PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
IEC – Instituto de Educação Continuada
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Luiza Castelar Rodrigues Pereira

CLUSTERIZAÇÃO DE HOSPEDAGENS AIRBNB

Belo Horizonte
2022

Luiza Castelar Rodrigues Pereira

CLUSTERIZAÇÃO DE HOSPEDAGENS AIRBNB

Belo Horizonte

2022

SUMÁRIO

1. Introdução

1.1. Contextualização

O Airbnb é uma empresa que permite o compartilhamento de hospedagem, aonde o anfitrião cadastra um imóvel ou parte dele para hospedagem de um ou mais hóspedes, para estadia de curta, média ou longa duração. Está presente em diversos países do mundo.

Uma das principais ferramentas do Airbnb consiste em um sistema de filtragem aonde é possível o viajante filtrar por preço, localidade, tipo de imóvel, quantidade de comodidades que o imóvel possui, entre outros atributos para melhor escolher o imóvel e condições que o agrada.

O presente trabalho se baseia na mineração, análise e agrupamento de hospedagens por características semelhantes, a fim de trazer novos insights que permitirão ao Airbnb:

- Fazer peças publicitárias mais especificas para a região analisada
- Possibilitar ao Airbnb classificar também usuários por comportamentos semelhantes com relação às escolhas das hospedagens, uma vez que, ao agrupar as hospedagens, descobriremos também mais insights com relação à quem as usa.
- Possibilitar recomendações/avisos aos anfitriões com relação ao grupo em que sua hospedagem está inserida, de forma a melhorar sua hospedagem (ex: uma hospedagem está no grupo nomeado de “hospedagem pet friendly”. O Airbnb pode sugerir ao anfitrião investir em móveis e facilidades que sejam ainda mais compatíveis com essa realidade).
- Poder sugerir novas acomodações para o usuário, como algo complementar ao match perfeito alcançado através dos filtros (como um sistema de recomendações de hospedagens), de forma passiva.

1.2. O problema proposto

Ao realizar o trabalho olhando pelo lado de implementar um sistema de recomendações de hospedagens em alternativa ou complementar ao match perfeito, um dos resultados do mesmo será potencializar o consumo de acomodações não pela localidade mas sim pelos atributos que a mesma tem, o que favorecerá acomodações e anfitriões em locais/bairros não tão turísticos ou locais desconhecidos. Ao mesmo tempo, esse fator propiciará o sistema do Airbnb ficar mais igualitário/balanceado entre hospedagens. Com essa abordagem, encorajaremos novos anfitriões de locais não tão turísticos a se filiarem, teremos mais acomodações disponíveis, atraindo consequentemente mais turistas e finalmente gerando mais renda para a empresa, que ganha "comissão" em cada hospedagem que é reservada.

Já ao somente agrupar as hospedagens, teremos também condições de apontar características que outras hospedagens de um determinado grupo tem e que poderiam ser pontos de melhora para um anfitrião cuja hospedagem se encontra nesse grupo, de forma a atrair mais usuários e gerar mais renda também para o anfitrião.

1.3. Objetivos

O objetivo desse trabalho será agrupar acomodações com características semelhantes em clusters para poder retirar o máximo de insights para promover diversas ações de melhoria nas acomodações/na propaganda de atração de visitantes e posteriormente, se os clusters se mostrarem interessantes, implantar o sistema de recomendações de hospedagens, agregando valor para o usuário e o anfitrião e gerando novos atrativos para a entrada de novos usuários.

Para isso, passarei pelas etapas de limpeza dos dados, treinamento e avaliação do modelo escolhido, parametrizações diversas do cluster para achar os melhores resultados, comparar o modelo escolhido com outros modelos de aprendizado não supervisionado para verificar qual é o mais adequado para esse tipo de dado (por exemplo, comparar um K-means com um cluster de densidade), apresentar os dados de forma visual (gráficos) a fim de descobrir insights iniciais e por fim, interpretar os resultados finais como conclusão.

Utilizarei aprendizado não supervisionado pois é um problema sem labels/classes definidas e a divisão dos grupos que serão criados será feita pelo algoritmo. Não dividirei a base de dados em treino e teste, como é feito em aprendizado supervisionado, mas sim testaremos os diferentes parâmetros inseridos no cluster para chegar na melhor divisão.

2. Coleta de Dados

Os dados analisados foram coletados do Airbnb, por um grupo independente de pessoas (não ligadas à empresa) que se "opõe" [1] ao Airbnb[2].

Todos os dados das bases foram coletados de forma legal, uma vez que todos esses dados são públicos e acessíveis a qualquer um no app (os dados vem basicamente das páginas que o próprio anfitrião escreve, descrevendo o seu imóvel), e foram agrupados no insideAirbnb**.**

As bases consistem em:

- uma base de listagem de hospedagens,
- uma base de avaliações (chamada de reviews) de hospedagens,
- uma base com datas de disponibilidade para hospedagens (chamada de calendário)
- uma base de nome de bairros do Rio de Janeiro.

Todas as bases analisadas estão no contexto de airbnbs localizadas na cidade do Rio de Janeiro.

Embora apresente outras bases na categoria Rio de Janeiro no site insideAirbnb, elas são bases resumidas das anteriores ou são bases com dados desatualizados em relação à citadas anteriormente.

---

[1] Essas pessoas se opõe ao Airbnb, pois embora o mesmo venha como uma alternativa à indústria hoteleira, com preços competitivos e tenha como slogan "economia compartilhada", a grande quantidade de Airbnb em uma localidade trás uma série de problemas para os moradores locais em que se instala, a exemplo:

a. Diminuição de locais para locação (aluguel) de longo prazo e diminuição de aptos/casas para venda, uma vez que os proprietários dessas podem preferir fazer parte do airbnb, contribuindo também para que os preços do setor imobiliário desse local sofram alterações negativas para os residentes

b. A grande rotatividade de pessoas do mundo todo numa comunidade pode trazer insegurança (roubos, depredação de espaço público, festas barulhentas/inconvenientes em locais residenciais, etc).

[2] Os autores acreditam que, ao coletar dados, dão poder às comunidades locais para analisarem, protestarem, discutirem e acharem soluções para os problemas advindos de airbnbs de estadia curta nas suas comunidades.

Para esse trabalho, utilizaremos somente as bases de listagem de hospedagem, nomes de bairros e uma base de nomes por gênero, pois:

- A base de Calendário possui mais de 9 milhões de registros (número esse que excede o Excel e o Google Drive para armazenamento de linhas de tabela de forma gratuita), tornando-a uma base que teria que ser dividida em vários arquivos menores para manipulação.

- A base de Avaliações possui inúmeros elementos diferentes, difíceis/demorados de serem tratados, devido ao grande número de símbolos (como símbolos de ☺, 👌, ☹), acentuações, línguas diferentes (comentários em chinês, russo, alemão, francês e espanhol), TAGS (como por exemplo,
, br/>) e elementos não legíveis/não identificáveis na sua coluna principal (chamada de “comments”). Uma vez que a coluna “comments” no airbnb é um campo textarea aberto para o usuário comentar, são esperados quaisquer tipos de caracteres. Devido à essas características, não consegui manipular o .csv no Google Drive e nem abrí-lo corretamente com o Pandas.

- Por fim, e **principalmente**, justifico utilizar somente o dataset de hospedagens, juntamente com o dataset de bairros e de nomes/gênero, pois o de hospedagem possui aproximadamente **74 colunas** com 24.881 registros bem variados, o que me permite ter um bom número de atributos para trabalhar em cima, sem precisar de novos enriquecimentos de atributos por parte das outras tabelas.

Os arquivos dos datasets estão em formato CSV, que foram coletados pelo autor na sua totalidade em junho de 2022, mas possui hospedagens/hosts cadastrados desde 2008 até 2022. Fiz o download dos dados em 01/10/2022. As bases estarão disponíveis (além do link do insideAirbnb) em **driveDaLuiza** pois serão essas bases que importarei diretamente no Colab Python. Além disso, esses links evitam quaisquer problemas que venham a surgir nas bases originais (por exemplo: ficar indisponível ou mudar de link) que não permita a visualização das mesmas.

A base **hospedagens** possui **74 colunas**, organizadas em apenas um worksheet, com **24.881 registros**, que serão descritas mais abaixo. A base de dados contém uma documentação, porém nem todos os campos estão descritos de forma clara.

Nº linha	Nome da coluna/campo	Descrição	Tipo
1	ID	Identificador único de cada imóvel	Inteiro
2	listing_url	URL única de cada hospedagem no Airbnb	String
3	scrape_id	ID único para cada **conjunto** de dados de scrape. Identificadores de versão.	Inteiro
4	last_scraped	Data em que o scrape aconteceu	Data
5	Name	Nome/título dado ao imóvel pelo proprietário	String
6	Description	Descrição livre sobre a hospedagem	String
7	neighborhood_overview	Descrição do host a respeito do bairro em que sua hospedagem se localiza	String
8	picture_url	URL de uma das fotos da hospedagem. A foto da URL é a foto principal, ou de destaque, quando vemos a URL da hospedagem	String
9	host_id	Identificador único de cada proprietário	Inteiro
10	host_url	URL para a página do host no Airbnb	String
11	host_name	Nome do proprietário	String
12	host_since	Data em que o host se cadastrou no Airbnb	Data
13	host_location	A localidade aonde o host está (o host fornece esse dado).	String
14	host_about	Descrição de quem o host é (personalidade, o que ele gosta, etc)	String
15	host_response_time	Tempo de resposta do host	String
16	host_response_rate	Taxa de resposta do host (em porcentagens). Se ele respondeu ou não mensagens	Float
17	host_acceptance_rate	Taxa de aceitação de agendamentos/reservas de hóspedes (em porcentagem).	Float

18	host_is_superhost	Se o host é superhost (verdadeiro ou falso)	Boolean
19	host_thumbnail_url	URL contendo a foto redimensionada do host	String
20	host_picture_url	URL contendo a foto original do host	String
21	host_neighbourhood	Não é explicado na documentação o que se trata, mas imagino que seja o bairro aonde o host mora	String
22	host_listings_count	Número de acomodações o host está disponibilizando [3]	Inteiro
23	host_total_listings_count	Número de acomodações totais que o host está disponibilizando[3]	Inteiro
24	host_verifications	Verificação de meios de comunicação do host (verificação do número de celular, do email pessoal, do email de trabalho, etc)	Array de strings
25	host_has_profile_pic	O host tem foto? Verdadeiro ou falso	Boolean
26	host_identity_verified	Se a identidade do proprietário é verificada ou não (pede-se o upload de alguns documentos como comprovação). Verdadeiro ou Falso.	Boolean
27	Neighbourhood	Nome do bairro aonde está a hospedagem[4]	String
28	neighbourhood_cleansed	Identificação do bairro de acordo com a latitude e longitude fornecida (campos abaixo). Calculado pelo Airbnb	String
29	neighbourhood_group_cleansed	Grupo ao qual pertence o bairro, calculado pela latitude/longitude. Ex: bairro Jaraguá pertence à região da Pampulha (Pampulha sendo o grupo). Cálculo pelo Airbnb	String
30	Latitude	Latitude aonde se encontra a propriedade	Numérico
31	Longitude	Longitude aonde se encontra a propriedade	Numérico
32	property_type	Tipo de hospedagem ofertado (ex: prédio, casa)	String

[3] A documentação disponibilizada juntamente com as tabelas não informa a diferença entre host_listings_count e host_total_listing_count. Porém pela análise no python, vi que os valores nas duas colunas são exatamente iguais, me levando a crer que significam a mesma coisa

[4] A documentação não especifica esse campo. Portanto infiro que deve ser o bairro aonde está a hospedagem

33	room_type	Tipo de quarto ofertado (quarto compartilhado, quarto privado, casa inteira ou quarto hotel?)	String
34	Accommodates	Quantas pessoas pode acomodar dentro da hospedagem oferecida	Inteiro
35	Bathrooms	Número de banheiros na hospedagem	Inteiro
36	bathrooms_text	Número de banheiros na hospedagem, só que em texto invés de número	String
37	Bedrooms	Número de quartos	Inteiro
38	Beds	Números de camas disponíveis no Airbnb	Inteiro
39	Amenities	Quais facilidades a hospedagem oferece (eletrodomésticos, utensílios, etc)	Array de String
40	Price	Preço da acomodação por dia/noite	Float
41	minimum_nights	Número de noites mínimas que a pessoa tem que ficar para se hospedar	Inteiro
42	maximum_nights	Número de noites máximas que a pessoa pode ficar para se hospedar	Inteiro
43	minimum_minimum_nights	O valor mínimo dentro do range minimo de noites do calendário, considerando 365 dias no futuro[5]	Inteiro
44	maximum_minimum_nights	O valor máximo dentro do range minimo de noites do calendário, considerando 365 dias no futuro	Inteiro
45	minimum_maximum_nights	O valor mínimo dentro do range máximo de noites do calendário, considerando 365 dias no futuro	Inteiro
46	maximum_maximum_nights	O valor máximo dentro do range máximo de noites do calendário, considerando 365 dias no futuro	Inteiro
47	minimum_nights_avg_ntm	A média do valor mínimo de noites considerando 365 dias no futuro	Float
48	maximum_nights_avg_ntm	A média do valor máximo de noites considerando 365 dias no futuro	Float
49	calendar_updated	Data de atualização do calendário pelo host com	Data

[5] O mínimo e máximo de noites pode ser personalizado uma vez que o local pode ter alta temporada, baixa temporada, comemorações festivas que atraem mais gente, entre outros, modificando o fluxo de usuários. Assim, o host pode adequar o número mínimo e máximo de noites com relação à datas futuras

		relação à hospedagem, se houver.	
50	has_availability	A hospedagem está disponível no momento? Verdadeiro ou falso.	Boolean
51	availability_30	A disponibilidade dessa hospedagem pelos próximos 30 dias. Ex: se o número indicado é 23, então no próximo mês, há 23 dias disponíveis e 7 dias não disponíveis (seja porque os 7 dias foram agendados ou o host não quer os 30 dias completos disponíveis)	Inteiro
52	availability_60	A disponibilidade dessa hospedagem pelos próximos 60 dias (igual a coluna anterior).	Inteiro
53	availability_90	A disponibilidade dessa hospedagem pelos próximos 90 dias (igual a coluna anterior).	Inteiro
54	availability_365	A disponibilidade dessa hospedagem pelos próximos 365 dias (igual a coluna anterior).	Inteiro
55	calendar_last_scraped	Data em que ocorreu o scrape do calendário atrelado à hospedagem	Data
56	number_of_reviews	Número de avaliações que a hospedagem tem	Inteiro
57	number_of_reviews_ltm	Número de avaliações que a hospedagem tem nos últimos 12 meses	Inteiro
58	number_of_reviews_l30d	Número de avaliações que a hospedagem tem nos últimos 30 dias	Inteiro
59	first_review	Data da primeira avaliação feita por um usuário	Data
60	last_review	Data da última avaliação feita por um usuário	Data
61	review_scores_rating	Nota geral da hospedagem feita pelo usuário	Float
62	review_scores_accuracy	Nota dada à acurácia que o host descreveu a sua hospedagem	Float
63	review_scores_cleanliness	Nota dada à limpeza da hospedagem	Float
64	review_scores_checkin	Nota dada para o momento do checkin do usuário	Float
65	review_scores_commun	Nota dada pela comunicação entre host e usuário	Float

	ication		
66	review_scores_location	Nota dada pelo usuário a respeito da localização da hospedagem	Float
67	review_scores_value	Nota de custo/beneficio da hospedagem. A hospedagem valia o preço que se pagou nela?	Float
68	License	Se a hospedagem tem licença (e qual é ela) pelos órgãos governamentais ou reguladores para funcionar (caso precise na região/localidade)	String
69	instant_bookable	A hospedagem precisa ter aprovação da reserva ou pode-se reservar direto sem precisar passar pelo host	Boolean
70	calculated_host_listings_count	Número de acomodações o host tem nesse scrape, considerando a análise dessa região (Rio de Janeiro)	Inteiro
71	calculated_host_listings_count_entire_homes	Número de acomodações o host tem que são de casas inteiras nesse scrape/região	Inteiro
72	calculated_host_listings_count_private_rooms	Número de acomodações o host tem que são de quartos privados nesse scrape/região	Inteiro
73	calcula-ted_host_listings_count_shared_rooms	Número de acomodações o host tem que são de quartos compartilhados nesse scrape/região	Inteiro
74	reviews_per_month	Valor calculado a partir do número de avaliações que a hospedagem tem por mês[6]	Float

A base de **Bairros** possui **2 colunas** e **160 registros** não repetidos de nomes de bairros no Rio de Janeiro. Ela serve como uma padronização de nomes para bairros que irei utilizar para corrigir/comparar com os nomes presentes nas colunas de hospedagens aonde aparecem as localidades delas.

[6] Na documentação a definição está diferente, porém a definição da documentação não parece correta com o nome da variável.

Nº linha	Nome da coluna/campo	Descrição	Tipo
1	grupoBairro	Caso os bairros pertençam a um grupo. Ex: a região da Pampulha contém os bairros Jaraguá, Dona Clara, Ouro Preto e São Luiz	String
2	nomeBairro	Nome de cada bairro	String

A base de **Nomes/Gênero** é uma base criada usando duas listas de nomes separados por gênero em PDF disponível em www.dn.pt, retirada do Instituto dos Registos e Notariado. Acrescentei outros nomes além dos existentes nas listas através de algoritmo explicado na próxima seção.

