UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

André Warschauer de Crescenzo

Otimização de precificação dinâmica para aluguel de temporada

Florianópois
2021

André Warschauer de Crescenzo

Otimização de precificação dinâmica para aluguel de temporada

Relatório final da disciplina DAS5511 (Projeto de Fim de Curso) como Trabalho de Conclusão do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Santa Catarina em Florianópolis.
Orientador: Prof. Eduardo Camponogara, Dr.

Florianópois
2021

Ficha de identificação da obra

A ficha de identificação é elaborada pelo próprio autor.

Orientações em:

http://portalbu.ufsc.br/ficha

André Warschauer de Crescenzo

Otimização de precificação dinâmica para aluguel de temporada

Esta monografia foi julgada no contexto da disciplina DAS5511 (Projeto de Fim de Curso) e aprovada em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação

Florianópolis, 22 de 12 de 2022.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Jomi Fred Hübner, Dr.
Avaliador
UFSC/CTC/DAS

Eduardo Camponogara, Eng.
Orientador
UFSC/CTC/DAS

Bruno Benetti.
Avaliador
Instituição Seazone

Prof. Ricardo José Rabelo, Dr.
Presidente da Banca
UFSC/CTC/DAS

AGRADECIMENTOS

É difícil compilar em uma página todas as pessoas a quem sou muito agradecido pelo apoio e suporte durante a execução desse trabalho. Caso eu deixe de mencionar algum nome, não é por falta de carinho ou apreciação, mas porque escrevo com alguma pressa pois o prazo de entrega está mais próximo do que eu gostaria.

Primeiramente agradecer aos meus avós, meus pais e ao meu irmão, que estiveram do meu lado me incentivando em todos os momentos da faculdade.

Também preciso agradecer ao Pedro, que trilhou todo esse caminho comigo e que, sem ele, tudo teria sido muito pior.

A Maria, que também se juntou a mim nesse último ano e me deu todo o carinho e afeto que eu poderia pedir, e que mesmo quando eu sentia que não aguentava mais, não me deixou sozinho.

Ao Bill, que acompanhou mais perto do que qualquer um todos os desafios e soluções, e que é um dos poucos que de fato compreende de forma ampla e completa todas as facetas desse problema, e que confiou em mim completa liberdade para desenvolvimento de algorítimos e modelos.

Ao Eduardo Camponogara, que conferiu todos os passos e todas as contas com tamanha atenção que nunca vi nenhum professor dedicar a um aluno.

A toda equipe da Seazone e da Khanto, presentes e passados, desde sócios até os estágiários, que criaram e mantém toda a estrutura para que esse projeto fosse viável.

A toda a equipe do DAS, em especial ao Rodrigo e à Livia, que fazem todo o labirinto que é a graduação se tornar navegável.

Seja como for, no conjunto eu alcanço o que queria alcançar.
Não se diga que o esforço não valeu a pena.
No mais não quero nenhum julgamento dos homens,
quero apenas difundir conhecimentos.
(Kafka, 1917)

RESUMO

Esta monografia aborda o problema de precificação de diárias no Airbnb, e propõe um algoritmo baseado em dados de ocupação e preço do mercado. Partindo de dados limpos, utiliza métodos de aprendizado de máquina supervisionado, para fazer a modelagem e análise da probabilidade de ocupação de apartamentos. A monografia também explora diversos métodos diferentes, e faz a escolha de hiper parâmetros a partir de ferramentas de AutoML. Utiliza e detalha o funcionamento de diversas ferramentas de análise para se assegurar que a qualidade dos dados não foi comprometida, e encontra erros no processo de limpeza dos dados. Faz uso de teorias econômicas para modelar o faturamento de um imóvel, a partir da sua probabilidade de ocupação em cada dia.

Palavras-chave: Aprendizado de máquina, Aluguel de temporada, Precificação

ABSTRACT

This report addresses the problem of pricing of Airbnb listings, and proposes an algorithm based on occupancy and pricing data from the market. Starting from clean data, it uses supervised machine learning methods to model and analyze the probability of occupancy of apartments. It explores several different methods, and chooses hyper parameters with AutoML tools. The report details the functioning of various analysis tools to ensure that data quality has not been compromised, and finds errors in the data cleaning process. It uses economic theories to model the revenue of a property, based on its occupancy probability in each day.

Keywords: Machine Learning. Short Stay. Pricing.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

A automação de processos de precificação dinâmica teve impactos gigantescos em quase todas as indústrias, mas em especial no setor da aviação e de hotelaria foram talvez os mais afetados.

O mercado de hotelaria também tem sofrido, na última década, grandes impactos da economia compartilhada, com plataformas como o Airbnb focando em anúncios de apartamentos e casas para a hospedagem de temporada.

Nesse projeto de final de curso, vamos propor uma estratégia de precificação baseada em dados históricos do Airbnb. Queremos precificar os anúncios de forma a maximizar o faturamento dos imóveis.

O que será apresentado aqui é a parte do processo referente a modelagem e análise de dados. Ou seja, a obtenção e limpeza dos dados já foi feita, e não será apresentada nesse relatório. Queremos, a partir dos dados limpos de ocupação e preço do Airbnb, construir um modelo que consiga calcular a probabilidade de ocupação de um imóvel. Vamos experimentar com diferentes algoritmos de aprendizado de máquina – como redes neurais, árvores de decisão, e random forests – escolher a que tiver o melhor desempenho, fazer a escolha de hiper-parâmetros e analisar os resultados com ferramentas agnósticas ao modelo como SHAP e LIME.

Vamos contruir um modelo analítico discreto do faturamento de imóveis, onde o preço ao longo do tempo é uma série que pode ser determinada a partir da probabilidade de ocupação e o preço. Utilizaremos os modelos obtidos para resolver uma equação de diferenças que retorna o preço ótimo de um apartamento ao longo do tempo.

Embora o preço resultante dos algoritmos não tenha sido satisfatório, o processo de modelagem e análise apontou para erros na limpeza dos dados que, quando corrigidos, deve melhorar bastante o desempenho da precificação proposta. Além disso, o modelo teve boa capacidade de predição de ocupação, considerando principalmente a antecedência das reservas.

A análise do modelo também trouxe bastante conhecimento a respeito do impacto da pandemia na ocupação dos imóveis. Após a conclusão desse projeto, com o aprendizado obtido, a empresa começou o desenvolvimento de um produto de precificação.

Tudo aqui apresentado foi obra do autor desse documento, mas vale ressaltar que os dados são propriedade da empresa, e que a limpeza – embora idealizada em conjunto como equipe – não foi realizada pelo autor.

1.1 ESTRUTURA DO DOCUMENTO

O trabalho foi estruturado da seguinte maneira:

- O Capítulo 2 aborda o contexto e conceitos de mercado de aluguel de temporada e precificação dinâmica. Aqui introduzimos a fundo o problema a ser resolvido.
- O Capítulo 3 explica os conceitos técnico-teóricos utilizados na realização do trabalho. É aqui que explicamos os modelos de faturamento utilizados e as simplificações postas.
- O Capítulo 4 explica o passo a passo a ser realizado para se solucionar o problema proposto.
- O Capítulo 5 apresenta a implementação da solução e resultados obtidos.
- O Capítulo 6 apresenta a conclusão, ou seja, síntese pessoal, objetiva, sucinta e interpretada dos resultados. É um resumo do que foi feito, dos resultados globais (frente aos objetivos inicialmente traçados). Também é aqui que colocamos sugestões para trabalhos futuros.

2 MOTIVAÇÃO

O contexto desse PFC se dá na Seazone, uma empresa de administração de imóveis focada em aluguel de temporada. Mais especificamente, aqui buscamos responder:

"Por que preço anunciar a diária?"

Ou, de forma ligeiramente mais completa:

"Como precificar um anúncio?"

Neste capítulo, exploraremos os conceitos necessários para se entender completamente essas perguntas.

2.1 ALUGUEL DE TEMPORADA - SEAZONE

Em Florianópolis, devido ao alto índice de turismo da cidade, muitos imóveis conseguem um retorno maior fazendo aluguel por temporada em sites como o Airbnb, do que fazendo o aluguel anual tradicional. Mas a administração do imóvel para o aluguel de temporada requer processos diferentes do aluguel anual, desde checkin e checkout de hóspedes até a Precificação das diárias - tópico deste trabalho.

A Seazone surge para atender esse nicho, percebendo que os anúncios no Airbnb em Jurerê eram feitos de forma amadora: Fotos desfocadas, escuras e/ou borradas, e descrições incompletas eram comuns.

Além disso, os preços cobrados durante a baixa temporada estavam inflados: Muitos proprietários preferiam deixar seus imóveis vazios do que alugar a um preço baixo, em parte devido ao trabalho de se realizar o checkin e o checkout do hóspede, além da limpeza do imóvel.

Não foi incomum um crescimento de 30% no faturamento dos imóveis que começavam a ser administrados pela Seazone, o que justificava facilmente a comissão de 10% a 20% que a empresa cobrava sobre o faturamento do imóvel.

Com o bom desempenho veio o tamanho, e com o tamanho vieram problemas. Enquanto era razoável decidir o preço de 1, ou até 10 imóveis, olhando o preço de imóveis parecidos e tentando sempre se manter competitivo, fazer isso para os 200 anúncios que a empresa hoje possui já é uma tarefa que demanda bastante tempo. Com o crescimento no portfólio de imóveis de 10% ao mês que vem sido observado, mesmo durante a pandemia, se faz necessário o desenvolvimento de soluções mais escaláveis de precificação.

Além disso, a qualidade da precificação tem se mostrado essencial para o faturamento dos imóveis. Apenas seguindo o comportamento do mercado, principalmente durante a pandemia, se mostrou uma estratégia subótima. Boa parte do mercado

segurou os preços altos apesar da baixa procura, resultando em imóveis vazios. A demanda ainda existia, mas os imóveis precisaram abaixar seus preços para voltar a obter retorno. Ou seja, se faz necessária uma estratégia de precificação – objetivo deste trabalho.

A seguir serão introduzidos conceitos relevantes para a resolução do problema, primeiramente os relevantes aos processos da empresa, específicos do setor de aluguel de temporada, e depois conceitos mais genéricos e paralelos com outras indústrias – como a da aviação – de onde boa parte da fundamentação teórica para a solução advém.

2.1.1 Online Travel Agencies (O.T.A.s)

Uma O.T.A (Online Travel Agency) é um canal de vendas que busca auxiliar hóspedes a se conectarem com uma hospedagem de curta duração, como um hotel ou pousada. O.T.A.s também podem conectar viajantes a outros serviços, como bilhetes aéreos ou aluguel de carros, mas esses não serão tratados neste documento.

A primeira O.T.A., Expedia, foi fundada em 1996 como parte da Microsoft, seguida em 1997 pela a Booking Holdings Inc. Ainda hoje, essas são as duas principais empresas do setor, gerando quase 200 bilhões de dólares em vendas [`https://www.travelweekly.com/Power-List-2019`].

Em 2008, a Airbnb entra no mercado, com uma proposta um pouco diferente: fazendo uso da recente economia do compartilhamento, a plataforma permite que proprietários disponibilizem seus imóveis para locação de temporada, desde que se responsabilizem pelo checkin e o checkout. Hoje conta com mais de 500 mil anúncios em mais de 35.000 cidades e 192 países, e já afeta a economia do mercado imobiliário, com um estudo em 2017 correlacionando um aumento em 10% no número de listings em uma região com um aumento de 0,42% no preço dos aluguéis e de 0,76% no custo das propriedades (BARRON; KUNG; PROSERPIO, 2020).

Rapidamente outras O.T.A.s também passaram a aceitar apartamentos como forma de hospedagem. A Seazone utiliza O.T.A.s para gerar demanda para seus apartamentos.

2.1.2 Diária

A diária é o objeto vendido, através das O.T.A.s, pela seazone, para os hóspedes. Uma diária representa uma estadia de uma noite em um apartamento. Ela é considerada como um bem perecível, dado que uma vez passada a data da diária, ela não pode ser mais vendida. A diária possui um preço que pode mudar ao longo do tempo. A diária pode estar disponível ou indisponível para venda, a depender de vários critérios.

2.1.2.1 Reserva

A reserva representa a venda de uma diária, ou um conjunto de diárias, para um hóspede. A partir da venda, as diárias passam a estar indisponíveis e não podem ser vendidas novamente a menos que haja um cancelamento dessa reserva. A data de início da reserva é a data de checkin, e a data final é a data de checkout. É possível fazer checkin no mesmo dia do checkout de outra reserva no mesmo apartamento. Quando isso ocorre, dizemos que é uma "cama quente". Para fins de maximização de receita, a cama quente é ideal, pois não deixa nenhuma diária livre. Todavia, do ponto de vista da operação, a cama quente é difícil de executar pois requer muita agilidade na limpeza do apartamento.

2.1.2.2 Bloqueio

Um bloqueio representa uma diária indisponível, mas que não foi vendida. Isso pode ocorrer por diversos motivos, como manutenções no imóvel, restrições impostas pela pandemia, ou até uso do proprietário.

2.1.2.3 Políticas de Cancelamento

Ao oferecer uma reserva, também se explicita uma política de cancelamento. Na Seazone, hoje, optamos pela política rigorosa sempre, e embora não caiba neste documento, sabemos que a política de cancelamento pode influenciar na venda da diária, e portanto sua escolha é também relevante para se maximizar a receita. No Quadro 1 podemos ver as diferentes políticas de cancelamento que podemos oferecer no Airbnb.

2.1.3 Taxa de Limpeza

Também é comum que haja uma taxa de limpeza associada a cada reserva. Essa taxa é fixa por anúncio, e independe da quantidade de diárias vendidas na reserva.

