Análise Exploratória Multimetodológica de Dados do Airbnb da cidade do Rio de Janeiro

André L. M. Dutra[1], **Gabriel L. C. Lira**[2], **Renato S. Santos**[3]

[1]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte – MG – Brazil

cienciandre@dcc.ufmg.br, renato.silva@dcc.ufmg.br, gabriel.lopes@ufmg.br

1. Introdução

O mercado de aluguéis de imóveis tem experimentado um crescimento contínuo, impulsionado pela valorização de áreas urbanas e pela popularização de plataformas como o Airbnb. No Rio de Janeiro, essa tendência é evidente na ampla oferta e demanda por imóveis para aluguel, que variam de apartamentos modestos a luxuosas residências. O Airbnb se destaca ao fornecer uma vasta gama de opções de hospedagem, atendendo a diversos perfis de visitantes.

Com isso, neste trabalho nos propomos a realizar uma análise exploratória multimetodólogica de dados de anúncios de imóveis no Airbnb na cidade do Rio de Janeiro entre as datas de 26/12/2023 e 30/12/2023. Dentre os dados disponíveis, temos informações sobre o imóvel (número de quartos, banheiros, localização, preço, etc.), informações sobre o host (nome, localização, taxa de aceitação, se é superhost, etc.), e informações de reviews de usuários sobre o imóvel (nota geraln e notas específicas sobre limpeza, localização, etc.).

A partir destes dados, buscamos explorar padrões em seus diferentes âmbitos usando diferentes abordagens: com o objetivo de encontrar relações entre as diferentes notas de review, aplicamos um agrupamento sobre estes dados. Com o objetivo de realizar uma análise preditiva sobre o tipo de quarto do imóvel, aplicamos uma classificação sobre os dados dos imóveis. Por fim, buscando encontrar padrões que diferenciam superhosts e não-superhosts, aplicamos uma mineração de items frequentes, utilizando uma técnica de discretização especializada para itemsets e posteriormente aplicando uma análise supervisionada sobre os itemsets baseada em outros artigos da literatura.

Os tópicos 2 e 3 introduzem a metodologia utilizada e o dataset sobre o qual a análise foi realizada, estabelecendo o contexto e os fundamentos necessários para entender as etapas subsequentes. Em seguida, os tópicos 4, 5 e 6 abordam as tarefas principais do estudo: clustering, classificação e mineração de itemset, respectivamente. Cada um desses tópicos explora uma técnica específica de análise de dados e os resultados obtidos a partir dessas abordagens. Finalmente, o tópico 7 oferece uma conclusão, resumindo os principais achados e implicações dos resultados das análises realizadas.

2. Metodologia

Nesta seção, abordaremos quatro principais tópicos: o processo de pré-processamento dos dados, a análise de clustering, a aplicação de técnicas de regressão e mineração de itemsets. Primeiramente, discutiremos o pré-processamento e o dataset em questão, que

envolveu a exclusão de valores ausentes e ajustes na formatação dos dados para garantir a precisão e consistência. Em seguida, exploraremos a análise de clustering, detalhando como as avaliações das propriedades foram agrupadas para identificar padrões e características comuns. Finalmente, abordaremos a aplicação de técnicas de regressão e mineração de itemsets para entender as influências das características das propriedades no preço e identificar padrões entre as características dos anfitriões e seu status de superhost. Cada tópico será tratado de forma a oferecer uma visão clara das metodologias utilizadas e dos resultados obtidos.

3. Dataset utilizado e Pré-Processamento

Escolhemos utilizar dados de imóveis do Airbnb para este trabalho devido à plataforma ser uma das maiores e mais conhecidas plataformas de hospedagem de curto prazo, proporcionando um extenso histórico de dados sobre disponibilidade, preços e avaliações de imóveis em diversas localizações, pela internet. Ao optar por esses dados, podemos obter uma visão detalhada e atualizada do mercado de aluguel, o que é crucial para uma análise precisa e informada. A escolha desses dados nos permite explorar tendências, padrões de preços e feedback de hóspedes, proporcionando uma base sólida para nossa pesquisa e conclusões.

Utilizamos o site Inside Airbnb para obter informações gerais trimestrais sobre 36008 imóveis disponíveis no Airbnb na cidade do Rio de Janeiro, coletadas do dia 26/12/2023 ao dia 30/12/2023, referentes ao quarto trimestre de 2023.

O site Inside Airbnb não está associado nem endossado pela Airbnb ou por quaisquer concorrentes da empresa. Os dados disponíveis neste site são baseados em informações públicas coletadas diretamente do site da Airbnb, abrangendo o calendário de disponibilidade para 365 dias futuros e as avaliações de cada anúncio. Estas informações são verificadas, limpadas, analisadas e agregadas para fornecer uma visão detalhada e precisa das listagens disponíveis na plataforma.

No processo de pré-processamento dos dados, o principal passo foi a exclusão das linhas com valores NaN, já que essas entradas não eram úteis e representavam uma pequena porcentagem do total e, portanto, não teriam um impacto significativo na análise geral. Além disso, fizemos a conversão de colunas textuais para colunas numéricas, como a coluna de preço e as colunas de taxa de resposta e taxa de aceitação, eliminando elementos textuais desnecessários e mantendo os dados consistentes.

Por fim, realizamos uma extração de features relacionadas ao número de banheiros de cada imóvel a partir de suas descrições. Uma das colunas da base, ”bathrooms_text”, traz uma descrição textual sobre o número de banheiros do imóvel. Este número pode incluir banheiros privados, banheiros compartilhados ou lavabos (half-bathrooms), e a descrição é escrita de maneira livre pelo proprietário. Com isso, de maneira a tornar os dados inconsistentes, analisamos todos os textos diferentes dentro desta feature e extraímos o número de banheiros, banheiros compartilhados e lavabos de cada imóvel (analisando palavras-chave como ”half”, ”shared”, etc.), gerando três novas features.

