INSTITUTO UNIVERSITÁRIO DE LISBOA

Ano Letivo 2022/2023

8 de abril de 2023

Expedia Hotel *Recomendations*

Base de Dados de Hotéis na Expedia

UC Processamento de Big Data

Licenciatura Ciência de Dados

Grupo 17, CDB1

Docentes
João Oliveira
Adriano Lopes

André Silvestre Nº104532
Rita Matos Nº104936

Índice

Introdução

Apache Spark

O Apache Spark é uma plataforma de processamento de dados distribuída que foi criada para lidar com grandes volumes de dados de forma rápida e eficiente. Esta plataforma é compatível com várias linguagens de programação, incluindo *Python*.

Através do seu uso, podemos processar, analisar e modelar enormes quantidades de dados numa infraestrutura de computação distribuída, como um *cluster* de servidores, para realizar operações em grande escala. [1]

Base de Dados

Neste sentido, foi proposto no âmbito da UC de Processamento de *Big Data* inserida na licenciatura de Ciência de Dados, um projeto que visa implementar uma solução computacional para estudo e análise de dados de grande dimensão.

Assim, optámos por trabalhar com os dados da competição do *Kaggle* intitulada ***Expedia Hotel Recommendations*** [2], dado o desafio que a temática e a base de dados apresentam.

Para compreender melhor os dados, começámos por observar a descrição das várias colunas que se encontram acompanhadas dos dados no *Kaggle.* [1]

Adicionalmente, após percebermos que a competição é referente a dados de pesquisas de usuários que reservam hotéis nas plataformas da *Expedia,* considerámos pertinente aprofundar mais o *Bussiness Understanding* e dedicar algum tempo a perceber efetivamente como foi obtida cada *feature* /variável.

Para tal fizemos uma pesquisa de como estava estruturado o site (Fig.1) em 2016, ano em que os dados foram disponibilizados.

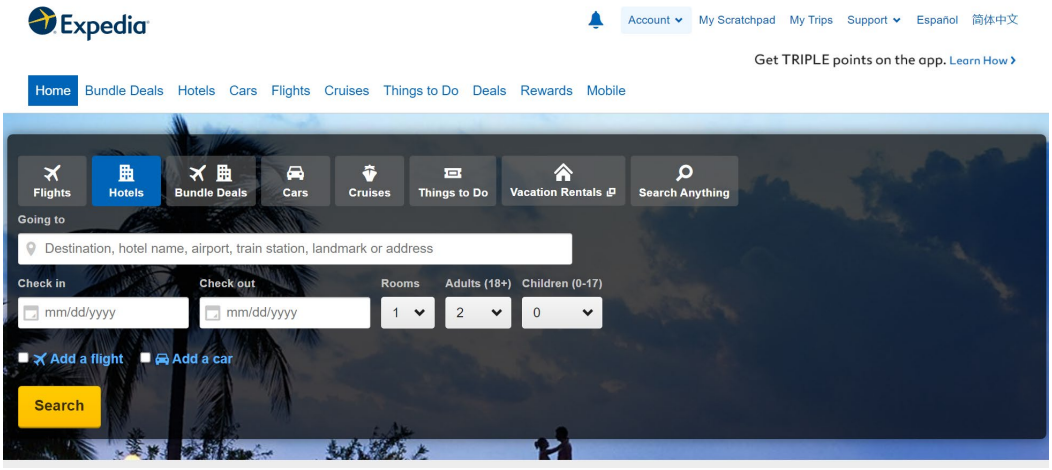

Figura 1 | Separador *Hotel* da página *expedia.com* em 2016.
Fonte: [3]

Podemos visualizar que o separador de pesquisa rotulado como *Going To* corresponde às *features* **`srch_destination_id`**, **`srch_destination_type_id`**, **`hotel_continent`**, **`hotel_country`** e **`hotel_market`**; as caixas de seleção de datas *Check in* e *Check out* correspondem, respetivamente, às *features* **`srch_ci`** e **`srch_co`**. O separador de pesquisa rotulada como *Guests* corresponde às *features* **`srch_adults_cnt`**, **`srch_children_cnt`** e **`srch_rm_cnt`**; e, por último, a caixa de seleção *Add a Flight* e *Add a Car* corresponde à *feature* **`is_package`** nos dados.