2.1.4 Regras de calendário

É possível impor algumas regras no calendário de cada anúncio, a fim de maximizar a receita:

2.1.4.1 Bloqueio de checkin e checkout

Podemos impedir o checkin ou o checkout em dias específicos. É comum em Florianópolis, por exemplo, impedir o checkin no dia primeiro de janeiro, pois se hou-

Quadro 1 – Políticas de cancelamento do Airbnb

Flexível	Reembolso integral 1 dia antes da chegada
Flexível ou Não reembolsável	Além da política de cancelamento Flexível, ofereça uma opção não reembolsável — os hóspedes pagam 10% a menos, mas você mantém seu pagamento mesmo se eles cancelarem. Saiba mais
Moderada	Reembolso integral 5 dias antes da chegada
Moderada ou Não reembolsável	Além da política de cancelamento Moderada, ofereça uma opção não reembolsável — os hóspedes pagam 10% a menos, mas você mantém seu pagamento mesmo se eles cancelarem. Saiba mais
Rigorosa	Reembolso completo para cancelamentos feitos até 48 horas após a reserva, se faltar pelo menos 14 dias para a data de check-in. Reembolso de 50% para cancelamentos feitos pelo menos 7 dias antes do check-in. Sem reembolso para cancelamentos feitos até 7 dias antes da data de check-in.
Rigorosa ou Não reembolsável	Além da política de cancelamento Rigorosa, ofereça uma opção não reembolsável — os hóspedes pagam 10% a menos, mas você mantém seu pagamento mesmo se eles cancelarem.

Fonte – https://www.airbnb.com/help/topic/1359/payments-pricing-and-refunds

vesse um checkin nesse dia temos grandes chances da noite de reveillon – a diária mais valorizada do ano – ficar vazia.

2.1.4.2 Estadia Mínima

Também podemos exigir que, para que ocorra uma reserva, ela tenha uma duração mínima. Isso é comum em feriados, onde não desejamos que ocorra uma reserva de apenas um dia, impedindo que outro hóspede realize uma reserva pelo feriado inteiro.

2.1.4.3 Descontos por Duração da Estadia

Para incentivar estadias mais longas, podemos dar descontos que dependem do tamanho da estadia, por exemplo descontos de 10% para reservas de mais de 7 dias ou de 20% para reservas de mais de 30.

2.1.5 Antecedência

Quando uma reserva para o dia 20 de um mês é feita no dia 10 do mesmo mês, dizemos que ela tem uma antecedência de 10 dias. A antecedência é um fator importante para a probabilidade de ocupação, tendo em vista que, para reservam em Florianópolis, os hóspedes, principalmente em tempos de incerteza como a pandemia, tendem a fazer reservas com uma antecedência de 1 a 7 dias.

2.1.5.1 Desconto *Early Bird* e *Last Minute*

Caso haja o interesse em incentivar reservas com alta antecedência, podemos oferecer discontos "Early Bird", que serão aplicados apenas a reservas com checkin depois de x dias a partir de hoje. Por outro lado, se o interesse é de incentivar reservas com baixa antecedência, o desconto "Last Minute"permite alterar o preço apenas de diárias com checkin antes de x dias a partir de hoje.

2.1.6 Faturamento

Para o cálculo do Faturamento do imóvel, consideramos apenas os ganhos sobre as reservas. Ou seja, desconsideramos as taxas de limpeza, e desconsideramos quaisquer gastos. Portanto, definimos o faturamento f_n de uma diária d_n, como:

$$\begin{cases} p, & \text{se a diária foi alugada por um preço p} \\ 0, & \text{caso contrário} \end{cases}$$

Dessa forma, podemos definir um conjunto finito de diárias = $\{d_1, d_2, d_3, ..., d_n\}$, com faturamento $F = f_1 + f_2 + f_3 + \cdots + f_n$. Em um exemplo prático, podemos tomar D como sendo o conjunto de todas as diárias de Janeiro de 2021. Nesse caso, F seria o faturamento referente a todas as diárias do mês. Importante notar que, por essa definição, uma reserva pode contribuir para o faturamento de mais de um mês, se, por exemplo, começar no dia 15 de Janeiro e terminar no dia 15 de Fevereiro.

2.1.7 Airbnb

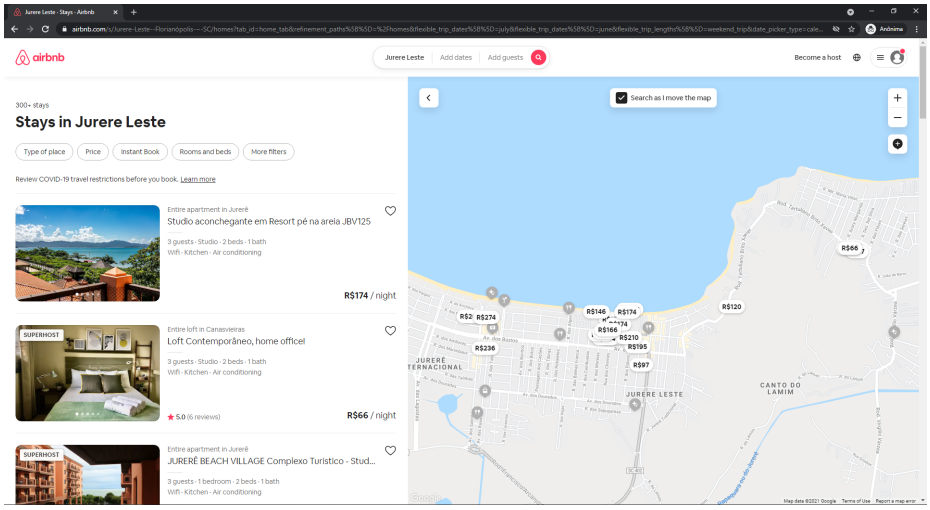
Como a maior parte das reservas feitas na Seazone ocorrem pelo Airbnb, vale a pena examinar algumas propriedades específicas dessa plataforma que são extremamente relevantes para a precificação.

2.1.7.1 Search Engine Optimization - S.E.O.

Ao pesquisar por uma localização no Airbnb, ele retornará até 300 anúncios, divididos em 10 páginas de 30.

É interessante para o Airbnb que os primeiros anúncios a serem mostrados no site sejam os anúncios que o hóspede tem maior chance de acessar e reservar. Para isso, existe um algoritmo de ranqueamento cujos parâmetros são mantidos em segredo pela empresa. Todavia, é razoável de se assumir que, se o objetivo é oferecer os anúncios mais competitivos, então anúncios baratos e com boa taxa de conversão (quantidade de clicks necessários para se gerar uma reserva) tendem a ser bem ranqueados. Embora não possa ser afirmada a causalidade, podemos examinar correlações entre variáveis do anúncio e seu ranqueamento. Um estudo feito na África do Sul, em 2017, levantou as 10 principais variáveis correlacionadas com o ranqueamento

Figura 1 – Print da página de busca do Airbnb



Fonte: Autor

de um anúncio. As 5 principais podem ser vistas na Figura 5. Podemos notar que a satisfação do hóspede é a variável com maior correlação, seguida pelo preço.

O impacto desse algoritmo nas estratégias de precificação é grande. Primeiramente, um anúncio barato terá mais chances de ser mostrado na primeira página e, portanto, mais chances de ser alugado. Ao ser alugado, esse anúncio receberá avaliações, que também colaboram com o seu ranqueamento. Se inicia um ciclo de feedback positivo, onde reservas melhoram o ranqueamento que gera mais reservas. O oposto pode ser verdade também: reviews negativos têm grande impacto no ranqueamento do anúncio, o que pode diminuir o fluxo de reservas e, portanto, impedir que novos reviews aconteçam.

2.1.7.2 Preço Inteligente

O Airbnb possui uma ferramenta de precificação inteligente, mas, paradoxalmente, recomenda que empresas que administram imóveis na plataforma não a utilize. De fato, existe um conflito de interesse entre o Airbnb e o proprietário do imóvel. O primeiro prefere apartamentos baratos que traga mais clientes para a plataforma, enquanto o faturamento do segundo depende não apenas do aluguel, mas também do preço alugado. Não é surpreendente, portanto, que a ferramenta de precificação inteligente do Airbnb sofra críticas por abaixar demasiadamente o preço das diárias.

Figura 2 – Tabela com as 5 variáveis mais correlacionadas com o ranqueamento de anúncios no Airbnb

Page #	# of listings	Guest Satisfaction	Price (USD)	Word Count	Minimum Nights	Days since Cal Update
1	109	83.7	$ 243.82	617	3.6	5.0
2	111	72.7	$ 281.39	538	4.3	13.2
3	106	72.6	$ 322.48	547	4.3	29.0
4	104	69.8	$ 356.59	447	4.8	25.9
5	101	65.4	$ 332.18	443	4.6	17.5
6	104	56.8	$ 339.46	390	4.8	27.5
7	109	59.8	$ 406.96	373	5.0	19.1
8	95	58.2	$ 427.70	380	4.9	22.5
9	99	52.1	$ 394.62	370	5.0	44.5
10	93	60.5	$ 327.96	364	4.8	38.6
11	92	49.3	$ 536.07	349	5.3	34.3
12	101	47.0	$ 408.89	301	5.3	46.1
13	105	45.2	$ 444.71	336	5.5	35.2
14	96	43.8	$ 542.61	290	5.5	49.3
15	84	57.5	$ 622.71	366	5.0	42.4
16	86	40.4	$ 559.47	270	5.5	59.6
17	43	42.9	$ 590.27	299	6.0	74.6
Grand Total	**1638**	**58.5**	**$ 407.63**	**401**	**4.9**	**32.2**

Fonte: Nicholas Child, Hacking Airbnb's search rank algorithm

2.2 REVENUE MANAGEMENT - HISTÓRICO E CONCEITOS

À pratica de mudar os preços e regras de venda a depender da situação, se dá o nome de Revenue Management. Nessa seção vamos explorar brevemente a história dessa prática, não apenas para entender de onde podemos buscar inspiração para a nossa solução, mas também trazendo contexto e motivação, apresentando outros setores onde esse projeto pode impactar.

2.2.1 História da precificação dinamica

A precificação dinâmica é tão antiga quanto o próprio comércio. A "pechincha", prática de negociar preços no varejo através de descontos é uma prática centenária, se não milenar. Todavia, o verdadeiro potencial dessa prática só foi revelado em 1985, quando a American-Airlines colocou no mercado o DINAMO.

2.2.1.1 DINAMO

Em no início da década de 1980, começaram a surgir nos Estados Unidos novas empresas no setor de transporte aéreo, focadas no trasporte de baixo custo. Elas ofereciam preços mais baixos e, assim, conseguiam captar uma boa parcela do

mercado. Um exemplo foi a People-Express, que nasce em 1981 para, já em 1984, ter um faturamento anual de US$ 60M.

A American Airlines sentiu no bolso a perda desse público, e focou seus esforços na criação do DINAMO, um software que seria capaz de alterar os preços das passagens em tempo real, baseado em variáveis de mercado. De repente, eram capazes de oferecer descontos para pessoas sensíveis ao preço e, ao mesmo tempo, manter o preço mais alto para o público que está disposto a pagar.

Em 1985, o DINAMO entra no mercado, e a People-Express vai de uma empresa lucrativa para perder mais de US$ 50M por mês. A People-Express, todavia, não mudou. Era a mesma empresa. Quem havia mudado – e capturado todo o mercado – foi a American Airlines.

Após a falência, o CEO da People-Express disse:

> Obviamente, a PeopleExpress falhou... Fizemos muitas coisas certas, mas não conseguimos solucionar os problemas de Yield Management e automação... [Se eu fosse fazer tudo de novo,] a minha prioridade número 1 todo dia seria que meus funcionários tivessem a melhor tecnologia da informação. Do meu ponto de vista, é isso que movimenta o faturamento do setor de aviação hoje, mais do que qualquer outro fator – mais do que serviço, mais do que aviões, mais do que rotas. (TALLURI; RYZIN, 2004, p. 9).

Hoje, não existe nenhuma empresa de aviação que não utilize de tecnologias semelhantes para a precificação de rotas. As tecnologias implementadas na American Airlines também não tardaram a entrar em outros mercados, notavelmente a indústria hoteleira tem muitas similaridades com o setor de aviação e foi um dos grandes utilizadores dessa nova tecnologia. A seguir, vamos explorar algumas das principais estratégias de Revenue Management utilizadas no mercado.

2.2.2 Estratégias de Revenue Management

Diferentes setores utilizam diferentes estratégias de Revenue Management, a depender de suas necessidades. Além do grande sucesso recente quando houve automação de obtenção de dados e mudanças de preços no setor de aviação, a teoria é aplicada também em hotelaria, aluguel de carros, no varejo, na geração e transmição de eletricidade, no armazenamento e transporte de gás natural, entre outros setores.

O que está sempre embutido nas estratégias de Revenue Management é a ideia de "vender para a pessoa certa, pelo preço certo". Ou seja, para que seja possível aplicar uma dessas estratégias é necessário que diferentes pessoas estejam dispostas a pagar diferentes preços pelo mesmo produto, o que é extremamente frequente. Fosse sabido quanto que cada cliente está disposto a pagar, a estratégia ideal seria vender por esse preço para cada um. Claro que isso, na prática, não é conhecido. Por isso, se buscam padrões de comportamento nos clientes para identificar quanto que cada um é sensível ao preço. No caso da indústria de aviação, um dos exemplos mais claros

é o preço do bilhete aumentando conforme a data se aproxima, pois o cliente que compra de última hora normalmente está disposto a pagar mais caro pelo assento, em comparação ao cliente que reservou o voo com 6 meses de antecedência.

2.2.2.1 Revenue Management Baseado em Quantidades

Revenue management baseado em quantidades utiliza o número de vendas como uma forma de realimentação para o preço. Para ser utilizado, é necessário que haja um inventário fixo a ser vendido. Esse é o caso na hotelaria, na aviação, no varejo, entre outros.

A diária em um hotel é considerada um bem perecível, como se estragasse após a data a ser vendida, afinal é impossível vender uma diária passada. Por isso, é interessante ao gerente do hotel tentar vender todos os quartos.

Suponha que, para uma data de interesse, um hotel de 100 quartos quer saber como precificar os quartos. Uma estratégia possível seria, para minimizar os riscos, vender metade dos quartos por um preço um pouco mais barato, digamos R$100,00. Estando vendidos esses quartos, podemos subir o preço para R$150,00 até que esteja vendido 80% dos quartos. Os últimos 20% podem ser vendidos a R$200,00. Dessa forma, idealmente, os clientes mais sensíveis ao preço tiveram chances de comprar suas diárias com maior antecedência por um valor menor, e hóspedes que estariam dispostos a comprar por qualquer preço puderam ser cobrados um pouco mais caro.

