767 172.567 254.4]					10 - 65.0	0.892	0.902	0.897	4960.0
[339.333 150.7 3690.533 317.433]					130.0 - 274	0.854	0.849	0.852	3258.0
[140.433 265.4 341.6 3399.567]]					65.0 - 91.0	0.824	0.820	0.822	4498.0
					91.0 - 130.0	0.824	0.820	0.822	4147.0

Figura 8- 30-Run Average DT Confusion Matrix and Classification Report

30-Run Average RandomForests Confusion Matrix:					30-Run Average Forest Classification Report:				
[[4454. 47. 356. 133.]						precision	recall	f1-score	support
[83. 2667. 165. 294.]					10 - 65.0	0.888	0.893	0.890	4990.0
[348. 119. 3714. 369.]					130.0 - 274	0.860	0.831	0.845	3209.0
[132. 267. 401. 3314.]]					65.0 - 91.0	0.801	0.816	0.809	4550.0
					91.0 - 130.0	0.806	0.806	0.806	4114.0

Figura 9- 30-Run Average RandomForest Confusion Matrix and Classification Report

Mais uma vez, a diferença de valores não é muito significativa. No entanto, a *DT* mostra uma melhor performance na classificação das categorias intermédias, tanto a nível de *precision* como *recall*. No nosso caso, interessa-nos um modelo que consiga prever corretamente com maior frequência as classes mais difíceis de classificar, ou seja as classes intermédias, como os preços entre 65.0–91.0 € e 91.0–130.0 €.

Além disso, a *DT*, é menos exigente a nível computacional e tem um tempo de execução mais rápido que o *RandomForest*. Assim aplicando o conceito de Occam's razor: "All Things being equal, the simples solution tends to be the best one", escolhemos a *DT* como o algorithmo para o nosso modelo. No entanto, no teste de 5 categorias, permanecemos os outros algoritmos como opções, para ver se a subida de categorias poderá influenciar os valores destes.

Testagem do modelo usando estratégia de regressão

Além dos algoritmos de classificação referidos, testou-se uma estratégia de regressão para prever o preço por noite dos *listings*, usando um algoritmo de regressão linear, obtendo os seguintes resultados (também para 30 execuções).

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Mean Squared Error (MSE)	R Squared (R2)
Regression	Linear Regression	0.318	0.305	1747.945	0.305

Analisando os resultados obtidos em cima, conclui-se que uma abordagem de regressão não seria a mais adequada, tendo em conta os dados disponíveis. A *accuracy* tanto no *dataset* de treino como de teste é muito baixa, apresentado valores de 0.318 e 0.305, respetivamente.

Além disso, apresenta um alto valor de *MSE*, o que indica que as previsões geradas pelo modelo estão distantes dos valores reais. Por fim, um *R2* de 0.305, significa que 30.5% da variabilidade da variável dependente (preço por noite) é explicada pelas variáveis independentes incluídas neste modelo. Isto leva-nos a concluir que, as variáveis independentes incluídas no modelo, não conseguem explicar totalmente a variação dos preços das casas, o que poderá estar relacionado com a ausência de correlação entre os atributos. Deste modo, descartamos a opção de fazer um modelo de regressão para prever o preço por noite dos alojamentos.

Testagem do modelo para 5 categorias (melhoria contínua)

Nesta fase do processo de *Data Minig*, tentamos melhorar o modelo, introduzindo mais categorias de preço e assim aumentando granularidade dos dados. No entanto, temos de ter em conta o *trade-off* entre aumentar o número de categorias e a performance do modelo em termos de *accuracy*, *precision*, *recall* e *f1-score*.

O processo de testagem é semelhante ao explicado nos pontos anteriores. A tabela seguinte representa os resultados obtidos para cada um dos algoritmos testados, considerando agora 5 categorias de preço (10.0–59.0 €; 59.0–80.0 €; 80.0–104.0 €; 104.0–141.0 €; 141.0–274.0 €).

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Simple Classifier	Decision Trees	0.979	0.844	0.844	0.843	0.844
	KNN	0.975	0.754	0.752	0.751	0.752
Ensembles	RandomForests	0.980	0.820	0.818	0.816	0.817

	AdaBoost	0.424	0.429	0.420	0.418	0.414
--	----------	-------	-------	-------	-------	-------

Tal como na testagem anterior para 4 categorias, os algoritmos *KNN* e *AdaBoost*, apresentam os piores resultados. Destaca-se ainda que a performance destes piorou com o aumento das categorias. Isto leva-nos a descartar os mesmos como opções viáveis para as próximas etapas de testagem com mais categorias.

Mais uma vez, o algoritmo com melhor performance foi o *Decision Tree*. Os valores de *accuracy*, *precision*, *recall* e *f1-score*, são muito semelhantes aos resultados anteriores. Segue-se em baixo, a *confusion_matrix* e uma visão detalhada do *classification_report*:

30-Run Average DT Confusion Matrix:						30-Run Average DT Classification Report:				
							precision	recall	f1-score	support
[[3492.367 58.267 26.667 176.367 101.333]						10 - 59.0	0.889	0.906	0.897	3855.0
[48.867 2563.5 184.7 130.9 192.033]						104.0 - 141.0	0.803	0.822	0.812	3120.0
[37.567 224. 2117.233 82.067 144.133]						141.0 - 274	0.842	0.813	0.827	2605.0
[233.367 136. 80.233 3115.867 212.533]						59.0 - 80.0	0.824	0.825	0.824	3778.0
[117.033 210.933 104.333 275.667 2797.033]]						80.0 - 104.0	0.811	0.798	0.805	3505.0
						macro avg	0.834	0.833	0.833	16863.0
						weighted avg	0.835	0.835	0.835	16863.0

Figura 10- 30-Run Average DT Confusion Matrix and Classification Report

Em comparação com os resultados para 4 categorias, a categoria referente ao preço por noite mais baixo (10.0–59.0 €) mantém o mesmo *f1-score* e por isso continua a ser a mais fácil de classificar. No entanto, a categoria referente ao preço mais alto (141.0–274.0 €), tornou-se mais difícil de classificar. Com 4 categorias tínhamos um *f1-score* de 0.852 e agora temos de 0.827.

A *confusion_matrix* leva-nos a querer que isto está relacionado com a adição de uma nova categoria (104.0–141.0 €), pois o modelo classificou incorretamente esta 276 vezes como sendo a categoria de preço mais alto. Do outro lado, o modelo classificou incorretamente 213 vezes a categoria de preço mais alto, como sendo a categoria de 104.0–141.0 €. Isto indica que o modelo tem dificuldade em distinguir as duas categorias. Por fim, tal como era esperado, a categoria que o modelo teve mais dificuldade em classificar, foi a categoria intermédia (80.0–104.0 €).

Como o aumento das categorias para 5 não comprometeu as métricas de desempenho, continuamos com este processo de melhoria contínua.