Adicionalmente, **`date_time`** corresponde ao dia e hora em que a pesquisa no site foi realizada; **`site_name`** é o nome do site visitado, seja o site principal *Expedia.com*, ou outro; **`posa_continent`** é o continente associado ao site; **`orig_destination_distance`** é a distância física entre um hotel e um cliente no momento da pesquisa; **`user_location_country`**, **`user_location_region`**, **`user_location_city`**, **`user_id`**, **`is_mobile`**, **`channel`**, **`is_booking`** e **`cnt`** são atributos determinados pelo local onde o utilizador se encontra, pelo seu dispositivo, ou pela sua sessão no site.

Questão Problema do Trabalho

A principal questão que tencionamos responder prende-se com:

Como será que podemos prever os *clusters* de hotéis que um utilizador escolherá, tendo por base o histórico de pesquisas e reservas na plataforma da *Expedia*?

Análise e Pré-Processamento dos Dados

Com o propósito de obter alguns *insights* dos dados, prosseguimos para o pré-processamento dos dados e análise exploratória dos mesmos, com base em estatísticas descritivas e visualizações.

Primeiramente, decidimos que apenas vamos trabalhar com o *dataset* **`train.csv`** e não com o **`test.csv`** porque apenas é possível avaliar a qualidade dos modelos que desenvolveremos com este, dado que a variável target é **`hotel_cluster`**, e o **`test.csv`** não apresenta esta variável. Isto ocorre, porque sendo uma competição, o objetivo seria submeter os resultados no *Kaggle*.

De seguida, verificámos que além destes *datasets*, estava disponível ainda o ficheiro **`destinations.csv`** com 150 *features* que, segundo o *host* da competição [4], correspondem a diferentes *latent factors* (por exemplo, praia, esqui, ...), sendo os valores probabilidades (log) de um cliente considerar ir para um hotel num destino para um determinado fator. Para extrair os valores apresentados foi utilizada a análise de sentimento baseada em avaliações de utilizadores, razão pela qual as informações que foram usadas para criar as colunas **`D1-D149`** não estão incluídas no *dataset* de treino.

A fim de ser possível utilizar esta informação nas análises, optámos por utilizar a técnica de PCA (*Principal Component Analysis*), tendo como objetivo reduzir a dimensionalidade através de combinações lineares entre as variáveis, de modo a reduzir o seu número e aumentando a sua interpretabilidade, sem perder muita informação.

Adicionámos colunas com datas decompostas (dados derivados das colunas **`date_time`**, **`srch_ci`**, **`srch_co`**) para mais fácil compreensão e visualização dos dados, separando a data em dia, mês, ano e dia da semana. O dia da semana do *Check-In* e *Check-Out* foi criado pois acreditamos ser algo importante a analisar, dado que estamos a trabalhar com o ramo da hotelaria. Ainda referente a estas variáveis, adicionámos a *feature* **`tempo_estadia`** como sendo a diferença entre o *Check-Out* e *Check-In*, com o objetivo de compreender a procura de estadias que os clientes mais desejam.

Também procedemos à limpeza destas variáveis após encontrarmos 827 valores anormais ($\sim 0{,}000027\%$ das observações), como por exemplo uma pesquisa para o ano 2558. Da mesma forma, limitámos a data do *Check-In* e *Check-Out* até 2020, pois como os dados são referentes a 2016 não achámos a pesquisa plausível; e retirámos as observações com tempo de estadia negativo.

De seguida prosseguimos à análise exploratório dos dados em gráficos com a visualização de gráficos e criação de tabelas de frequências. É de notar que acrescentámos no *notebook* respostas a questões que considerámos relevante explorar.