2.2.2.1.1 Overbooking

Se for sabido que 10% dos clientes vão cancelar uma reserva, é possível vender mais reservas do que temos disponíveis. Isso também é um risco, pois não havendo os cancelamentos previstos, mais de um cliente aparecerá esperando o mesmo quarto.

No Brasil, existe legislação que proíbe essa prática, mas mesmo assim ela ainda é recorrente na indústria de aviação. Na Seazone, não utilizamos overbooking para tentar maximizar receita, pois é vendido cada quarto individualmente, e não um quarto genérico em um hotel ou um assento qualquer no avião.

2.2.2.2 Revenue Management Baseado em Preço

Em outras situações, não é possível ou desejável fazer a precificação baseada em quantidades. Por exemplo, quando se tem apenas um exemplar a ser vendido. Nesses casos ainda é possível utilizar práticas de Revenue Management.

2.2.2.2.1 Precificação Dinâmica

Na precificação dinâmica, os preços mudam ao longo do tempo para se ajustar a flutuações na oferta e na demanda. Por exemplo, no varejo, no setor de roupas,

roupas de verão serão vendidas por um preço no verão, quando existe demanda, e ficarão mais baratas no inverno, quando a demanda diminuí. Assim, os clientes que estão dispostos a pagar mais pelas roupas podem comprar no verão, enquanto clientes que são mais sensíveis ao preço podem comprar no inverno.

Leilão

Leilões são muito utilizados para se maximizar o lucro de bens onde o preço ”justo” não é muito claro. Embora os mais conhecidos sejam leilões de peças de arte ou antiguidades, hoje é muito comum leilões online em sites de compra e venda como E-bay.

Existe uma quantidade considerável de pesquisa em teoria de leilões, incluindo o Nobel de 2020, dedicado a Paul Milgrom e Robert B. Wilson por melhorias na teoria do leilão e invenções de novos formatos de leilão.

Existem diferentes formatos de leilão possíveis. O que mais se aproxima de algo utilizável pelo Airbnb, é o "leilão holandês": O objeto é oferecido, primeiramente, por um preço acima do mercado, para o qual não é esperado que um comprador feche negócio. O preço então é sucessivamente diminuído até que haja um comprador interessado. Isso encerra o leilão, e esse comprador leva pelo preço em que demonstrou interesse.

No Airbnb, podemos oferecer uma diária primeiramente por um preço mais alto, e diminuir sucessivamente para que, assim que houver um interessado, ele compre a diária. Isso não é exatamente um leilão, pois não são as mesmas pessoas visualizando o anúncio todo dia, e nem mesmo a mesma quantidade de pessoas, mas é próximo o suficiente para que possa ser utilizado como estratégia de Revenue Management pela Seazone.

2.2.2.3 Microeconomia

Para escolher uma estratégia de Revenue Management, também é necessário estudar o mercado para entender como ele funciona. Assim, podemos modelar o problema de forma mais concreta, mesmo que fazendo uma ou outra simplificação.

Oferta e Demanda

Podemos definir uma curva de oferta de um produto como $O = o(p)$, onde O é a quantidade de pessoas dispostas a vender esse produto a um preço p. Similarmente, a curva de demanda desse produto é $D = d(p)$ como o número de pessoas dispostas a comprar esse produto pelo preço p.

Essas curvas são fundamentais para análises de mercados, e possuem algumas propriedades importantes:

- A curva de demanda é decrescente, pois o número de pessoas n_1d ispostas a comprar o produto pelo preço p_1 será sempre igual ou menor do que o número de pessoas n_2 dispostas a comprar por um preço p_2 se, e somente se, $p_1 gt; p_2$. Pela mesma lógica, a curva de oferta será sempre crescente.

- Podemos considerar que $o(0) \simeq 0$, e que $d(0) \simeq \infty$. Essas são, todavia, hipóteses simplificadoras e que devem ser utilizadas com cuidado. De fato o número de pessoas dispostas a comprar um produto por R$0,00 é muito grande, e a vender por R$0,00 é muito pequeno, mas, a depender do produto isso pode não ser verdade. Notavelmente, o preço de alguns barris de petróleo ficaram brevemente negativos em abril de 2020, pois com o início da pandemia a demanda caiu drásticamente e o custo de armazenamento do petróleo não compensava a compra – mesmo que de graça.

- Também podemos considerar que $\lim_{p\to\infty} o(p) = \infty$ e $\lim_{p\to\infty} d(p) = 0$.

- $o(p)$ e $d(p)$ são contínuas, pois a quantidade de pessoas dispostas a comprar ou vender por um preço $p + \epsilon$ é muito próxima da quantidade de pessoas dispostas a comprar ou vender por um preço p

- Segue dos itens anteriores que, pelo teorema do valor médio, essas duas curvas se intersectam em $o(p) = d(p)$. Quando a demanda e a oferta são iguais, o mercado é dito em equilíbrio e os preços tendem a ser constantes no tempo. Note que caso a curva de oferta ou de demanda mude, o ponto de equilíbrio também mudará.

Um mercado onde um produto está sendo oferecido a um preço p_1 tal que $o(p_1) gt; d(p_1)$ é um mercado ineficiente, pois a demanda é menor que a oferta e haverá excedente – mais pessoas procurando vender do que comprar. Dessa forma, alguns produtos não serão vendidos, sendo um potencial prejuízo.

Similarmente, se esse produto estivesse sendo oferecido a $p_2 < p_1$, tal que $o(p_2) < d(p_2)$, então o mercado também é ineficiente pois essa mesma quantidade de produtos ofertados poderia estar sendo vendida por um preço mais caro.

Por isso várias estratégias de Revenue Management procuram modelar as curvas de oferta e demanda, a fim de encontrar esse ponto de equilíbrio onde o mercado é eficiente.

Comportamento do Consumidor

Modelar o comportamento do consumidor também pode auxiliar em estratégias de Revenue Management. Um exemplo prático ocorre na indústria da aviação, onde

o segmento de consumidores que viajam a turismo tem um comportamento muito diferente daqueles que viajam a trabalho.

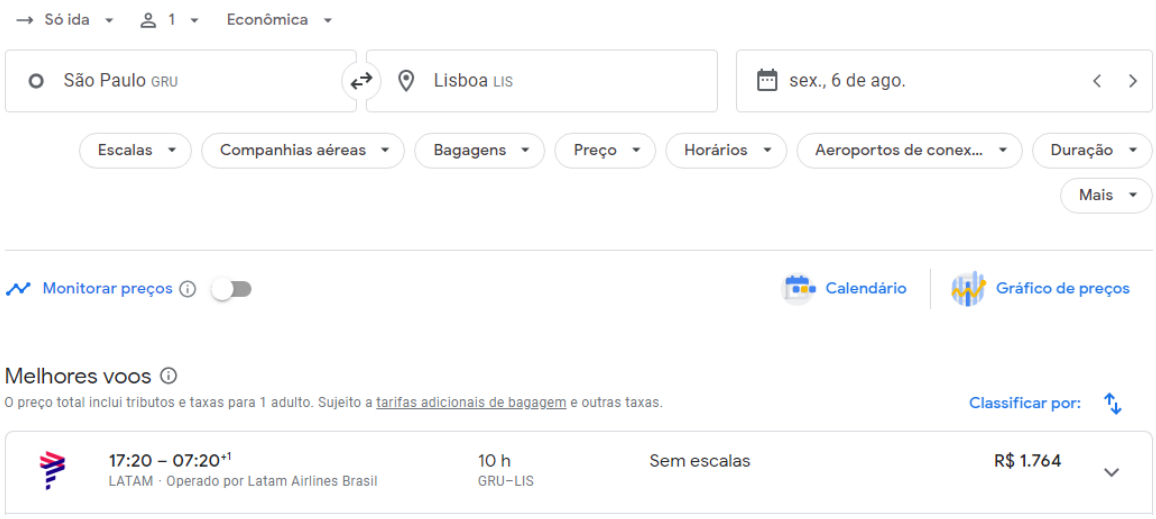
Por exemplo, uma pessoa que viaja numa sexta feira para retornar na segunda, provavelmente viaja a turismo, enquanto uma que viaja na segunda para retornar na sexta provavelmente viaja a trabalho.

Quem viaja a trabalho também tende a comprar as passagens em cima da hora, enquanto quem viaja com fins turísticos costuma se planejar com meses de antecedência.

Além disso, quem viaja a trabalho provavelmente é mais insensível ao preço – tendo em vista que quem paga a passagem tende a ser a empresa, e não o passageiro, enquanto quem viaja a turismo pode desistir facilmente da viagem a depender do preço da passagem.

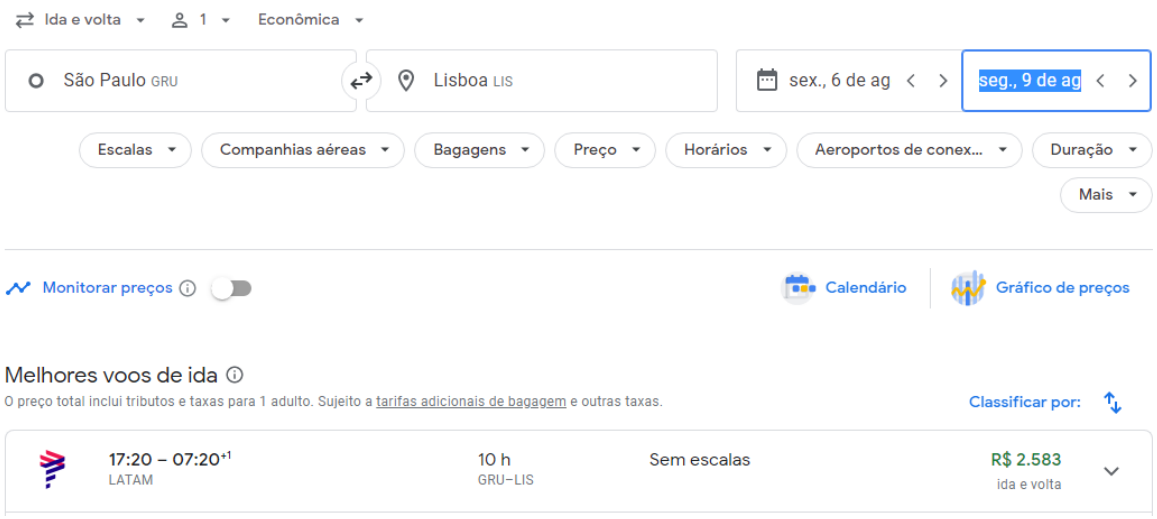
Por isso empresas aéreas tendem a mudar seus preços a depender de padrões de comportamento. Uma passagem comprada em cima da hora, para uma segunda com retorno na sexta, tende a ser muito mais cara do que uma passagem comprada com antecedência para um final de semana. Isso gera situações um pouco contra-intuitivas, onde passagens de ida e volta podem ter o mesmo preço, ou ser até mais barata, do que passagens só de ida, como podemos ver nas imagens abaixo:

Figura 3 – Preço de passagem de Guarulhos para Portugal, apenas ida



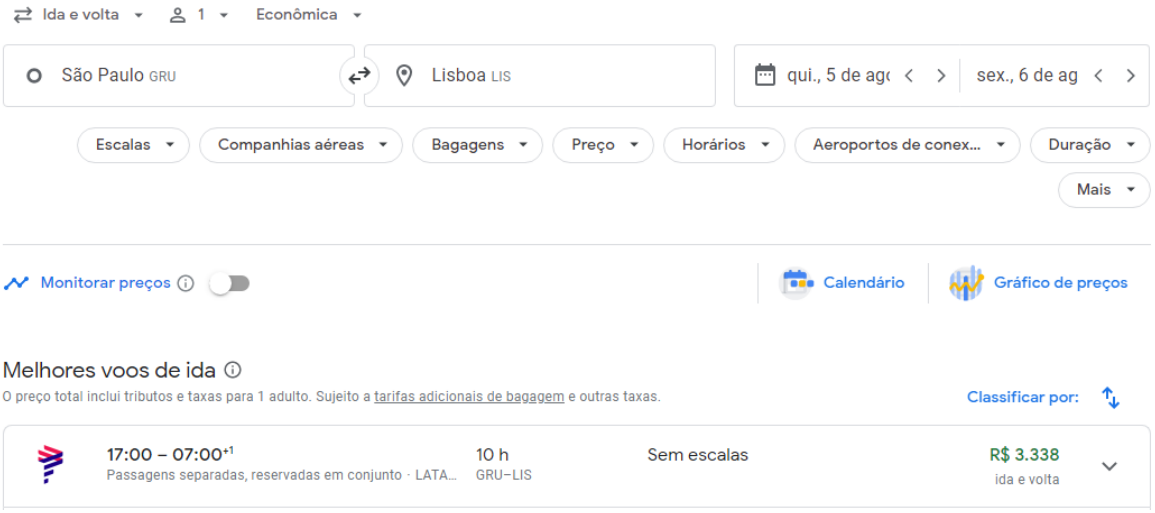
Fonte: Autor

Figura 4 – Preço de passagem de Guarulhos para Portugal, ida e volta em um final de semana



Fonte: Autor

Figura 5 – Preço de passagem de Guarulhos para Portugal, ida e volta durante a semana



Fonte: Autor

Competição

Quando temos um mercado onde um grande número de pessoas tenta vender um produto, para um grande número de compradores, e todos os vendedores estão

em situações muito simétricas – ou seja, o produto ou serviço oferecido por eles é muito parecido – dizemos que é um mercado de competição perfeita. Nesse cenário, os preços oferecidos entre todos os vendedores tende a ser muito próximo, pois o cliente pode trocar por um concorrente sem nenhum custo, e os produtos são similares. É importante que, para ser caracterizado como um mercado de competição perfeita, os compradores tenham conhecimento de vários competidores, e facilidade para compará-los.

Em um outro extremo temos uma situação de monopólio. Nesse mercado, temos apenas um vendedor, que tem liberdade de escolha de preços. É comum que, em situações de monopólio, o estado intervenha para facilitar a competição, ou para evitar o abuso de poder da empresa com monopólio.