Testagem do modelo para 6 categorias (melhoria contínua)

O processo de testagem é semelhante ao explicado nos pontos anteriores, no entanto estamos apenas a considerar o algoritmo *Decision Tree*. A tabela seguinte representa os resultados obtidos para cada um dos algoritmos testados, considerando agora 6 categorias de preço (10.0–55.0 €; 55.0–74.0 €; 74.0–91.0 €; 91.0–115.0 €; 115.0–150.0 €; 150.0–274.0 €).

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Simple Classifier	Decision Trees	0.978	0.823	0.822	0.822	0.822

Mais uma vez, os valores de *accuracy*, *precision*, *recall* e *f1-score*, são muito semelhantes aos resultados anteriores. No entanto, já se nota alguma diferença na ordem das cientíssimas em relação ao teste base com 4 categorias. Segue-se em baixo, a *confusion_matrix* e uma visão detalhada do *classification_report*:

30-Run Average DT Confusion Matrix:							30-Run Average DT Classification Report:				
[[3059.433 29.1 13.033 169.467 68.6 48.367]								precision	recall	f1-score	support
[35.433 2188.967 110.933 71.967 104.967 160.733]							10 - 55.0	0.894	0.903	0.899	3388.0
[13.4 152.533 1610.1 34.467 64.8 77.7]							115.0 - 150.0	0.821	0.819	0.820	2673.0
[167.833 57.8 42.567 2451.867 171.7 103.233]							150.0 - 274	0.845	0.824	0.834	1953.0
[85.767 106.633 49.933 201.2 2465.133 166.333]							55.0 - 74.0	0.806	0.819	0.812	2995.0
[58.533 132.667 79.7 112.733 182.7 2212.667]]							74.0 - 91.0	0.806	0.802	0.804	3075.0
							91.0 - 115.0	0.799	0.796	0.798	2779.0
							macro avg	0.829	0.827	0.828	16863.0
							weighted avg	0.829	0.830	0.829	16863.0

Figura 11- 30-Run Average DT Confusion Matrix and Classification Report

Tal como nos testes para 4 e 5 categorias, a categoria mais fácil de classificar continua a ser a referente ao preço por noite mais baixo, sendo que esta tem um *f1-score* de 0.899. Tanto a *confusion_matrix* como o *classification_report* seguem um padrão semelhante ao anterior. As categorias intermédias 74.0–91.0 € e 91.0–115.0 €, são as mais difíceis de classificar tendo os piores valores de *f1-score*.

Continuamos este processo iterativo de aumentar o número de categorias e analisar a performance, até chegarmos aquilo que consideramos o valor limite para as métricas (0.80).

Testagem do modelo para 10 categorias (melhor resultado)

O processo de testagem é semelhante ao explicado nos pontos anteriores. A tabela seguinte representa os resultados obtidos para cada um dos algoritmos testados, considerando agora 10 categorias de preço (10.0–45.0 €; 45.0–59.0 €; 59.0–70.0 €; 70.0–80.0 €; 80.0–91.0 €; 91.0–104.0 €; 104.0–120.0 €; 120.0–141.0 €; 141.0–173.0 €; 173.0–274.0 €).

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Simple Classifier	Decision Trees	0.969	0.800	0.800	0.799	0.800

Como observado pelos resultados, as métricas ficam muito perto do valor limite (0.80). Consideramos que 10 categorias é por isso, o melhor balanceamento para o modelo entre granularidade de categorias e métricas de desempenho. Por fim, apresentamos a uma visão detalhada do *classification_report* para comparar com as testagens anteriores:

30-Run Average DT Classification Report:				
	precision	recall	f1-score	support
10 - 45.0	0.848	0.851	0.849	1826.0
104.0 - 120.0	0.760	0.781	0.770	1707.0
120.0 - 141.0	0.777	0.794	0.785	1666.0
141.0 - 173.0	0.805	0.789	0.797	1601.0
173.0 - 274	0.858	0.833	0.845	1618.0
45.0 - 59.0	0.764	0.777	0.770	1639.0
59.0 - 70.0	0.758	0.776	0.767	1805.0
70.0 - 80.0	0.750	0.751	0.750	1587.0
80.0 - 91.0	0.766	0.748	0.757	1647.0
91.0 - 104.0	0.786	0.765	0.775	1767.0
macro avg	0.787	0.787	0.787	16863.0
weighted avg	0.787	0.787	0.787	16863.0

Figura 12- 30-Run Average DT Classification Report

Tal como nos testes anteriores, a categoria referente ao preço por noite mais baixo é a com melhor *f1-score* e por isso é a mais fácil de classificar. A categoria de preço mais alto, é a segunda mais fácil de classificar e surpreendentemente, apresenta um *f1-score* superior aos do modelo com 5 e 6 categorias. As restantes categorias apresentam todas valores muito semelhantes de *f1-score*, sendo que as categorias mais “centradas” na gama de valores (70.0–80.0 €;80.0–91.0 €) tem os valores mais baixos.

Acreditamos que por haver uma maior segmentação, vai existir uma maior dificuldade em distinguir as categorias intermédias entre si. Isto explica haver uma diferença tão grande entre os valores precisão das categorias intermédias (entre 0.75 a 0.78), com o referente a categoria de baixo preço (0.848) e de alto preço (0.858).

Numa iteração seguinte, corremos a mesma experiência, mas para 11 categorias, no entanto, as métricas passavam para valores entre 0.74 e 0.76, estando abaixo do limite delineado.

Resultados obtidos com integração na interface GUI

Numa fase final do projeto, implementamos uma interface gráfica para o utilizador (GUI) de modo a facultar a sua experiência com a utilização do nosso modelo e possibilitando uma certa experimentação do mesmo.

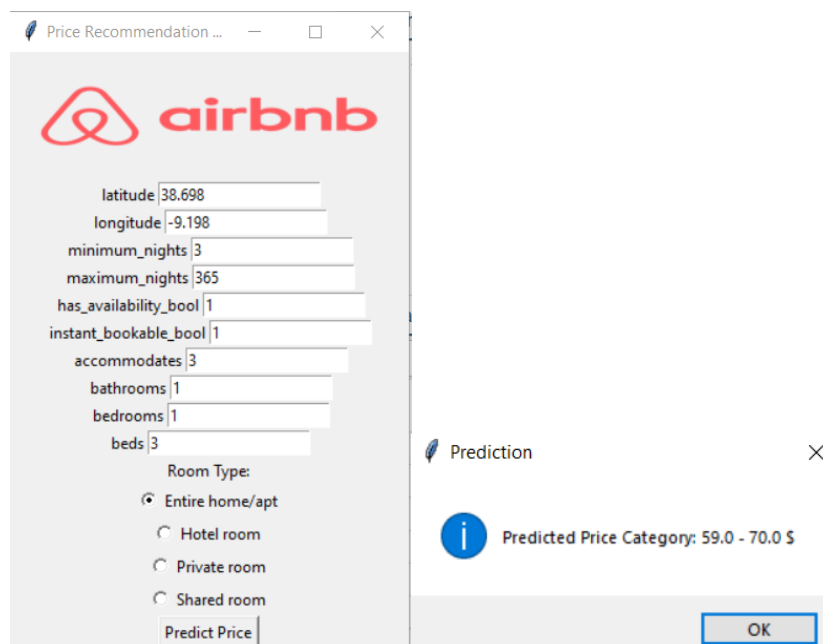


Figura 13 - GUI Interface for Price Recommendation Lisbon

Inserimos na interface GUI os dados referentes a um *listing* real e obtivemos uma previsão correta da categoria de preço.