Os dados foram então divididos em três bases de dados, relativas às três tarefas realizadas neste trabalho:

- A tabela clustering, onde foram agrupadas as colunas relativas às notas de re-

view (*review_scores_accuracy*, *review_scores_cleanliness*, *review_scores_checkin*, *review_scores_communication*, *review_scores_location* e *review_scores_value*);
- A tabela classification, onde foram agrupadas as colunas relativas a dados do imóvel (*room_type*, latitude, longitude, *accommodates*, *beds* e *price*, mais as três colunas geradas *bathrooms*, *half_bathrooms* e *shared_bathrooms*);
- E a tabela itemsets, onde foram agrupadas as colunas relativas a características do host (*host_response_time*, *host_response_rate*, *host_acceptance_rate*, *host_is_superhost*, *host_verifications*, *host_has_profile_pic* e *host_identity_verified*).

4. Análise de agrupamentos de Notas de Review

Para a tarefa de clustering, selecionamos as colunas review_scores_rating, *review_scores_accuracy*, *review_scores_accuracy*, *review_scores_cleanliness*, *review_scores_checkin*, *review_scores_communication*, *review_scores_location* e *review_scores_value* da tabela de reviews. Essas colunas fornecem uma visão detalhada das avaliações feitas pelos hóspedes sobre as propriedades.

O objetivo é responder à seguinte pergunta: Como as diferentes avaliações das hospedagens agrupam os anúncios coletados e quais são as características comuns das propriedades com classificações altas e baixas?

Para atingir esse objetivo, vamos explorar e analisar os dados de avaliações de hospedagens. Vamos agrupar os anúncios com base em suas avaliações e identificar as características que são comuns entre propriedades com classificações altas e baixas.

4.1. Descrição das Colunas de Review

A coluna review_scores_rating representa a avaliação geral dada pelos hóspedes, refletindo uma média das notas recebidas. A coluna review_scores_accuracy avalia a precisão das descrições das propriedades, indicando o quanto elas correspondem à realidade.Já review_scores_cleanliness mede a satisfação dos hóspedes com a limpeza e arrumação da propriedade. A coluna review_scores_checkin refere-se à experiência de check-in, abrangendo a facilidade e a eficiência do processo de entrada. review_scores_communication avalia a qualidade da comunicação entre o anfitrião e o hóspede, considerando a clareza e a rapidez das respostas. review_scores_location oferece uma avaliação da localização da propriedade, incluindo aspectos como acessibilidade e proximidade de pontos de interesse. Por fim, review_scores_value reflete a percepção dos hóspedes sobre o valor da propriedade em relação ao preço pago. Utilizamos essas colunas para entender os perfis das propriedades com base em suas avaliações e analisar como esses perfis estão distribuídos. Agrupando os imóveis de acordo com suas pontuações em diferentes aspectos, buscamos identificar padrões e clusters de propriedades com características similares em termos de satisfação dos hóspedes.

4.2. Pré-processamento para Agrupamento

Primeiramente, focamos apenas nas colunas que fornecem pontuações relacionadas às características dos anúncios. Esse filtro permitiu concentrar nossa análise nas informações mais pertinentes ao agrupamento. Além disso, decidimos remover a coluna *review_score_rating*, pois ela representa uma média dos valores de avaliação das outras

colunas Dado que essa coluna tende a estar fortemente correlacionada com as demais, como visto na matriz de correlação (Figura 1), sua presença poderia gerar redundâncias e complicar o processo de agrupamento. Assim, sua exclusão ajudou a simplificar a análise.

Outro passo importante foi a eliminação dos anúncios que não possuíam avaliações. Esses anúncios nunca foram alugados, o que impossibilita a atribuição de uma nota. Como não há possibilidade de imputar dados para essas instâncias, optamos por removê-las do dataset. Essa decisão resultou na exclusão de aproximadamente 10.000 registros, refinando o conjunto de dados para a análise.

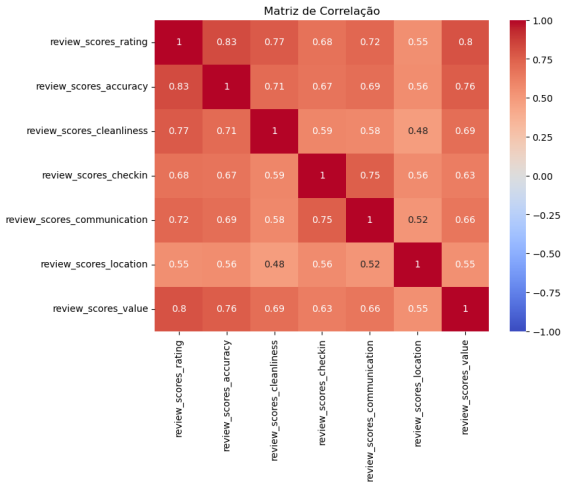

Figura 1. Matriz de correlação das features selecionadas

4.3. Escolha do Algoritmo de Agrupamento

Para a escolha do algoritmo de agrupamento, consideramos que o tamanho do dataset é relativamente pequeno. Além disso, dado que estamos lidando com avaliações, que por natureza são frequentemente enviesadas e podem conter outliers significativos para a análise, optamos por utilizar o algoritmo aglomerativo hierárquico. Esse algoritmo é particularmente adequado para conjuntos de dados pequenos devido à sua complexidade computacional, que é mais fácil de gerenciar em escala reduzida.

Além disso, foi feita a normalização dos dados, mas não a remoção de outliers, uma vez que, por se tratar de dados de reviews, estes contêm naturalmente outliers (geralmente poucos pontos negativos ou poucos pontos positivos) que queremos presentes para avaliar. A presença desses outliers é importante para a análise, pois eles representam avaliações extremas que podem fornecer insights valiosos sobre a satisfação dos hóspedes e a qualidade das propriedades.

Após aplicar o algoritmo e analisar o dendrograma resultante (Figura 2), identificamos três clusters distintos: avaliações altas, avaliações moderadas e avaliações baixas. O agrupamento revelou que:

- Cluster 0 representa avaliações altas, com 17.884 instâncias.
- Cluster 1 corresponde a avaliações moderadas, com 7.652 instâncias.
- Cluster 2 abrange avaliações baixas, com apenas 182 instâncias.