Nº linha	Nome da coluna/campo	Descrição	Tipo
1	nomesMasculinos	Nomes de homens	String
2	nomesFemininos	Nomes de mulheres	String

3. Processamento/Tratamento de Dados

Dividi as colunas em grupos de análise, por serem muitas colunas. Todas elas são colunas válidas para se analisar, porém pelo tempo que demoraria, irei analisar as colunas que de fato acho que são prioritárias e que mais definem as hospedagens (em vermelho), e caso dê tempo, analisarei as demais:

Base de Bairros:

- Apenas modifiquei o nome das colunas, para ser mais intuitivo e removi a coluna grupoBairros, pois todos seus valores eram nulos.

Base de Nomes por Gênero:

- Para transformar o conteúdo dos PDFs para .csv, utilizei o sublime com o regex **[0-9]*$** para remover quaisquer números.

- Além dos nomes das pessoas separados por gênero nos PDFS, para complementar utilizei um algoritmo em cima da coluna “host_names” da base de hospedagens.

O algoritmo consistia em procurar palavras com o final “a” para mulher (ex: Adriana, Barbara, Clara) e final “o” para homens (Adriano, João, Caio). Posterior a isso eu triava os nomes que vieram de resultado do algoritmo olhando rapidamente um a um e removendo os resultados que eu considerava incorretos. O algoritmo utilizado era:

for item in lista:

```
if item[-1] == 'a':       # e “o” para homem
  print (item)            # copiei os itens do print e coloquei no .cvs de nomes
```

Base de Hospedagens:

Remoção de colunas vazias:

- **neighbourhood_group_cleansed** pois o único valor que possuía era NaN, não representando nenhuma informação relevante.
- A coluna **bathrooms** está toda vazia, mas também é representada por **bathrooms_text**, portanto são colunas redundantes.
- A coluna **license** só tem valores NaN, portanto não tem informação relevante, tornando-a apta à remoção do dataset.
- A coluna **calendar_updated** só tem valores NaN, portanto também não tem informação relevante.

Remoção de colunas sem informação relevante para a resolução do problema:

- A coluna **scrape_id** só possui um valor (id 20220620202144), uma vez que todos os registros fazem parte do mesmo conjunto de scrape. O valor só mudaria para uma versão diferente de coleta, sendo essa versão coletada antes, versão coletada depois (mais atualizada que essa) ou de outro conjunto (ex: dados de Nova York). Portanto não tem relevância para o problema.
- A coluna **last_scraped**, embora tenha 2 valores (2022-06-20 e 2022-06-21), apenas sinaliza se o dado foi pegado na mesma data, não tendo relevância para a resolução do problema. A coluna **calendar_last_scraped** tem exatamente o mesmo valor da **last_scraped**, mas se refere ao calendário. Também não tem relevância para o problema.
- A coluna **host_thumbnail_url** tem os mesmos valores da **host_picture_url** (única diferença são os parâmetros de cada uma, uma vez que esses parâmetros indicam se a

imagem virá pequena ou grande). Por terem os mesmos valores, posso excluir a **host_thumbnail_url**, pois é dado redundante.

- A coluna **host_picture_url** também será removida, pois a url da foto não é relevante, e sim se o host tem ou não foto de perfil, representado pela coluna **host_has_profile_pic**.

- A coluna **id** será removida pois não agrega nenhuma característica a respeito das hospedagens.

- Como a coluna **host_listing_count** e **host_total_listing_count** tem exatamente os mesmos valores para todos os registros do dataset, a host_listing_count será removida.

Remoção de colunas APÓS análise gráfica e exploração do dado (são colunas que servem para insights iniciais, porém não são uteis como parâmetros para o cluster):

- A coluna **listing_url** será removida pois não agrega nenhuma característica a respeito das hospedagens (todas as hospedagens tem o link da sua página). Não analisei se quando esse trabalho foi desenvolvido as URLs ainda eram válidas ou não, porém todas as URLs eram válidas quando o scrape foi feito.

- A coluna **host_name** será removida, pois já temos **host_id** para representar/diferenciar cada host.

- A coluna **host_url** será removida. Todos os registros tem **host_url**, porém essa coluna não trás informações relevantes uma vez que a tabela possui diversas outras colunas que destrincham e descrevem melhor o host do que o host_url.

- As colunas **name**, **description**, **neighborhood_overview** e **host_about** serão removidos pois são campos abertos e o dataset já possui outros campos com dados que sintetizam bem as informações advindas deles, de forma bem mais fácil de fazer operações. Exemplos:

    - Vimos que na coluna **name** (análise na próxima seção), que é o título que o anfitrião dá para sua hospedagem, muitos anfitriões nomeiam as hospedagens com o nome da localização, juntamente com o que tem no local de atra-

tivo e o tipo de acomodação. Temos a coluna de neighborhood e review_scores_location para “sintetizar” esses pontos, além de colunas como “room_type” que descreve o tipo de acomodação.

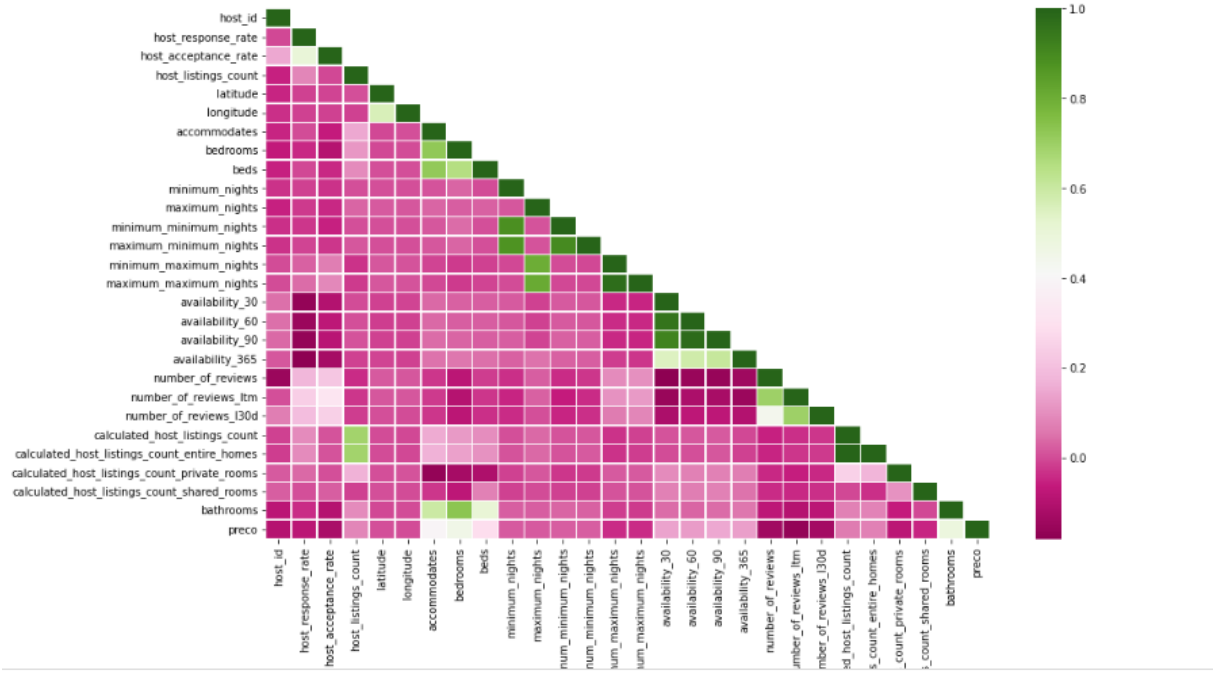
- A coluna **host_about** pode ser explicada pelas colunas review_scores_cleanliness, review_scores_rating e principalmente review_scores_communication. A primeira, quanto a limpeza do ambiente (a pessoa é asseada), a segunda é o score geral do airbnb e a terceira é a nota de comunicação entre usuário e host.

- A coluna **description** pode ser descrita pelas colunas review_scores_accuracy, review_scores_checkin e review_scores_value, ou seja, a acurácia quanto a descrição do ambiente pelo host é avaliada, o checkin descrito pelo host é avaliado e é dada uma nota relativa ao custo/beneficio da hospedagem. Temos também a coluna “amenities” que é o que a hospedagem oferece de eletrodomésticos, utensílios, etc.

Remoção de colunas pela análise de correlação

Pelo gráfico de correlação, verificamos que há várias variáveis bem correlacionadas entre si, seja de forma linear positiva ou linear negativa. A alta correlação entre 2 variáveis nos indica que uma variável explica a variabilidade e os dados da outra (e vice-versa). Assim sendo, podemos remover uma das 2 variáveis.

Após as remoções das seções acima, temos ainda 59 colunas. Dessas, 28 são numéricas e podem ser correlacionadas e são mostradas no gráfico a seguir:



Quanto mais escura a cor, seja rosa ou verde, mais correlacionada a variável está com a outra.

Verifiquei que não havia nenhuma variável correlacionada negativamente na escala -0.9. Porém, verifiquei que as variáveis abaixo se relacionavam positivamente acima de 0.9, o que as torna aptas a remoção:

- maximum_minimum_nights **X** minimum_minimum_nights
- availiabilty_30 **X** availiabilty_60
- availiabilty_30 **X** availiabilty_90
- availiabilty_60 **X** availiabilty_90
- minimum_maximum_nights **X** maximum_maximum_nights
- calculated_host_listings_count **X** calculated_host_listings_count_entire_homes

	maximum_minimum_nights	availability_90	availability_60	maximum_maximum_nights	calculated_host_listings_count_entire_homes
minimum_minimum_nights	0.902991	NaN	NaN	NaN	NaN
availability_30	NaN	0.920141	0.956201	NaN	NaN
minimum_maximum_nights	NaN	NaN	NaN	0.973519	NaN
availability_60	NaN	0.981081	NaN	NaN	NaN
calculated_host_listings_count	NaN	NaN	NaN	NaN	0.996392

Decidi remover as colunas **maximum_minimum_nights**, **minimum_maximum_nights**, **availability_30**, **availability_60** e **calculated_host_listings_count**, (de 59 colunas caiu para 54 colunas).

Tratamento de missing values em linhas inteiras

- Ao analisar as colunas **host_name** e **host_since**, percebi que 117 hospedagens não tinham diversos atributos (sem host_name, sem host_since, sem host_location, sem host_verifications, sem host_about, sem foto, sem identidade verificada, sem tempo de resposta, etc). Ao analisar essas hospedagens, consegui perceber, ao agrupar elas, que todas pertenciam a 5 anfitriões, sendo o anfitrião com mais hospedagens uma empresa (não uma pessoa física).

ost_location	host_response_time	host_is_superhost	host_neighbourhood	host_verifications	host_has_profile_pic	host_identity_verified
0	0	0	0	117	0	0
0	0	0	0	1	0	0
NaN	NaN	NaN	NaN	None	NaN	NaN
NaN	NaN	NaN	NaN	117	NaN	NaN

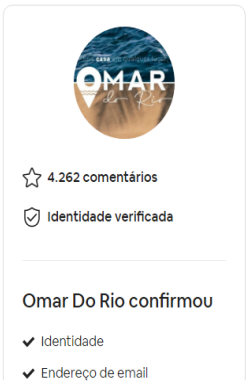
Além dos atributos do tipo String estarem vazios, haviam erros em certos atributos que deveriam ser inteiros ou floats e estavam como datetime (com frequência em que aparecia o datetime em 111 registros):

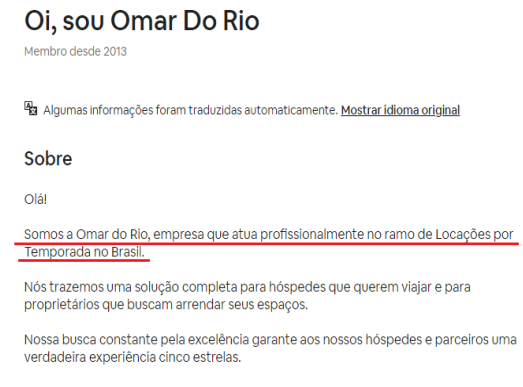
.describe()

menities	price	minimum_nights_avg_ntm	maximum_nights_avg_ntm	has_availabi
117	117	117	117	
116	45	6	5	
Dedicated rkspace", crowave", 'Fire ext...	$281.00	2022-02-02 00:00:00	90.0	
2	14	111	111	

Devido à grande quantidade de falta de informações nos atributos e a presença de atributos com valores em formatos errados, sendo a grande maioria dessas hospedagens de um anfitrião apenas (a empresa), decidi por remover essas linhas. De 24881 hospedagens, irei retirar 117, que não é um número significativo a ponto de modificar o dataset.

A empresa:





Além desses registros, localizei mais 7 registros através da análise da coluna accommodates em que várias colunas também estavam com valores missing. Invés de tratar uma a uma, preferi excluir as 7 uma vez que é um número muito pequeno. As colunas faltantes eram a “description”, “bedroms”, “beds”, “review_score_checkin”, “bathroom_text”, entre outras.

	accommodates	bathrooms_text	bedrooms	beds
4212	0.0	NaN	NaN	NaN
11262	0.0	NaN	NaN	NaN
13195	0.0	NaN	NaN	NaN
13681	0.0	NaN	NaN	NaN
16103	0.0	NaN	NaN	NaN
17069	0.0	NaN	NaN	NaN
21629	0.0	NaN	NaN	NaN

...	review_scores_checkin	review_scores_communication	review_scores_location	review_scores_value	in
...	NaN	NaN	NaN	NaN	
...	NaN	NaN	NaN	NaN	
...	NaN	NaN	NaN	NaN	
...	NaN	NaN	NaN	NaN	

De 24.881 registros, temos agora 24757 registros.

Transformação de colunas, pois o valor original não se adequa ao problema:

- As colunas **name**, **description**, **host_about**, **neighborhood_overview** foram tratadas com a criação de funções que transformam todas as palavras em minúsculas, retiram acentos e retiram elementos como preposições ‘a’, ‘e’, ‘o’, ‘da’, ‘de’, ‘do’, ‘para’, ‘com’, entre outras preposições, que não tem significado algum.

- **Coluna Banheiro**

  - A coluna **bathroom_text** apresentava valores como 1.5 banheiros. Isso significa que o “meio-banheiro” tem apenas pia e vaso sanitário, sem ter chuveiro,

ou seja, estilo lavabo. Modifiquei esse meio banheiro para ser contado como um banheiro inteiro.

- Também tinha 46 valores nulos, o que é baixo diante do dataset de mais de 20 mil registros. Substitui os valores nulos pelo valor mais frequente (1 banheiro) com frequência de 10431 banheiros únicos. Essa transformação será feita pois os 46 nulos não são significativos e colocar eles com o valor da maioria não afetará negativamente a contagem do restante de banheiros.

- Além disso, separei as palavras “shared” e “private” para banheiro privado ou compartilhado da coluna bathroom_text, colocando elas em uma nova coluna chamada **bathroom_type**. Ou seja, bathroom_text agora ficaria apenas com números e bathroom_type com os valores “shared” e “private”. Por fim, re-nomeei bathroom_text para **bathroom** apenas.

- A nova coluna gerada (**bathroom_type**) possui 20371 banheiros em que os anfitriões não se preocuparam em definir se eram ou não compartilhados. Na análise dessa coluna (próxima seção), não analisarei bathroom_type com o preenchimento do valor. Somente preencherei os valores vazios na hora de colocar no cluster. Preencherei o NaN dessa coluna com “shared” pois, além de ser o valor com mais ocorrências (ou seja, olhando pela moda), pelo senso comum, se um host tem uma hospedagem com um banheiro privado, essa característica é um agregador de valor que não deve ser deixado de fora na hora do host detalhar a hospedagem, pois valoriza bastante a hospedagem. Se ele não coloca, é porque não há relevância quanto aos banheiros, muito provavelmente por serem compartilhados, seja entre os hospedes ou entre o host e hospede.

- **Coluna Property_type**

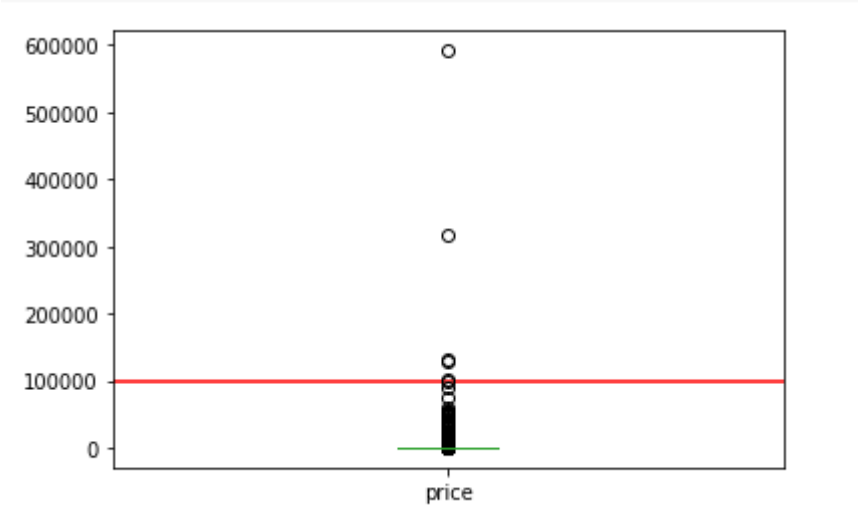
  - A transformação feita foi descobrir quais strings únicas que essa coluna tinha (que eram muitas) e padronizar/agrupar em um número de string menores, para poder plotar em gráfico e facilitar a análise. Então, strings como: “shared

room in home” e “private room in home” viravam simplesmente “home”, já que havia a coluna room_type que descreveria se a hospedagem era privada ou compartilhada.

