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

Márcio Vinícius Almeida Fazolin

# Classificação binária com redes neurais para automação do processo de detecção de bloqueios em imóveis de aluguel de temporada

Florianópolis
2023

Márcio Vinícius Almeida Fazolin

# Classificação binária com redes neurais para automação do processo de detecção de bloqueios em imóveis de 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. Eric Aislan Antonelo
Supervisor: Bruno Eduardo Benetti

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Fazolin, Márcio Vinícius Almeida
Classificação binária com redes neurais para automação do processo de detecção de bloqueios em imóveis de aluguel de temporada / Márcio Vinícius Almeida Fazolin ; orientador, Eric Aislan Antonelo, 2023.
65 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia de Controle e Automação, Florianópolis, 2023.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Aprendizado de Máquina. 3. Redes Neurais. 4. Detecção de Bloqueios. 5. Análise de Dados. I. Antonelo, Eric Aislan. II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Márcio Vinícius Almeida Fazolin

**Classificação binária com redes neurais para automação do processo de detecção de bloqueios em imóveis de 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, 19 de Dezembro de 2023.

Prof. Hector Bessa Silveira, Dr.
Coordenador do Curso

**Banca Examinadora:**

Prof. Eric Aislan Antonelo, Dr.
Orientador
UFSC/CTC/DAS

Luis Fernando Nazari.
Avaliador
Instituição IFC

Prof. Bruno Benetti.
Avaliador
Instituição Seazone

Prof. Eduardo Camponogara, Dr.
Presidente da Banca
UFSC/CTC/DAS

Este trabalho é dedicado aos meus amigos e família,
sempre presentes e me ajudando.

## AGRADECIMENTOS

A conclusão deste projeto marca o encerramento de uma jornada enriquecedora, e muitos foram os indivíduos que desempenharam algum papel em tornar este projeto uma realidade. Expresso minha sincera gratidão a todos aqueles que, de diferentes maneiras, contribuíram para o sucesso deste trabalho.

Primeiramente, minha mais profunda gratidão vai para minha família, em especial para minha mãe, Suelene Fazolin, e meu pai, Márcio Tadeu Fazolin. Sem o apoio incansável deles, desde os primeiros passos até fim dessa jornada acadêmica da graduação, nada disso seria possível. Suas palavras de incentivo, compreensão e amor foram minha rocha durante todo o percurso.

Estendo meus agradecimentos calorosos aos demais membros da minha família, incluindo avós, tios e primos. Sendo filho único, a presença constante de vocês foi um apoio inestimável e fez toda a diferença.

À comunidade acadêmica e ao Departamento de Automação e Sistemas (DAS) da UFSC, expresso minha gratidão. Em particular, gostaria de destacar o professor Eric Aislan Antonelo, cuja generosidade e disponibilidade surpreenderam-me positivamente, tornando esta jornada ainda mais enriquecedora.

Agradeço também a Bruno Benetti e André Crescenzo, cuja parceria desde os primeiros dias na Seazone foi fundamental. Seu apoio e orientação foram essenciais para navegar e entender as complexidades do mundo dos dados.

Meus agradecimentos se estendem a todo o time de dados da Seazone. André Padilha, Artur Brito, Augusto Hideki, Francisco Burigo, Lucas Abel e Nícolas Campana, agradeço por compartilharem seus conhecimentos e experiências, enriquecendo este trabalho com sua colaboração.

Por último, mas não menos importante, agradeço aos meus amigos, sejam aqueles que conheci na faculdade, aqueles que acompanho desde a infância ou qualquer pessoa que, de alguma forma, contribuiu para o sucesso deste projeto. Sua amizade e apoio foram pilares essenciais durante esta jornada.

A todos mencionados e a muitos outros que, de alguma forma, deixaram sua marca neste percurso, meu mais profundo obrigado.

*O objetivo é transformar dados em informação*
*e informação em entendimento.*
*(Carly Fiorina, 2020)*

**DECLARAÇÃO DE PUBLICIDADE**

Florianópolis, 19 de Dezembro de 2023.

Na condição de representante da Seazone Serviços LTDA na qual o presente trabalho foi realizado, declaro não haver ressalvas quanto ao aspecto de sigilo ou propriedade intelectual sobre as informações contidas neste documento, que impeçam a sua publicação por parte da Universidade Federal de Santa Catarina (UFSC) para acesso pelo público em geral, incluindo a sua disponibilização *online* no Repositório Institucional da Biblioteca Universitária da UFSC. Além disso, declaro ciência de que o autor, na condição de estudante da UFSC, é obrigado a depositar este documento, por se tratar de um Trabalho de Conclusão de Curso, no referido Repositório Institucional, em atendimento à Resolução Normativa n° 126/2019/CUn.