É importante notar que existe todo um espectro entre mercados de competição perfeita e mercados de monopólio. Mercados onde existem poucos vendedores ou onde o acesso a diferentes vendedores não é fácil. Quando apenas algumas poucas empresas dominam a maior parte do mercado, dizemos que se trata de um Oligopólio. Nesses casos a dinâmica de preços é bem complicada, e precisamos de um pouco de Teoria dos Jogos para tentar modelar.

No caso do Airbnb, o acesso a diferentes vendedores é muito fácil, pois estaremos sempre anunciando na mesma plataforma que nossos competidores. Todavia, os produtos oferecidos tendem a ser bem diferentes. Dois apartamentos de dois quartos, no mesmo bairro, podem ter preços ótimos completamente diferentes, a depender de características do apartamento – por exemplo, tem vista para o mar? Geladeira de duas portas ou frigobar?

Além disso, apesar da alta oferta de apartamentos, na prática apenas os apartamentos das primeiras páginas são facilmente visualizados. Por isso, caracterizamos nosso mercado como um Oligopólio, onde os competidores são os apartamentos nas primeiras páginas com características parecidas.

Teoria dos Jogos

Teoria dos Jogos é o campo da matemática que estuda situações estratégicas, onde jogadores escolhem diferentes ações em um jogo com regras pré definidas. É muito aplicado em economia, onde podemos definir regras de jogos e modelar o comportamentos complexos de jogadores através de premissas mais simples – como, por exemplo, que o comprador vai sempre procurar a alternativa mais barata.

Um exemplo clássico de teoria dos jogos é o dilema do prisioneiro. Dois suspeitos de um crime estão sendo interrogados, e são oferecidos o seguinte acordo: caso denunciem seu parceiro, terão sua pena de 2 anos reduzida para 1 ano, enquanto o parceiro terá a pena aumentada para 10 anos. Caso ambos se denunciem, todavia, a pena será de 5 anos para os dois. Todavia nesse caso os prisioneiros não podem coo-

perar, e não sabem se seu parceiro aceita ou recusa o acordo. Nesse caso podemos desenhar a seguinte matriz de Nash:

Matriz de nash	Prisioneiro "A"recusa acordo	Prisioneiro "A aceita o acordo
Prisioneiro "B"recusa acordo	Ambos são condenados a 2 anos	"A"é condenado a 1 ano, enquanto "B"a 10
Prisioneiro "B"aceita acordo	"A"é condenado a 10 anos, enquanto "B"a 1	Ambos são condenados a 5 anos

Dizemos que essa matriz possui equilíbrio de Nash competitivo (a resposta esperada de jogadores lógicos) quando ambos aceitam o acordo, pois ambos se beneficiam de denunciar seu colega independente da cooperação dele. Note que, caso ambos os prisioneiros fossem capazes de cooperar, poderíamos definir um equilíbrio de Nash cooperativo onde ambos recusam o acordo, diminuindo a pena total de 20 para 4 anos.

Esse conceito é importante para definir se nossos apartamentos estarão competindo entre si, ou cooperando. Suponha que, para Janeiro, a Seazone possúi os dois últimos apartamentos no Airbnb, que podem manter ou subir seu preço. Caso o preço de um apartamento suba, a demanda tenderá a ir para o apartamento mais barato. Todavia, ambos podem subir o preço juntos e aumentar seu faturamento em conjunto. A matriz de Nash resultante é:

Matriz de nash	Apto 1 mantém preço	Apto 1 sobe o preço
Apto 2 mantém o preço	Fat 1: R$1000 Fat 2: R$1000	Fat 1: R$500 Fat 2: R$3000
Apto 2 sobe o preço	Fat 1: R$3000 Fat 2: R$500	Fat 1: R$2000 Fat 2: R$2000

Nesse caso, no equilíbrio de Nash cooperativo, ambos sobem o preço para faturar mais no total. Todavia, em um equilíbrio competitivo ambos mantém o preço pois subir significa diminuir seu faturamento, independente da escolha do outro apartamento. Isso está altamente atrelado à dinâmica de mercados perfeitamente competitivos – que mantém seu preço mais baixo, em oposição a mercados monopolistas onde o preço tende a subir. No escopo desse trabalho, estaremos considerando apenas o equilíbrio competitivo, pois, pelo modelo de negócio da empresa, diminuir o faturamento de um apartamento ”A” em prol do faturamento total pode levar ao proprietário do apartamento ”A” a remover seu apartamento da administração da Seazone.

3 FUNDAMENTAÇÃO TEÓRICA E MODELAGEM

Nesse capítulo vamos construir sobre o contexto dado no capítulo anterior, e buscar uma modelagem do problema que permita uma solução elegante.

3.1 MODELAGEM DO FATURAMENTO

Como nosso objetivo é maximizar o faturamento, o primeiro passo é encontrar uma função que descreva o faturamento do imóvel, com base nas variáveis que temos controle (como preço, estadia mínina, etc.).

Vamos partir das seguintes hipóteses simplificadoras, que podem ser questionadas em futuras análises:

- O tempo é discreto em dias. Isso facilitará a implementação de uma solução pois o “Sample Time” da scrappagem (ato de obter dados de websites) do Airbnb é de 1 dia. Caso haja algum sistema capaz de detectar mudanças no Airbnb instantaneamente, as equações de diferenças aqui tratadas serão equações diferenciais ordinárias.
- Para todo apartamento, para toda diária, existe um preço máximo P_{max} a partir do qual ninguém estará disposto a comprar essa diária. Isso é dado pela curva de demanda discutida no capítulo anterior.
- Como segue da curva de demanda que sempre existem mais pessoas dispostas a comprar por um preço p_1 do que p_2 se, e somente se, $p_1 < p_2$, podemos deduzir que a probabilidade de alugar um apartamento por um preço p dada por $Prob_{oc}(p)$ é $Prob_{oc}(p_1) > Prob_{oc}(p_2)$
- Cada data é vendida individualmente. Ou seja é possível calcular o faturamento de cada data independentemente. Isso é falso, mas acarreta em uma grande simplificação do problema, ainda mantendo boa aproximação da realidade.

3.1.1 Da existência de um preço ótimo

Seja o faturamento esperado p para a data n, no dia $n-1$, dado pela fórmula:

$$Fat(p, a, c) = p * Prob_{oc}(p, a, c) \tag{1}$$

onde p é o preço, a é o apartamento e c é um conjunto de variáveis que não temos controle (previsão do tempo, final de semana, feriados, etc.). Note que a probabilidade de ocupação é decrescente no preço, e tende a 0 quando o preço tende a P_{max} Nesse caso,temos:

$$\begin{aligned} Fat(P_{max}, a, c) = P_{max} * Prob_{oc}(P_{max}, a, c) = 0 \\ Fat(0, a, c) = 0 * Prob_{oc}(0, a, c) = 0 \end{aligned} \tag{2}$$

Portanto, se existe algum preço para o qual alguém compraria a diária, então a função $F(p, a, c)$ possui pelo menos um ponto máximo para cada apartamento e cenário. Como a função $Prob_{oc}(p, a, c)$ é decrescente, então o conjunto de preços que maximizam o faturamento para um dado apartamento e cenário é conexo.

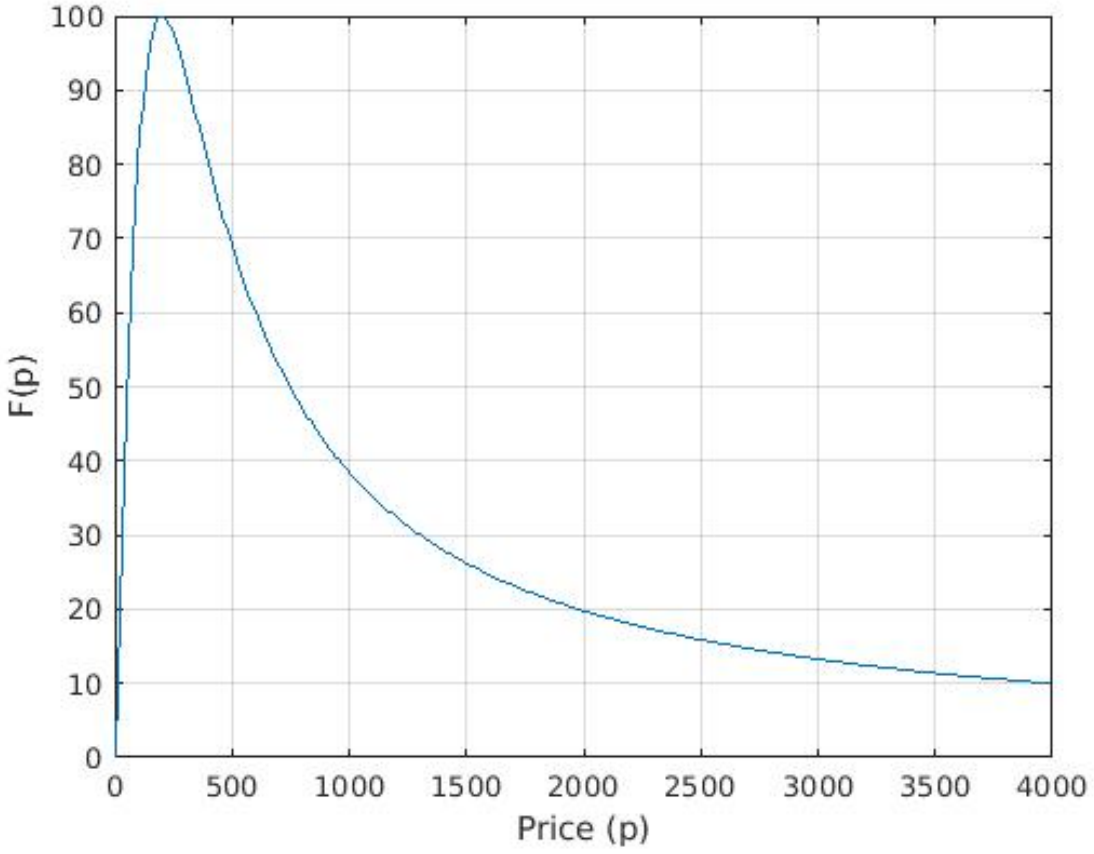
Em um exemplo concreto, suponha que temos um apartamento disponível para amanhã. Isso significa que precisamos vendê-lo hoje se esperamos receber algo. Todavia, não queremos vender por um preço muito barato. Suponha que a equação que descreve a probabilidade de ocupação do imóvel para um determinado apartamento e cenário seja:

$$Prob_{oc}(p) = \frac{1}{(\frac{p}{200})^2 + 1}$$

Então a minha função de faturamento será:

$$F(p) = \frac{p}{(\frac{p}{200})^2 + 1}$$

Figura 6 – Gráfico do faturamento em função do preço ofertado



Fonte: Autor

Que possui máximo quando $p = 200$, com a probabilidade de ocupação de 50% e faturamento esperado de R$100.

3.1.2 Da existência de um caminho ótimo de preços

Sabemos que existe um preço ótimo para o dia $n-1$, com faturamento esperado F_{n-1}. Nesse caso, basta provar que se é sabido o faturamento esperado no dia $n-m-1$ então existe um preço ótimo para o dia $n-m$, que está provado por indução finita que existe um caminho ótimo de preços $P_{n-m}, P_{n-m-1}, \ldots, P_{n-1}$ que maximiza o faturamento esperado.

Nesse caso, o faturamento esperado F_{n-m} é dado pela fórmula:

$$F_{n-m}(p, a, c) = p * Prob_{oc}(p) + F_{n-m-1} * (1 - Prob_{oc}(p))$$

Em palavras, o meu faturamento esperado de uma data n, com antecedência m é a chance de alugar essa data hoje pelo preço de hoje ($p * Prob_{oc}(p)$), mais o meu faturamento esperado de amanhã, se eu não alugar hoje [$F_{n-m-1} * (1 - Prob_{oc}(p))$]. Note que, para um apartamento em um cenário, podemos calcular o faturamento esperado de P_max e F_{n-m-1} para mostrar que são iguais:

$$F_{n-m}(P_{max}) = P_{max} * 0 + F_{n-m-1} * (1 - 0) = F_{n-m-1}$$

$$F_{n-m}(F_{n-m-1}) = F_{n-m-1} * Prob_{oc}(F_{n-m-1}) + F_{n-m-1} * (1 - Prob_{oc}(F_{n-m-1})) = F_{n-m-1}$$

Portanto, se existe algum preço $P_{max} > P > F_{n-m-1}$ então a função possui máximo. Isso é equivalente a dizer que se $P_{max} > F_{n-m-1}$ a função possuirá máximo, o que será sempre verdade dado que existe algum preço pelo qual alguém está disposto a comprar a diária. Fica provado então, por indução finita, que existe um caminho ótimo de preço, como queríamos demonstrar.

Um exemplo didático para explicar o que está acontecendo seria um jogo com dados, composto das seguintes regras:

- O jogador escolhe um número de 1 a 6. Essa escolha será equivalente a escolha do preço do apartamento.
- O jogador então joga um único dado. Caso o valor do dado seja estritamente maior que o valor escolhido, o jogo acaba e o jogador pontua o número que escolheu. Isso é equivalente ao apartamento alugar pelo preço escolhido.
- O jogo tem um número máximo de 5 rodadas. Se após 5 rodadas o jogador não conseguir nenhum ponto, ele fica com pontuação 0.