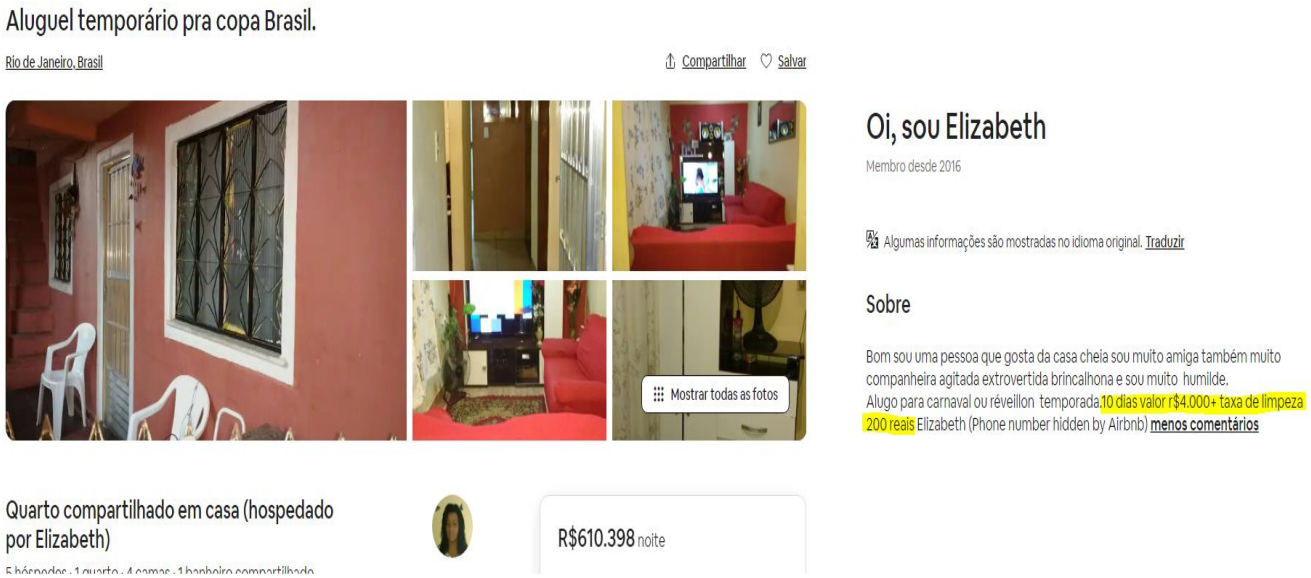
- **Coluna Price**

  - Estava em dólar. Removi o sinal de dólar, virgula e ponto (os decimais após o ponto poderiam ser os centavos, mas ao analisar vi que todos os registros estavam com os centavos zerados, portanto podia transformar tudo em inteiro).

  - Há outliers (5 hospedagens tem valores superiores a R$100.000 **NA DIÁRIA** e 119 hospedagens, incluindo as 5 anteriores, tem valores superiores à R$10.000,00 **NA DIÁRIA**), o que me faz constatar que com certeza os hosts erraram nos valores. Removi esses valores, uma vez que 119 registros de 24547 hospedagens é um valor insignificante (colocar o valor médio não é aconselhável nesse caso, primeiro porque o valor da média com os outliers fica muito acima do valor real e colocar média somente é aconselhável quando mais da metade ou um grande número de valores daquela coluna está incorreto ou faltante, uma vez que tentaríamos “salvar” a coluna invés de descartá-la). A análise por gráfico de outliers também demonstra o quão fora do normal os valores estavam.



Um exemplo de hospedagem (esse é o de valor máximo no outlier encontrado). A moça realmente cobra R$610.398 por noite, mas no perfil dela, está escrito “10 dias por R$ 4000 + taxa de limpeza de 200 reais”. O que mostra o equívoco na hora de criar o anúncio.



- **Colunas Bedrooms e Beds**

  - Os missing values das colunas **bedrooms** são 5,5% do total de dados e **beds** são 1% do total. Não são valores tão expressivo a ponto de mudar a configuração dos dados. Portanto, irei completar os missing values com o valor que mais aparece (1.0 bedrooms e 1 para beds). Esses valores não podem ser descartados, pois a coluna irá fazer parte da próxima etapa de inserção no machine learning e para descartar o valor eu teria que descartar o registro, que é importante.

  - Já para outliers em bedroom, irei remover os seus registros, pois eles não representam a realidade das hospedagens e seu valor modifica de forma errônea a média. Removerei outliers que representam hospedagens com mais de 10 bedrooms (são 16 registros), embora, ainda vão sobrar alguns outros outliers. Em beds, também haverá a remoção de outliers (são 194 registros) em valores com mais de 10 beds.

- Na imagem abaixo, vemos a análise de **bedroom** antes da remoção dos outliers, em que mais de 50% dos seus dados são o valor 1.

```
bedrooms
1.0     14453
2.0      6245
3.0      2941
4.0       700
5.0       222
6.0       115
7.0        32
8.0        11
9.0        11
10.0       11
11.0        5
12.0        3
15.0        2
17.0        1
18.0        1
20.0        2
30.0        1
47.0        1
Name: bedrooms, dtype: int64
```

```
Bedrooms Nulos: 1361
Porcentagem de nulos: 5.497435068869411 %
count    23396.000000
mean         1.701103
std          1.094016
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max         47.000000
Name: bedrooms, dtype: float64
```

- **Coluna Amenities** (comodidades):

  - Composta de uma string simulando um array de palavras/sentenças para cada registro. Quebrei em palavras/sentenças menores, juntei num array enorme e agrupei a ocorrência de cada sentença. Exemplo:

    Linha 1 = "['TV', 'maquina de lavar', 'conta netflix para celular, desktop e dá pra colocar na TV']"

    Linha 2 = "['Sabão', 'Pia', 'conta netflix para celular, desktop e dá pra colocar na TV']"

    A linha 1 inteira é uma string. Transformei em array de verdade. Depois, contei as ocorrências:

    ['TV': 3, 'Sabão':1, 'Pia': 1, 'máquina de lavar':1]

    A frase "conta netflix para celular, desktop e da pra colocar na TV" agrupei sobre a categoria "TV" para diminuir o número de facilidades e agrupar em grupos maiores para análise.

- Criei um dicionário para poder agrupar as palavras que mais apareciam para poder visualizar melhor o gráfico de facilidades.

- **Coluna host_since**

  - Agrupei datas por ano. Não precisei remover nulos, pois já não tinham depois dos tratamentos anteriores com outras variáveis.
  - Uma vez analisado os dados, modifiquei a coluna para ser somente o ano de entrada, por ser um dado mais fácil de analisar do que o dia e mês.

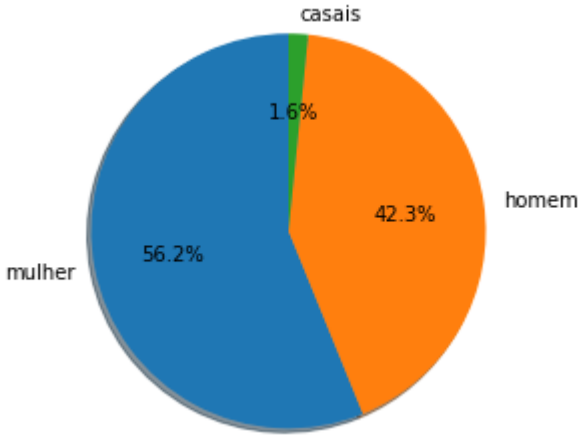
4. Análise e Exploração dos Dados

- A coluna **license** está toda vazia. Essa coluna representaria “Licença dada aquele imóvel de ser airbnb por alguns governos/prefeituras (nem todos os locais aonde o imóvel está exigem essa licença)” o que me faz inferir que no Rio de Janeiro não é preciso qualquer licença para ter uma hospedagem Airbnb.

- Analisando o **host_name**, percebi que há mais mulheres ofertando hospedagens que homens, na média. Também verifiquei que há empresas/companhias de hotéis ofertando quartos no aibnb. Ex:

```
'Carolina Lis',
'Gilles',
'Luciane & Milton',
'Linkhouse',
'Estadia',
'Giselle & Jorge',
'Andres & Manu',
"Sam'S Home",
'Gricel',
'Sonali',
'Micheline & Jorge',
'Gessé',
'Alves',
'Yes Temporada',
'Flat Residence',
'Apart-Hotel Convention Rio Stay',
'一麟',
'Tupiniquim Hostel',
'Gustavo & Valesca',
'Jacque',
'Goya',
```

Também há usuários que são casais (e portanto colocam ambos os nomes, da mulher e do homem) como se fosse um perfil só.

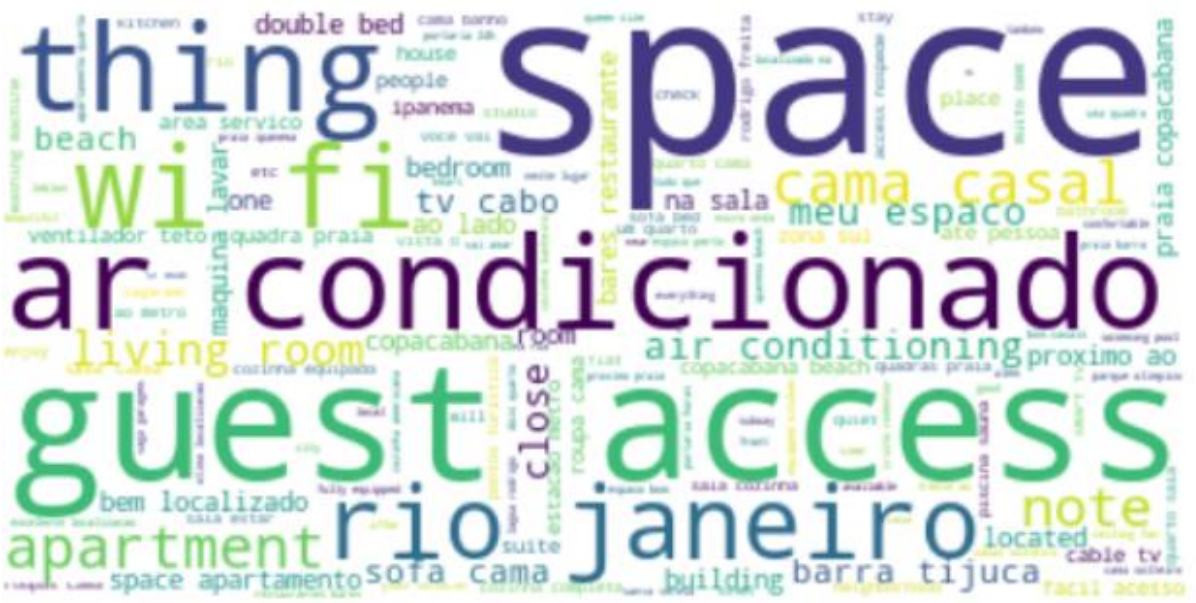


Para colunas abertas, em que o host pode digitar tudo, optei por utilizar nuvens de palavras com a contagem das palavras que mais apareceram para retirar insights:

- A coluna **name** (que é o nome que os proprietários colocam nas acomodações) mostra que a maioria das hospedagens se localiza nos bairros de Copacabana, Ipanema, Barra da Tijuca e Leblon. São usadas muitas palavras como praia/beach, suíte, quarto e apartamento, o que sugere que muitas suítes são ofertadas, hospedagens próximas ao mar, e são ofertados tanto quartos (modalidade somente quarto e não o apto inteiro) e a modalidade apartamento (inteiro).

- Já para a coluna **description** vemos que há várias palavras em inglês, o que denota que as hospedagens devem receber muitos turistas estrangeiros. Também é bem visível

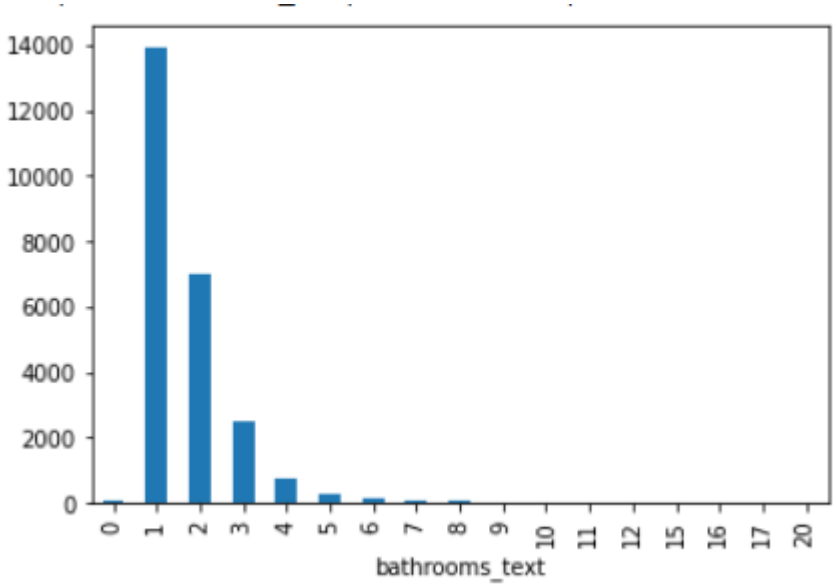
que grande parte das hospedagens tem ar condicionado, uma vez que o Rio de Janeiro é muito quente, grande maioria tem wi-fi, TV a cabo, cama de casal, e deve ter um tamanho razoável ou bom, pois a palavra “space” se destaca entre todas as outras.



- Pela coluna **host_about** dá para perceber que embora seja uma coluna para falar de si, há muitos hosts que enfatizam a localidade Rio de Janeiro (seria porque nasceram no Rio, ou até para falar sobre a própria hospedagem?), enfatizam o fato de ser uma “cidade maravilhosa”. Pelo que parece, grande parte gosta de viajar, são cariocas, são pessoas que gostam de ajudar (ou se o usuário precisar de ajuda em algo, simbolizado pela palavra “help”). São pessoas com algum grau de conhecimento em inglês. Gostam de viver a vida (pelas palavras “live” e “love”).

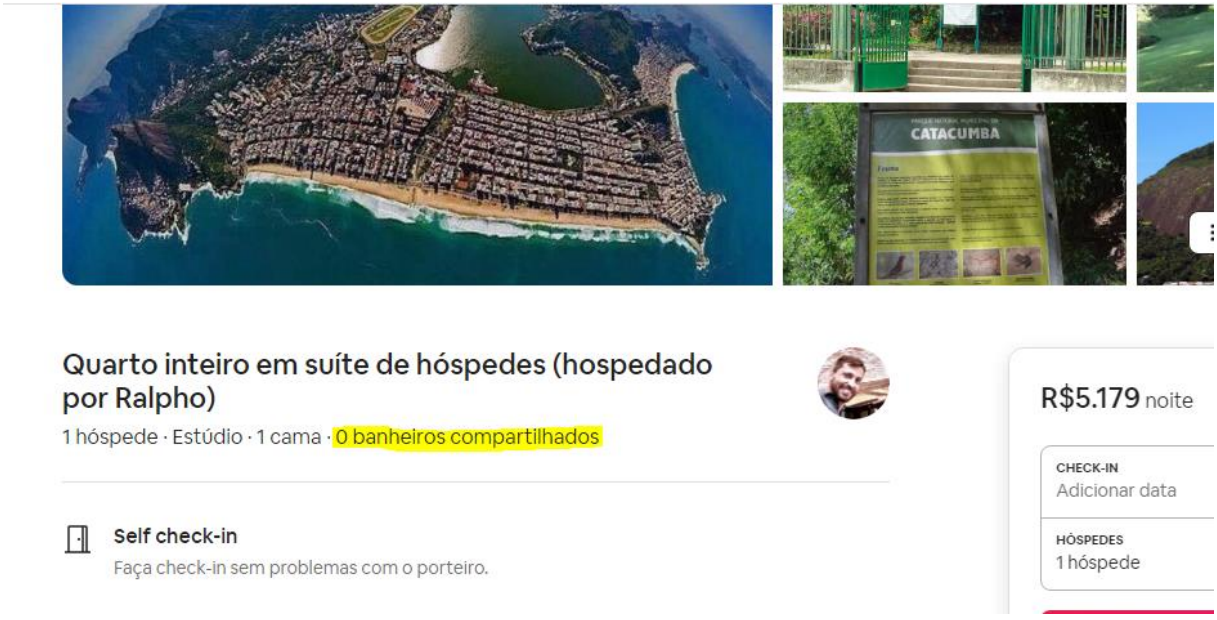
- Por fim, temos a última coluna de texto aberto, **neighborhood_overview**, que mostra que na vizinhança das hospedagens é muito mencionada as palavras beach (praia), shopping, restaurantes e bares. Além disso, há a referência de “fácil acesso” dos locais em relação às hospedagens, talvez também simbolizada pela palavra “metrô”, e novamente aparecem as palavras Barra da Tijuca, Ipanema, Copacabana, Zona Sul, etc (igual apareceu na coluna “name”), indicando os locais em que podem estar as hospedagens ou da onde elas podem ser vizinhas ou bem próximas.