Por estar de acordo com esses termos, subscrevo-me abaixo.

GOV.BR

Documento assinado digitalmente
**BRUNO EDUARDO BENETTI**
Data: 19/12/2023 08:05:51-0300
CPF: ***.335.709-**
Verifique as assinaturas em https://v.ufsc.br

Bruno Benetti
Seazone Serviços LTDA

## RESUMO

Esta monografia propõe a criação de um modelo baseado em redes neurais para prever a natureza de períodos ocupados em imóveis de aluguel de temporada, distinguindo-os entre reservas e bloqueios. O modelo utiliza dados internos da empresa Seazone para realizar o treinamento supervisionado e depois a predição é feita em cima de dados provindos de Web Scraping do Airbnb. O intuito é gerar predições de faturamento em cima dos imóveis raspados. A monografia inicia com a análise exploratória dos dados, envolvendo tabelas de reservas, bem como dados que detalham as qualidades físicas dos imóveis. Depois de explorados, é realizada a limpeza, criação de novos atributos e normalização dos dados. A monografia aborda então a fase de treinamento, onde é criado modelos iniciais de redes neurais que fazem a classificação binária entre reservas e bloqueios. Também é utilizado a otimização bayesiana para ajuste fino dos hiperparâmetros. São utilizadas técnicas para mitigar o risco de overfitting, como o uso de regularizadores L1 e L2 e camadas de dropout. O uso de métricas de validação, como f1-score, foram usadas para avaliar o desempenho do modelo. No final o modelo é posto em ambiente de produção onde é realizado cálculos de previsões de faturamento.



**Palavras-chave**: Aprendizado de Máquina, Redes Neurais, Detecção de Bloqueios.

## ABSTRACT

This dissertation proposes the creation of a neural network-based model to predict the nature of occupied periods in seasonal rental properties, distinguishing between reservations and blocks. The model uses internal data from the Seazone company for supervised training, and the prediction is then made on data obtained through Airbnb web scraping. The goal is to generate revenue predictions for scraped properties. The dissertation begins with exploratory data analysis, involving reservation tables, as well as data detailing the physical qualities of the properties. After exploration, data cleaning, the creation of new attributes, and data normalization are performed. The dissertation then addresses the training phase, where initial neural network models are created for binary classification between reservations and blocks. Bayesian optimization is also used for fine-tuning hyperparameters. Techniques to mitigate the risk of overfitting, such as the use of L1 and L2 regularizers and dropout layers, are employed. Validation metrics, such as the f1-score, are used to evaluate the model's performance. In the end, the model is deployed in a production environment where revenue forecast calculations are performed.



**Keywords**: Machine Learning, Neural Networks, Block Detection.

## LISTA DE FIGURAS

## LISTA DE TABELAS

**SUMÁRIO**

# 1 INTRODUÇÃO

O presente Projeto de Final de Curso (PFC) tem como foco a abordagem de um desafio significativo na área de previsão de faturamento, especificamente no contexto do Airbnb. O problema em questão reside na complexidade associada à distinção entre bloqueios e reservas, o que tem sido uma tarefa desafiadora na análise de dados dessa plataforma.

## 1.1 CONTEXTO DO AIRBNB

O Airbnb, uma plataforma líder no setor de aluguel por temporada, tem testemunhado um crescimento notável, consolidando-se como uma escolha preferida para viajantes em todo o mundo. De acordo com dados fornecidos pela (IMPACTO. . . , 2022), em 2021, os gastos dos hóspedes atingiram a marca impressionante de 4 bilhões de dólares apenas no Brasil. Esse montante representa uma significativa parcela de 5.6% de toda a atividade turística no país, destacando a crescente influência e participação do Airbnb no cenário turístico brasileiro.

Além dos números impressionantes de faturamento gerados, o estudo detalhado de dados do Airbnb iria oferecer uma visão valiosa do comportamento do mercado em diversas cidades do Brasil. Ao analisar padrões de reserva, preferências de acomodação e tendências sazonais, as possíveis informações extraídas desses dados não apenas refletem a dinâmica do setor de hospitalidade, mas também se tornam uma fonte valiosa para estrategistas de turismo e profissionais do ramo que buscam compreender e adaptar-se às demandas em constante evolução dos viajantes.