Deste modo, vamos de encontro ao principal objetivo desta etapa, que era conseguir fornecer recomendações do preço para um alojamento tendo em conta as características apresentadas no *listing*. Assim os utilizadores finais (anfitriões de alojamento) podem:

- compreender como as diferentes características como por exemplo, localização e número de camas, podem afetar o preço por noite dos seus alojamentos;
- estimar os potenciais ganhos pois tem uma noção do preço praticado por noite para alojamentos com aquelas características;

Exemplificação com a cidade do Porto

Além de ser possível testar o modelo usando os dados referentes a cidade de Lisboa, este foi adaptado para poder fazer o mesmo processamento para *datasets* de outras cidades, desde que estes tenham uma estrutura semelhante à dos presentes no site *InsideAirbnb*. No entanto, no caso da cidade do Porto, optamos por reduzir o número de categorias de 10 para 6, uma vez que quando testado para 10 categorias o modelo apresentou baixos valores de *accuracy*.

Apresentamos na tabela seguinte, os valores de *accuracy*, *precision*, *recall* e *f1-score*, do nosso modelo, mas no caso da cidade do Porto para 6 categorias.

Tipo de Algoritmo	Algoritmo	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Simple Classifier	Decision Trees	0.930	0.701	0.699	0.698	0.698

Destaca-se que no caso da cidade do Porto, o desempenho do modelo é inferior, sendo na ordem de 0.70 na *accuracy* de teste, na *precision*, no *recall* e no *f1-score*. Esta caída no desempenho poderá estar relacionada com fatores como a correlação entre os atributos ser menor ou o próprio *dataset* não ser tão adequado para o modelo como o anterior. Uma possibilidade de trabalho futuro, seria treinar o modelo de modo a este ter uma maior flexibilidade e adaptar-se melhor a outras localizações.

Segue-se em baixo uma demonstração da interface *GUI*, mas no caso concreto da cidade do Porto.

Figura 14-GUI Interface for Price Recommendation Porto

Previsão de ocupação (Estudo de análise de séries temporais)

Objetivos do estudo

Sendo um mercado bastante volátil ao longo do tempo (marcado por questões meteorológicas, turísticas, etc.), é muito relevante que se tenha uma noção do que se pode esperar da procura por alojamento ao longo das diferentes alturas do ano. Assim, esta técnica de extração de informação foi escolhida por ser a mais adequada para prever valores futuros tendo em conta dados que são parte do histórico de funcionamento da plataforma.

Gráficos temporais

De modo a escolher o estudo mais adequado possível aos nossos objetivos, procurou-se fazer diversos gráficos que representassem diversos tipos de características referentes ao funcionamento da plataforma ao longo do tempo. Para isto, recorremos a dois novos ficheiros (também disponibilizados pelo mesmo projeto da plataforma) que consistem em informações sobre o calendário (a disponibilidade e preço referente a cada alojamento nos próximos 365 dias) e as avaliações (data e avaliação de todas as avaliações referentes a todos os alojamentos atualmente listados na plataforma).

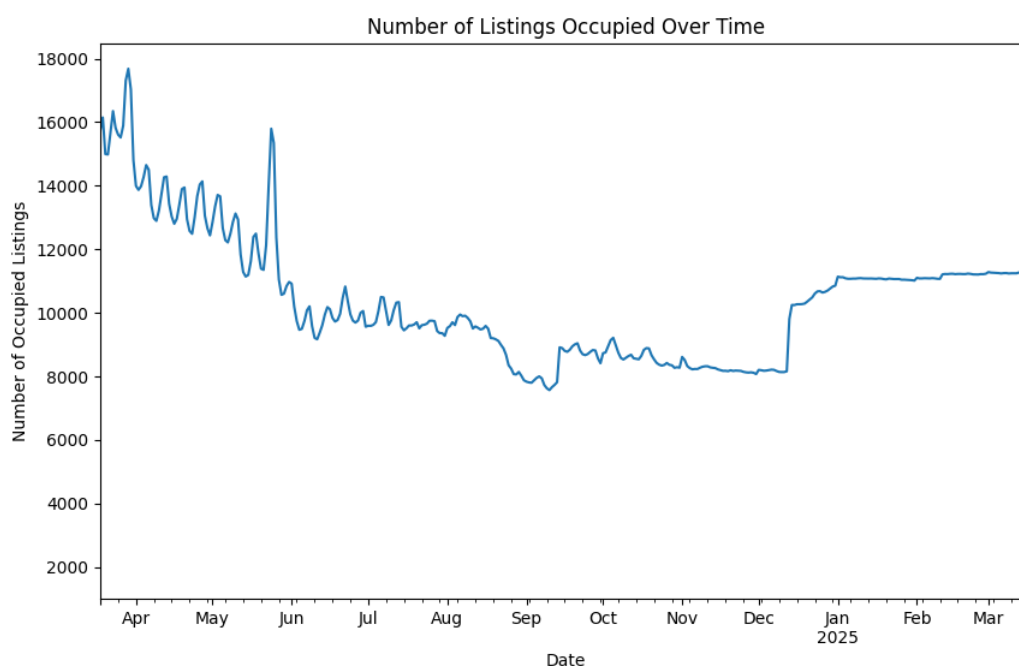


Figura 15 - Número de alojamentos ocupados ao longo do tempo

Como seria de esperar, em geral há um maior número de alojamentos ocupados junto à data referente ao ficheiro de dados devido aos utilizadores reservarem os alojamentos para datas mais próximas em comparação a reservas nos meses seguintes, por exemplo.

Apesar disso, dois picos podem ser reconhecidos com maior destaque: um no final de março (provavelmente devido ao facto de o dia 29 de março, que assinala a Sexta-Feira Santa, ser seguido da Páscoa, no domingo de 31 de março, existindo assim a possibilidade de três dias seguidos de descanso levando a uma maior atividade turística na capital) e outro nos dias 24 e dia 25 de maio (altura do concerto da artista Taylor Swift em Lisboa, sendo um evento que origina uma grande onda de turismo à cidade). Este último pico está representado com mais detalhe na imagem seguinte.