- Fiz análise univariada da coluna **bathrooms_text.** Embora seja evidente a análise que a grande maioria das hospedagens tem 1 ou 2 banheiros, vemos pelo gráfico que existem hospedagens que não possuem banheiro (uma pequena linha azul perto do 0):

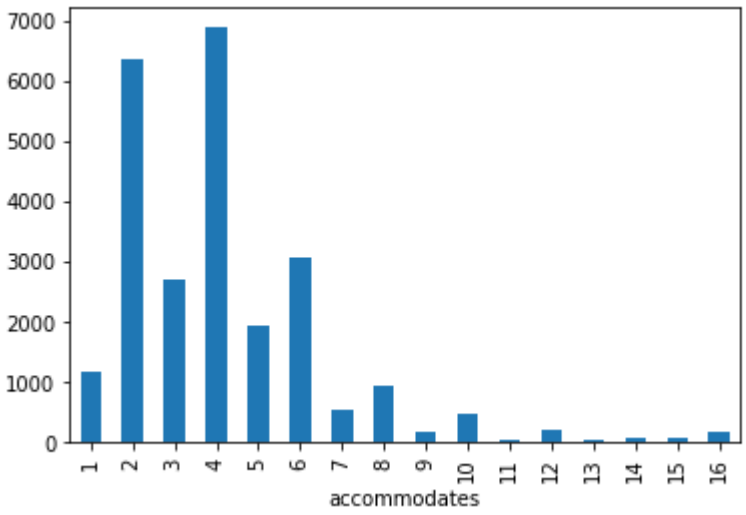


Vemos também, pela tabela abaixo, que somente o banheiro único é privado. O restante é todo compartilhado. Também vemos que certos anfitriões colocaram erroneamente que possuem 0 banheiros compartilhados, uma vez que isso abre interpretações diferentes (eles não possuem nenhum banheiro compartilhado, então, todos os banheiros são privados? Ou eles não possuem nem um banheiro e o anfitrião colocou a palavra “shared” erroneamente?). Vemos que existem hospedagens com 20 banheiros.

	bathrooms_text	bathrooms_type	size
0	0	shared	29
1	1	private	1472
2	1	shared	1905
3	2	shared	769
4	3	shared	144
5	4	shared	35
6	5	shared	17
7	6	shared	6
8	7	shared	7
9	9	shared	1
10	20	shared	1



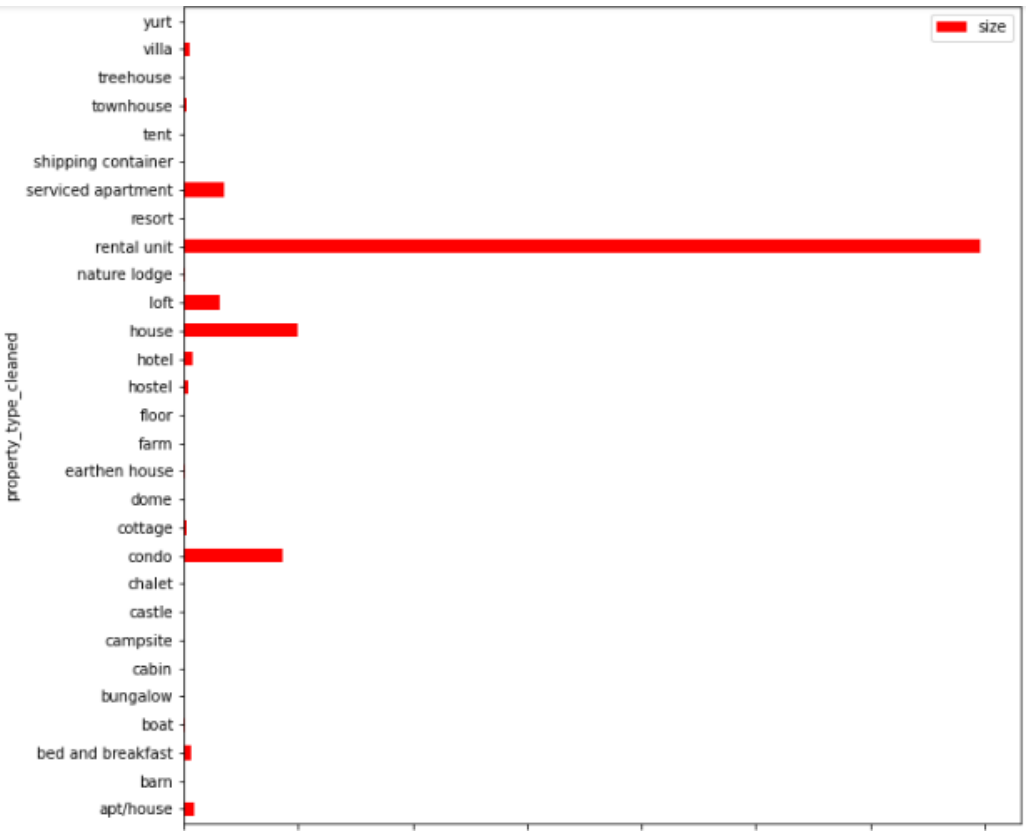
- A coluna **acommodates** (número de pessoas que a hospedagem comporta) tem valor máximo de 16 pessoas, sendo em sua maioria hospedagens que acomodam 2, 4 ou 6 pessoas.



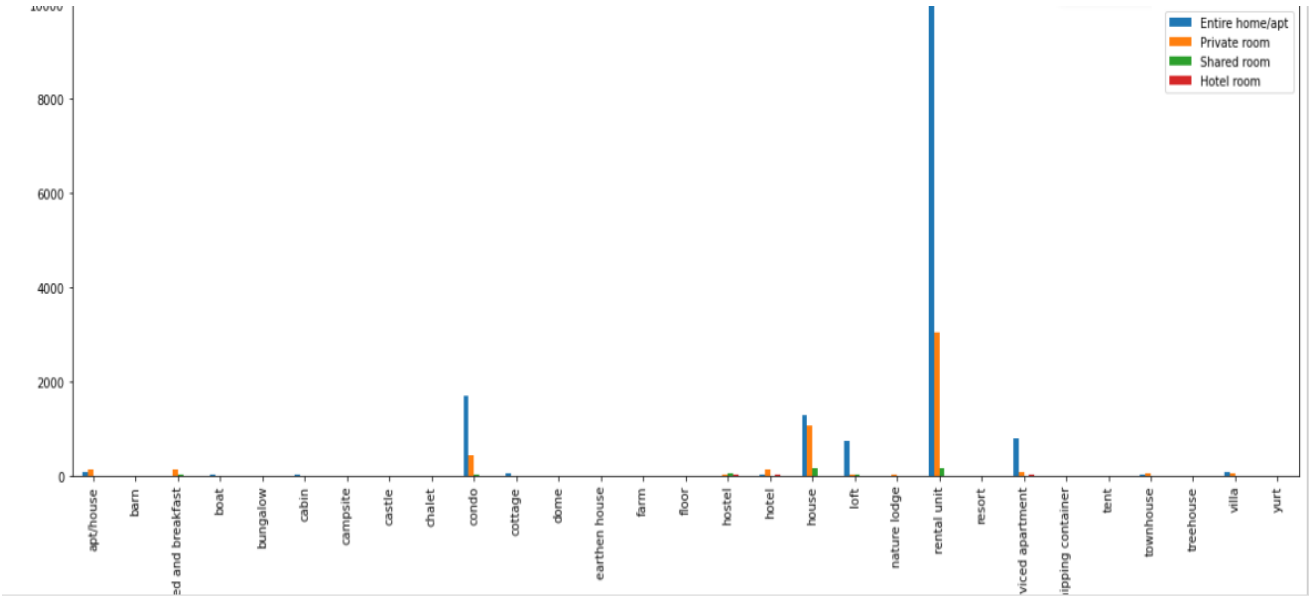
- Na coluna **propery_type** (tipo de propriedade):
    - Ao agrupar as propriedades em um número de categorias menores, verifiquei que as “rental unit”, ou unidades de locação eram as mais presentes, seguidas de casas, apartamentos e serviced apartment (uma mistura de hoteis-apartamento), respectivamente.

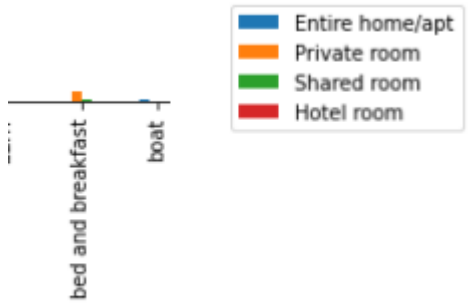
    - Verifiquei que o tipo de propriedade que um anfitrião pode declarar é extremamente variada.

    - Também verifiquei pela relação **property_type** com **room_type** (no gráfico mais abaixo) que é mais comum em todas as propriedades a hospedagem ser inteira do que compartilhada.



- Também verifiquei que a única modalidade em que a grande maioria das hospedagens é de quartos privados é na modalidade “bed and breakfast”, como mostra o recorte do gráfico mais abaixo. Bed and breakfast, de acordo com o google são: “pequenos estabelecimentos de hospedagem que oferecem pernoite e café da manhã. Bed and breakfasts são muitas vezes casas familiares privadas (...). Além disso, um B&B geralmente tem os anfitriões morando na casa”, o que justificaria o porquê deles serem hospedagens compartilhadas.

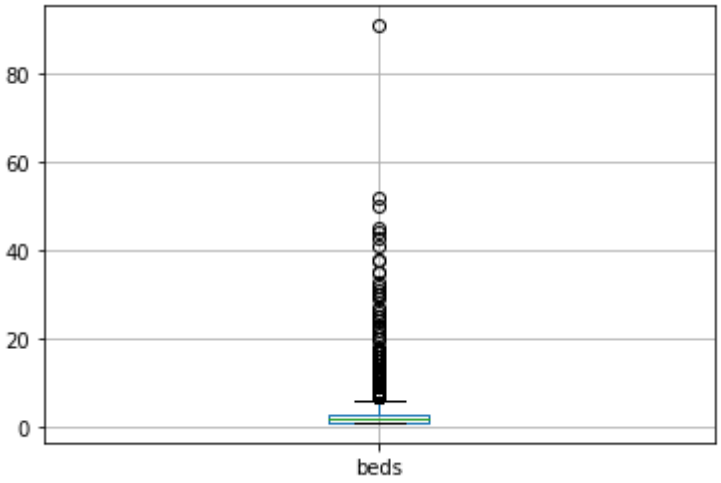




- A coluna **bedrooms e bed** nos mostra novamente que há hotéis e grandes estruturas com muitos quartos e camas, alguns talvez desproporcionais (47 quartos e apenas 1 cama???), o que leva a crer que o anfitrião colocou os números errados (a interpretação pode ser que tenha 47 quartos, cada um com uma cama):

	name	bedroom	beds
655	Clube pontal festas passeios e hospedagens	30	11
1341	Hotel Yoo2 Rio De Janeiro By Intercity - Yoo2 ...	47	1
18863	FLATS PONTAL COUNTRY CLUB	20	4
19118	PONTAL COUNTRY CLUB	20	6

Para a coluna **beds**, vemos que há vários outliers, mas um se destaca com 91 camas.

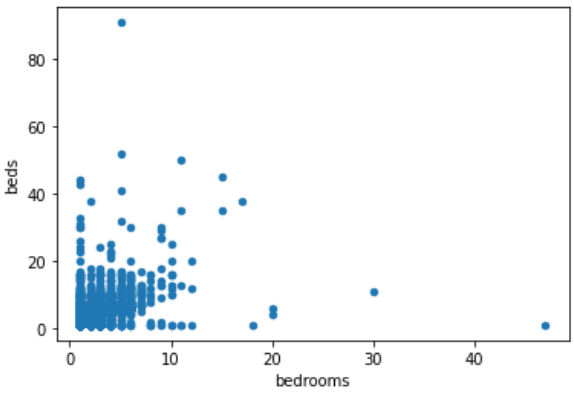


Ao olhar o anuncio do outlier da hospedagem com 91 camas, vemos (como mostra a imagem abaixo) que de fato é possivel o estabelecimento ter ofertado corretamente:

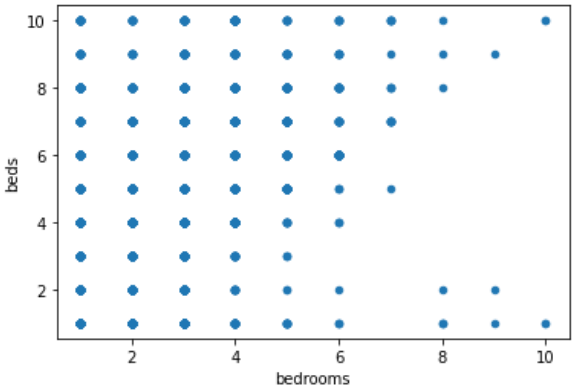
A foto abaixo mostra outro registro, em que há 50 camas e 11 quartos. Na última foto é possivel ver 3 camas em um quarto, para a gente ter a noção de como está essa relação cama X quarto:

A relação entre quartos e camas nos mostram que a grande maioria das hospedagens tem no máximo 20 camas e menos de 10 cômodos (considerando o gráfico com outliers).

Quando retiramos os outliers (deixando tanto beds e bedrooms com no máximo o valor de 10 cada), vemos que a hospedagem que tem 7 bedrooms, tem pelo menos de 7 até 10 camas; se ela tem 8 bedrooms, ela tem 8 a 10 beds, se ela tem 9 bedrooms, tem 9 camas e se tem 10 bedrooms tem 10 camas (quase uma relação 1 cama por quarto). Já para abaixo de 7 bedrooms (de 1 a 6 bedrooms) temos de 1 a 10 camas, ou seja, temos locais com mais de uma cama por bedroom.



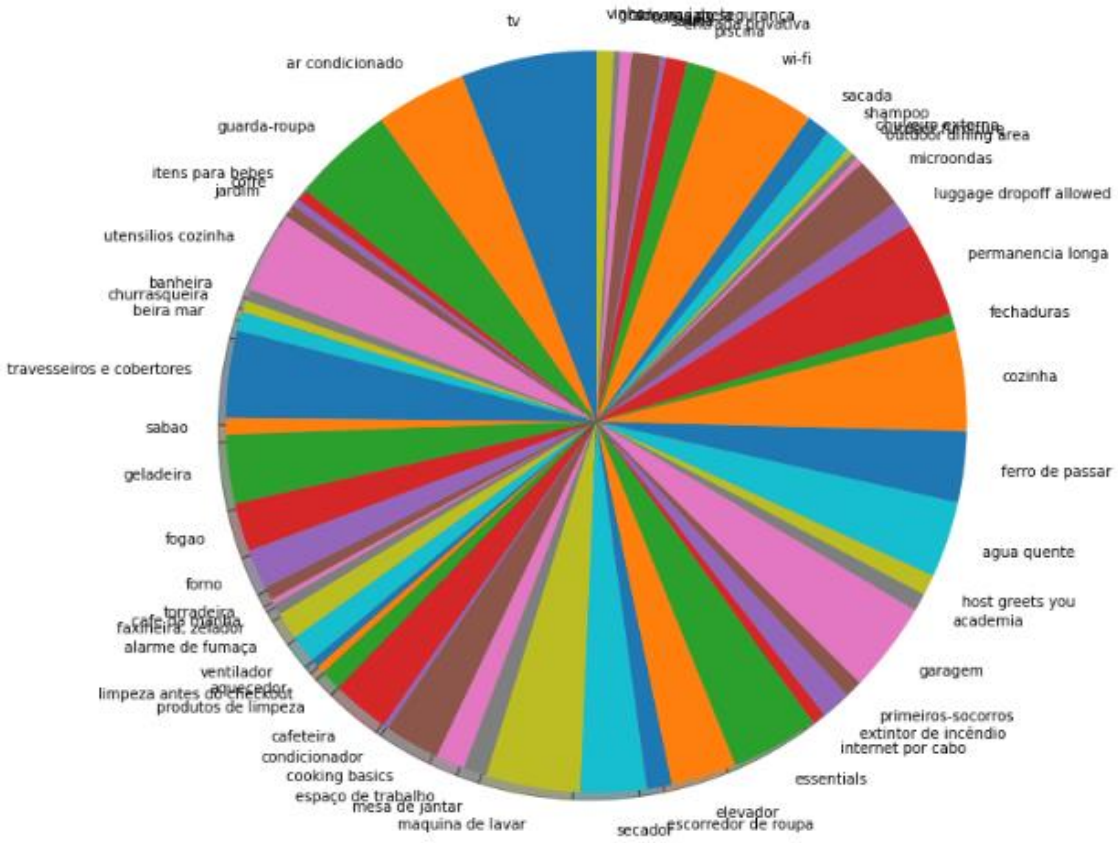
Com outliers



Sem tantos outliers tanto na variavel beds como bedrooms

- **Coluna Amenities (facilidades):**

- São 532600 palavras totais (ou seja, itens em cada hospedagem listado) separadas em 2697 tipos de itens. Decidi analisar itens que tinha mais de 20 unidades. Desses, caiu para 285 tipos de itens com mais de 20 unidades.