Nesse jogo, caso houvesse duração máxima de 1 rodada, a melhor escolha de número para o jogador seria 3, onde existe uma probabilidade de 50% dele conseguir 3 pontos, ou seja, uma expectativa de 1.5 pontos. Todavia, como o jogo proposto tem 5 rodadas, escolher o número 3 todas as 5 rodadas teria um retorno esperado de apenas 2.9, enquanto escolher o número 4 teria um retorno de 3.47. Como o jogador pode mudar o número escolhido em cada rodada, ele pode tentar começar com números de maior risco - como o 5 - e gradualmente abaixar o número para garantir um retorno mais seguro. A tabela com as jogadas ótimas ressaltadas em verde pode ser vista abaixo:

valor dado	1	2	3	4	5	6
prob >	83,33%	66,67%	50,00%	33,33%	16,67%	0,00%
retorno	1	2	3	4	5	6
retorno t-5 escolha constante	0,9998713992	1,991769547	2,90625	3,473251029	2,99061214	0
retorno esperado t-1	0,8333333333	1,333333333	1,5	1,333333333	0,8333333333	0
retorno esperado t-2	1,083333333	1,833333333	2,25	2,333333333	2,083333333	1,5
retorno esperado t-3	1,222222222	2,111111111	2,666666667	2,888888889	2,777777778	2,333333333
retorno esperado t-4	1,314814815	2,296296296	2,944444444	3,259259259	3,240740741	2,888888889
retorno esperado t-4	1,37654321	2,419753086	3,12962963	3,50617284	3,549382716	3,259259259

3.2 MODELAGEM DA OCUPAÇÃO

Encontradas as fórmulas de faturamento, fica claro que precisamos de um modelo que seja capaz de encontrar a probabilidade de ocupação para um dado preço, apartamento e cenário. Isso feito podemos substituir os resultados do modelo na equação de diferenças do faturamento e assim encontrar o caminho de preços ótimo para uma data.

3.2.1 Aprendizado de máquina para modelagem

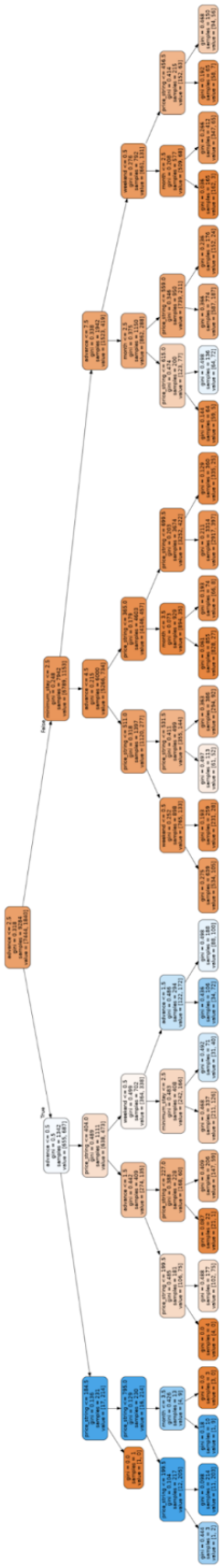
Quando é necessário modelar uma função baseada em dados passados, a abordagem mais comum é a de aprendizado de máquina supervisionado. Nessa subseção vamos abordar diferentes métodos de aprendizado de máquina.

3.2.1.1 Árvores de Decisão

Árvores de decisão é um método de aprendizado de máquina supervisionado, baseado em árvores binárias de decisão. A árvore é composta por nós, galhos e folhas. Um nó é um ponto de intersecção do grafo, caracterizado por uma decisão (daí o nome árvore de decisão). Em cada nó a árvore se divide em dois galhos. Um galho que não se divide mais é chamado de uma folha. A árvore é construída baseada em todo o conjunto de dados, com o objetivo de separá-los em conjuntos homogêneos de variável de saída nas folhas. No nosso caso, a variável de saída é booleana (1 ou 0) e indica se um quarto foi ou não alugado. As variáveis de entrada compreendem todas as variáveis medidas que podem afetar a locação do imóvel, como antecedência, dia da semana,

mês, etc. As decisões dos nós são como perguntas baseadas nas variáveis de entrada, como *"é fim de semana?"* ou *"estamos em janeiro?"*. Com isso, cada decisão separa os dados em dois conjuntos - o conjunto que respondeu afirmativamente e o conjunto negativo.

Figura 7 – Pequena árvore de decisão, gerada sobre os dados de ocupação de imóveis no AirBnb



Fonte: Autor

Na Figura 7 cada nó está colorido de acordo com a probabilidade de ocupação. Ou seja, a partir da primeira decisão, uma pergunta foi feita e os dados foram separados em dois conjuntos. O da esquerda, pintado em branco, respondeu positivamente para a pergunta e tem probabilidade de ocupação próxima de 50%, enquanto o da direita respondeu negativamente e tem probabilidade de ocupação próxima de 0. Isso significa que diárias que respondem positivamente para essa pergunta tem chance maior de ocupar do que diárias que respondem negativamente. Esse processo se repete quantas vezes necessário, de acordo com a escolha de hiper-parâmetros que serão discutidos mais a frente no texto. A partir dessas perguntas, essa árvore simples já conseguiu identificar conjuntos de diárias que tem probabilidade de ocupação próxima de um, depois de responder 5 perguntas.

3.2.1.2 Random Forest

Uma *"Random Forest"* é um método de aprendizado de máquina supervisionado, baseado em várias árvores de decisão geradas em diferentes subconjuntos de dados, tomando a média do resultado dessas árvores. Esse método é mais complexo, e demanda mais recursos tanto em tempo de treinamento quanto em espaço em disco, mas tende a produzir melhores resultados do que utilizar apenas uma árvore de decisão, principalmente por evitar *"overfitting"*.

Overfitting ocorre quando o modelo obtido está adequado apenas ao conjunto de treinamento, e seria inadequado em outros conjuntos. Isso é bastante comum em árvores de decisão com alto número de perguntas consecutivas que levam a folhas de subconjuntos muito pequenos. Como a random forest utiliza diferentes árvores de decisão em diferentes subconjuntos, efeitos de overfitting de uma árvore não geram um overfit da floresta inteira.

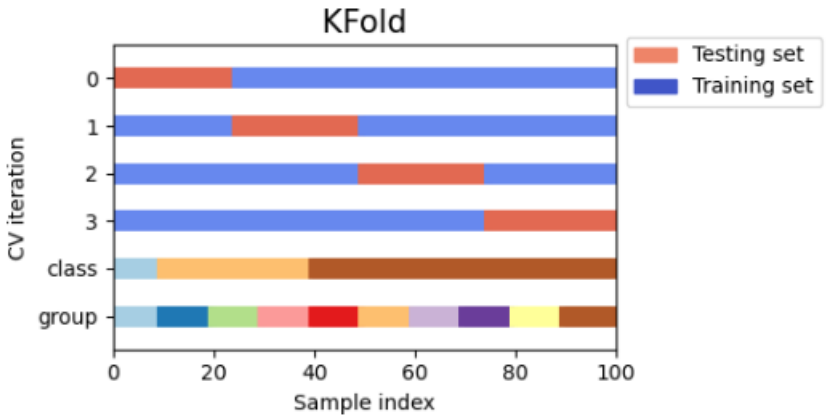
3.2.2 Cross Validation

Para validar se o modelo obtido tem poder preditivo – ou seja, se ele pode ser aplicado a dados nunca antes vistos e mesmo assim ter um bom desempenho – é necessário treinar o modelo com um conjunto de dados, chamado de conjunto de treinamento, e testar o modelo em um conjunto diferente, denominado de conjunto de teste. Esse processo é chamado de *Cross Validation*. Existem diferentes formas de separar os dados em diferentes subconjuntos, cada uma com seus prós e contras.

A K-fold Cross-Validation separa os dados da seguinte forma:

Ou seja, em uma primeira iteração de testes, as primeiras $\frac{1}{n}$ linhas serão utilizadas como conjunto de testes, e o resto será utilizado como conjunto de treinamento. Para cada iteração de testes, o conjunto de treinamento é alterado para as $\frac{1}{n}$ linhas seguintes, até o enésimo teste. Em todos estes testes serão obtidas diferentes métricas de desempenho (que serão discutidas mais a frente no texto) que podem ser compara-

Figura 8 – K-fold Cross validation

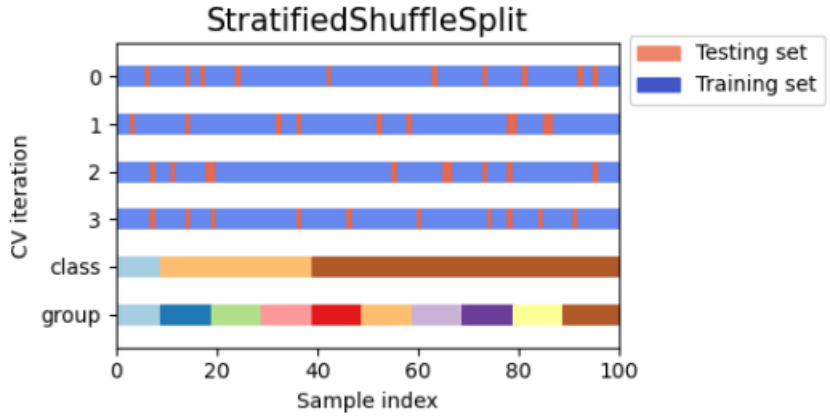


Fonte: (PEDREGOSA *et al.*, 2011)

das com diferentes algoritmos de aprendizado de máquina, ou diferentes escolhas de hiperpârametros, para determinar qual o ideal para o seu projeto. Um dos problemas desse método está no fato das linhas poderem estar ordenadas de acordo com, por exemplo, tempo. Nesse caso, seria possível escolher como dados de teste todo o mês de janeiro, por exemplo, e treinar o modelo em dados que não continham o mês de janeiro, o que não seria ideal.

A Stratified Shuffle Split Cross-Validation busca resolver esse problema, separando os dados da seguinte forma:

Figura 9 – Stratified Shuffle Split Cross validation



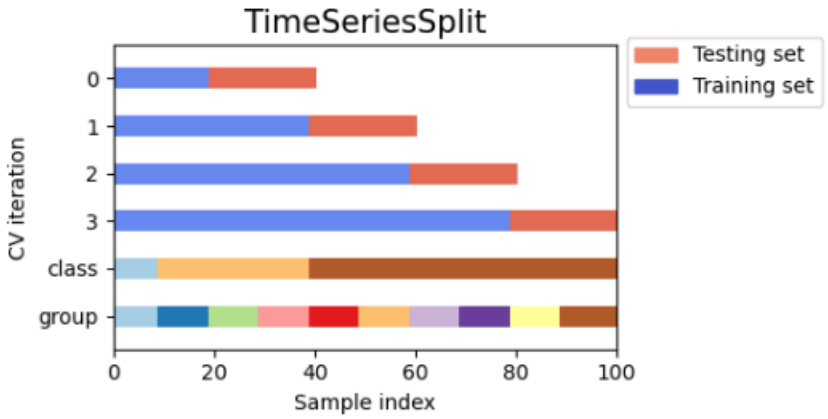
Fonte: (PEDREGOSA *et al.*, 2011)

Ou seja, garantindo que cada conjunto de treinamento seja homogêneo em

características de entrada e saída. No nosso caso, isso significaria que em todos os testes nós estaríamos utilizando aproximadamente o mesmo número de diárias de janeiro em todos os testes feitos.

Embora esses métodos de Cross Validation sejam estremamente adequados para avaliar o desempenho de diferentes métodos de aprendizado de máquina, eles não indicam como a performance desses métodos melhora com o crescimento da base de dados. Para isso se utiliza o Time Series Split, que separa os dados da seguinte forma:

Figura 10 – Time Series Split Cross validation



Fonte: (PEDREGOSA *et al.*, 2011)

Aqui queremos determinar quanto que nosso modelo melhora conforme aumentamos os dados no nosso conjunto de treinamento. Ordenamos nosso conjunto de dados por data, e treinamos, por exemplo, o modelo com os dados de janeiro de 2021 para predizer os dados de fevereiro. Depois utilizamos janeiro e fevereiro para predizer março, e assim em diante.

3.2.3 Métricas de avaliação do modelo

Um algoritmo de aprendizado de máquina supervisionado busca encontrar uma função que correlacione variáveis de entrada com variáveis de saída. O quão bem essa função aproxima a realidade pode ser mensurado de acordo com métricas de avaliação desse modelo. Existem diferentes métricas que buscam mensurar a performance em diferentes aspectos, como taxa de falso positivos, falso negativos, etc. Essas métricas podem ser organizadas em uma matriz de confusão:

Figura 11 – Matriz de confusão

		Predicted condition		Sources: [13][14][15][16][17][18][19][20] view · talk · edit	
	Total population = P + N	**Positive (PP)**	**Negative (PN)**	Informedness, bookmaker informedness (BM) = TPR + TNR − 1	Prevalence threshold (PT) = $\frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Positive (P)**	**True positive (TP)**, hit	**False negative (FN)**, type II error, miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power = $\frac{TP}{P}$ = 1 − FNR	False negative rate (FNR), miss rate = $\frac{FN}{P}$ = 1 − TPR
	Negative (N)	**False positive (FP)**, type I error, false alarm, overestimation	**True negative (TN)**, correct rejection	False positive rate (FPR), probability of false alarm, fall-out = $\frac{FP}{N}$ = 1 − TNR	True negative rate (TNR), specificity (SPC), selectivity = $\frac{TN}{N}$ = 1 − FPR
	Prevalence = $\frac{P}{P+N}$	Positive predictive value (PPV), precision = $\frac{TP}{PP}$ = 1 − FDR	False omission rate (FOR) = $\frac{FN}{PN}$ = 1 − NPV	Positive likelihood ratio (LR+) = $\frac{TPR}{FPR}$	Negative likelihood ratio (LR−) = $\frac{FNR}{TNR}$
	Accuracy (ACC) = $\frac{TP+TN}{P+N}$	False discovery rate (FDR) = $\frac{FP}{PP}$ = 1 − PPV	Negative predictive value (NPV) = $\frac{TN}{PN}$ = 1 − FOR	Markedness (MK), deltaP (Δp) = PPV + NPV − 1	Diagnostic odds ratio (DOR) = $\frac{LR+}{LR-}$
	Balanced accuracy (BA) = $\frac{TPR+TNR}{2}$	F_1 score = $\frac{2PPV \times TPR}{PPV+TPR}$ = $\frac{2TP}{2TP+FP+FN}$	Fowlkes–Mallows index (FM) = $\sqrt{PPV \times TPR}$	Matthews correlation coefficient (MCC) = $\sqrt{TPR \times TNR \times PPV \times NPV}$ − $\sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI) = $\frac{TP}{TP+FN+FP}$

Fonte: Wikipedia https://en.wikipedia.org/wiki/Confusion_matrix

A métrica mais simples da matriz de confusão é accuracy (exatidão), que consiste na porcentagem de acertos que o modelo obteve. Ou seja, se de 100 samples o modelo identificou corretamente 80 delas, a accuracy desse modelo é de 80%. Todavia essa não é uma medida boa para conjuntos de dados não balanceados, onde a porcentagem de positivos é diferente da porcentagem de negativos. Para corrigir isso é possível utilizar a balanced accuracy (exatidão balanceada), dada pela fórmula:

$$PB = \frac{Se + Sp}{2}$$

Onde:

$$Se = \frac{VP}{VP + FN}$$

$$Sp = \frac{VN}{VN + FP}$$

Sendo que VN são os verdadeiros negativos, FP os falsos positivos, VP os verdadeiros positivos e FN os falso negativos, e Se denota a sensitividade do modelo, enquando Sp denota a Specificidade. Intuitivamente, o que está sendo feito é tirando a média de quantos acertos foram obtidos em conjuntos onde o verdadeiro resultado era positivo, e quantos acertos foram obtidos em resultados onde o verdadeiro resultado era negativo. Se em 100 samples o modelo identifica corretamente 80 delas, todas negativas, mas o modelo falha em identificar outras 20 positivas, o balanced accuracy desse modelo é então 50% ((80/(80+0) + 0/(20+20))/2). Tando a specificidade quando a sensibilidade também podem ser utilizadas como métricas de avaliação do modelo.