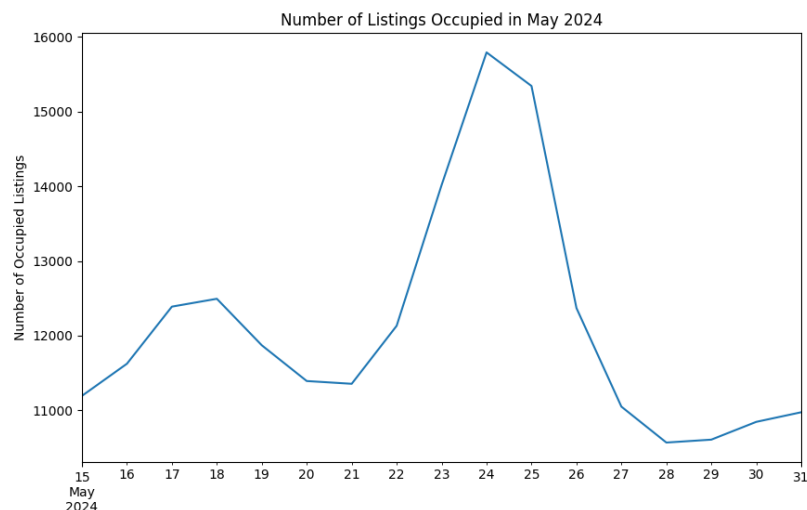


Figura 16 - Pico de uso da plataforma em maio de 2024

Além dos picos analisados, é possível notar um padrão de oscilação entre os valores ao longo de um mês, mais realçado ao olhar para o mês de abril, sendo que a hipótese que surgiu foi que estes picos menores ao longo do mês representassem os fins de semana. Para verificar essa afirmação procurámos perceber a evolução do número de alojamentos ocupados ao longo dos diferentes dias da semana. Com a análise do gráfico seguinte foi possível perceber que existem dois dias da semana (sexta-feira e sábado) que se destacam dos restantes.

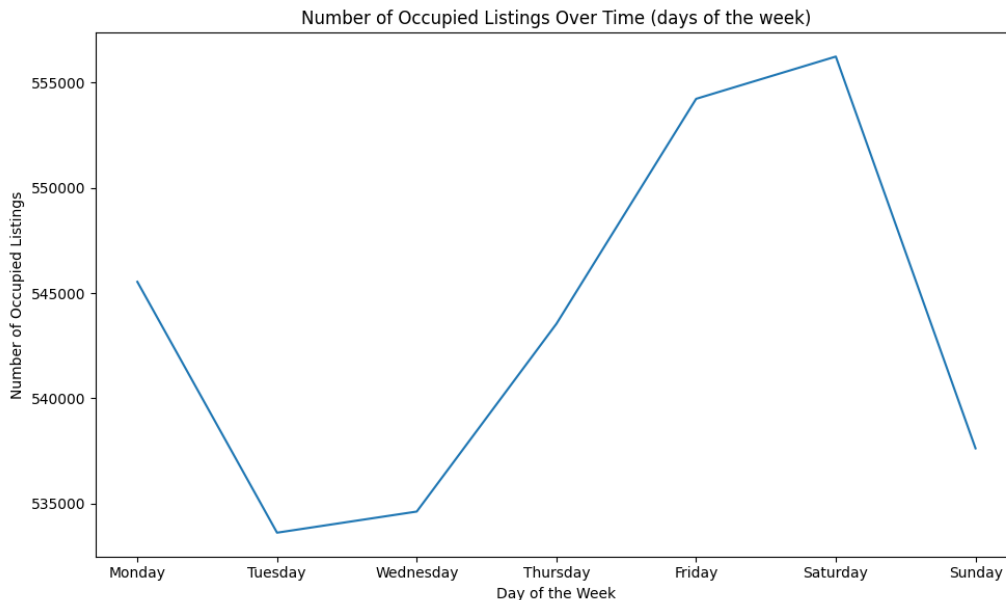


Figura 17 - Número de alojamentos ocupados por dia da semana

Após ter observado gráficos relacionados ao número de alojamentos, tentámos detetar alguma tendência na flutuação de preços ao longo das datas de importação dos quatro ficheiros importados. Isso permitiu, no gráfico seguinte, perceber que existe uma queda relevante no preço médio durante o mês de dezembro comparativamente aos meses de verão.

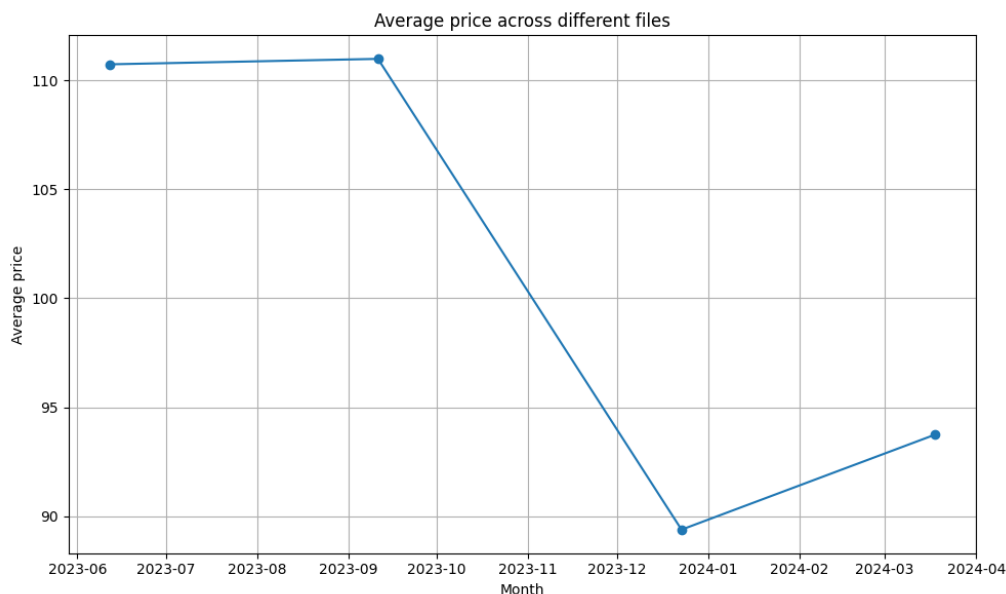


Figura 18 - Preço médio das diferentes datas dos ficheiros

De seguida, usámos os dados presentes no ficheiro referente às avaliações deixadas pelos clientes com o propósito de detetar tendências de procura por alojamento por parte dos utilizadores. De realçar que ao contrário dos dados do ficheiro anterior referente ao calendário (que continha informação do estado atual e futuro da plataforma), através do uso do ficheiro das avaliações é possível aceder a factos passados do funcionamento da plataforma, sendo isto útil para a realização de uma análise de evolução temporal.