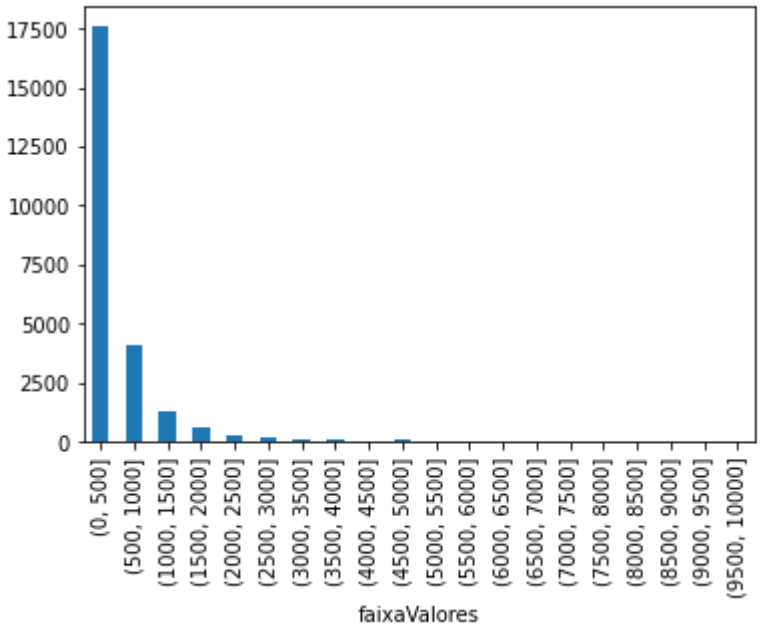


- Podemos verificar que ar condicionado, presença de TV ou elementos associados a TV, garagem, máquina de lavar, cozinha, travesseiros e cobertores, wi-fi, água quente, itens essenciais (Essentials, como itens de higiene), guarda-roupa, geladeira e ferro de passar, são os itens que mais se destacam em vários airbnb.

- Relembrando a coluna **description**, em que eu descobri que os anfitriões descreviam as suas hospedagens como possuindo wi-fi, TV a cabo e ar condicionado, vemos novamente na coluna **amenities** essas palavras aparecendo fortemente.

- **Coluna Price (preço)**

    - Ao dividir a coluna em 20 bins, vemos que mais da metade das hospedagens cobram entre R$33 a R$500 a diária. O menor valor é R$33 e o maior é R$10.000 (uma vez que removi outliers inconsistentes).

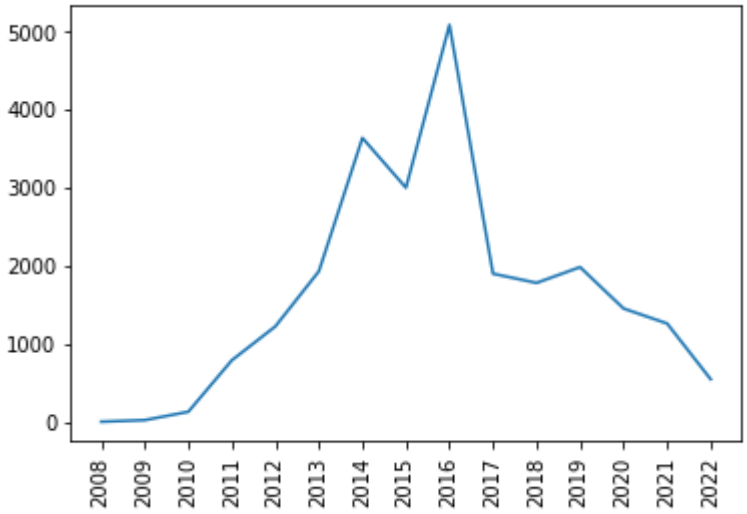


- **Coluna host_since (anfitrião desde quando)**

    - O gráfico nos mostra que os primeiros anfitriões entraram em 2008, e que desde então entravam cada vez mais anfitriões, até 2016 que foi o pico. De 2016 a 2017 teve uma queda brusca, e desde então, até 2022, não houveram entradas maciças de novos anfitriões igual antes.

    - Hipótese: isso poderia ter algo haver com a economia do Brasil, que em 2016 passava pela pior recessão da história do país? Ou seja, o carioca poderia tentar renda extra através do Airbnb, caso estivesse passando dificuldades em 2016, ou o contrário, parar de participar do Airbnb em 2017, pois perdeu a casa/apto “principal” e precisa do imóvel que deixava no airbnb (virando agora a única moradia)? Ou pode ter relação com os Jogos Olimpicos de 2016, que aconteceram em Agosto no Rio de Janeiro, e, através do Airbnb, o carioca

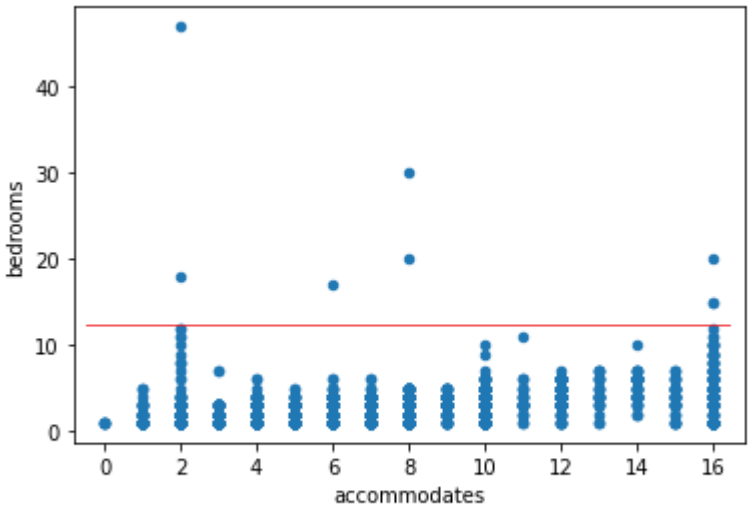
tentou ganhar renda extra? Uma vez que os Jogos acabassem, não haveria grandes motivos para continuar ofertando várias hospedagens.



Análises de 2 ou mais variáveis

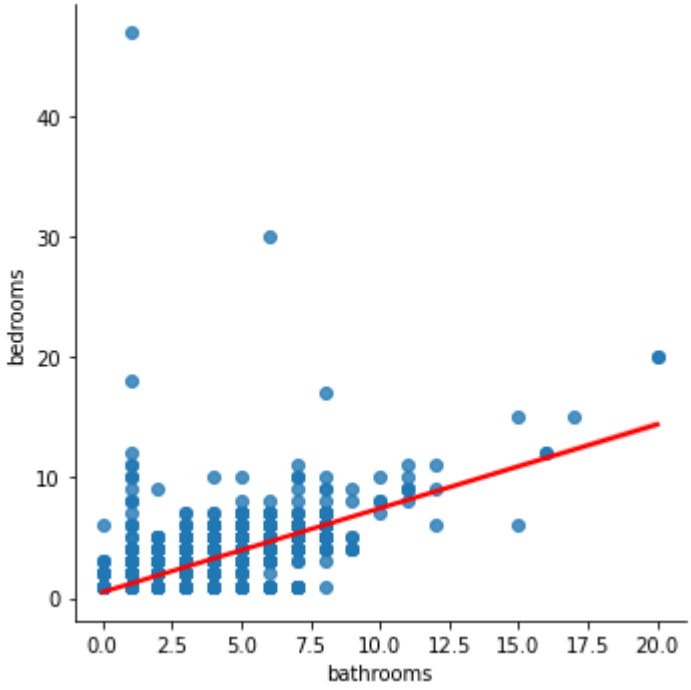
Uma vez que as principais variáveis estão tratadas, podemos relacionar elas entre si formando mais gráficos para tirar insights.

- **Bedrooms x accommodates (número de pessoas)**



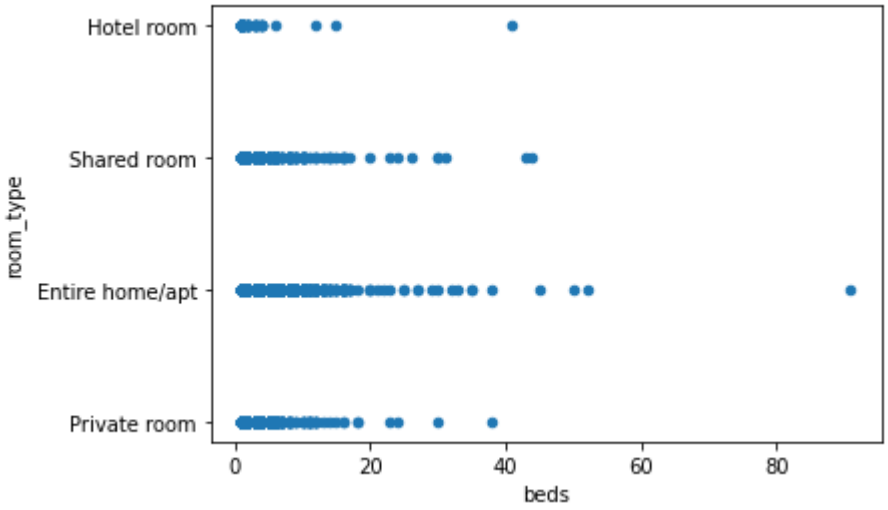
O número de quartos não parece se diferenciar muito mesmo quando a hospedagem aceita mais hospedes (vide 2 hospedes com relação à 16 hospedes, pela linha vermelha, o número de quartos se mantém muito semelhante, próximo a 12). Isso poderia me indicar um esquema “mais camas por quarto”, estilo hostel, ou seja, por exemplo, um quarto com 16 camas acomoda 2 viajante mas acomoda também 16 pessoas (seja grupo ou 16 pessoas separadas). Pode não haver distinção para o anfitrião se ele esta acomodando 2 ou 16.

- **Bedrooms x bathroom**



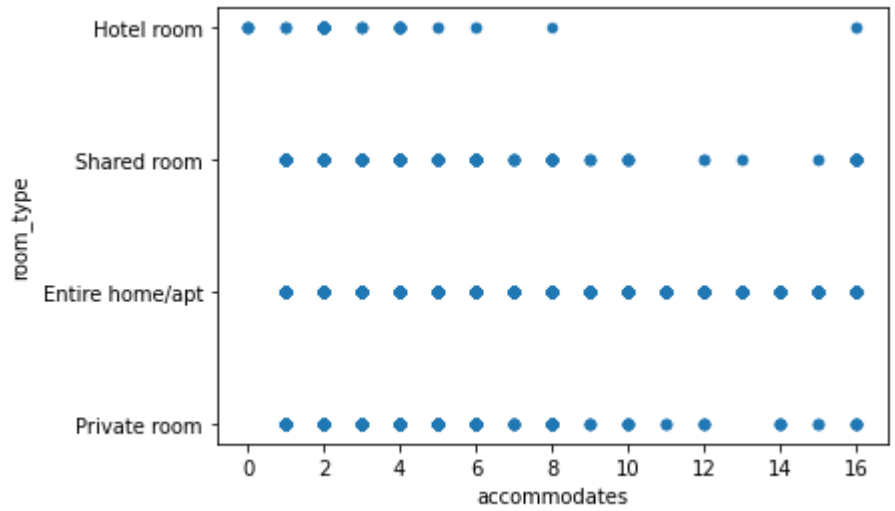
Ao aumentar o número de banheiros, o número de quartos aumenta proporcionalmente, de forma a seguir a reta vermelha, com exceção à quando se tem 1 banheiro, que permeia praticamente várias hospedagens desde as sem quartos até as com mais de 10.

- **Beds x room_type**



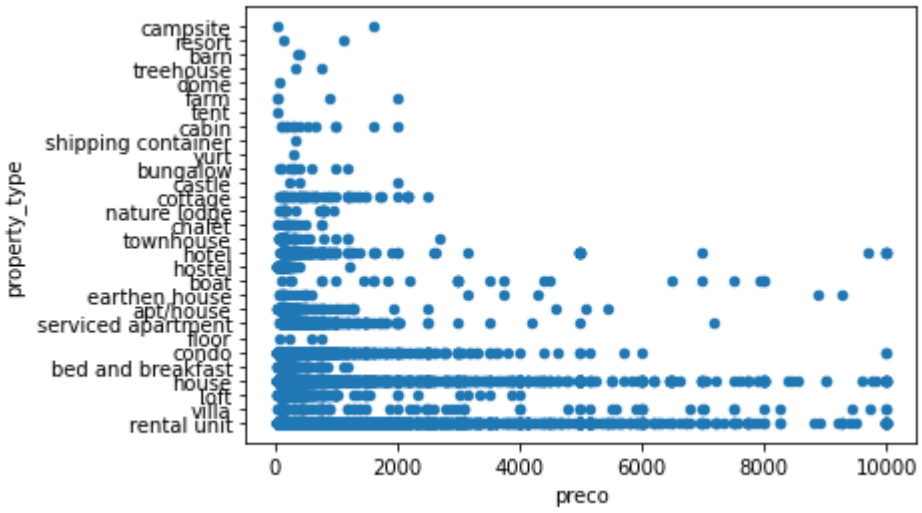
Vemos pelo gráfico que quando a hospedagem é de apt/casa inteira, o número de camas pode ser ainda maior que 20 camas. Vemos que nas três modalidades (quarto compartilhado, casa inteira ou quarto privado), há grandes concentrações de hospedagens de 0 a 20 camas. Vemos que quartos de hotel não passam de aproximadamente 5 camas por hospedagem.

- **Accomodates (número de pessoas) x room_type**



Vemos que os dados estão bem distribuidos, exceto quando analisamos quartos de hoteis, em que mais de 8 pessoas até 16, não existem registros, apenas um outlier de hotel com 16 pessoas. Também é notado que pode ter um erro/"sujeira" no dado, pois a primeira marcação de hotel é para 0 pessoas.

- **Preço x accommodates (número de pessoas)**



As propriedades do tipo rental_unit, além de serem grande maioria, são as que variam mais de valor, chegando à valores máximos de R$10.000 por diária.

As propriedades mais baratas estão em sua maioria de cima para baixo (exemplo: campsite (casa de campo), resort, barn (celeiro?), treehouse (casas na árvore), dome (casa circular), farm (fazenda), tent (tenda), floor, yurt (lembra uma tenda), castle (castelo), shipping container (casa container).

Verificações de suposição do(s) modelo(s) de Aprendizado de Máquina

Para o problema utilizando clusterização, é necessário que todas as colunas sejam transformadas em números, uma vez que o modelo só aceita números. Para tanto, uma técnica a ser utilizada é a transformação em dummies das colunas (transformação em 0 ou 1 e a consequente extensão do número de colunas). Algumas colunas no dataset já são naturalmente 0 ou 1, numéricas ou foram transformadas em número na hora do tratamento.

Será feita a transformação com 0 e 1 ao invés da categorização ordinal pois as variáveis analisadas para transformação não tem uma ordem, ou seja, se eu categorizasse os valores, por causa da ordem, elas iriam ganhar peso (por exemplo: ruim - 1, médio – 2, bom - 3). Ao pegarmos os valores que assumem a variável property_type, por exemplo, uma

hospedagem do tipo apartamento, casa ou outro tipo de estrutura não a faz ser mais ou menos importante dentre os demais valores dessa variável. Todos têm o mesmo valor e peso entre si.

As variaveis que não estão em número mesmo após o tratamento, dentre o conjunto analisado (property_type, room_type, accommodates, bathroom, bathroom_type, bedrooms, beds, amenities e price) são**: property_type, room_type, amenities e bathroom_type**.

	bathrooms_type	bathrooms	accommodates	bedrooms	beds	property_type	room_type	preco
0	shared	1	1	1	1	private room in rental unit	Private room	150.0
1	shared	2	6	3	5	entire rental unit	Entire home/apt	774.0
2	shared	1	2	1	2	entire rental unit	Entire home/apt	1136.0
3	shared	3	5	2	3	entire rental unit	Entire home/apt	500.0
4	shared	9	12	5	7	entire villa	Entire home/apt	10000.0

Após a transformação em dummies, temos 113 colunas com 24428 registros. Verificamos que o número de colunas novamente aumentou consideravelmente. Na imagem abaixo está um preview do dataset de dummies e as novas colunas (não cabe todas as colunas na imagem).

permanencia longa	piano	piscina	preco	primeiros-socorros	produtos de limpeza	property_type=apt/house	property_type=barn	property_type=bed and breakfast	property_type=boat	property_
1.0	0.0	0.0	150.0	0.0	0.0	0.0	0.0	0.0	0.0	
1.0	0.0	0.0	774.0	0.0	0.0	0.0	0.0	0.0	0.0	
1.0	0.0	0.0	1136.0	0.0	0.0	0.0	0.0	0.0	0.0	

```
['academia',
 'accommodates',
 'acesso lago',
 'agua quente',
 'alarme de fumaça',
 'aquecedor',
 'ar condicionado',
 'banheira',
 'bathrooms',
 'bathrooms_type=private',
 'bathrooms_type=shared',
 'bedrooms',
 'beds',
 'beira mar',
 'bicicletas',
 'bidê',
 'brinquedos',
 'cadeira',
 'cafe da manha',
 'cafeteira',
 'caixa de som',
 'cameras de segurança',
 'churrasqueira',
 'chuveiro externo',
 'cofre',
 'condicionador',
 'cortina',
 'cozinha',
 'elevador',
 'entrada privativa',
 'escorredor de roupa',
 'espaço de trabalho',
 'extintor de incêndio',
 'faxineira, zelador',
 'fechaduras',
 'ferro de passar',
 'fogao',
 'forno',
 'garagem',
 'geladeira',
 'grade na janela',
 'guarda-roupa',
 'internet por cabo',
 'itens de higiene',
 'itens de praia',
 'itens para bebes',
 'itens para crianças',
 'jardim',
 'jogos de tabuleiro',
 'lareira',
 'limpeza antes do checkout',
 'maquina de lavar',
 'mesa de jantar',
 'mesa de ping-pong',
 'microondas',
 'permanencia longa',
 'piano',
 'piscina',
 'preco',
 'primeiros-socorros',
 'produtos de limpeza',
 'property_type=apt/house',
 'property_type=barn',
 'property_type=bed and breakfast',
 'property_type=boat',
 'property_type=bungalow',
 'property_type=cabin',
 'property_type=campsite',
 'property_type=castle',
 'property_type=chalet',
 'property_type=condo',
 'property_type=cottage',
 'property_type=dome',
 'property_type=earthen house',
 'property_type=farm',
```

5. Criação de Modelos de Machine Learning

O algoritmo base de clusterização a ser utilizado é o K-means. Escolho ele por ser um modelo clássico, fácil de ser implementado, fácil de ser entendido e com poucos parâmetros de teste.