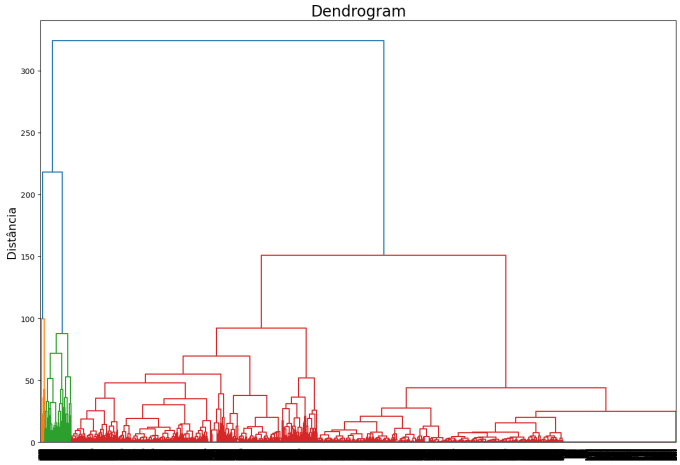

Figura 2. Dendograma dos clusters

4.4. Avaliação do Modelo

Para avaliar a qualidade da clusterização, utilizamos a pontuação de silhueta, uma métrica amplamente reconhecida para medir o quão bem os objetos foram classificados em seus clusters. A pontuação de silhueta é calculada para cada amostra e combina a proximidade média da amostra aos outros pontos do mesmo cluster e a proximidade média ao ponto mais próximo de um cluster diferente.

A pontuação varia de -1 a 1, onde um valor próximo de 1 indica que as amostras estão bem agrupadas e claramente distintas de outros clusters, enquanto uma pontuação próxima de -1 sugere que as amostras podem ter sido agrupadas no cluster errado.

A pontuação de silhueta obtida foi de 0.72, o que sugere que a maioria das amostras estão corretamente atribuídas aos seus respectivos clusters, estando relativamente bem separadas umas das outras.

4.5. Conclusões

A distribuição das instâncias entre os clusters mostra um enviesamento significativo. O Cluster 0, que contém avaliações altas, possui a maioria das instâncias. O Cluster 1, com avaliações moderadas, também tem um número considerável de instâncias, enquanto o Cluster 2, que agrupa avaliações baixas, é bastante pequeno (Figura 6). Este enviesamento reflete a distribuição real dos dados, em que a maioria das propriedades recebe pontuações elevadas, algumas recebem pontuações moderadas e muito poucas recebem pontuações baixas. É comum que os dados de avaliação sejam enviesados, com a maioria das avaliações sendo baixa ou alta.

Para identificar as características mais importantes que diferenciam os clusters com maiores pontuações dos clusters com menores pontuações, calculamos a diferença entre as médias das características para os clusters com a maior e a menor pontuação.

De acordo com a tabela 1, as características mais importantes para determinar as pontuações das hospedagens foram as seguintes:

- **Valor da estadia (review_scores_value)**: Esta característica apresenta a maior diferença entre as classificações altas e baixas, indicando que os hóspedes dão grande prioridade à obtenção de uma boa relação custo-benefício.

Tabela 1. Diferenças de Características entre os Clusters com Maior e Menor Pontuação

Característica	Diferença
review_scores_value	2.909227
review_scores_communication	2.747582
review_scores_accuracy	2.745853
review_scores_cleanliness	2.614229
review_scores_checkin	2.521688
review_scores_location	2.008231

- **Comunicação (review_scores_communication) e Exatidão (review_scores_accuracy)**: Estas características são cruciais, uma vez que têm um impacto significativo nas experiências e percepções dos hóspedes.
- **Limpeza (review_scores_cleanliness), Check-in (review_scores_checkin) e Localização (review_scores_location)**: Embora importantes, possuem um impacto ligeiramente menor em comparação com as outras características.

4.6. Discussão dos Resultados

Os resultados mostram que o *valor da estadia* é a característica mais decisiva para obter avaliações altas, sugerindo que os hóspedes valorizam uma boa relação custo-benefício. Além disso, a *comunicação* e a *exatidão* são igualmente importantes, uma vez que influenciam diretamente a experiência e a satisfação dos hóspedes. Embora as características de *limpeza*, *check-in* e *localização* também sejam relevantes, seu impacto é um pouco menor em comparação com as outras características analisadas. Estes insights podem ser utilizados para melhorar as estratégias de gestão das propriedades, focando nas áreas que mais influenciam a satisfação dos hóspedes.

4.7. Análises Adicionais

Nesta subseção, apresentamos análises adicionais que fornecem insights complementares sobre os clusters identificados na nossa análise de avaliações. As seguintes análises foram realizadas: a distribuição de superhosts por clusters, a distribuição espacial dos anúncios por cluster e o preço médio por cluster.

4.7.1. Distribuição de Superhosts por Clusters

Uma análise da distribuição de superhosts entre os clusters (Figura 3) revelou que apenas o Cluster 0, com avaliações altas, possui uma quantidade significativa de superhosts, enquanto nos outros clusters essa quantidade é quase nula. Isso indica que apenas hosts com ótimas avaliações podem alcançar o status de superhost. Esta observação sugere que a qualidade das avaliações é um fator crítico para que os anfitriões sejam reconhecidos como superhosts, evidenciando a correlação entre a excelência no serviço prestado e o reconhecimento pelos hóspedes.