Assim, o gráfico seguinte mostra a diferença do número de avaliações ao longo dos diferentes meses do ano, onde se percebe que o pico é durante o mês de agosto. Além disso, é perceptível uma maior aderência à plataforma durante os meses de verão e outono (meses que coincidem com o maior fluxo turístico da cidade).

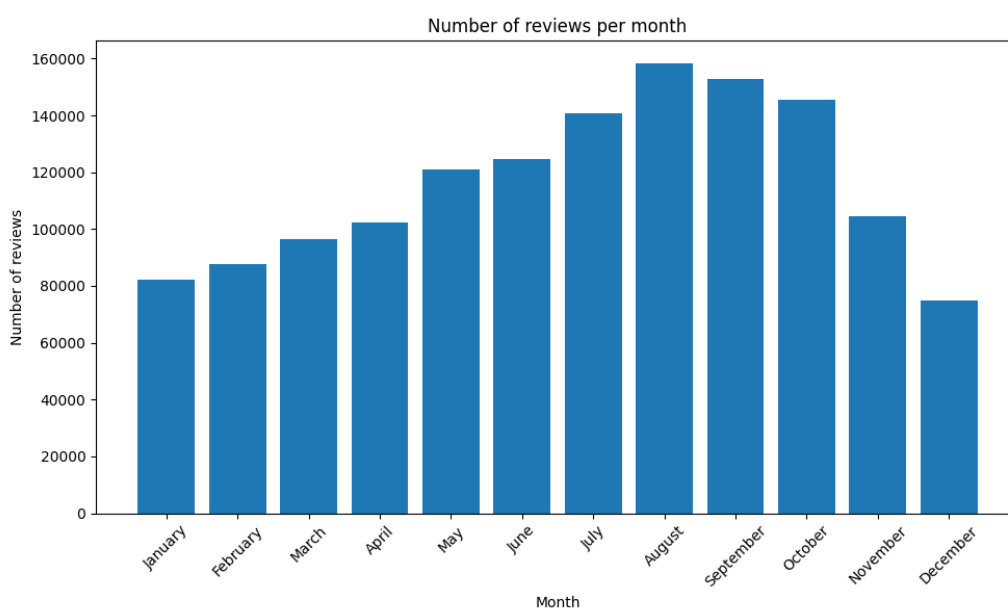


Figura 19 - Número de avaliações por mês

Após a visualização do gráfico acima, questionamos a hipótese deste padrão ser detetável ao longo dos anos de funcionamento da plataforma. Desse modo, decidimos desenvolver uma visualização que permitisse observar o número médio de avaliações por alojamento por mês ao longo dos anos.

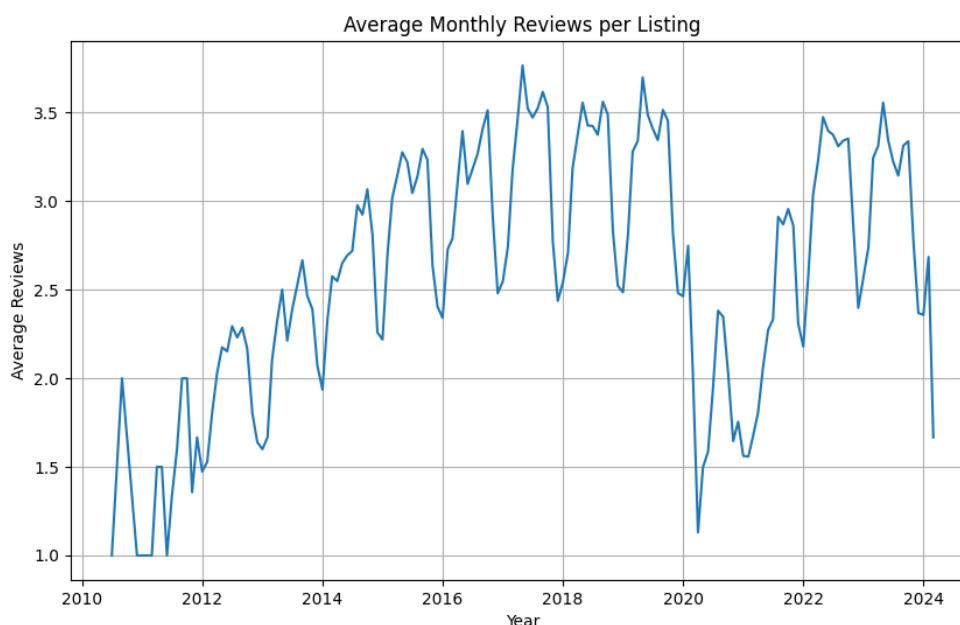


Figura 20 - Número médio de avaliações por alojamento ao longo do tempo

Um primeiro olhar para o gráfico permite destacar inicialmente dois aspetos: o padrão detetado entre os anos de 2016 e 2020 (que se volta a repetir entre 2022 e 2024) e os anos influenciados pela pandemia COVID-19 em que o país sofreu diversas consequências, sendo o setor do turismo um dos mais afetados devido às diversas restrições relacionadas à mobilidade turística e saúde da sociedade. Foi através destas observações que decidimos aplicar uma análise temporal nos dados referentes a esta visualização no processo de data mining.

Escolha do algoritmo - ARIMA

De modo a alcançar os objetivos propostos para esta etapa, seria necessário um algoritmo que tornasse possível prever valores futuros a partir da aprendizagem de uma série temporal relativa a dados passados da plataforma. Assim, optou-se pelo uso de uma média móvel integrada autoregressiva (ARIMA). Este consiste num modelo de análise estatística que usa dados de séries temporais para compreender melhor o conjunto de dados ou para prever tendências futuras.

Com isto, devido a ser um modelo estatístico autoregressivo, consegue prever valores futuros com base em valores passados, sendo muito utilizado na previsão pois combina a facilidade de utilização com a capacidade para representar aproximadamente o comportamento de qualquer série temporal. Assim, um modelo ARIMA pode ser entendido descrevendo cada um de seus componentes da seguinte forma:

- Autorregressão (**AR**): refere-se a um modelo que mostra uma variável que regride nos seus próprios valores anteriores;
- Integrado (**I**): representa a diferenciação de observações para permitir que a série temporal se torne estacionária (ou seja, os valores dos dados são substituídos pela diferença entre os valores dos

dados e os valores anteriores de modo a tornar a série temporal numa cujas propriedades estatísticas (média, variância, etc.) não dependem do tempo em que a série é observada);

- Média móvel (MA): incorpora a dependência entre uma observação e um erro residual de um modelo de média móvel aplicado a observações defasadas.

Apesar disso, teve-se em conta que os modelos autorregressivos assumem implicitamente que o futuro será semelhante ao passado. Deste modo, ainda que o mercado turístico tenha algumas alturas do ano em que se verifique com frequências certas tendências de maior fluxo, esta suposição pode fazer com que o modelo se revele impreciso em determinadas condições de mercado, tais como crises financeiras, períodos de rápidas mudanças tecnológicas ou, tal como verificado, condições epidemiológicas.