Em cima dele, irei utilizar a técnica do cotovelo para termos o melhor número de clusters para nosso problema. Utilizarei também métricas como Silhouette Index e Davies Bouldin para confirmar o melhor número de clusters. Além disso, irei utilizar normalização via MinMaxScaler, comparando o resultado sem ele e com ele no cluster. Por fim, também utilizarei PCA, com e sem MinMaxScaler para verificar se há melhora no desempenho e no resultado.

O K-means posteriormente será comparado com outro tipo de cluster (de densidade DBScan) e esses 2 clusters serão avaliados novamente por Silhouette Index e Davies Bouldin para saber qual é o melhor.

Abaixo temos o algoritmo de cluster do k-means, aonde no random_state eu coloquei 0 para que sempre que for rodada a célula do cluster, ter os mesmos resultados e não ficar completamente aleatório (é uma semente).

Já o max_iter é o número de iterações que o k-means irá fazer para descobrir clusters. Coloquei 3000 (um número mais alto) para tentar evitar o problema do k-means ficar rodando infinito (pois podem ter pontos equidistantes do centro de diferentes clusters, o que faz com que esses pontos a cada iteração mudem de cluster).

Por fim, uso fit_predict que é um padrão de método para clusterização (ao contrário de análises supervisionadas, que você pode fazer o fit separado do predict).

Esses parâmetros colocados no K-means (max_iter, random_state, init e fit_predict) serão utilizados para todos os testes de k-means (o que altera é o número de clusters).

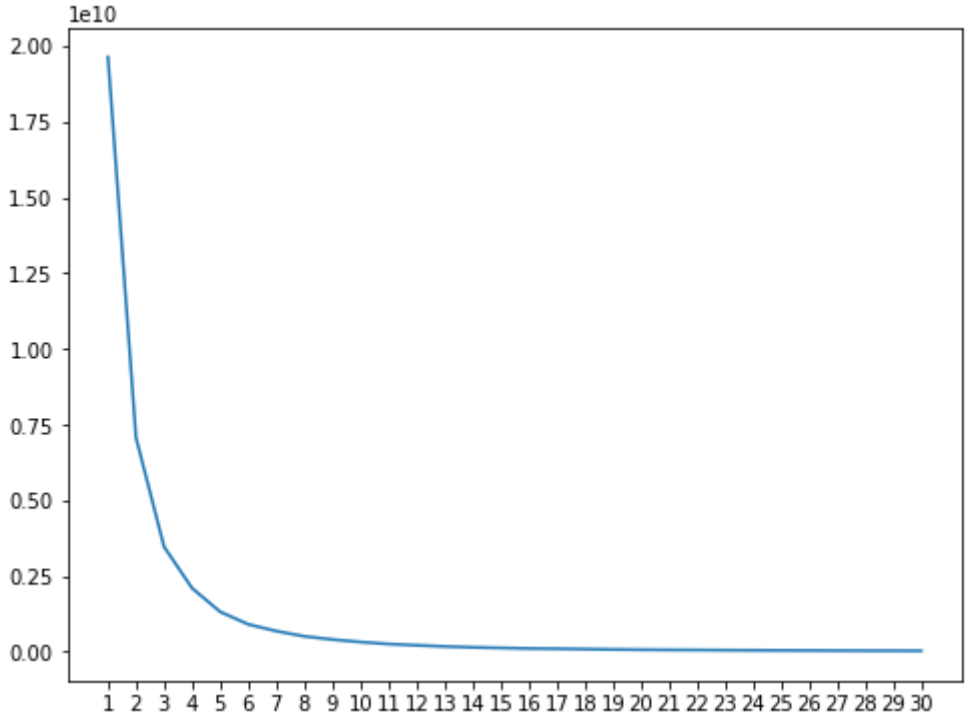
```
%%time
values = []
for k in range(1,31):
  kmeans = KMeans(init='k-means++', n_clusters=k, max_iter=3000, random_state=0)

  clusters = kmeans.fit_predict(dummies)

  values.append(kmeans.inertia_)
```

Para o método do cotovelo, escolhi testar clusters de tamanho 1 até 30. Usando o parâmetro inertia_, que nos dá um valor relativo à distância dos pontos entre si nos clusters, plotei o gráfico e vi que as distâncias dão uma queda acentuada entre os valores de cluster 1 a 5. Para verificar dentre 1 a 5 qual o melhor número de clusters, utilizei o Silhouette Index e Davies Bouldin, descobrindo que o melhor número de clusters é 2.

Abaixo a foto dos valores de inertia_.

```
0 cluster, com valor:19634022366.741364
1 cluster, com valor:7067842216.982395
2 cluster, com valor:3465228641.9873924
3 cluster, com valor:2096223715.7866888
4 cluster, com valor:1317775693.0887923
5 cluster, com valor:903989411.8843827
6 cluster, com valor:679902660.6149509
7 cluster, com valor:506778092.17768085
8 cluster, com valor:401519070.6812443
9 cluster, com valor:319061776.62374413
10 cluster, com valor:253431131.38823724
11 cluster, com valor:211111442.26518756
12 cluster, com valor:172255327.88085932
13 cluster, com valor:142107683.26993662
14 cluster, com valor:124138397.83608457
15 cluster, com valor:104946917.99002239
16 cluster, com valor:97549778.62629716
17 cluster, com valor:84368433.67005368
18 cluster, com valor:72651405.11070265
19 cluster, com valor:65209777.74921429
20 cluster, com valor:58670667.93669039
21 cluster, com valor:55287223.74136859
22 cluster, com valor:49821765.99364107
23 cluster, com valor:45649573.296357095
24 cluster, com valor:41297760.54496303
25 cluster, com valor:38307448.74629762
26 cluster, com valor:34820588.698729776
27 cluster, com valor:32693275.822340455
28 cluster, com valor:30756760.62974774
29 cluster, com valor:28135532.608548477
```

Plotando o gráfico correspondente ao _inertia:



A próxima análise consiste de padronizar os dados do dataset antes de passar para o cluster, utilizando métodos como MinMaxScaler, StandScaler, RobustScaler, etc, pois como temos valores com escalas muito distantes entre si (compare os valores da coluna **preço** com os valores da coluna **accomodates** por exemplo, em que preço vai até 10.000 e accomodates vai até 16) as colunas com maiores valores irão influenciar bem mais o modelo, mesmo que o peso real dessas colunas devesse ser o mesmo para o algoritmo.

De todos os normalizadores citados acima, utilizarei o MinMaxScaler, pois ele age em cima de cada coluna de forma independente, ficando todos os valores no range de 0 a 1 ou -1 a 1 caso haja dados negativos. Ele preserva a distribuição original.

Abaixo mostro alguns registros após o dataset ser passado pelo MinMaxScaler:

accommodates	acesso lago	agua quente	alarme de fumaça	aquecedor	ar condicionado	banheira	bathrooms	bathrooms_type=private	bathrooms_type=shared	bedrooms	beds	b
0.000000	0.0	0.0	0.0	0.0	0.333333	0.0	0.066667	0.0	1.0	0.000000	0.000000	
0.333333	0.0	0.0	0.0	0.0	0.333333	0.0	0.133333	0.0	1.0	0.222222	0.444444	
0.066667	0.0	0.0	0.0	0.0	0.333333	0.0	0.066667	0.0	1.0	0.000000	0.111111	

Utilizei Silhouette e Davis Bouldin novamente com o novo dataset normalizado:

Normalizado

```
clusters = 2  silhouette: 0.2053117932247313 davies bouldin: 2.2882718358904763
clusters = 3  silhouette: 0.1178862312415381 davies bouldin: 2.4307360657492905
clusters = 4  silhouette: 0.08541226183594232 davies bouldin: 2.5812147946064656
clusters = 5  silhouette: 0.08650972288457302 davies bouldin: 2.961107329433699
CPU times: user 1min 11s, sys: 7.01 s, total: 1min 18s
Wall time: 49.8 s
```

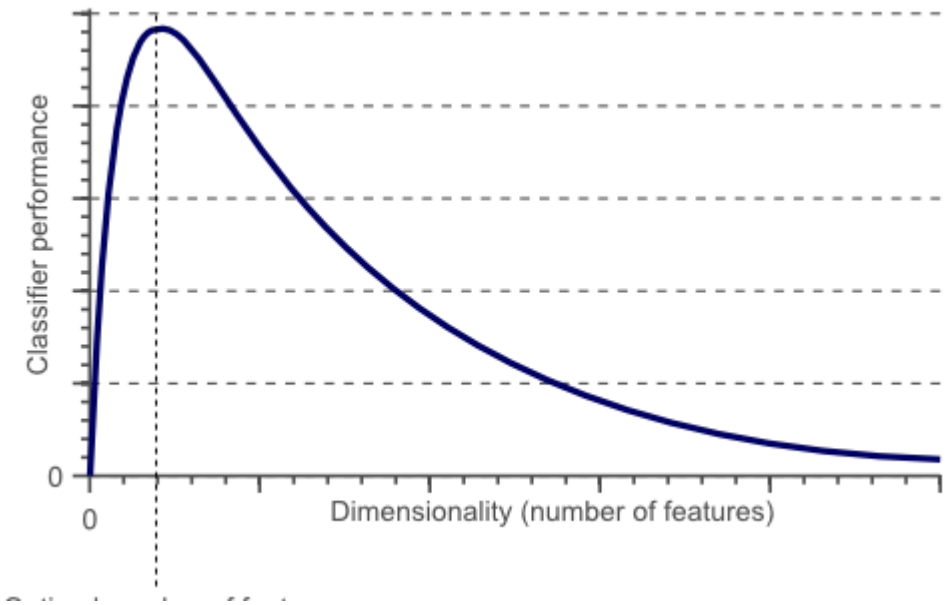
Sem normalização

```
clusters = 2  silhouette: 0.8739324364132001 davies bouldin: 0.4702484094232248
clusters = 3  silhouette: 0.7713963876254026 davies bouldin: 0.4818703010685885
clusters = 4  silhouette: 0.7204360705714008 davies bouldin: 0.5080671342354715
clusters = 5  silhouette: 0.6987737378195366 davies bouldin: 0.48454990318213076
CPU times: user 1min 10s, sys: 6.8 s, total: 1min 17s
Wall time: 50.7 s
```

Vemos que o valor do cluster sem normalização é melhor do que o normalizado justamente porque variáveis como “preço” que tendem a ter uma variação enorme de valores estão influenciando totalmente o modelo, em detrimento das outras variáveis.

Por fim, para o K-means, vamos passar o dataset pelo PCA. Embora pareça intuitivo pensar que quanto mais colunas temos, melhor para o modelo descobrir clusters, nem sempre esse pensamento é correto, uma vez que quanto mais features possuirmos, podemos estar sujeitos à Maldição da Dimensionalidade, o que acarreta em dados mais esparsos e uma piora no desempenho do cluster. Devido a isso, utilizarei a técnica de PCA (Análise do Componente Principal) para diminuir o número de colunas, mantendo a acurácia do modelo, como se estivesse de fato ainda usando todas as colunas.

Abaixo uma imagem que ilustra um dos problemas da Maldição da Dimensionalidade (performance cai a medida que o número de colunas aumenta):



Fonte: https://medium.com/@fabiolenine

Ao passar o PCA sem normalizar, vemos que somente uma variável explica 99% dos dados. Minha hipótese é que isso se dá provavelmente novamente pela influência enorme que a variável **preço** tem, ou seja, valores enormes, destoantes do resto, pois um dos passos para descobrir os componentes principais se passa por operações matemáticas envolvendo a média dos valores.

```
getVariabilidadePCA(dummies)

[ 0.99997251  0.00000834  0.00000483  0.00000119  0.000001    0.0000009
  0.00000075  0.00000071  0.00000058  0.00000054  0.00000045  0.00000041
  0.00000039  0.00000035  0.00000033  0.00000027  0.00000026  0.00000026
  0.00000024  0.00000023  0.00000023  0.00000021  0.00000021  0.00000021
  0.0000002   0.00000019  0.00000019  0.00000017  0.00000016  0.00000016
  0.00000015  0.00000014  0.00000014  0.00000014  0.00000013  0.00000013
  0.00000012  0.00000012  0.00000011  0.00000011  0.00000011  0.0000001
  0.0000001   0.0000001   0.00000009  0.00000009  0.00000009  0.00000009
  0.00000008  0.00000008  0.00000008  0.00000007  0.00000007  0.00000007
  0.00000007  0.00000007  0.00000006  0.00000006  0.00000005  0.00000005
  0.00000005  0.00000005  0.00000005  0.00000004  0.00000004  0.00000004
  0.00000004  0.00000003  0.00000003  0.00000003  0.00000002  0.00000002
  0.00000002  0.00000002  0.00000002  0.00000002  0.00000001  0.00000001
  0.00000001  0.00000001  0.00000001  0.00000001  0.00000001  0.00000001
  0.00000001  0.00000001  0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.          0.          0.          0.
  0.          0.          0.         -0.         -0.        ]
```

Ao passar o PCA novamente pelo dataset, porém dessa vez normalizado, vemos que várias outras variáveis explicam os dados.

```
[ 0.19415895  0.07886073  0.05426392  0.05301963  0.03615628  0.03327555
  0.02959879  0.02697663  0.02472422  0.0235792   0.01976695  0.01878154
  0.01756883  0.01732993  0.01623652  0.01484752  0.01452925  0.01441464
  0.01387332  0.01307805  0.01294003  0.01198778  0.01192477  0.01131704
  0.01098034  0.01065539  0.01039356  0.0098438   0.0093901   0.00921743
  0.00855814  0.00820552  0.00803688  0.00742615  0.00712546  0.00687342
  0.00650114  0.0063223   0.00602952  0.00590446  0.00548268  0.00526822
  0.0051648   0.00473632  0.00465757  0.0044874   0.00416741  0.00406269
  0.00399605  0.00364214  0.00336713  0.0033008   0.00316716  0.00310035
  0.00281224  0.00278532  0.00251958  0.00235668  0.00229889  0.00219301
  0.00212488  0.00208583  0.00181154  0.00176949  0.00164698  0.00160789
  0.00143998  0.00140083  0.00135235  0.00129983  0.00117372  0.00106284
  0.00104069  0.00100719  0.00099672  0.00088769  0.00079727  0.00073611
  0.00066893  0.0006405   0.00059652  0.00058836  0.00058485  0.00047538
  0.00046461  0.00045281  0.00042856  0.00039611  0.00038636  0.00032705
  0.00027847  0.00023705  0.00017215  0.00015363  0.00013561  0.00011832
  0.0001043   0.00008989  0.00006342  0.00002919  0.00002241  0.00002022
  0.00001542  0.00001348  0.00001346  0.00001333  0.00000754  0.00000674
  0.00000671  0.00000668  0.         -0.         -0.        ]
```

São necessárias pelo menos 60 variáveis para explicar mais de 95% dos dados:

```
variabilidade = getVariabilidadePCA(normalizado)

variabilidade[0:60].sum()
```

```
0.9682401295336588
```

De 113 variáveis caiu para 60. Passando essa última configuração no cluster, vemos o seguinte resultado:

```
clusters = 2  silhouette: 0.20856673098074882 davies bouldin: 2.2453053284632087
clusters = 3  silhouette: 0.1213470201551845 davies bouldin: 2.38180159234371
clusters = 4  silhouette: 0.09026627903344582 davies bouldin: 2.5321187532661305
clusters = 5  silhouette: 0.09146625105492456 davies bouldin: 2.8974106546310807
CPU times: user 56.2 s, sys: 8.06 s, total: 1min 4s
Wall time: 43 s
```

Ou seja, não melhorou em relação ao primeiro caso, em que não tínhamos normalizado o cluster, mas ficou um pouco melhor do que simplesmente normalizar. Na figura abaixo vemos também que o tempo total de processamento do PCA com processamento foi ligeiramente melhor.