Se desejamos minimizar os erros de tipo 1 (Falso positivo) e erros de tipo 2 (Falso negativo), então desejamos que se a specificidade OU a sensibilidade do

modelo sejam 0, então que a nossa métrica objetivo também seja 0. Ao mesmo tempo, se a especificidade e a sensibilidade são 1, então desejamos que a nossa métrica objetivo seja 1. Uma solução possível é tirar a média harmônica entre a especificidade e a sensibilidade, dada por:

$$F1 = 2\frac{Se * Sp}{Se + Sp}$$

Essa métrica é comumente conhecida como F1, e é uma das mais utilizadas para avaliar o desempenho de modelos.

3.2.4 Dummy Estimators

O valor em si das métricas de avaliação não necessariamente indica um bom modelo. Por exemplo, quando temos uma base de dados não balanceada, onde de 100 linhas, temos 90 com saída 1 e apenas 10 com saída 0, por exemplo, um modelo que sempre prediz uma saída 1 estará certo 90% das vezes. Isso não quer dizer que esse é um bom modelo para se utilizar. Por isso, quando testamos um modelo contra uma base de dados é uma boa prática testar também um Dummy Estimator, ou seja, um modelo "burro"que não foi treinado nos dados. Um exemplo de Dummy estimator seria um modelo que prediz sempre o mesmo valor de saída. Outro exemplo seria um que prediz sempre valores aleatórios. Assim podemos ter uma ideia de quanto que o modelo treinado é "melhor"do que um modelo "burro".

3.2.5 Escolha de hiper-parâmetros

Ao construir uma árvore de decisão, é necessário realizar a escolha de alguns parâmetros da árvore, como quantas folhas são desejadas? Qual o número máximo de nós da árvore? Qual o valor mínimo de samples que deve estar contido em uma folha? A escolha desses parâmetros vai alterar drasticamente o comportamento do modelo. Árvores com um grande número de nós e folhas, que contêm poucos samples por folha, tendem a overfittar o conjunto de dados de treinamento, enquanto árvores com poucos nós tendem a ter pouco poder preditivo.

Existem diferentes hiper-parâmetros em diferentes algoritmos de aprendizado de máquina. Em redes neurais é possível escolher o número de camadas e o número neurônios por camada. Em random forests, além de determinar os hiperparâmetros das árvores contidas na floresta, também é necessário especificar o número de árvores.

Não existe fórmula ou regra para a escolha desses paramêtros, pois escolhas que funcionam bem em um conjunto de dados podem não funcionar para outro. Existem alguns métodos para a escolha, onde hiper-parâmetros são testados e iterativamente alterados na busca de otimizar uma dada métrica.

Como esse processo é extremamente complicado, existem ferramentas de AutoML, onde o usuário apenas especifíca um tempo de treinamento e um limite de memória RAM para o algoritmo e ele retorna uma escolha otimizada de hiperparâmetros.

3.2.6 Análise de resultados

Após todo o processo de validação do modelo, escolha de hiper-parâmetros para otimizar métricas de desempenho, o modelo obtido é uma caixa preta, ou seja, não é possível identificar claramente qual o efeito das variáveis de entrada nas variáveis de saída. Mesmo que o modelo funcione em dados passados stakeholders podem não confiar na solução. Para explicar como funciona um modelo caixa preta existem Explainers.

Um exemplo de uso de algoritmos de aprendizado de máquina é na identificação de objetos em imagens. Suponha um algoritmo que foi treinado para separar imagens de gatos e cachorros. Se as fotos de cachorro foram obtidas em um parque, e as de gato em um apartamento, é possível que o algoritmo esteja separando corretamente as fotos com base no fundo, e não no animal. Por exemplo, se o fundo é majoritariamente verde, então provavelmente se trata de uma foto de cachorro, enquanto um fundo cinza se trata de uma foto de gato. Como descobrir quais pixels estão sendo utilizados para tomar essa decisão? Estudamos duas abordagens, LIME e SHAP.

3.2.6.1 LIME

LIME é sigla para Local Interpretable Model-agnostic Explanations. A idéia é fazer uma aproximação linear da função no ponto e encontrar as derivadas parciais com relação as diferentes variáveis de entrada. Em outras palavras, quanto varia o valor da probabilidade de existir um cachorro ou um gato nessa imagem, conforme eu altero diferentes pixels? O resultado é que quando se remove um gato da imagem, deve-se observar grande variação na variável de saída, enquanto quando se remove uma árvore ou uma luminária, isso não deve alterar a saída.

Figura 12 – LIME identificando um gato e um cachorro em uma imagem

Fonte: (RIBEIRO; SINGH; GUESTRIN, 2016)

3.2.6.2 SHAP

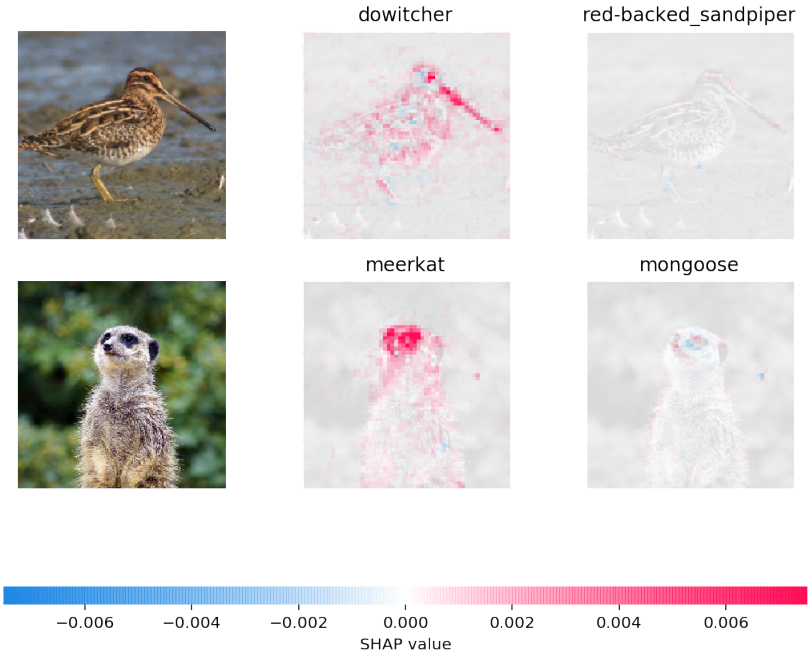
SHAP é sigla para SHapley Additive exPlanations, e busca trazer uma abordagem de teoria dos jogos para determinar o impacto de cada variável na saída do modelo. O SHAP vai considerar que o modelo funciona como um jogo cooperativo, onde cada variável de entrada é um jogador e a variável de saída é o resultado do jogo. O objetivo do algorítimo é definir quanto que cada jogador impactou no resultado do jogo.

Suponha um jogo cooperativo com n jogadores, com um resultado R. O Shapley Value S_n de um jogador representa o quanto que esse jogador impactou no resultado do jogo. Como o resultado do jogo é a soma dos impactos de cada jogador, temos:

$$S_1 + S_2 + \cdots + S_n = R$$

Podemos considerar que cada variável em uma linha no modelo é um jogador que contribui para um resultado: a predição do modelo. Podemos então calcular o Shapley Value de cada uma das variáveis para cada uma das linhas, a fim de entender o impacto das variáveis na predição do modelo. Existe uma fórmula para o Shapley Value, mas aqui basta a intuição de que ele representa o valor agregado por cada variável de entrada na saída do modelo. Ou seja, se a saída predita pelo modelo é 1, e temos 3 variáveis, a, b e c, e o Shapley Value delas é 0.7, 0.5, -0.2 respectivamente, então sabemos que a variável que mais impactou nessa decisão foi a, e que o valor da variável c teve um impacto negativo, ou seja, está ”tentando levar” a saída para 0.

Figura 13 – SHAP identificando uma ave e um suricato em duas imagem



Fonte: (LUNDBERG; LEE, 2017)

4 PROPOSTA DE SOLUÇÃO

Sabendo que existe um preço ótimo teórico, podemos tentar estimar a função de probabilidade de ocupação a partir de métodos numéricos de modelagem a partir de algoritmos de aprendizado de máquina. Uma vez obtida essa função, podemos resolver uma equação de diferenças para obter os preços ótimos de diárias a cada dia até a reserva.

4.1 OBTENÇÃO E LIMPEZA DE DADOS

A Seazone possui dados de disponibilidade e preço do Airbnb por aproximadamente 2 anos. É necessário fazer uma limpeza e transformação nesses dados, para que eles possam ser utilizados para o treinamento de modelo. É necessário que cada linha represente uma diária com uma antecedência específica. Como queremos identificar uma “borda de transição”, ou seja, o dia em que aquela diária vai alugar, fizemos o seguinte procedimento com os dados:

- Removemos todas as diárias com cancelamentos de reserva.
- Marcamos como 1 a borda de transição, ou seja, quando uma diária passa de disponível para alugada.
- Removemos todas as linhas referente a mesma diária após uma borda de transição

Após a limpeza e a transformação dos dados, teremos uma tabela onde cada diária é representada por uma sequência de zeros, até que ela alugue, a data do aluguel está marcada com um 1, e datas seguintes não estão representadas . Por exemplo, se o Natal (25/12) foi alugado no dia 01/12, os dados para essa diária seriam:

Figura 14 – Exemplo de organização de dados

date	advance	rented
25/12	30	0
25/12	29	0
25/12	28	0
25/12	27	0
25/12	26	0
25/12	25	0
25/12	24	1

Fonte: Autor

A cada uma dessas linhas podemos associar quaisquer variáveis que acreditamos influenciar na decisão de alugar ou não um imóvel. Nesse caso, estaremos

avaliando o impacto de variáveis temporais como dia da semana e mês, e também o preço e estadia mínima. Após o início da pandemia, em março, até novembro de 2020, marcamos uma variável Booleana “pandemic” como verdadeira.

4.2 ESTIMAR A FUNÇÃO DE PROBABILIDADE DE OCUPAÇÃO

Escolhemos um conjunto de apartamentos muito similares, a fim de diminuir o impacto que a escolha do apartamento pode ter no consumidor. A partir dos dados de ocupação e preço do último ano desses apartamentos, podemos treinar um modelo que encontre a probabilidade de ocupar em cada dia. Aqui é necessário testar diferentes métodos de aprendizado de máquina (como Árvores de Decisão, Redes Neurais, Random Forest, etc.) para determinar o método que possui o melhor desempenho. Para medir o desempenho vamos utilizar a métrica F1, e o Stratified Shuffle Split, e também o Time Series Split. Vamos comparar o modelo treinado com um Dummy Estimator “uniforme”, que prediz 0 ou 1 de forma proporcional à nossa base de dados, ou seja, se temos 90 zeros na saída, e apenas 10 uns, o dummy estimator terá uma chance de 90% de predizer 0, e apenas 10% de predizer 1. Uma vez encontrado o modelo com o melhor desempenho, vamos analisar esse modelo com o SHAP, a fim de obter conclusões de funcionamento do modelo.

4.3 CÁLCULO DE PREÇO ÓTIMO

Depois de obtido o modelo que calcula $Prob_{oc}(p)$, podemos utilizar desse resultado para resolver a seguinte equação de diferenças:

$$F_{n-m}(p, a, c) = p * Prob_{oc}(p) + F_{n-m-1} * (1 - Prob_{oc}(p))$$

Onde $F_{n-m}(p, a, c)$ é o faturamento esperado de um imóvel para a diária n, com m dias de antecedência, F_{n-m+1} é o faturamento esperado do imóvel para a diária n com m-1 dias de antecedência. p é o preço ofertado para a diária n, com m dias de antecedência, e $Prob_{oc}(p)$ é a probabilidade de a diária n, alugar com exatamente m dias de antecedência pelo preço p.

Podemos comparar os preços obtidos com os preços utilizados historicamente e obter conclusões a respeito de possíveis vieses no modelo.