Processo de Data Mining

Inicialmente decidimos dividir os dados referentes à série temporal em conjuntos de treino do algoritmo e teste. Para isso, definimos o intervalo entre 2016 e 2019 para representar os dados de treino devido ao padrão detetado que referimos acima (sazonalidade) e como teste os anos de 2022 e 2023 devido a se ter verificado um padrão semelhante, sendo uma boa aproximação para avaliar a qualidade do modelo.

Para este processo recorreu-se ao atributo da data em que as avaliações foram feitas, assim como o identificador único do alojamento referente a cada avaliação. Isto permitiu calcular o total de avaliações por alojamento, levando assim ao possível cálculo da média ao longo do tempo.

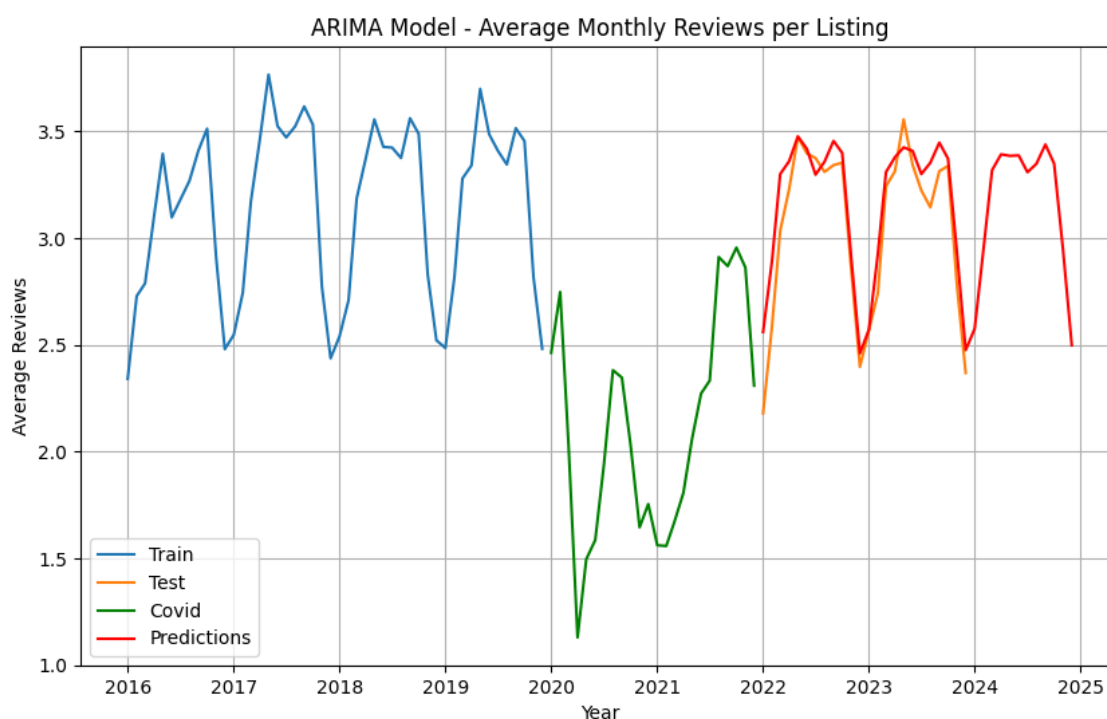


Figura 151 - Previsão da média de avaliações mensais por alojamento em Lisboa

Assim, o gráfico representado contém 4 linhas com diferentes significados. A azul e a laranja estão representados os dados de treino e de teste, respetivamente, como já explicados acima. Além desses, a cor verde

realça a parte dos dados que não foi considerada para os conjuntos devido a não seguir o padrão detetado (devido às consequências da COVID-19). Por fim, a vermelho, estão representadas as previsões que o modelo fez para os anos de 2022 a 2024, destacando-se o ano de 2024 pois é o primeiro que não tem dados (previsão para o futuro).

Após o desenvolvimento da visualização anterior, de modo a complementar as conclusões possíveis a retirar do gráfico, decidimos representar uma estimativa da percentagem de ocupação de alojamentos na cidade ao longo do tempo.

Para isso, após alguma pesquisa em fontes acerca do turismo em Lisboa, assumimos uma estimativa de que a duração média de estadia turística na cidade é de 3.2 noites (considerando a que as diversidades de valores encontrados durante o processo de pesquisa se encontravam em maioria dentro do intervalo de 3 a 4). Além disso, considerámos que cada avaliação deixada pelos utilizadores correspondia diretamente a uma estadia no alojamento.

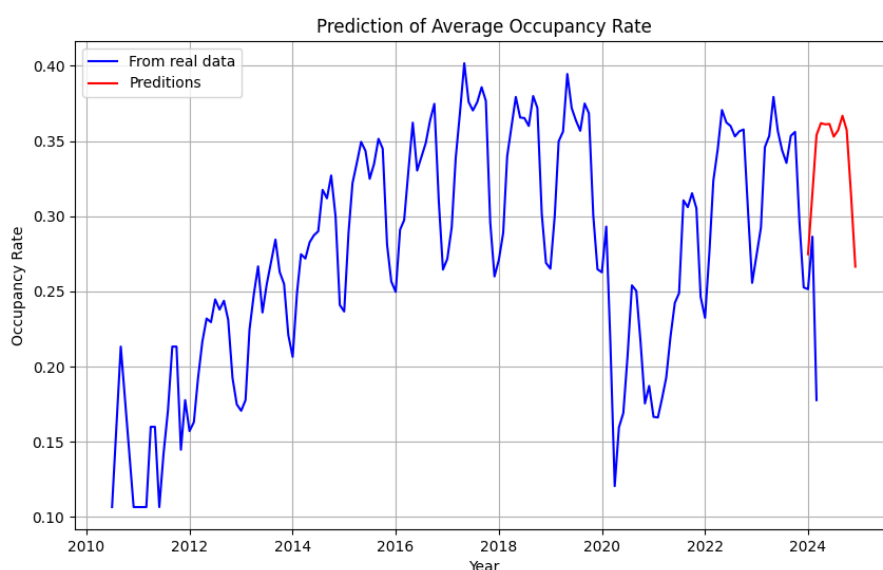


Figura 22 - Taxa de ocupação dos alojamentos em Lisboa

Através do gráfico acima é possível ter uma noção de que, ao considerar uma estadia média de 3.2 dias, a taxa de ocupação dos alojamentos listados na plataforma da Airbnb na cidade de Lisboa é inferior a 50% ao longo dos vários anos. Além da semelhança visual com o gráfico anterior, é importante realçar que, além do seguinte gráfico já conter uma componente inerentemente preditiva, como nem todos os utilizadores deixam uma avaliação do serviço, esta abordagem torna-se uma estimativa relativamente “conservadora”.