Sem nada

```
clusters = 2  silhouette: 0.8739324364132001 davies bouldin: 0.4702484094232248
clusters = 3  silhouette: 0.7713963876254026 davies bouldin: 0.4818703010685885
clusters = 4  silhouette: 0.7204360705714008 davies bouldin: 0.5080671342354715
clusters = 5  silhouette: 0.6987737378195366 davies bouldin: 0.48454990318213076
CPU times: user 1min 13s, sys: 6.98 s, total: 1min 20s
Wall time: 55.7 s
```

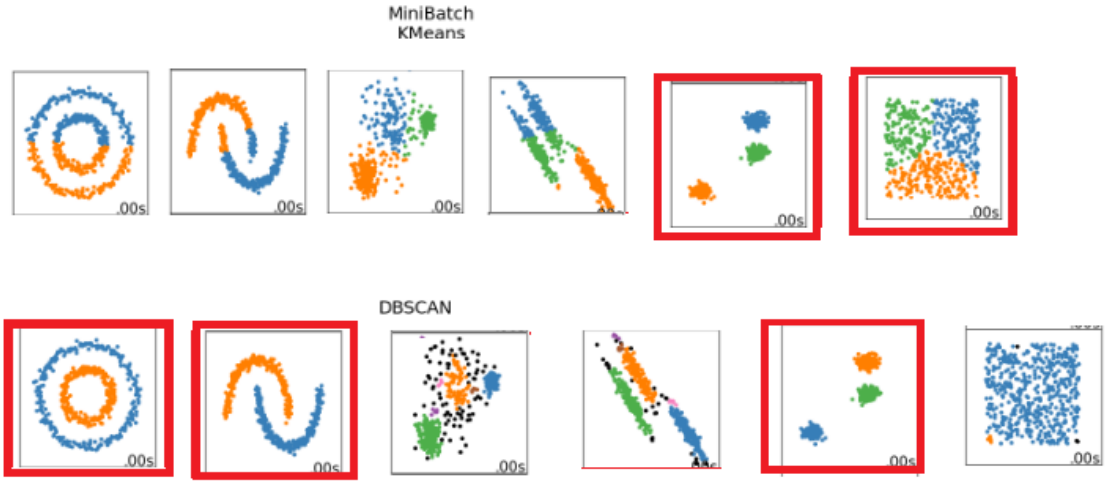
normalizado

```
clusters = 2  silhouette: 0.2053117932247313 davies bouldin: 2.2882718358904763
clusters = 3  silhouette: 0.1178862312415381 davies bouldin: 2.4307360657492905
clusters = 4  silhouette: 0.08541226183594232 davies bouldin: 2.5812147946064656
clusters = 5  silhouette: 0.08650972288457302 davies bouldin: 2.961107329433699
CPU times: user 1min 14s, sys: 5.9 s, total: 1min 20s
Wall time: 54.8 s
```

PCA normalizado

```
clusters = 2  silhouette: 0.20856673098074882 davies bouldin: 2.2453053284632087
clusters = 3  silhouette: 0.1213470201551845 davies bouldin: 2.38180159234371
clusters = 4  silhouette: 0.09026627903344582 davies bouldin: 2.5321187532661305
clusters = 5  silhouette: 0.09146625105492456 davies bouldin: 2.8974106546310807
CPU times: user 56.2 s, sys: 8.06 s, total: 1min 4s
Wall time: 43 s
```

Comparando o DBScan com o K-means, vemos que cada um desses tipos de clusterização reconhece tipos de figuras diferentes (abaixo em vermelho o que cada algoritmo é especialista).



O DBScan também tem a capacidade de detectar registros que ele considera ruídos, que são mostrados na imagem 3 acima como pontos pretos.

O primeiro teste com DBScan foi com os parâmetros default do modelo em cima do dataframe não normalizado e sem PCA. Vemos que embora o tempo de processamento do DBScan seja bem melhor que qualquer tempo do K-means, o resultado dele é muito pior.

```
] %%time

  dbscan = DBSCAN()
  labels = dbscan.fit_predict(dummies)

  CPU times: user 12.4 s, sys: 1.27 s, total: 13.7 s
  Wall time: 12.7 s
```

```
[205] ruido = len(visualizacaoDBScan[visualizacaoDBScan['clusters']==-1])

      amostras = len(visualizacaoDBScan[visualizacaoDBScan['clusters'] != -1])

      print('Ruido:', ruido, ' Amostras:', amostras)

      Ruido: 24375  Amostras: 53

[206] print("silhouette:", silhouette_score(dummies, labels), 'davies bouldin:', davies_bouldin_score(dummies, labels))

      silhouette: -0.5993510867442907 davies bouldin: 1.125316375342217

[210] len(set(labels)) - 1 # -1 para remover ruidos

      8
```

Ele dividiu os dados em 24375 ruídos (ou seja, quase a base inteira ele detectou como ruído), considerou apenas 53 amostras válidas, divididas em 8 clusters diferentes, com Silhouette de -0.6 e Davies Bouldin de 1.12.

Ao usar o dataframe normalizado, vemos que as métricas melhoraram um pouco, mas continuam péssimas (Silhouette de -0.29 e Davies Bouldin de 1.48). O algoritmo dividiu o dataframe em 249 clusters diferentes, em um tempo total de 12.6 segundos. Ele considerou como ruído 20441 e como amostras válidas 3987.

Ao utilizar o dataframe passado pelo PCA, melhora mais um pouco em relação ao dataframe normalizado, mas continua ruim no geral, tanto no que considera ruído como nas métricas de Silhouette e Davies Bouldin.

Como uma tentativa de melhorar o algoritmo, utilizei o RandomizedSearchCV para modificar os parâmetros do cluster. Usei RandomizedSearchCV pois ele pega randomicamente os parâmetros a partir de uma lista de parâmetros que eu forneci, tentando achar a melhor combinação de parâmetros também se baseando em quantas iterações eu quero. Caso eu usasse o GridSearch, ele iria procurar entre todos os parâmetros, porém o tempo de processamento seria muito grande.

Na imagem abaixo, inseri apenas os valores de EPS e MIN_SAMPLES (embora o DBS-can tenha outros), pois além de serem os parâmetros mais importantes, modificar um número muito grande de parâmetros geraria um processamento muito pesado e demorado. O **EPS** é a distância máxima entre pontos para serem considerados vizinhos, e o **min_samples** é o número mínimo de pontos na vizinhança para um ponto contido nela ser considerado central. O CV significa se haverá cross-validation no dataset, o que nesse caso não precisa. O scoring é exatamente o mesmo que é implementado no Silhouette Index, e o n_iter é o número de testes com parâmetros que será realizado.

```
def customSilhouette(model, X):
  #print(model)

  preds = model.fit_predict(X)
  return silhouette_score(X, preds) if len(set(preds)) > 1 else float('nan')

dbscan = DBSCAN()

parametros = {
      'eps':  np.arange(0.5, 5, 0.5),
      'min_samples': np.arange(1, 10, 1),
}

cv = [(slice(None), slice(None))] #sem cross-validation
```

```
%%time

n_iter_search = 5
random_search = RandomizedSearchCV(dbscan, param_distributions=parametros,
                                   n_iter=n_iter_search, cv=cv, random_state=0, scoring=customSilhouette)

random_search.fit(novosNormalizados)
```

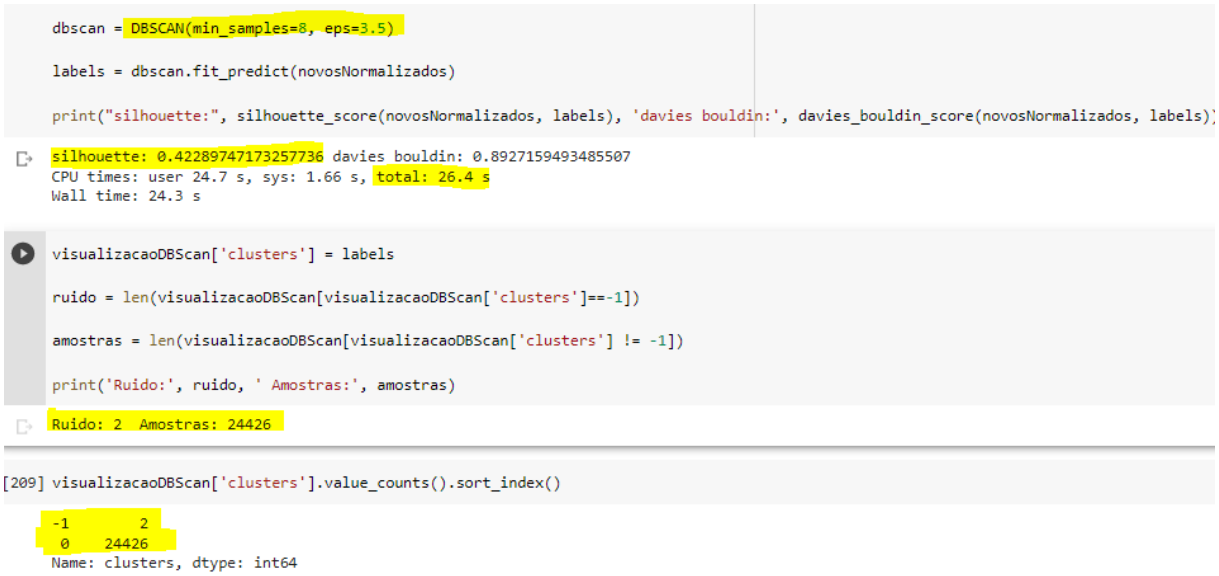
No DBScan acima usei o dataset normalizado e passado pelo PCA (ou seja, o dataset de 60 variáveis).

```
CPU times: user 2min 54s, sys: 18 s, total: 3min 12s
Wall time: 2min 28s
RandomizedSearchCV(cv=[(slice(None, None, None), slice(None, None, None))],
                   estimator=DBSCAN(), n_iter=5,
                   param_distributions={'eps': array([0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5]),
                                        'min_samples': array([1, 2, 3, 4, 5, 6, 7, 8, 9])},
                   random_state=0,
                   scoring=<function customSilhouette at 0x7fcd7945fcb0>)
```

O EPS fiz de 0.5 até 4.5, sempre somando 0.5 unidades ao anterior. O min_samples vai de 1 a 9. Esses valores vieram baseados nos valores default (o valor default seria um "valor central" de cada um).

Vemos que o tempo de processamento é de 3 minutos, os melhores parâmetros encontrados são de EPS = 3.5 e min_samples = 8, com ruído de apenas 2 e silhouette index de 0.4. Parece ser o melhor valor de silhouette se não víssemos que não houveram divisões

no cluster. Essa “melhor” parametrização indicou apenas a classe de ruídos e uma classe de cluster.



```
dbscan = DBSCAN(min_samples=8, eps=3.5)

labels = dbscan.fit_predict(novosNormalizados)

print("silhouette:", silhouette_score(novosNormalizados, labels), 'davies bouldin:', davies_bouldin_score(novosNormalizados, labels))
```

```
silhouette: 0.42289747173257736 davies bouldin: 0.8927159493485507
CPU times: user 24.7 s, sys: 1.66 s, total: 26.4 s
Wall time: 24.3 s
```

```
visualizacaoDBScan['clusters'] = labels

ruido = len(visualizacaoDBScan[visualizacaoDBScan['clusters']==-1])

amostras = len(visualizacaoDBScan[visualizacaoDBScan['clusters'] != -1])

print('Ruido:', ruido, ' Amostras:', amostras)
```

```
Ruido: 2  Amostras: 24426
```

```
[209] visualizacaoDBScan['clusters'].value_counts().sort_index()
```

```
-1        2
 0    24426
Name: clusters, dtype: int64
```

A análise do dataset com a clusterização me mostrou que provavelmente o dataset não é adequado para DBScan. Embora o K-means também não tenha dado ótimos resultado, ele é melhor que o DBScan, por tratar todos os dados como válidos, por conseguir dividir o cluster em pelo menos 2 e por ter o melhor silhouette encontrado dada a situação analisada.

Acredito que a imagem geral do dataset seja mais adequada para tratamento por K-means.

6. Interpretação dos Resultados

Pela análise dos dados

Vemos que mais mulheres ofertam hospedagens que homens e que há presença de muitos anúncios de hospedagens em inglês. Muitas hospedagens são próximas ou estão em Cobacabana e Ipanema e os anfitriões estão preocupados em ter como principais facilidades TV, wi-fi e ar-condicionado. As hospedagens são próximas à praia, com acesso via metro, bares e restaurantes em sua volta. Grande maioria das hospedagens tem apenas 1 banheiro, e são poucos o número de banheiros privados.

As hospedagens permitem no máximo 16 pessoas e há variados tipos de hospedagens (desde barcos a castelos), sendo em grande maioria hospedagens do tipo rental unit (locais de locação). A grande maioria são de locais que você aluga a hospedagem inteira.

Os anfitriões em sua maioria tem personalidade expansiva (gostam de viajar, curtir a vida).

Pelas colunas camas e quartos, vemos que há hoteis, hostels, pousadas e grandes estruturas cadastradas no Airbnb. Também vemos que não é incomum ter várias camas por quarto. O preço médio das hospedagens fica entre R$ 33 até R$500 por noite.

A medida que aumenta o número de quartos, o número de banheiros aumenta proporcionalmente. O número de quartos independe do número anunciado de pessoas (2 pessoas podem alugar uma propriedade de 10 quartos por exemplo, ou 16 pessoas podem alugar uma propriedade de 1 quarto).

O Airbnb teve seu pico de entrada de anfitriões em 2014 e 2016.

Pelo K-means e DBScan

Entre DBScan e K-means, o melhor é o K-means, com número de clusters igual a 2, com dataset normalizado e passado pelo PCA.

Mesmo tentando encontrar os melhores valores de parâmetros do DBScan, sempre havia uma falha, ou seja, o silhouette index não estava com boa pontuação, ou tinha muito ruído ou o cluster não era dividido como esperado.

Pela interpretação dos clusters

Rodei o K-means com N=2 sem PCA mas com normalização para criar os clusters. Peguei o dataframe original sem normalização para visualização e interpretação, juntando ele com os clusters gerados anteriormente. A interpretação foi:

- Vemos que a divisão dos clusters se deu sendo 16823 registros na classe 0 e 7605 registros na classe 1, não sendo bem divididos os clusters, uma vez que seus valores de Silhouette e Davis Boulding não são bons.

- Análise por quartil (análise no final, abaixo de todas as figuras).

CLUSTER 0

	academia	accommodates	acesso lago	agua quente	alarme de fumaça	aquecedor	ar condicionado	banheira	bathrooms
count	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
mean	0.144980	4.031267	0.004637	0.465196	0.174285	0.070796	0.800571	0.079712	1.670213
std	0.352091	2.336123	0.067936	0.503429	0.504939	0.256491	0.401068	0.279709	1.019438
min	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000
50%	0.000000	4.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000
75%	0.000000	5.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	2.000000
max	1.000000	16.000000	1.000000	2.000000	2.000000	1.000000	2.000000	2.000000	11.000000

CLUSTER 1

	academia	accommodates	acesso lago	agua quente	alarme de fumaça	aquecedor	ar condicionado	banheira	bathrooms
count	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
mean	0.206969	4.129915	0.060224	1.199474	0.416437	0.078238	0.885339	0.162919	1.656805
std	0.405160	2.163116	0.237916	0.468132	0.756364	0.268563	0.338254	0.433528	0.987445
min	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	1.000000
50%	0.000000	4.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000	1.000000
75%	0.000000	5.000000	0.000000	1.000000	1.000000	0.000000	1.000000	0.000000	2.000000
max	1.000000	16.000000	1.000000	2.000000	2.000000	1.000000	3.000000	2.000000	15.000000

CLUSTER 0

bathrooms_type=private	bathrooms_type=shared	bedrooms	beds	beira mar	bicicletas	bidê	brinquedos	cadeira
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.067586	0.932414	1.627118	2.436248	0.088985	0.000476	0.001783	0.013850	0.010878
0.251042	0.251042	0.919023	1.622452	0.314114	0.021802	0.042192	0.116872	0.103732
0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	2.000000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	1.000000	10.000000	10.000000	2.000000	1.000000	1.000000	1.000000	1.000000

CLUSTER 1

bathrooms_type=private	bathrooms_type=shared	bedrooms	beds	beira mar	bicicletas	bidê	brinquedos	cadeira
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
0.043130	0.956870	1.621696	2.554241	0.358054	0.027876	0.057462	0.072452	0.036949
0.203162	0.203162	0.897760	1.608143	0.647703	0.164629	0.232739	0.259252	0.188650
0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	1.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	2.000000	3.000000	1.000000	0.000000	0.000000	0.000000	0.000000
1.000000	1.000000	8.000000	10.000000	2.000000	1.000000	1.000000	1.000000	1.000000

CLUSTER 0

cafe da manha	cafeteira	caixa de som	cameras de segurança	churrasqueira	chuveiro externo	cofre	condicionador	cortina	cozinha	elevador
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.055698	0.284670	0.000476	0.080544	0.071688	0.031504	0.026274	0.006004	0.057362	0.889437	0.547524
0.229344	0.471625	0.021802	0.272142	0.257978	0.174682	0.161432	0.077253	0.232540	0.313599	0.497751
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000
1.000000	3.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	1.000000	1.000000	1.000000