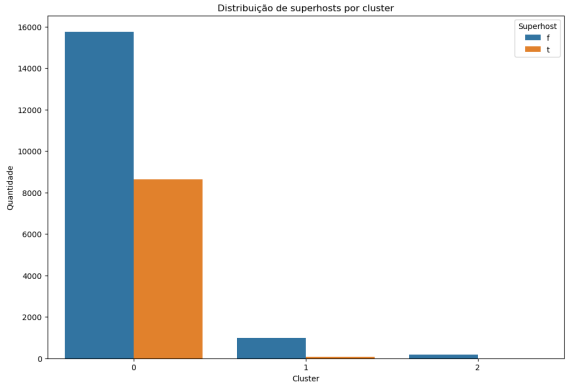

Figura 3. Distribuição de Superhosts por Cluster

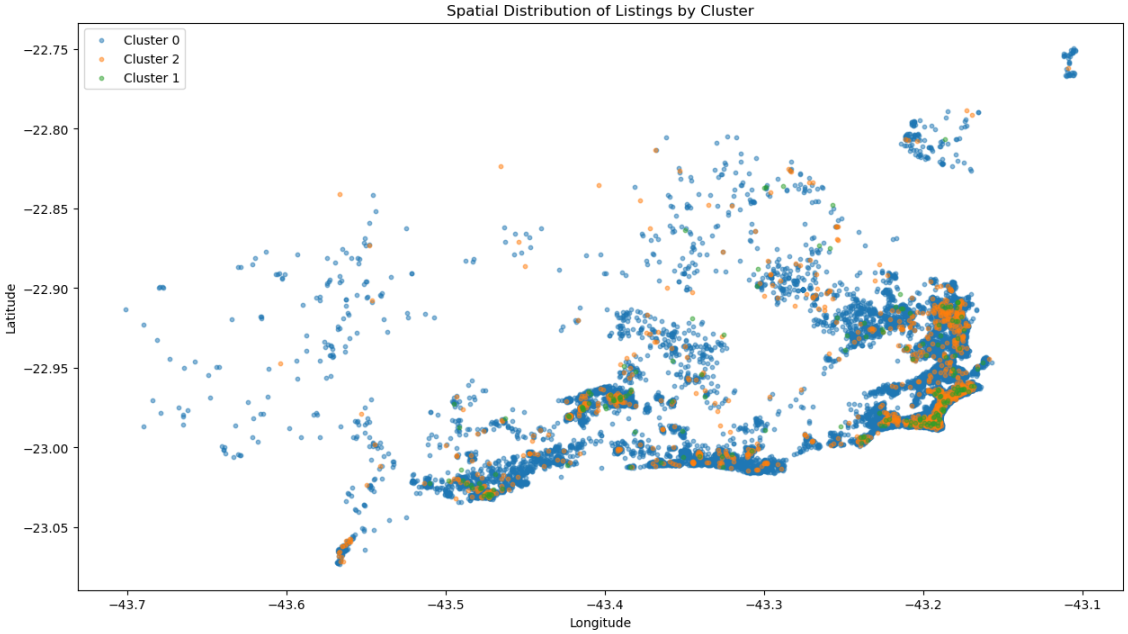

Figura 4. Distribuição Espacial dos Imóveis por Cluster

4.7.2. Distribuição Espacial de Anúncios por Cluster

A análise da distribuição espacial dos anúncios (Figura 4) revelou que os pontos verdes, atribuídos ao Cluster 2 de baixa pontuação, estão agrupados em regiões específicas. Este padrão espacial pode indicar que a qualidade da hospedagem está relacionada ao bairro onde as propriedades estão localizadas. Áreas com uma concentração maior de anúncios de baixa pontuação podem refletir problemas específicos dessas regiões, como segurança, acessibilidade, ou infraestrutura, que afetam negativamente as avaliações dos hóspedes.

4.7.3. Preço Médio por Cluster

Ao analisar o preço médio dos anúncios por cluster (Figura 5), observamos que o preço médio do Cluster 2, de avaliações baixas, é o menor de todos, com uma diferença de aproximadamente 200 reais em relação aos outros clusters. Este resultado é interessante por-

que, apesar de ser o cluster mais barato, as avaliações mostram uma grande insatisfação no quesito custo-benefício. Isso sugere que, mesmo com preços mais baixos, as propriedades no Cluster 2 não estão atendendo às expectativas dos hóspedes, refletindo uma percepção de baixa qualidade que não compensa o valor pago.

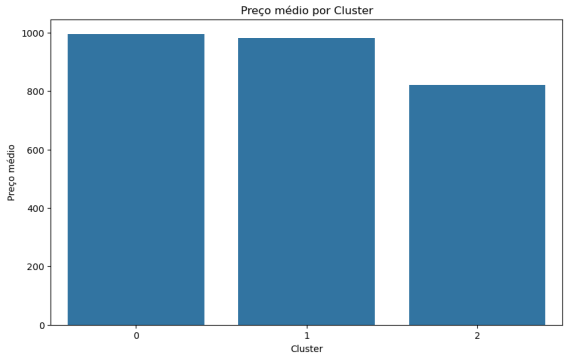

Figura 5. Preço Médio do Imóvel por Cluster

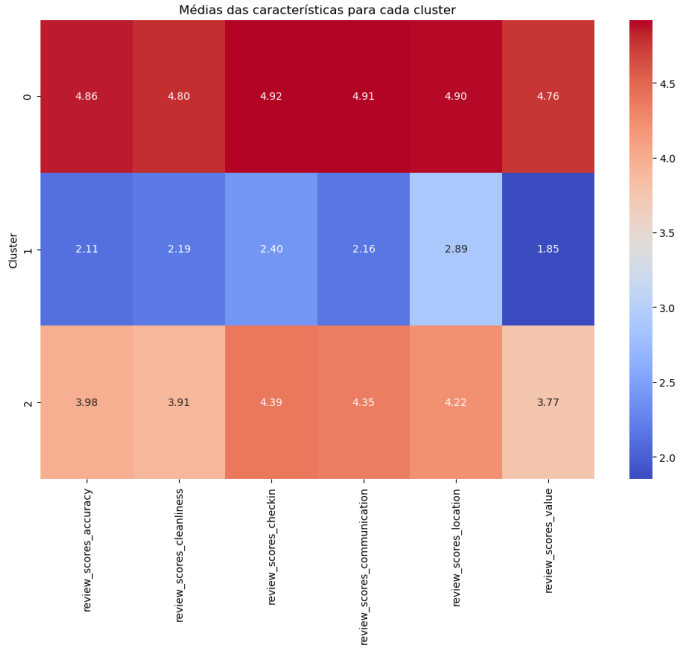

Figura 6. Médias das Características para cada Cluster

5. Análise Preditiva de Tipos de Quarto

Para esta análise foram separadas as colunas relativas às características intrínsecas aos imóveis. São elas a localização (em latitude e longitude), o número de camas, número de hóspedes que podem ser recebidos, número de banheiros (privados, compartilhados e lavabos), preço e o tipo de quarto (apartamento/casa completa, quarto privado, quarto compartilhado ou quarto de hotel). A partir destes dados dos imóveis, desejamos provar a hipótese de que ”O tipo de quarto pode ser previsto a partir das demais características do imóvel”. Para isso, realizamos uma classificação com a coluna *room_target* como target, e as demais colunas numéricas como features.