Processo de procura e escolha dos melhores parâmetros

Cada componente no ARIMA funciona como um parâmetro com uma notação padrão de p , d e q . Estes valores inteiros substituem os parâmetros para indicar o tipo de modelo utilizado. Assim, um modelo é classificado como um modelo "ARIMA (p , d , q)" em que:

- p é o número de termos auto-regressivos;
- d é o número de diferenciações para que a série se torne estacionária;
- q é o número termos de médias móveis;

sendo que estes termos são todos inteiros maiores ou iguais a zero.

Sendo assim, um intervalo de valores desses termos foi definido para os quais se testaram o desempenho do modelo (de 0 a 10 em cada termo). Para isso, recorreu-se a uma lista que contém todas as combinações desses parâmetros (total de 1000 elementos). Assim, para cada combinação, treinou-se o modelo com esses valores.

Com isto, foi possível obter a melhor combinação avaliando a previsão para os dados de teste e comparando consecutivamente o resultado de MSE das diversas configurações: ('p': 9, 'd': 0, 'q': 7).

Exemplificação usando a cidade do Porto

A forma de como se implementou todo o projeto, permite que se tenha em conta diferentes cidades além da capital portuguesa. Assim, de seguida, estão representados os gráficos referentes à previsão da média de avaliações mensais por alojamento e da taxa de ocupação dos alojamentos na cidade do Porto.

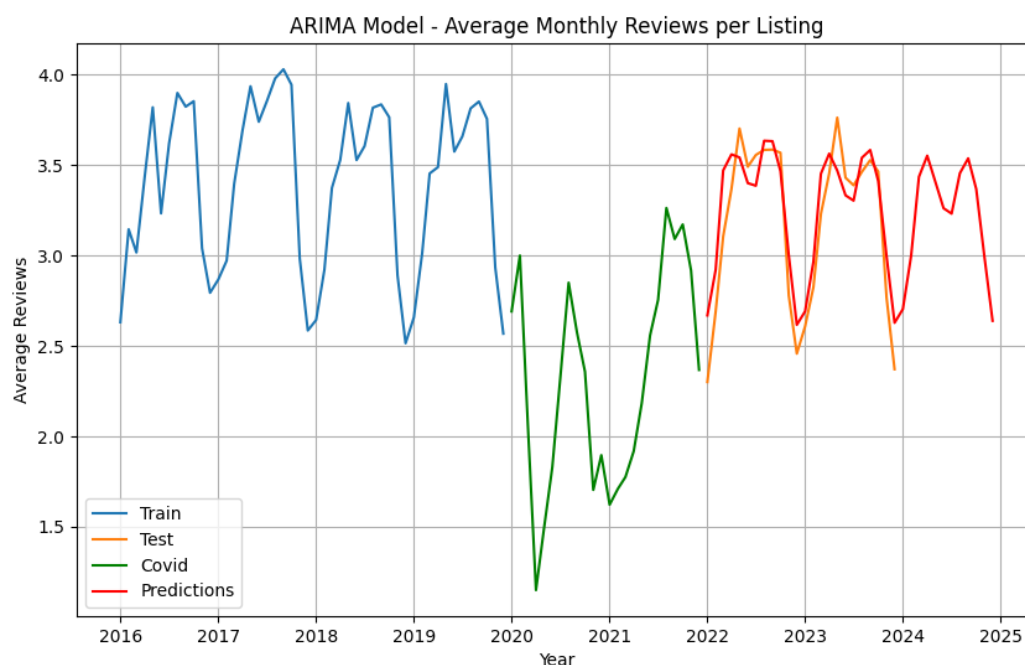


Figura 23 - Previsão da média de avaliações mensais por alojamento no Porto

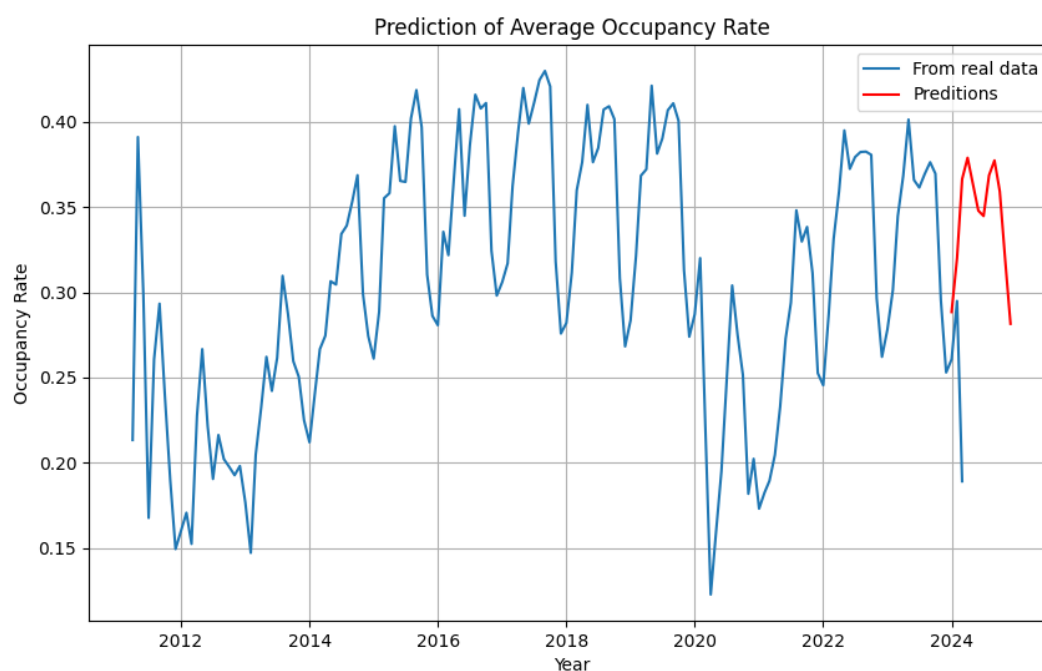


Figura 24 - Taxa de ocupação dos alojamentos no Porto

Para esta cidade, os parâmetros do algoritmo mais adequados foram: ('p': 9, 'd': 2, 'q': 6). Através da visualização dos gráficos, é possível observar uma sazonalidade entre 2016 e 2019 semelhante à observada em Lisboa. É possível destacar que, apesar de existir menos alojamentos listados na plataforma na zona do Porto, os utilizadores nessa zona tendam a deixar mais avaliações pois a média de avaliações mensais por alojamento da cidade atinge picos (4 avaliações mensais) que não foram alcançados na capital.

Comparação dos resultados com outras fontes

A tabela seguinte representa dados retirados de outras fontes de informação que correspondem a taxas de ocupação por quarto. Estas são referentes ao desempenho do turismo e habitação em diferentes zonas de Portugal durante o ano de 2022, de modo a possibilitar a sua comparação com os dados obtidos durante a realização do presente projeto.