CLUSTER 1

cafe da manha	cafeteira	caixa de som	cameras de segurança	churrasqueira	chuveiro externo	cofre	condicionador	cortina	cozinha	elevador
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
0.030901	1.051414	0.012229	0.172124	0.167258	0.114398	0.184221	0.168047	0.715187	0.954241	0.716897
0.173060	0.726997	0.109913	0.377513	0.373231	0.318315	0.423673	0.373933	0.451355	0.209605	0.450536
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000
1.000000	4.000000	1.000000	1.000000	1.000000	1.000000	3.000000	1.000000	1.000000	2.000000	1.000000

CLUSTER 0

entrada privativa	escorredor de roupa	espaço de trabalho	extintor de incêndio	faxineira, zelador	fechaduras	ferro de passar	fogao	forno	garagem	geladeira
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.142067	0.021994	0.144267	0.233787	0.075016	0.168817	0.551982	0.281460	0.221482	0.614338	0.385900
0.349130	0.146667	0.351370	0.423251	0.263426	0.383074	0.497305	0.450121	0.415400	0.664674	0.507741
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000
1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	2.000000	2.000000	5.000000	4.000000

CLUSTER 1

entrada privativa	escorredor de roupa	espaço de trabalho	extintor de incêndio	faxineira, zelador	fechaduras	ferro de passar	fogao	forno	garagem	geladeira
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
0.312163	0.678764	0.532676	0.352926	0.160947	0.244313	0.849704	0.896252	0.725707	1.388955	1.244181
0.463406	0.466982	0.498964	0.477911	0.367506	0.442969	0.357385	0.345015	0.484345	0.968838	0.677429
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000
0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000	1.000000	2.000000	2.000000
1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	3.000000	3.000000	7.000000	5.000000

CLUSTER 0

grade na janela	guarda-roupa	internet por cabo	itens de higiene	itens de praia	itens para bebes	itens para crianças	jardim	jogos de tabuleiro	lareira	limpeza antes do checkout
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.015931	0.633656	0.031921	0.007965	0.009808	0.052369	0.020270	0.066219	0.000773	0.009511	0.029900
0.125211	0.628610	0.175794	0.088895	0.098551	0.275076	0.165379	0.248672	0.027788	0.101843	0.170315
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	6.000000	1.000000	1.000000	1.000000	5.000000	2.000000	1.000000	1.000000	2.000000	1.000000

CLUSTER 1

grade na janela	guarda-roupa	internet por cabo	itens de higiene	itens de praia	itens para bebes	itens para crianças	jardim	jogos de tabuleiro	lareira	limpeza antes do checkout
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.00000	7605.000000	7605.000000
0.141091	2.308218	0.316502	0.098751	0.105588	0.229980	0.148192	0.214070	0.05194	0.012755	0.174753
0.348139	1.081474	0.465142	0.348371	0.307330	0.675992	0.441470	0.410202	0.22192	0.119046	0.379781
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
0.000000	3.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000
1.000000	6.000000	1.000000	4.000000	1.000000	7.000000	2.000000	1.000000	1.00000	2.000000	1.000000

CLUSTER 0

maquina de lavar	mesa de jantar	mesa de ping-pong	microondas	permanencia longa	piano	piscina	preco	primeiros-socorros	produtos de limpeza
16823.00000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.66522	0.015098	0.001011	0.290852	0.869881	0.000654	0.263449	635.801522	0.120549	0.022945
0.52022	0.121948	0.031774	0.454169	0.336445	0.025563	0.454988	982.816312	0.325612	0.149732
0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	33.000000	0.000000	0.000000
0.00000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	180.000000	0.000000	0.000000
1.00000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	320.000000	0.000000	0.000000
1.00000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	680.000000	0.000000	0.000000
3.00000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	10000.000000	1.000000	1.000000

CLUSTER 1

maquina de lavar	mesa de jantar	mesa de ping-pong	microondas	permanencia longa	piano	piscina	preco	primeiros-socorros	produtos de limpeza
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
1.287311	0.646943	0.035503	0.867587	0.890072	0.011308	0.306772	423.706903	0.231295	0.569494
0.690092	0.477952	0.185059	0.338961	0.312820	0.105745	0.525682	643.525397	0.421688	0.495180
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	34.000000	0.000000	0.000000
1.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	167.000000	0.000000	0.000000
1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	0.000000	259.000000	0.000000	1.000000
2.000000	1.000000	0.000000	1.000000	1.000000	0.000000	1.000000	450.000000	0.000000	1.000000
3.000000	1.000000	1.000000	1.000000	1.000000	1.000000	4.000000	10000.000000	1.000000	1.000000

CLUSTER 0

room_type=Entire home/apt	room_type=Hotel room	room_type=Private room	room_type=Shared room	sabonete	sacada	sauna	secador	shampoo	sistema de som
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000
0.731201	0.002556	0.243892	0.022350	0.029900	0.125721	0.005231	0.489627	0.202104	0.002378
0.443348	0.050494	0.429441	0.147825	0.170315	0.331544	0.072138	0.636538	0.401730	0.048705
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	2.000000	1.000000

CLUSTER 1

room_type=Entire home/apt	room_type=Hotel room	room_type=Private room	room_type=Shared room	sabonete	sacada	sauna	secador	shampoo	sistema de som
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.00000	7605.000000
0.850888	0.001841	0.141749	0.005523	0.514267	0.403550	0.133465	0.847863	0.32426	0.094675
0.356223	0.042869	0.348815	0.074114	0.499829	0.490642	0.340099	0.691401	0.47065	0.292784
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000
1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.00000	0.000000
1.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.00000	0.000000
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000	2.00000	1.000000

CLUSTER 0

teclado	torradeira	travesseiros e cobertores	tv	utensilio churrasco	utensilios cozinha	ventilador	video game	vinho	wi-fi	c
16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	16823.000000	
0.006895	0.009749	0.403733	1.177792	0.004934	0.421150	0.050169	0.002199	0.006004	0.909469	
0.082754	0.098858	0.686757	0.696955	0.070069	0.525147	0.236596	0.046847	0.077253	0.332084	
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	
0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	
0.000000	0.000000	1.000000	2.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	
1.000000	2.000000	2.000000	7.000000	1.000000	3.000000	2.000000	1.000000	1.000000	2.000000	

CLUSTER 1

teclado	torradeira	travesseiros e cobertores	tv	utensilio churrasco	utensilios cozinha	ventilador	video game	vinho	wi-fi
7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000	7605.000000
0.032479	0.370677	1.650099	1.684024	0.089678	1.410651	0.710322	0.019987	0.493886	1.055095
0.177279	0.522012	0.550203	1.077709	0.285738	0.614627	0.657858	0.150818	0.499995	0.327568
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000
0.000000	0.000000	2.000000	2.000000	0.000000	1.000000	1.000000	0.000000	0.000000	1.000000
0.000000	1.000000	2.000000	2.000000	0.000000	2.000000	1.000000	0.000000	1.000000	1.000000
1.000000	2.000000	2.000000	8.000000	1.000000	4.000000	2.000000	2.000000	1.000000	2.000000

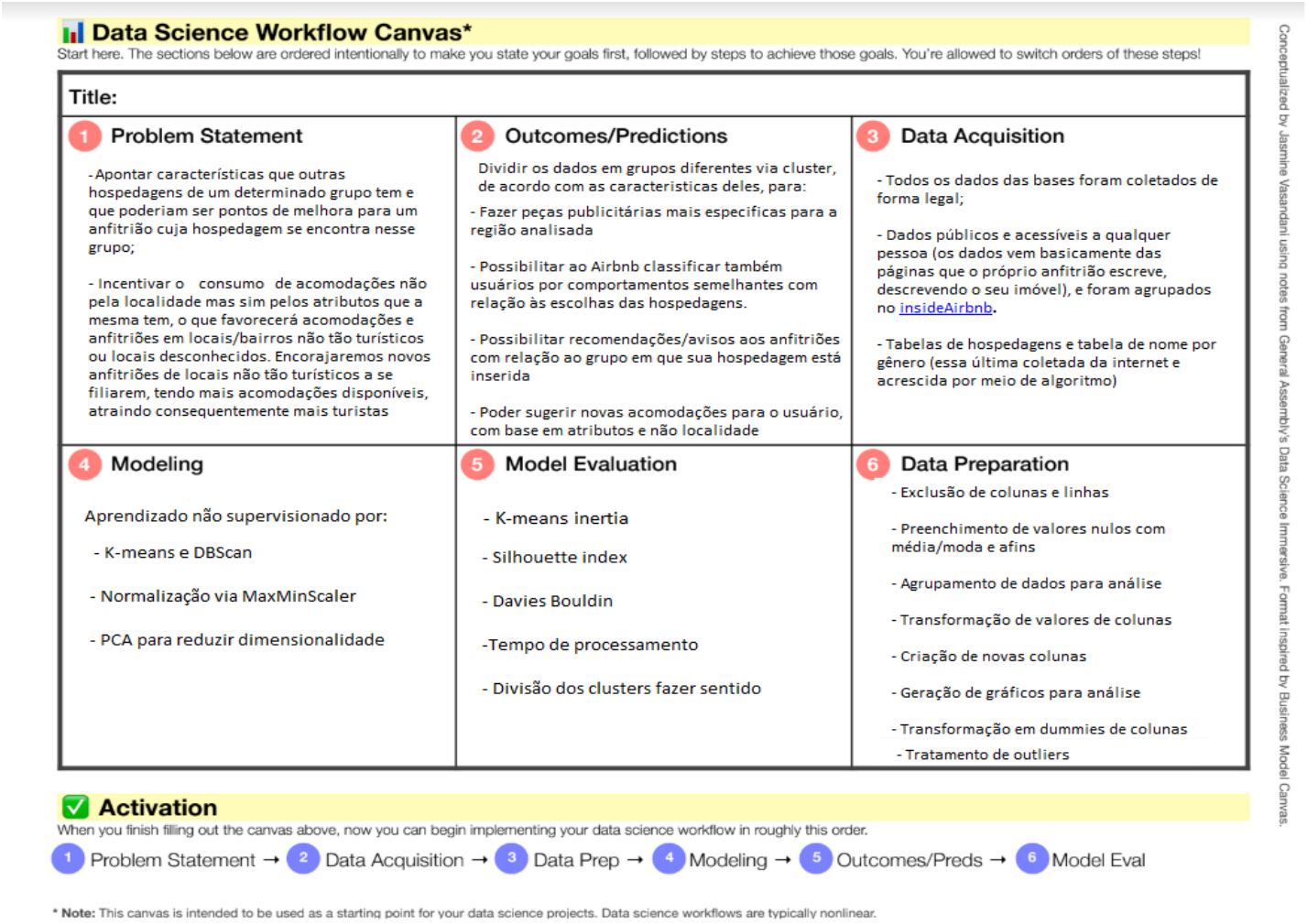
Fiz a análise marcando primeiramente as diferenças por quartil. Aqueles que eram completamente diferentes em pelo menos 2 quartis, assumi como característica marcante do cluster. Levando isso em consideração, o cluster 0 tem de diferente do 1:

- **Cluster 0**: sem água quente, sem cafeteira, sem cortina, sem escorredor de roupa, sem espaço de trabalho, sem fogão, sem forno, com garagem para no máximo 1 carro, 1 geladeira no máximo, um guarda roupa, 1 máquina de lavar, sem mesa de jantar, sem microondas, com preço médio de até 630, sem produto de limpeza, sem sabonete, com um conjunto de travesseiro e cobertor, com 1 item de cozinha, sem ventilador (os demais itens que não estão listados, o K-means não fez diferenciação entre o cluster 0 e 1, ou seja, tem o mesmo número).

- **Cluster 1**: com água quente, com cafeteira, com cortina, com escorredor de roupa, com espaço de trabalho, com fogão, com forno, obrigatório garagem para pelo menos 1 carro, no mínimo 1 geladeira (obrigatoriamente), mínimo de um guarda roupa (obrigatoriamente, mas geralmente mais de 1), mínimo de uma máquina de lavar (obrigatoriamente), com preço médio de 420, obrigatório no mínimo 1 conjunto de travesseiro e cobertor.

7. Apresentação dos Resultados

Utilizando o modelo proposto por Vasandani (explicitado aqui), resumo todo o processo da seguinte forma:



Data Science Workflow Canvas*

Start here. The sections below are ordered intentionally to make you state your goals first, followed by steps to achieve those goals. You're allowed to switch orders of these steps!

Title:

1 Problem Statement

- Apontar características que outras hospedagens de um determinado grupo tem e que poderiam ser pontos de melhora para um anfitrião cuja hospedagem se encontra nesse grupo;

- Incentivar o consumo de acomodações não pela localidade mas sim pelos atributos que a mesma tem, o que favorecerá acomodações e anfitriões em locais/bairros não tão turísticos ou locais desconhecidos. Encorajaremos novos anfitriões de locais não tão turísticos a se filiarem, tendo mais acomodações disponíveis, atraindo consequentemente mais turistas

2 Outcomes/Predictions

Dividir os dados em grupos diferentes via cluster, de acordo com as caracteristicas deles, para:

- Fazer peças publicitárias mais especificas para a região analisada

- Possibilitar ao Airbnb classificar também usuários por comportamentos semelhantes com relação às escolhas das hospedagens.

- Possibilitar recomendações/avisos aos anfitriões com relação ao grupo em que sua hospedagem está inserida

- Poder sugerir novas acomodações para o usuário, com base em atributos e não localidade

3 Data Acquisition

- Todos os dados das bases foram coletados de forma legal;

- Dados públicos e acessíveis a qualquer pessoa (os dados vem basicamente das páginas que o próprio anfitrião escreve, descrevendo o seu imóvel), e foram agrupados no insideAirbnb.

- Tabelas de hospedagens e tabela de nome por gênero (essa última coletada da internet e acrescida por meio de algoritmo)

4 Modeling

Aprendizado não supervisionado por:

- K-means e DBScan

- Normalização via MaxMinScaler

- PCA para reduzir dimensionalidade

5 Model Evaluation

- K-means inertia

- Silhouette index

- Davies Bouldin

-Tempo de processamento

- Divisão dos clusters fazer sentido

6 Data Preparation

- Exclusão de colunas e linhas

- Preenchimento de valores nulos com média/moda e afins

- Agrupamento de dados para análise

- Transformação de valores de colunas

- Criação de novas colunas

- Geração de gráficos para análise

- Transformação em dummies de colunas

- Tratamento de outliers

Activation

When you finish filling out the canvas above, now you can begin implementing your data science workflow in roughly this order.

1 Problem Statement → 2 Data Acquisition → 3 Data Prep → 4 Modeling → 5 Outcomes/Preds → 6 Model Eval

* **Note:** This canvas is intended to be used as a starting point for your data science projects. Data science workflows are typically nonlinear.

Conceptualized by Jasmine Vasandani using notes from General Assembly's Data Science Immersive. Format inspired by Business Model Canvas.

8. Links

Todo o código foi gerado no Colab:

https://colab.research.google.com/drive/1sPwMyLQlqjaPDIBcsLLoA4I--0Lc9YkV?usp=sharing

Pasta dos datasets:

https://drive.google.com/drive/folders/1KQNUk4BAoVCwFuTxn1JhWKEEzsVam-Uo?usp=sharing

Apresentação do trabalho:

https://docs.google.com/presentation/d/1DePDkwdFd3ME0ZKdT-3O3bJACq_66Bsb8sNAkNxLRU0/edit?usp=sharing

REFERÊNCIAS

INSTITUTO DOS REGISTOS E NOTARIADO. **Nomes 2013 M (até 20 dez.)**. Portugal: Diário de Noticias, 2013. Fonte: www.dn.pt/DNMultimedia/DOCS+PDFS/2013/Nomes%202013%20M%20(at%C3%A9%2020%20dez.).pdf

INSTITUTO DOS REGISTOS E NOTARIADO. **Nomes 2013 F (até 20 dez.)**. Portugal: Diário de Noticias, 2013. Fonte: www.dn.pt/DNMultimedia/DOCS+PDFS/2013/Nomes%202013%20F%20(at%C3%A9%2020%20dez.).pdf

INSIDE AIRBNB. **Inside Airbnb: Get the Data**. 2022. Fonte: **insideairbnb.com/get-the-data**

IVANOVSKI, Antonio. **Maximize Value of AirBnb Rental**. Estados Unidos: 2017. Fonte: www.kaggle.com/code/ivanovskia1/maximize-value-of-airbnb-rental

AIRBNB. **Avaliações por estrelas**. Brasil. Fonte: https://www.airbnb.com.br/help/article/1257/avalia%C3%A7%C3%B5es-por-estrelas

AIRBNB. **Como os bairros são determinados**. Brasil. Fonte: https://www.airbnb.com.br/help/article/422/como-os-bairros-s%C3%A3o-determinados

HOSTAWAY. **Airbnb Minimum Nights: Everything you need to know**. Fonte: https://www.hostaway.com/airbnb-minimum-nights/

YOKOYAMA, Naoki. **ML Pre_processamento**. Brasil: 2020. Fonte: https://naokiyokoyama.medium.com/ml-pre-processamento-cc348e778d06

SUBRAMANIAN, Niranjan. **Introduction to Principal Component Analysis (PCA)**. Fonte: https://aiaspirant.com/introduction-to-principal-component-analysispca/