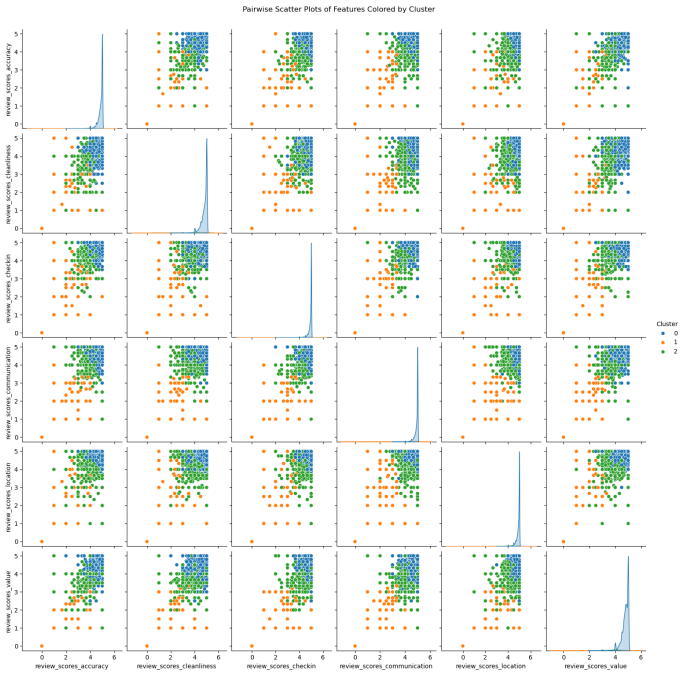

Figura 7. Gráficos de Dispersão Pareados das Características, por Cluster

Realizamos uma Z-normalização nos dados e uma separação aleatória em 80% das instâncias para treino e 20% das instâncias para teste, e validamos quatro classificadores: Random Forest, Gradient Boosting, Gaussian Naive Bayes e SVM, implementados usando a biblioteca sckit-learn. Para a validação, utilizamos validação cruzada com 5-folds e o score de acurácia médio das folds para validação. Por fim, treinamos o modelo de melhor desempenho na validação, Random Forest, com todos os dados de treino e o validamos com os dados de teste, coletando acurácia, precisão, recall e f1.

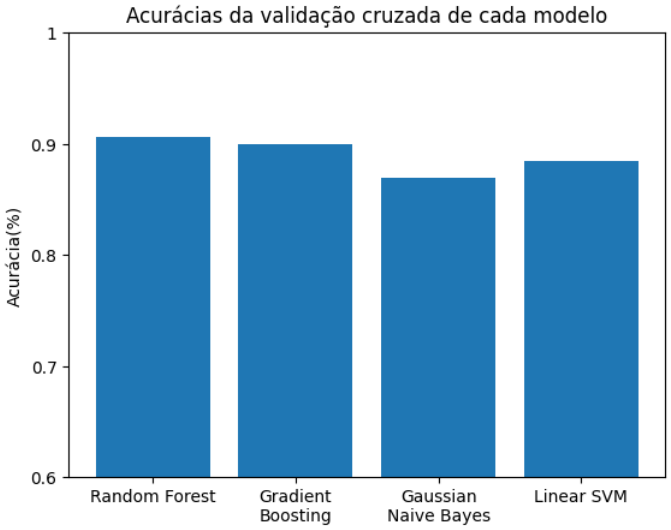

Figura 8. Acurácias da validação cruzada de cada modelo

Ao realizar a validação cruzada dos modelos, obtivemos a acurácia de cada uma, conforme mostrado na figura 8. Dentre todos os modelos, o que melhor se desempenhou

foi o Random Forest com cerca de 90.6% de acurácia média na validação cruzada, embora todos tenham apresentado um desempenho muito próximo ao ideal. Com isso, selecionamos o Random Forest e o treinamos com a base completa de treino, validando com a base de teste.

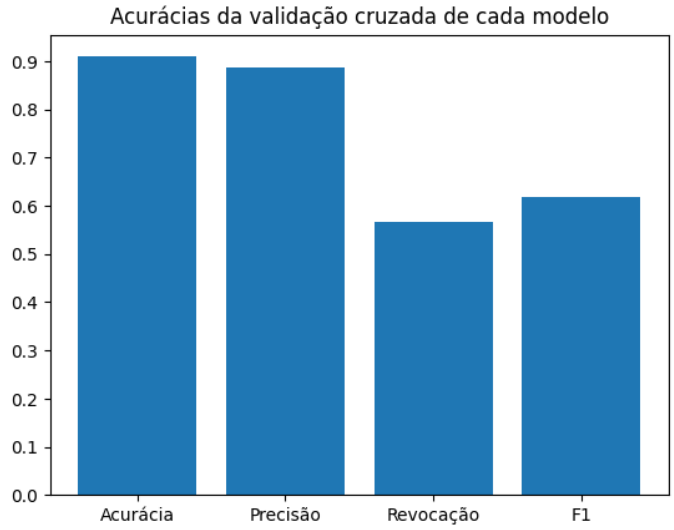

Figura 9. Scores finais de teste do Random Forest

O desempenho obtido pode ser visto na figura 9. Como podemos ver, o desempenho em termos de acurácia é bem semelhante ao obtido na validação, cerca de 90.9%. No entanto, o modelo varia bastante em termos de precisão e revocação. Enquanto a precisão média é de 88.68%, um valor alto e próximo ao da acurácia, indicando que o modelo possui um baixo número de falsos positivos para cada classe. No entanto, a revocação média é de 56.67%, um pouco mais baixa que a precisão, indicando que o modelo tem uma tendência a gerar falsos negativos, ou seja, alguma das classes tem uma tendência a ser menos predita que as demais. Embora este valor de recall não seja ideal, de maneira geral um recall de cerca de 60%, que é o nosso caso, ainda é considerado bom em um modelo de alta acurácia, então podemos concluir que, de maneira geral, o desempenho do modelo do modelo na previsão dos tipos de quarto foi suficientemente bom, confirmando a hipótese proposta.