	Projeto	INE	Turismo de Portugal
Lisboa	32,5 %	67,2 %	68,0 %
Porto	34,0 %	50,4 %	66,2 %

Figura 25 - Taxas de ocupação por quarto em diversas fontes

Os valores consideravelmente inferiores podem ter diversas causas. A mais relevante é que neste projeto só se está a ter em conta os quartos disponíveis na plataforma Airbnb, porém um dos motivos pode estar relacionado à estimativa da duração média de estadia, tal como referido anteriormente. Além destes temos de considerar o facto de que nem em todas as estadias os utilizadores deixam a sua avaliação.

Resultados obtidos e Integração no produto final

Tal como almejado nos objetivos, esta etapa do projeto permitiu fazer previsões futuras e, assim, ter ideia de como é que poderá ser o funcionamento na plataforma na cidade à escolha. Isto pode fazer com que os utilizadores finais (anfitriões de alojamentos) possam:

- adaptar os seus serviços a tendências de variação de procura sazonais (por exemplo: estar mais disponível na altura do verão ou alterar o seu preço ao longo do ano tendo em conta a procura esperada);
- estimar os potenciais ganhos pois podem ter ideia da procura por alojamento que vão ter;
- ter noção do estado do próprio negócio através da comparação dos seus resultados com as estatísticas médias obtidas pelos alojamentos da cidade.

Assim, esta parte do projeto pode ser integrada no produto final em duas vertentes: uma para a própria plataforma (por exemplo: para obtenção de dados estatísticos e antevisão de eventos de marketing da marca) e outra para os anfitriões da plataforma (dados estatísticos avaliadores de desempenho, onde seria possível comparar automaticamente os dados obtidos por cada anfitrião perante os dos restantes utilizadores). Esta última vertente até poderia ser aproveitada rentavelmente pela plataforma através de uma subscrição especial que disponibilizaria este tipo de serviço.

Adição de pontos de interesse próximos

Tal como referido no setor “etapas futuras” da parte anterior, foi considerada a hipótese da adição do número de pontos próximos de cada alojamento. Para isso, foi adicionado um passo no *Pentaho* que faz uma chamada à API do *OpenStreetMap*, retornando o número de pontos de interesse (POIs) num raio de 100 metros da localização do alojamento. Assim, os parâmetros passados à chamada API são a latitude e longitude do alojamento e o raio para o qual o cálculo é efetuado.

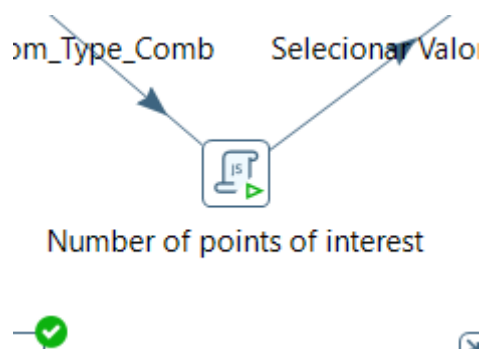
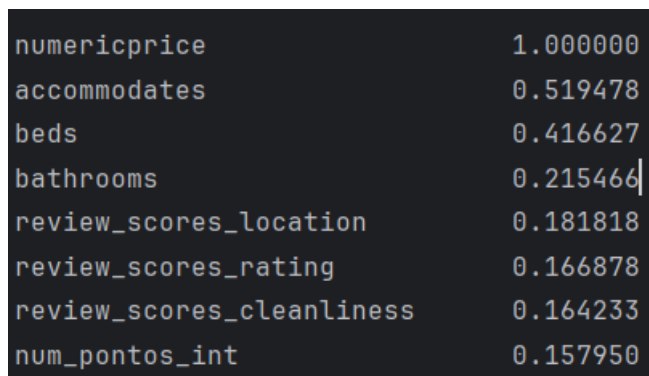


Figura 16 - Obtenção do número de pontos de interesse próximos

Apesar disso, um problema foi imediatamente identificado: o tempo de demora de chamada e consequente resposta ao serviço. A cada segundo são obtidos apenas cerca de 3 valores referentes a três alojamentos distintos. Assim, esse facto e considerando que a lista de alojamentos mais recentes continha mais de vinte mil entradas, uma

atualização à tabela da base de dados que necessitasse de rever essa informação, demoraria cerca de duas horas (número de alojamentos / taxa de receção de valores).

Além disso, foi calculada a matriz de correlação dessa contagem com o atributo do preço do alojamento (devido ao interesse em prever o preço do alojamento como referido acima) de modo a tentar perceber se esse valor teria uma correlação assim tão relevante que justificasse a sua inclusão nos dados.



numericprice	1.000000
accommodates	0.519478
beds	0.416627
bathrooms	0.215466
review_scores_location	0.181818
review_scores_rating	0.166878
review_scores_cleanliness	0.164233
num_pontos_int	0.157950

Figura 17 - Correlação do atributo da contagem dos pontos de interesse

Ao observar esta informação, foi possível perceber que atributos naturalmente inerentes a cada alojamento tinham uma correlação com o preço muito mais relevante do que o valor calculado. Assim, considerou-se que a sua adição não seria relevante para o projeto devido a esta acumulação de fatores.

Um dos fatores que mais pesaram para esta decisão é que, como este atributo teria efeito na previsão de preço do alojamento e um anfitrião não tem a obrigação de saber esse valor, este não poderia ser utilizado na previsão do preço aquando da utilização do produto final. Uma das alternativas de contornar esta propriedade seria fazer uma chamada API sempre que o anfitrião usasse o serviço (pois seria possível saber a localização do alojamento), porém não se considerou uma boa alternativa devido ao facto da solução se ter de tornar dependente do desempenho de um serviço externo sempre que esta fosse utilizada.

Conclusão – Parte 2

Ao realizar um estudo de classificação referente ao preço por noite dos *listings* e uma análise de series temporais que permite prever a taxa de ocupação dos *listings* para os próximos anos, complementamos a análise de dados OLAP feita na primeira parte, com uma dimensão descritiva e preditiva características do processo de *Data Mining*.

Assim, acreditamos que a nossa solução pode melhorar de forma significativa o processo de recomendações do Airbnb, criando um valor conjunto entre a empresa e os *hosts* que usam a plataforma para gerirem os seus anúncios. Para a plataforma Airbnb, desenvolvemos um sistema de recomendações personalizado as características dos *listings* colocados na plataforma e altamente escalável, podendo utilizar dados referentes a várias cidades, indo para além da cidade de Lisboa. Para o utilizador, oferecemos a possibilidade de aumentar os seus lucros através das recomendações sugeridas pelo sistema e ainda, a possibilidade de fazer previsões referentes a taxa de ocupação dos *listings* na localização do *host* para os próximos anos.