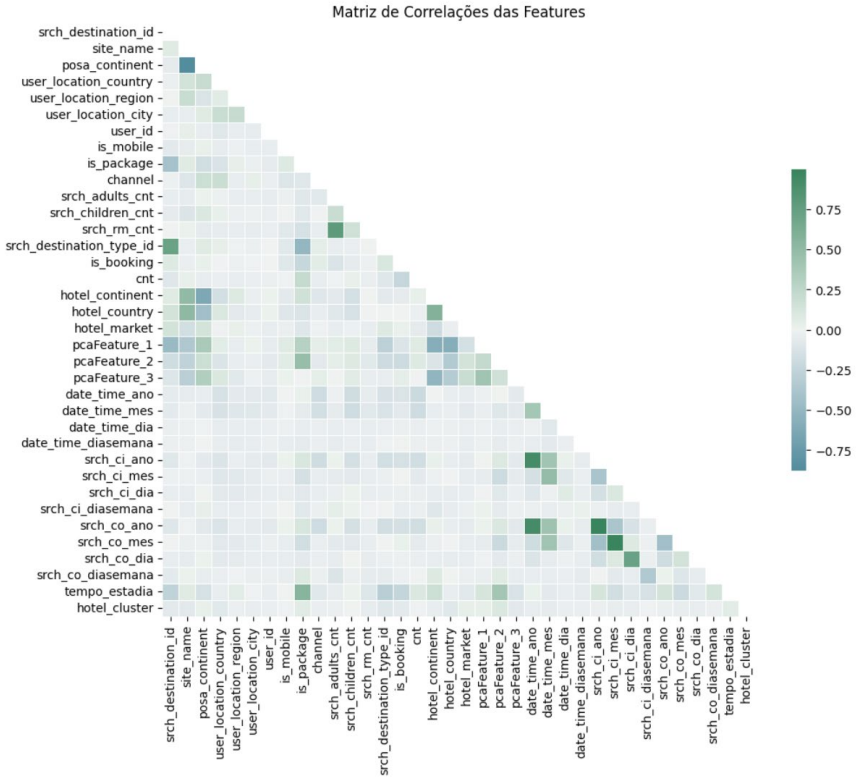

Figura 2 | Matriz de Correlação das *features*.

O gráfico apresentado mostra que as variáveis consideradas têm pouca ou nenhuma relação linear com a variável alvo **`hotel_cluster`**. Além disso, foi notada uma correlação significativa entre algumas das variáveis independentes utilizadas como preditores, o que deve ser considerado para evitar a presença de multicolinearidade, uma vez que esta pode impactar negativamente o desempenho do modelo.

Relativamente aos restantes gráficos que podem ser encontrados no *notebook*, foi notado que a distribuição da variável alvo **`hotel_cluster`** é homogênea, com média e mediana aproximadamente iguais. O tempo de estadia mais comum é de curta duração, exceto para aqueles que fazem *Check-In* e *Check-Out* no mesmo dia. Também foi observado que não há hotéis na área entre os mercados 400 e 600, sugerindo que esses mercados são nichos, exclusivamente localizados no país representado pelo número 50.

Pode-se ainda observar que há uma concentração maior de destinos entre os IDs $90\,000$ e $180\,000$, tanto para as pesquisas durante a semana quanto para as realizadas no fim de semana; e que a dispersão das reservas e dos cliques é semelhante nos dias do mês.

Na série temporal representada, apesar de as reservas e os cliques serem visualizadas em unidades diferentes, ambos seguem uma tendência semelhante, destacando-se uma elevada procura durante a época sazonal de verão e nos meses seguintes até a época natalícia.

Feature Engineering

Feature engineering é o processo de selecionar, transformar e criar características (ou atributos) a partir de dados, para melhorar o desempenho dos modelos de aprendizagem.

Dado que na etapa anterior limpámos os dados e criámos novas *features,* pois considerámos relevante fazer logo essas tarefas de modo a melhor visualizar, nesta etapa apenas voltámos a verificar valores omissos e possíveis *outliers.*

Deste modo, iniciámos por eliminar 166 801 observações (0.44%) com valores omissos de **`pcaFeature_1`**, **`pcaFeature_2`** e **`pcaFeature_3`**.

Atentámos ainda a possíveis valores *outliers*, porém não considerámos nenhum valor evidente.