6. Mineração Supervisionada de Padrões Frequentes entre Hosts

Para esta análise foram separadas as colunas relativas às características dos hosts. São elas booleanos indicando se o host tem a identidade verificada; se possui foto de perfil; se é superhost; a lista de tipos de verificação que o host possui (email, telefone, etc.); o tempo de resposta do host; a taxa de resposta do host; e a taxa de aceitação do host.

Dentre todas as características, a mais subjetiva é se o host é um ”superhost”, que se refere a uma premiação que o próprio Airbnb escolhe dar para alguns hosts da plataforma. Embora o site forneça alguns pré-requisitos para que a categoria seja premiada, a premiação ainda é feita de forma individual e arbitrária para cada indivíduo. Por isso, a partir dos dados dos hosts, desejamos responder a seguinte questão: ”Quais conjuntos de características definem um superhost? Quais definem um não superhost? Quais são

comuns a ambos?". Para isso, realizamos uma mineração de itemsets frequentes sobre os dados e aplicamos uma análise supervisionada dos itemsets sobre a coluna de superhosts, segundo uma abordagem semelhante para a tarefa observada em outras aplicações da literatura, e que será descrita em mais detalhes posteriormente.

Para a montagem das transações, cada instância da base se transformou em uma transação. Como a maioria das colunas neste caso são booleanas ou categóricas, para estes casos a montagem da transação foi simples. No caso das colunas booleanas, a feature foi adicionada à transação se, e somente se seu valor na tabela é verdadeiro. No caso das colunas categóricas, que é o caso do tempo de resposta do host (que por padrão já vem discretizado), o valor da coluna foi adicionado como item na transação. No caso da coluna contendo a lista de verificações, todos os elementos da lista foram adicionados como item na transação. Por fim, as colunas contínuas, contendo as taxas de resposta e taxas de aceitação do host, precisaram ser tratadas com uma técnica de discretização, conforme descrito no item seguinte.

Por fim, para a mineração em si utilizamos o algoritmo FP-Growth [Han and Pei 2000]. Isso foi feito pois a base, embora grande, ainda era pequena o suficiente, tanto em instâncias quanto em colunas, para que um algoritmo exaustivo, que testa todos os itemsets possíveis, fosse executado. Deste modo, o FP-Growth foi escolhido por possuir o menor tempo de execução de maneira geral dentre todas as técnicas de mineração de itemsets exaustivas. O FP-Growth foi aplicado utilizando a biblioteca pyfpgrowth em python [fpg], a principal biblioteca implementando este algoritmo na linguagem.

6.1. Discretização Utilizando Equi-Depth Patitioning

Dentre as colunas utilizadas, duas das colunas continham valores contínuos, relativos à taxa de resposta do host e à taxa de aceitação do host, em porcentagem. Para realizar a discretização destes valores para a montagem das transações, utilizamos a abordagem proposta por Srikant e Agrawal [Srikant and Agrawal 1996], denominada Equal Frequency Partitioning, também conhecida como Equi-Depth Partitioning. Nela, dado um valor de n, divide-se o intervalo dos valores em n intervalos de maneira que a frequência de instâncias em cada intervalo seja a mesma. No artigo, Srikant e Agrawal demonstram que esta divisão minimiza a entropia relativa entre cada intervalo, tornando-a ideal. Implementamos a equi-depth partitioning utilizando um algoritmo que, essencialmente, ordena todos os valores da coluna, divide-os em 10 setores de mesmo tamanho e usa o primeiro e o último valor de cada setor como limite inferior e superior de cada partição. Desta maneira, cada partição delimita setores de mesmo tamanho.

Por fim, para selecionar o número de bins, calculamos os cinco valores mais frequentes de taxa de resposta e taxa de aceitação. A ideia é selecionar o menor número de bins de modo que cada valor esteja completamente contido em um bin. A figura 10 mostra os resultados obtidos.

Como as taxas mais frequentes são de 100% em ambos, com porcentagens de 68% e 28% para taxa de resposta e taxa de aceitação, respectivamente. A partir destes valores, os números de bins ideais seriam 1 e 3, o piso do inverso das maiores porcentagens. No entanto, Como estes bins são muito baixos, de maneira a não afetar demais a expressividade dos dados escolhemos 5 bins para ambos os casos, optando por mesclar os

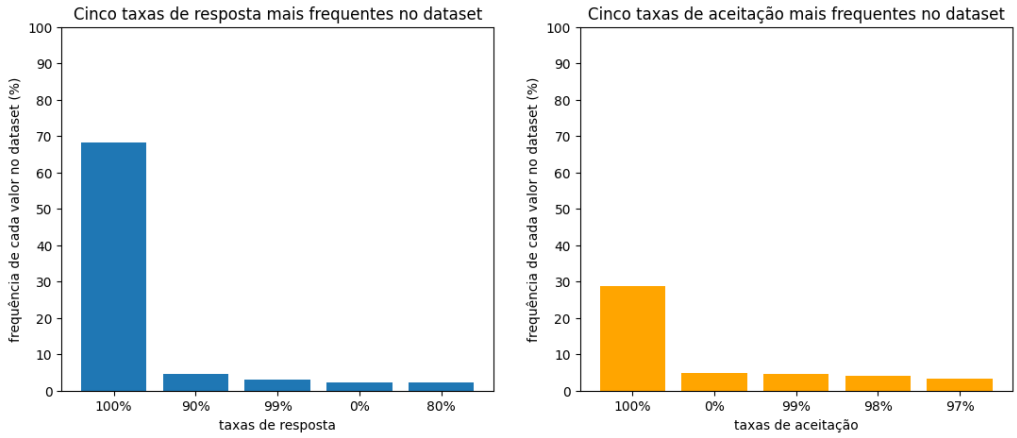

Figura 10. Taxas mais frequentes de resposta e aceitação no dataset

bins contendo somente taxas de 100%, caso houvessem. Com isso geramos os seguintes intervalos:

- host_response_rate: [0%-89%]
- host_response_rate: [90%-99%]
- host_response_rate: [100%-100%]
- host_acceptance_rate: [0%-57%]
- host_acceptance_rate: [58%-83%]
- host_acceptance_rate: [84%-96%]
- host_acceptance_rate: [97%-99%]
- host_acceptance_rate: [100%-100%]

Como podemos ver, a maior parte das taxas se concentra nos maiores valores, evidenciando a importância do particionamento por frequência ao invés do particionamento equidistante padrão.