5 IMPLEMENTAÇÃO DA SOLUÇÃO PROPOSTA E RESULTADOS

Para estimar a função de probabilidade de ocupação escrevemos um script em Python utilizando da bilbioteca scikit-learn. Para uma árvore de decisão com um mínimo de 100 amostras por folha, e para um Dummy Estimator uniforme obtivemos os resultados exemplificados na figura 15, em testes de Stratified Shuffle Split:

Figura 15 – Desempenho de uma árvore de decisão e um estimador Dummy balanceado

Scores						Média	Desvio Padrão
fit_time'	0,103	0,111	0,109	0,106	0,109	0,108	0,00303903
score_time'	0,991	0,972	0,971	0,977	0,956	0,973	0,01248028
test_accuracy'	0,957	0,957	0,957	0,958	0,957	0,957	0,00040854
test_balanced_accuracy'	0,794	0,796	0,799	0,798	0,793	0,796	0,00275681
test_precision'	0,769	0,767	0,752	0,769	0,768	0,765	0,00744811
test_recall'	0,602	0,606	0,615	0,610	0,600	0,606	0,00611262
test_f1'	0,675	0,677	0,676	0,680	0,673	0,677	0,00256538
Dummy							
fit_time'	0,125	0,833	0,845	0,830	0,826	0,692	0,31672375
score_time'	0,797	0,795	0,800	0,795	0,796	0,797	0,00201679
test_accuracy'	0,864	0,864	0,863	0,864	0,864	0,864	0,00014434
test_balanced_accuracy'	0,500	0,500	0,499	0,501	0,500	0,500	0,00052872
test_precision'	0,073	0,074	0,072	0,075	0,073	0,073	0,00098160
test_recall'	0,073	0,074	0,072	0,074	0,073	0,073	0,00097952
test_f1'	0,073	0,074	0,072	0,075	0,073	0,073	0,00098056

Fonte: Autor

Na figura 15, a coluna ’Scores’ representa o nome das variáveis computadas em cada linha. As 5 colunas sem nome são os 5 testes realizados de acordo com a metodologia do Stratified Shuffle Split. A coluna ’Média’ indica o resultado médio dos 5 testes, enquanto a coluna ’Desvio Padrão’ indica o desvio padrão dos 5 testes.

A variável ’fit_time’ indica o tempo de treinamento do modelo para esse conjunto de dados, em segundos. ’score_time’ é o tempo de avaliação, ou seja, para o teste e a computação das métricas de desempenho. ’test_accuracy’ é a exatidão, ’test_balanced_accuracy’ é a exatidão balanceada. ’test_precision’ é a specificidade, ’test_recall’ é a sensibilidade, e ’test_f1’ é a métrica objetivo F1.

As primeiras 7 linhas após o cabeçalho são referentes ao modelo treinado, enquanto as últimas 7 linhas são referentes ao ’Dummy Estimator’ uniforme.

Para a métrica objetivo F1, o desempenho foi significativamente melhor com a

árvore de decisão, onde houve um aumento de 7.3% para 67.7%. Aqui também fica claro que um bom resultado com algumas métricas não necessáriamente indica um bom modelo. O Dummy Estimator foi capaz de acertar se determinada data aluga ou não 86% das vezes, contra os 97% da árvore de decisão. Um leitor desavisado poderia olhar para essa métrica e acreditar que a melhora de desempenho não foi significativa, mas a métrica mais significativa – que leva em consideração o balanceamento dos dados e quantidade de falsos positivos e negativos – é a F1, e é por isso que estamos usando ela como métrica objetivo.

Também realizamos um teste utilizando o Time Series Split, para determinar como que o desempenho do modelo evolui no tempo. Os resultados estão na figura 16:

Figura 16 – Desempenho de uma Árvore de decisão ao longo do tempo

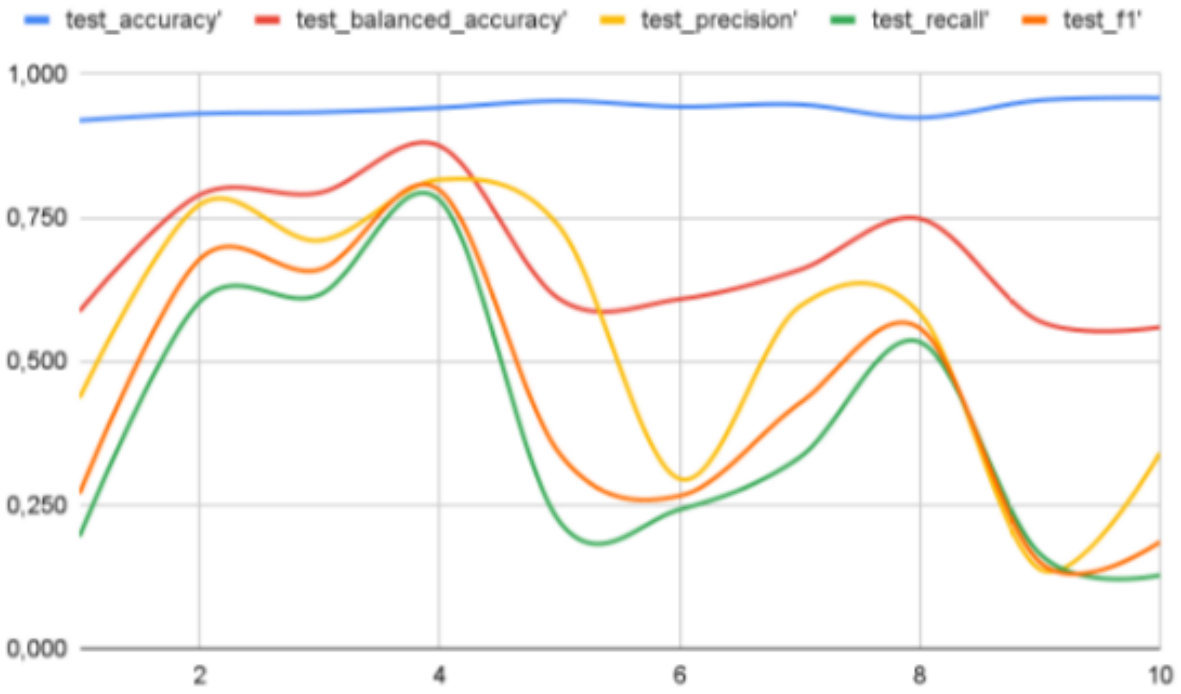
Time Scores:	1	2	3	4	5	6	7	8	9	10
fit_time'	0,102	0,245	0,350	0,527	0,597	0,741	0,896	1,164	1,393	1,538
score_time'	0,155	0,151	0,144	0,144	0,144	0,146	0,147	0,145	0,145	0,144
test_accuracy'	0,919	0,931	0,933	0,941	0,953	0,943	0,947	0,924	0,954	0,958
test_balanced_accuracy'	0,587	0,789	0,793	0,875	0,608	0,608	0,659	0,748	0,569	0,559
test_precision'	0,436	0,773	0,710	0,816	0,733	0,295	0,595	0,583	0,138	0,340
test_recall'	0,195	0,603	0,616	0,780	0,221	0,242	0,333	0,534	0,165	0,127
test_f1'	0,270	0,677	0,660	0,798	0,339	0,266	0,427	0,557	0,150	0,185

Fonte: Autor

Como pode ser visto nas figuras 16 e 18, na primeira iteração, o desempenho foi relativamente ruim pela baixa quantidade de dados para treinamento do modelo. O desempenho melhora drasticamente já na segunda iteração, e atinge o pico na quarta iteração. Todavia, na quinta iteração é onde temos os dados de março de 2020, ou seja, do início da pandemia. Aqui temos os primeiros indícios de que a pandemia está afetando negativamente o resultado do modelo. Entre a quinta e a oitava iteração o desempenho do modelo segue crescendo, mas o verão lotado seguido de uma forte segunda onda da pandemia foram novamente detrimentais para o desempenho do modelo.

Podemos tentar melhorar o desempenho do modelo realizando uma melhor escolha de hiperparâmetros. Para isso, vamos utilizar a biblioteca Scikit AutoML, com a qual podemos especificar um tempo de treinamento e ela poderá utililzar algoritmos de otimização para encontrar a melhor escolha. Para uma hora de treinamento temos:

Figura 17 – Gráfico do desempenho de uma Árvore de decisão ao longo do tempo



Fonte: Autor

Figura 18 – Comparação do desempenho de uma árvore de decisão, e de uma random forest, com parametros otimizados a partir do SciKitAutoML

	Scores						Média	Desvio Padrão
Decision tree, minSamplesLeaf = 100	fit_time'	0,103	0,111	0,109	0,106	0,109	0,108	0,0030390
	score_time'	0,991	0,972	0,971	0,977	0,956	0,973	0,0124803
	test_accuracy'	0,957	0,957	0,957	0,958	0,957	0,957	0,0004085
	test_balanced_accuracy'	0,794	0,796	0,799	0,798	0,793	0,796	0,0027568
	test_precision'	0,769	0,767	0,752	0,769	0,768	0,765	0,0074481
	test_recall'	0,602	0,606	0,615	0,610	0,600	0,606	0,0061126
	test_f1'	0,675	0,677	0,676	0,680	0,673	0,677	0,0025654
Scikit AutoML, randomForest, training time = 1h	fit_time'	3635,342	3644,500	3652,022	3635,334	3642,760	3641,992	7,0003262
	score_time'	396,839	354,651	343,545	358,240	420,420	374,739	32,4959394
	test_accuracy'	0,963	0,963	0,963	0,964	0,964	0,963	0,0001488
	test_balanced_accuracy'	0,820	0,821	0,818	0,818	0,817	0,819	0,0016037
	test_precision'	0,813	0,814	0,817	0,820	0,824	0,818	0,0043716
	test_recall'	0,653	0,653	0,647	0,648	0,645	0,649	0,0035768
	test_f1'	0,724	0,725	0,722	0,724	0,724	0,724	0,0009458

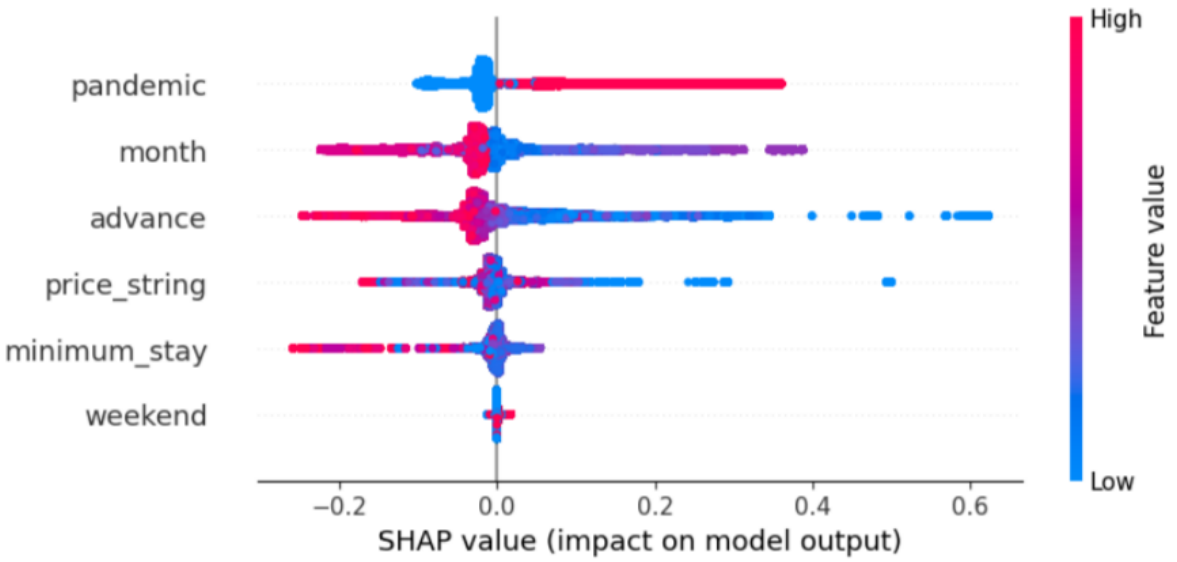
Fonte: Autor

Onde fica claro que o desempenho melhorou consideravelmente. A nossa métrica objetivo F1 melhorou em 5%, e o desvio padrão entre os testes diminuiu pela

metade – indicando que esse novo modelo tende a ser menos overfitado do que o anterior, ou seja, tem maiores chances de funcionar na prática. Isso é esperado de um modelo de random forest, que utiliza diversas árvores de decisão para evitar o overfitting.

Utilizando a biblioteca SHAP nesse modelo, podemos atribuir para cada variável qual impacto ela teve na saída do modelo. No gráfico abaixo, cada variável em cada linha dos dados está representada por um ponto. A cor de cada ponto representa o valor dessa variável. Como pandemic é uma variável booleana, ela apenas se apresenta como azul claro e vermelho claro, enquanto variáveis com maior variação de valores como month ou price possuem seus valores intermediários coloridos em um gradiente roxo. Quando há uma quantidade maior de pontos em uma mesma região, eles são empilhados a fim de demonstrar uma maior concentração e/ou quantidade. O gráfico abaixo é chamado de ”gráfico resumo” do modelo.