De seguida, eliminámos a *feature* **`orig_destination_distance`**, visto que um terço dos dados estão omissos, situação mencionada pelo *host* da competição [5].

```
df_plot["hotel_cluster"].sort_values(ascending=True)
```

```
srch_destination_type_id   -0.033594
pcaFeature_3               -0.031486
hotel_country              -0.024434
site_name                  -0.022475
is_booking                 -0.021557
hotel_continent            -0.013947
srch_destination_id        -0.012047
user_location_country      -0.010442
srch_rm_cnt                -0.005930
srch_co_mes                -0.003439
srch_ci_mes                -0.001542
date_time_diasemana        -0.000928
date_time_ano              -0.000833
date_time_mes              -0.000452
srch_ci_dia                -0.000150
date_time_dia               0.000037
srch_co_diasemana           0.000588
srch_co_dia                 0.000629
channel                     0.000743
user_location_city          0.000822
user_id                     0.001043
cnt                         0.002990
srch_ci_ano                 0.006848
user_location_region        0.007421
srch_co_ano                 0.008060
is_mobile                   0.008415
srch_ci_diasemana           0.008668
srch_adults_cnt             0.012333
pcaFeature_1                0.013699
posa_continent              0.015010
srch_children_cnt           0.016267
hotel_market                0.034179
tempo_estadia               0.038774
is_package                  0.038902
pcaFeature_2                0.065019
hotel_cluster               1.000000
Name: hotel_cluster, dtype: float64
```

Figura 3 | Valores de Correlação de *Pearson* das *features* com o *target*.

Adicionalmente, procedemos à escolha das variáveis para os modelos, sendo o critério de seleção as correlações de *Pearson*, realizadas na análise exploratória dos dados, com valores absolutos acima de 0.01, porque neste contexto são as que apresentam os valores mais elevados (valores assinalados na Fig.3).

Por fim reduzimos a dimensionalidade do *dataset* para 30% do seu tamanho original, restando 11 236 673 observações. Esta redução deve-se a questões de produtividade, tal como solicitado no enunciado do projeto, de forma a pudermos aplicar os modelos antes de os correrrmos com a totalidade do *dataset* em ambiente *cloud* AWS.

AWS

A AWS é uma plataforma de computação em nuvem da Amazon que oferece serviços de infraestrutura para armazenamento, processamento e análise de dados, permitindo que empresas e indivíduos executem aplicações e serviços de forma escalável, segura e eficiente, sem a necessidade de construir e manter a sua própria infraestrutura. [6]

Nesta perspetiva, utilizámo-la para correr toda a parte de *Machine Learning* que requer elevado processamento para modelar todo o *dataset*. Começámos por editar o *script* de *bootstrap* para que o *cluster* tivesse todas as bibliotecas necessárias. Quando tentámos corrê-lo deparámo-nos com dificuldades pela incompatibilidade de algumas bibliotecas, porém ultrapassámos o problema, sendo que não conseguimos instalar e usar as bibliotecas de visualização necessárias.

Assim, dos 4 *notebooks* corremos com sucesso todos os códigos na AWS utilizando o *kernel* do *PySpark*, exceto os de visualização, o que não demonstrou ser um problema visto que já os tínhamos observados no computador local.

Modelos

Utilizando apenas as variáveis selecionadas fizemos a divisão do conjunto de treino (90%) e de teste (10%), para ambos os *datasets* (o reduzido e completo), visto que existem $\sim 37M$ de observações, pelo que se justifica esta proporção na divisão, pois possibilita treinar o modelo numa quantidade substancial de dados e ainda manter uma quantidade adequada para avaliar o desempenho do modelo.

Assim trabalhámos com $10\,110\,707$ observações no conjunto de treino e $1\,125\,432$ no conjunto de teste no *dataset* pequeno; $33\,708\,161$ observações no conjunto de treino e $3\,747\,416$ no conjunto de teste no *dataset* completo.

Relativamente aos modelos, usámos o ***Random Forest*** (algoritmo que usa várias árvores de decisão para classificar dados) e ***Decision Forest*** (algoritmo que usa uma estrutura de árvore para fazer previsões a partir de dados).

Após observar os resultados, com os dois *dataset*, concluímos que ambos os modelos têm um baixo desempenho na classificação do *target*, com uma *accuracy, precision*, *recall* e *F1-score* muito baixos.