6.2. Análise Supervisionada de Itemsets

Em [Bunker et al. 2021], Bunker et al. propõem uma abordagem inédita para a tarefa de mineração supervisionada de padrões, que consiste essencialmente na tarefa de em minerar padrões distintos a uma única classe de um target definido em relação às demais classes. Embora a abordagem proposta tenha alta complexidade, utilizando geração de padrões por meio de podas de árvores e seleção de padrões e utilizando pesos de features em classificadores, o artigo também propõe uma segunda abordagem mais simples para a tarefa, utilizando técnicas não-supervisionadas clássicas de mineração de itemsets. Segundo eles, a segunda abordagem correspondia à melhor estratégia existente para a solução do problema antes da publicação de artigo, e por isso foi utilizada como comparativo dos resultados que eles obtiveram. Devido à sua simplicidade comparada ao âmbito deste trabalho, escolhemos implementar a abordagem mais simples para a tarefa de minerar padrões característicos apenas de hosts que são superhosts, adicionando um segundo passo na análise dos itemsets, tornando-a mais precisa.

A técnica em questão consiste em separar o dataset em dois datasets: um dataset contendo as instâncias com valor positivo no target e um dataset contendo instâncias com

valor negativo no target, e minerar padrões frequentes em ambos utilizando uma abordagem não-supervisionada clássica (como o Apriori, por exemplo. No nosso caso usamos o FP-Growth). Por fim, seleciona-se para cada base, somente os itemsets que são frequentes nela e não são frequentes na base oposta. Com isso, encontra-se os itemsets exclusivos de cada base, que correspondem aos itemsets exclusivos dos valores do target, que no nosso caso correspondem ao usuário ser ou não superhosts. Com isso, a implementação do método foi feita simplesmente separando a base original em uma base de superhosts e uma base de não superhosts e, ao fim, coletando os itemsets frequentes únicos aos superhosts, únicos aos não superhosts e comuns a ambos, e seus respectivos suportes.

6.3. Resultados

Após a aplicação da discretização descrita nos itens anteriores, geramos os itemsets para o conjunto de superhosts e o conjunto de não superhosts, totalizando 10 itemsets frequentes de superhosts e 3 . O threshold de suporte utilizado foi de 70% (em relação ao tamanho de cada base), após testes empíricos, mas também calculamos os itemsets com threshold de 40% para que possamos comparar as diferenças. Em seguida, segregamos os itemsets exclusivos de cada base e os itemsets comuns a ambas, gerando os itemsets das tabelas 2, 3 e 4 para o threshold de 70%.

Unique Superhost Itemsets	Percentage
('host_response_rate:[100%-100%]',)	89.5%
('host_has_profile_pic', 'host_response_rate:[100%-100%]')	89.05%
('host_has_profile_pic', 'host_response_time_within_an_hour')	80.95%
('host_identity_verified', 'host_response_rate:[100%-100%]')	79.5%
('host_has_profile_pic', 'host_identity_verified', 'host_response_rate:[100%-100%]')	79.26%
('host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	76.12%
('host_has_profile_pic', 'host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	75.74%
('host_identity_verified', 'host_response_time_within_an_hour')	72.99%
('host_has_profile_pic', 'host_identity_verified', 'host_response_time_within_an_hour')	72.71%

Tabela 2. Unique Superhost Itemsets (70% threshold)

A tabela 2 apresenta os padrões com frequência superior a 70% únicos aos superhosts. Primeiramente, é importante ressaltar que a implementação utilizada minera automaticamente os itemsets fechados, por isso os subconjuntos dos itemsets frequentes (que em teoria seriam mais frequentes pela propriedade de não-monotonicidade da tarefa) não aparecerão. É notável que as características mais comuns entre superhosts são uma alta taxa de resposta (de 100%, exatamente), um curto tempo de resposta (a opção de tempo de resposta "dentro de uma hora"é a mais curta dentro do dataset, com as demais sendo da ordem de múltiplas horas ou dias), com a identidade verificada e com foto de perfil no aplicativo. Todos os itemsets frequentes desta categoria apresentam combinações de dois ou mais destes elementos.

No entanto, é importante ressaltar que, dentre eles, um bom desempenho nas respostas aos clientes (seja no tempo ou na taxa de mensagens respondidas) são os mais importantes, estando obrigatoriamente presentes em todos os itemsets frequentes. As demais características provavelmente foram puxadas nos itemsets por serem comuns em toda a base, já que se tratam de características básicas de cadastro (identificação e foto). Além disso, é interessante observar que as demais características analisadas, as taxas de aceitação e o tipo de verificação, não tiveram muito impacto no caso dos superhosts.

Estes resultados são coerentes com a realidade, uma vez que, no caso do tipo de verificação, considerando que o aplicativo tenha um sistema de verificação consistente, não deveria importar o tipo de verificação feita contanto que o usuário seja verificado, que foi a cracterística que de fato apareceu entre os superhosts. Já no caso da taxa de aceitação, nem sempre um host ruim terá baixas taxas de aceitação especificamente. Seria inclusive plausível que hosts ruins tivessem altas taxas de aceitação, para lucrar com marcações mesmo com imóveis impróprios para receber visitantes, mas este resultado é meramente hipotético e não pode ser medido nos nossos dados, que não têm informações específicas sobre hosts ruins.