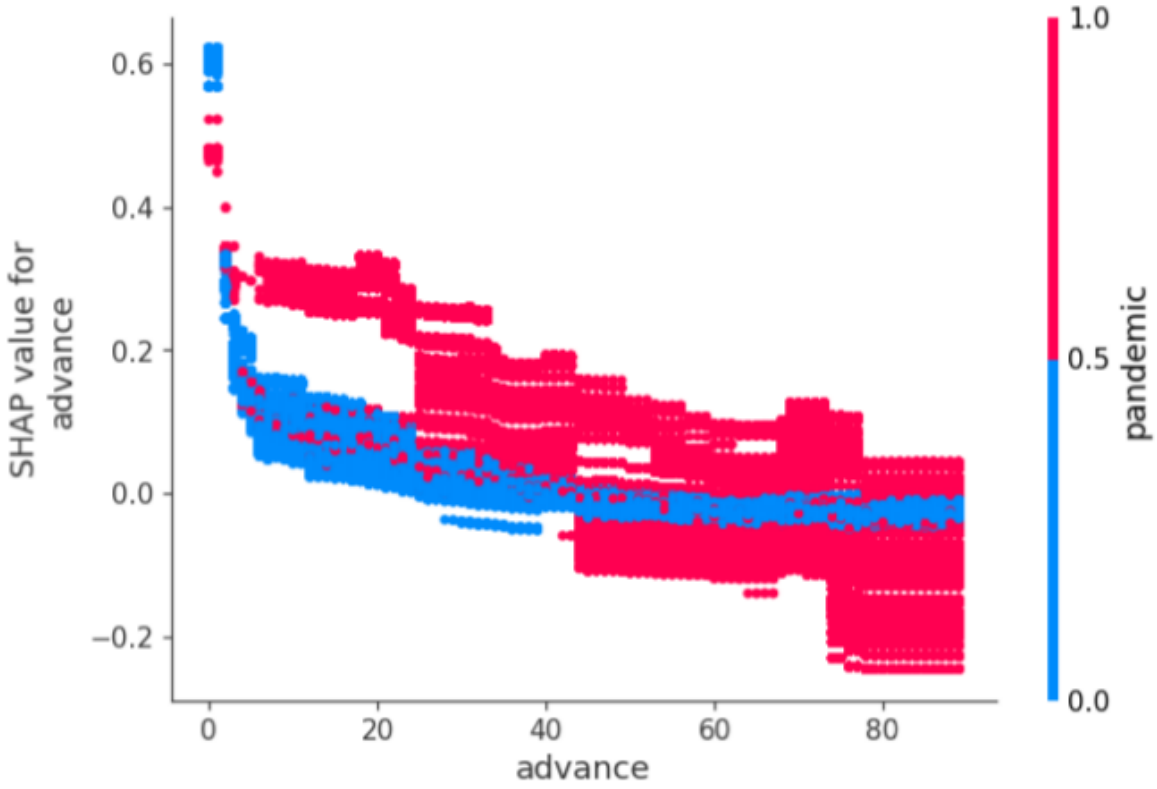
Figura 19 – Gráfico resumo do modelo



Fonte: Autor

O posicionamento de um ponto no eixo x representa o impacto dessa variável na saída do modelo. Ou seja, quanto que determinada variável aumenta ou diminui a probabilidade de ocupação. Vemos que advance (antecedência) pode ter um impacto de quase 60% na probabiidade de ocupação, e que linhas com o advance azul – valores baixos de antecedência – tem uma chance maior de serem alugados. Podemos examinar essa relação mais a fundo com um gráfico de disperção (scatter plot) da antecedência:

Figura 20 – Gráfico de dispersão da antecedência



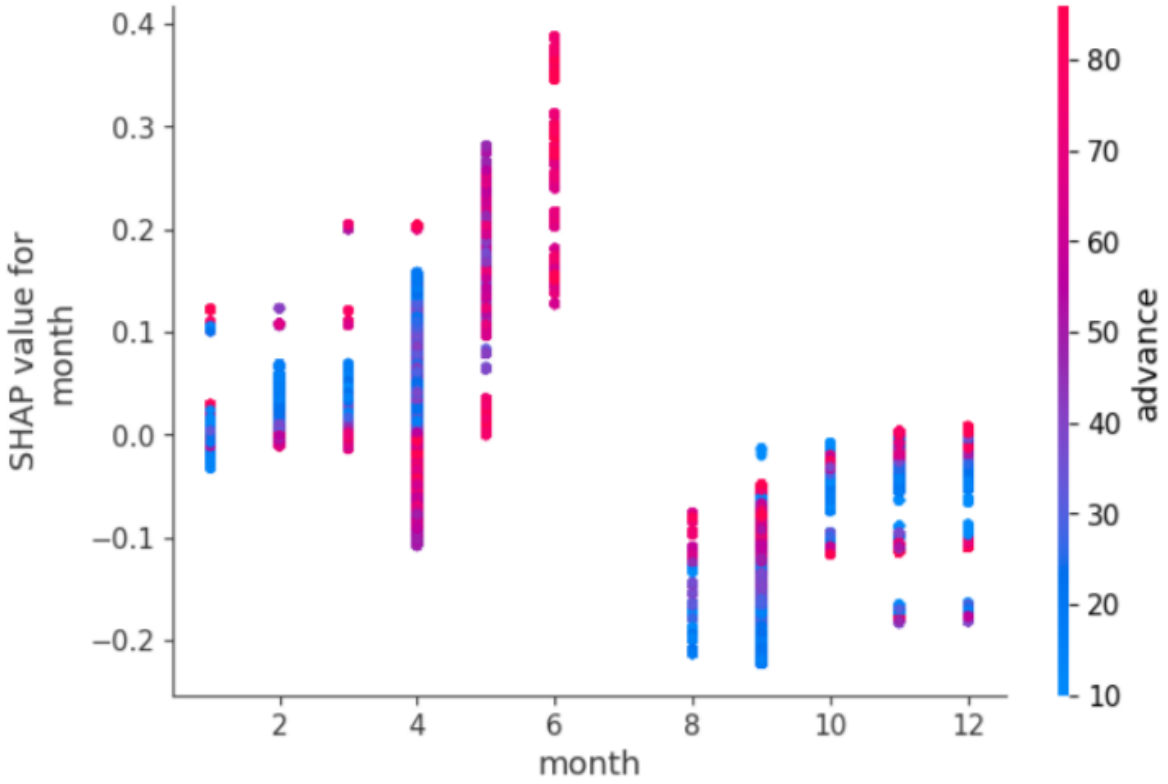
Fonte: Autor

Nesse gráfico os pontos onde pandemic é verdadeiro estão pindatos de vermelho, enquanto pontos onde pandemic é falso estão pintados de azul. O eixo X marca o valor de antecedência, enquanto o eixo Y marca o impacto da variável antecedência na probabilidade de ocupação.

Aqui fica claro que diárias com antecedência menor tem probabilidade maior de serem ocupadas.

Também fica claro que o comportamento de diárias com pandemic marcado como verdadeiro tem um comportamento muito diferente de diárias fora da pandemia. Isso começa a consolidar que a pandemia teve um grande impacto no resultado do modelo, e que talvez esteja prejudicando seu funcionamento. Podemos explorar mais a fundo o comportamento das diárias ao longo do tempo com um scatter plot da variável month (mês), conforme Figura 21:

Figura 21 – Gráfico de dispersão do mês



Fonte: Autor

Aqui os pontos estão coloridos de acordo com a sua antecedência.

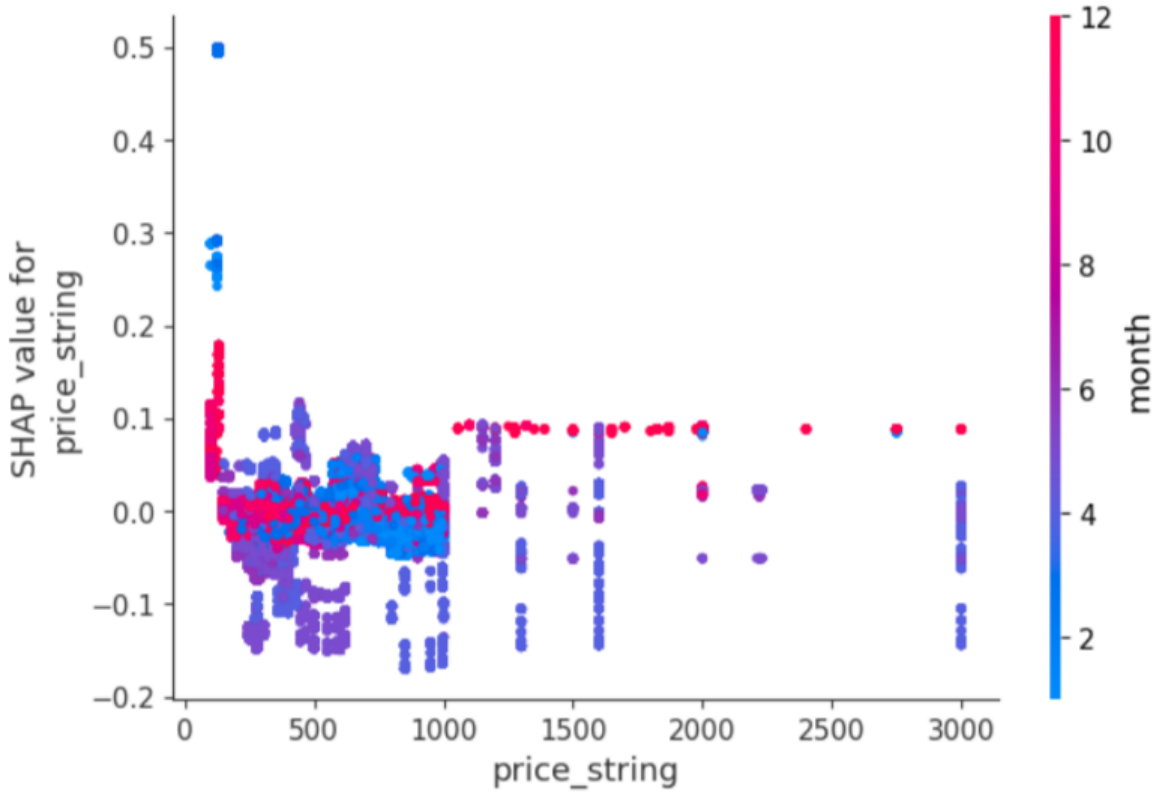
Não existe nenhum ponto no mês 7, pois todos eles foram removidos durante a limpeza de dados. Isso faz sentido, pois houve uma medida provisória que proibia alugueis nesse período por conta da pandemia. Todavia, percebemos nesse gráfico que durante o mês 6, e também durante o mês 5, a probabilidade de ocupação aumenta em função do mês. Também percebemos que não existem pontos nesses meses com baixa antecedência, o que indica que eles foram quase todos alugados com uma antecedência alta.

Isso ocorreu porque durante esses meses, apenas era permitido a operação desses apartamentos durante o final de semana. Como eles estavam indisponíveis no Airbnb durante a semana, eles foram identificados como ocupados, ou seja, alugados.

Isso explicaria o porquê de, na Figura 18, o desempenho do modelo ter piorado significativamente ao redor de março/abril.

Podemos também analisar o impacto da variável ”preço” na probabilidade de ocupação: afinal esse era o principal objetivo do modelo.

Figura 22 – Gráfico de dispersão do preço



Fonte: Autor

Na Figura 22 os pontos estão coloridos de acordo com o mês da diária. Aqui fica claro que o modelo não encontrou uma boa correlação entre o preço e a ocupação, mesmo considerando os meses.

É muito provável que a falta dessa correlação tenha acontecido por conta da pandemia e da falta da remoção dos dados entre março e junho, que já observamos terem impactado muito negativamente no desempenho do modelo. Também é possível que não tenham sido fornecidas variáveis suficientes para que o modelo identifique a correlação entre a locação e o preço. Se fosse fornecido também para o modelo a previsão do tempo para cada dia, então talvez ele fosse capaz de discernir quando que um apartamento deixou de alugar por conta do clima, ou por conta do preço. Também vale lembrar que a pandemia, na prática, não é uma variável Booleana, e que provavelmente é mais sensato utilizar uma variável numérica para representá-la (como número de mortos, infectado, lotação de UTI, etc.).

Mesmo sabido dos problemas com o modelo, e esperando que o resultado não seja satisfatório, também utilizamos esse modelo para resolver a equação:

$$F_{n-m}(p, a, c) = p * Prob_{oc}(p) + F_{n-m+1} * (1 - Prob_{oc}(p))$$

e encontrar o preço que maximiza o faturamento. Todavia, na maior parte dos testes realizados, o modelo não é capaz de identificar a variação na probabilidade de ocupação em função do preço. Ou seja, para o modelo de árvores de decisão, a diária a 100 ou a 600 reais tem a mesma probabilidade de ocupação. Com isso, o preço sugerido pelo algoritmo ficou muito acima de preços que são empiricamente razoáveis. Acreditamos que com uma melhor limpeza de dados, e com melhorias nos modelos, seja possível conseguir melhores resultados.

6 CONCLUSÃO

Nesse trabalho exploramos como o processo de análise e modelagem pode ser utilizado para encontrar erros no pipeline de dados. Isso é fundamental para evitar modelos comumente chamados de “garbage in, garbage-out” (lixo entra, lixo sai), onde falhas no início do pipeline comprometem o funcionamento do modelo. Esse tipo de problema tem se tornado muito comum na indústria, pois grande parte dos modelos obtidos são “caixas pretas”, ou seja, é muito difícil de compreender seu funcionamento.

Quando iniciei esse trabalho eu sabia que o problema proposto era talvez demasiado complexo para o tempo que nós teríamos disponível para o projeto. Fui alertado por alguns professores desde o início de que possivelmente o que estava sendo proposto seria uma tese de mestrado, e não um projeto de fim de curso. Nesse sentido, não me sinto surpreso ou decepcionado de o principal objetivo do trabalho – de conseguir encontrar preços otimizados – não ter sido cumprido. Todavia, com os conhecimentos obtidos, fizemos um enorme avanço e abrimos mais portas do que fechamos. Dentre os próximos passos para o algoritmo de precificação, destaco:

- Melhorar o processo de limpeza de dados. Principalmente referente a dados da pandemia.
- Enriquecer os dados com mais informações relevantes, como ocupação de apartamentos similares, preços dos concorrentes disponíveis, e previsão do tempo, entre outras.
- Utilizar de computação em nuvem distribuída para o treinamento de modelos por mais tempo e por máquinas com mais RAM.

Também tivemos a ideia de decompor a variável preço em diferentes componentes. Ou seja, construir modelos para determinar quando e quanto diminuir/aumentar o preço ao longo do tempo, com relação a um preço base. Isso seria como decompor um modelo mais complexo em uma sequência de modelos mais simples, que pode – ou não – ter um melhor desempenho.

A equipe está bem otimista com relação ao potencial desses modelos, e já começamos o desenvolvimento de um produto de precificação de diárias. Finalmente, nota-se que o problema de precificação é quase universal, e que muitos dos conceitos abordados e implementados nesse projeto podem ser transferidos para o varejo, aviação, e até para o mercado de ações.

REFERÊNCIAS

BARRON, Kyle; KUNG, Edward; PROSERPIO, Davide. The Effect of Home-Sharing on House Prices and Rents: Evidence from Airbnb. **SSRN**, mar. 2020. DOI: 10.2139/ssrn.3006832.

LUNDBERG, Scott M; LEE, Su-In. A Unified Approach to Interpreting Model Predictions. *In*: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems 30**. [*S.l.*]: Curran Associates, Inc., 2017. P. 4765–4774. Disponível em: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *In*: PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. [*S.l.*: *s.n.*], 2016. P. 1135–1144.

TALLURI, Kalyan T.; RYZIN, Garrett J. Van. **The Theory and Practice of Revenue Management**. [*S.l.*]: Springer US, 2004. DOI: 10.1007/b139000. Disponível em: https://doi.org/10.1007/b139000.