Conforme orientação do professor, realizámos ajustes de parâmetros no modelo com melhor desempenho, em ambiente *cloud* AWS, para tentar melhorá-lo com diferentes configurações. Esta etapa, o *Tunning* do modelo, foi realizada com o objetivo de aprimorar a performance do modelo selecionado, o *Decision Tree*. Desta forma, testámos com diferentes valores os parâmetros `maxDepth` (número máximo de níveis que a árvore pode ter) `minInstancesPerNode` (número mínimo de amostras necessárias para dividir um nó da árvore em dois nós filhos) [7]. No final, obtivemos resultados ligeiramente superiores aos obtidos anteriormente.

Conclusões

Para este projeto como nos debruçámos sobre diversos assuntos podemos dividir as conclusões tiradas em 4 áreas: conclusões na perspetiva da empresa; no modelo; na perspetiva do cliente da *Expedia* e sobre a pergunta a que nos prepusemos a responder no início do trabalho:

- Empresa - aconselhamos o grupo *Expedia* a continuar a usar os seus dados para melhor compreender as preferências e necessidades dos clientes, e comportamentos ao reservar hotéis e outras atividades de viagem. Isto pode ajudar a empresa a adaptar os seus serviços e ofertas para atender melhor às necessidades dos clientes, aumentar a fidelidade deles e, consequentemente, aumentar as suas receitas;

- Modelo - podemos concluir que alguns *clusters* na variável **`hotel_cluster`**, tal como definidos inicialmente no *dataset*, podem precisar de ser reavaliados para garantir que as propriedades deles sejam mais distintas e significativas. Sendo a área abrangida pelo *dataset* o mundo inteiro, consideramos que 100 cluster, em teoria deveria ser o número ideal, mas que a aglomeração efetuada não foi tão adequada como esperado, o que se corroborou nos resultados dos modelos efetuados. Assim, poderia ser considerado o uso de algoritmos diferentes ou técnicas de *clustering* diferentes para obter esta variável;

- Cliente - com base na análise dos dados efetuada sobre o número de reservas dos clientes (apenas 7.98% das observações são reservas), podemos concluir que muitos apenas pesquisam e não se exprimem em compras, o que pode indicar uma necessidade de oferecer incentivos ou promoções para incentivar reservas futuras, e que há uma preferência por tempos de estadia de curto período, face a estadias mais longas;

- Questão Problema – das 40 variáveis em estudo, as que mais influenciam a escolha do hotel pelos clientes são as 16 utilizadas nos modelos, tendo-se salientado as *features* **`pcaFeature_2`**, **`is_package`** e **`tempo_estadia`**. Portanto, a empresa pode-se concentrar nessas variáveis para melhor prever os *clusters* dos hotéis.

Em suma, cumprimos com sucesso o objetivo pretendido neste projeto de implementar uma solução computacional para estudo e análise de dados de grande dimensão, analisando *insights* úteis sobre a base de dados da *Expedia* e apresentando sugestões para futuras melhorias no modelo e na análise de dados.

Bibliografia

[1] Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). Learning Spark: Lightning-Fast Big Data Analysis. "O'Reilly Media, Inc."

[2] Adam, W. K. (2016). *Expedia Hotel Recommendations*. Kaggle. *https://www.kaggle.com/competitions/expedia-hotel-recommendations*

[3] Expedia. (2016). Expedia Travel: Search Hotels, Cheap Flights, Car Rentals & Vacations. Expedia.com. *https://www.expedia.com/*

[4] AISWARYARAMACHANDRAN. (2016). *Expedia Hotel Recommendations | Discussions - What does d1-d149 mean in destinations.* Kaggle.com. *https://www.kaggle.com/competitions/expedia-hotel-recommendations/discussion/20223*

[5] Woznica, A. (2016). *Expedia Hotel Recommendations - Discussion | Data leak*. Kaggle.com. https://www.kaggle.com/competitions/expedia-hotel-recommendations/discussion/20345~

[6] AWS. (n.d.). Serviços de computação em nuvem - Amazon Web Services (AWS). Amazon Web Services, Inc. *https://aws.amazon.com/pt/*

[7] Apache Spark. (n.d.-c). *ParamGridBuilder – PySpark 3.3.2 documentation.* Spark.apache.org. *https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.tuning.ParamGridBuilder.html*