Common Itemsets	Superhost/Not Superhost Percentage
('host_has_profile_pic', 'host_identity_verified')	88.57% / 81.72%
('host_has_profile_pic',)	99.39% / 97.07%

Tabela 3. Common Itemsets (70% threshold)

Unique Not Superhost Itemsets	Percentage
('host_identity_verified',)	84.0%

Tabela 4. Unique Not Superhost Itemsets (70% threshold)

No que diz respeito aos padrões frequentes entre não-superhosts e aos padrões comuns a ambos, foram gerados muito menos padrões frequentes. Isso provavelmente se deve ao fato de que a base de não-superhosts é muito mais diversa que a base de superhosts, contendo tanto hosts ruins, quanto médios, quanto potenciais superhosts que ainda não receberam a premiação no aplicativo. Com isso, não é esperado ver padrões de comportamento frequentes nestes grupos, como taxas de aceitação e resposta, e sim elementos mais comuns a todas as instâncias, que é o caso dos itens de cadastro mencionados anteriormente na análise. De fato, entre os padrões comuns a ambos os grupos, encontramos justamente a presença de foto de perfil e a foto de perfil com identidade verificada. E entre os padrões apenas de não-superhosts, temos somente a identidade verificada. Deste modo, como nenhum superhost possui identidade verificada sem possuir foto de rosto (do contrário o itemset somente com a identidade verificada seria comum a ambos), podemos concluir que provavelmente o Airbnb coloca como pré-requisito que todos os superhosts tenham foto de perfil.

Como forma de completar a análise, também coletamos as mesmas informações usando o threshold de 40% na mineração de itemsets. Com isso, podemos observar se as observações feitas inicialmente mudam muito considerando itemsets não tão frequentes quanto o limite utilizado originalmente. As tabelas 5, 6 e 7 apresentam os resultados obtidos.

Unique Superhost Itemsets	Percentage
('host_response_rate:[100%-100%]',)	89.5%
('host_has_profile_pic', 'host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	75.74%
('host_identity_verified', 'host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	68.31%
('host_has_profile_pic', 'host_identity_verified', 'host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	68.08%

Tabela 5. Unique Superhost Itemsets

Common Itemsets	Superhost/Not Superhost Percentage
('host_has_profile_pic', 'host_identity_verified')	88.57% / 81.72%
('host_has_profile_pic', 'host_response_rate:[100%-100%]')	89.05% / 56.71%
('host_has_profile_pic',)	99.39% / 97.07%
('host_has_profile_pic', 'host_identity_verified', 'host_response_time_within_an_hour')	72.71% / 44.45%
('host_response_rate:[100%-100%]', 'host_response_time_within_an_hour')	76.12% / 40.23%
('host_has_profile_pic', 'host_response_time_within_an_hour')	80.95% / 50.61%
('host_has_profile_pic', 'host_identity_verified', 'host_response_rate:[100%-100%]')	79.26% / 47.87%
('host_identity_verified', 'host_response_rate:[100%-100%]')	79.5% / 49.1%

Tabela 6. Common Itemsets

Unique Not Superhost Itemsets	Percentage
('host_identity_verified',)	84.0%
('host_identity_verified', 'host_response_time_within_an_hour')	45.41%

Tabela 7. Unique Not Superhost Itemsets

Como podemos observar, quase nenhum padrão novo surgiu entre as tabelas. O que aconteceu foi que, num geral, a maior parte dos padrões comuns a superhosts com threshold 70% passou a ser comum a ambos, já que o threshold baixo permitiu que não-superhosts apresentassem estes itemsets com frequência. Dentre os três itemsets novos (dois únicos a superhosts e um único a não-superhosts) todos consistem essencialmente em padrões que já eram frequentes no threshold de 70%, com a adição do tempo de resposta dentro de uma hora, que também foi apontado como uma característica de superhosts. Mesmo que ele tenha aparecido em um padrão exclusivo a não-superhosts, é de se esperar que, por ser uma característica positiva, a plataforma beneficie os hosts com tempo de resposta curto de outras formas, justificando sua presença mesmo entre os não-superhosts. Além disso, o padrão novo dos não-superhosts não possui foto de perfil, corroborando a observação levantada anteriormente sobre a provável obrigatoriedade de fotos de perfil para que um usuário seja condecorado superhost.

7. Conclusão

Na análise de agrupamento de avaliações, aplicamos o algoritmo de clustering aglomerativo hierárquico para identificar padrões nas avaliações dos imóveis. Observamos três clusters principais: avaliações altas, moderadas e baixas, com a maioria dos imóveis agrupados nas avaliações altas. A métrica de pontuação de silhueta, utilizada para avaliar a qualidade do clustering, confirmou a validade dos clusters, refletindo uma distribuição esperada e positiva das avaliações.

Na tarefa de classificação, investigamos a capacidade de prever o tipo de quarto a partir das características dos imóveis. Utilizando vários classificadores, o Random Forest demonstrou o melhor desempenho, com alta acurácia. Essa análise não só confirmou a hipótese de que as características dos imóveis podem prever o tipo de quarto, como também destacou a eficácia de modelos preditivos para essa finalidade.

Finalmente, exploramos a mineração de padrões para entender quais características definem um superhost no Airbnb. Através da mineração de itemsets frequentes e da análise supervisionada, identificamos padrões distintos entre superhosts e não-superhosts. A análise revelou que certos atributos, como o tempo de resposta e as taxas de aceitação dos hosts, são indicativos significativos do status de superhost, fornecendo insights valiosos sobre o perfil de hosts bem-sucedidos.

Referências

Welcome to fp-growth's documentation! — fp-growth 1.0 documentation. https://fp-growth.readthedocs.io/en/latest/.

Bunker, R., Fujii, K., Hanada, H., and Takeuchi, I. (2021). Supervised sequential pattern mining of event sequences in sport to identify important patterns of play: an application to rugby union. *PloS one*, 16(9):e0256329.

Han, J. and Pei, J. (2000). Mining frequent patterns by pattern-growth: methodology and implications. *ACM SIGKDD explorations newsletter*, 2(2):14–20.

Srikant, R. and Agrawal, R. (1996). Mining quantitative association rules in large relational tables. *SIGMOD Rec.*, 25(2):1–12.