Neste contexto, existe a empresa Seazone, cujo foco é aluguel de temporada. Ela administra diversos imóveis e os anuncia em múltiplos sites de aluguel, como Airbnb, Booking, entre outros. Ela também possui diversas "web scrappers", que são programas ou códigos de computador capazes de acessar uma ou mais páginas da internet a fim de coletar dados para salvá-los em tabelas internas. Na Seazone, esses "scrappers" adquirem dados do Airbnb, tornando possível uma análise a fundo de dados dessa plataforma. A partir deles, a empresa consegue fazer previsões de faturamento para diversos imóveis e regiões do Brasil. Por realizar decisões estratégicas com base em dados, a Seazone se diferencia de outras empresas no mesmo ramo.

Um dos muitos exemplos de uso destes dados é saber quais regiões faturam mais no Airbnb. A Seazone possui informações de localização e faturamento de cada imóvel do Airbnb, portanto, ela consegue saber qual cidade tem mais faturamento e, dentro dessa cidade, qual o melhor bairro. Esses dados podem ser utilizados para a empresa expandir em regiões onde já foi comprovado um rendimento superior.

Entretanto, os dados do Airbnb não vêm de forma limpa e, mesmo depois de tratados, eles possuem erros de faturamento. O maior problema é que os "scrappers" não

trazem informações suficientes para diferenciar automaticamente reservas e bloqueios realizados no site e esses dois dados se misturam nas previsões de faturamento. Uma reserva é quando um hóspede paga para usar o imóvel durante um intervalo de tempo, isso gera faturamento e é um dado importante de ser considerado nas previsões, entretanto, um bloqueio acontece quando o proprietário do imóvel bloqueia períodos de tempo, isso não gera faturamento e, portanto, deve ser descartado.

A empresa adotou um método para diferenciar esses casos, composto em aplicar uma série de regras de negócio que fazem sentido no papel, mas ainda sim a maior parte dos erros de faturamento vêm deste problema de diferenciar reservas e bloqueios. Por esses motivos, a Seazone está interessada em empregar um método novo e mais inteligente para distingui-los, com intuito de obter previsões de faturamento mais confiáveis. Aprimorar essa capacidade de predição desencadeará um impacto significativo em diversos aspectos de negócio para a empresa, pois dados mais confiáveis podem gerar decisões estratégicas melhores.

## 1.2 OBJETIVO DO PROJETO

No âmbito deste projeto, será explorada a aplicação de redes neurais como uma abordagem para a classificação binária entre reserva e bloqueio. O intuito é que, por ser a maior fonte de erro, melhorar essa diferenciação também melhoraria a confiabilidade dos dados de predição de faturamento. Vale ressaltar que a utilização de redes neurais não foi o primeiro método que a empresa utilizou para essa finalidade, mas é uma das alternativas selecionadas para este projeto.

A metodologia adotada segue métodos convencionais de aprendizado de máquina, compreendendo as etapas de entendimento dos dados, treinamento, predição, validação e otimização. Este trabalho se propõe a explorar essas fases em detalhes, apresentando resultados relevantes para o contexto em questão.

Os resultados obtidos até o momento revelaram melhorias notáveis na capacidade de previsão de faturamento da Seazone, com ganhos de até 7% em determinados meses, quando comparados com os métodos aplicados anteriormente que dependiam exclusivamente de heurísticas e regras de negócio.

Adicionalmente, serão discutidos os procedimentos específicos realizados durante a implementação do projeto, incluindo a criação de pipelines no Sagemaker da AWS para o treinamento do modelo e outros aspectos de implantação relevantes. Esses detalhes e as técnicas empregadas serão abordados de forma mais abrangente ao longo deste trabalho.

## 1.3 ESTRUTURA DO DOCUMENTO

No capítulo 2, é apresentada a motivação que impulsionou a realização deste projeto. Descreve-se o cenário envolvendo a empresa Seazone, como os "web scrappers" funcionam e o desafio relacionado à previsão de faturamento dos imóveis do Airbnb. Este capítulo estabelece o contexto fundamental para o entendimento do escopo do projeto.

No Capítulo 3, detalham-se as tecnologias e serviços utilizados na Seazone, que, por utilizar muitas ferramentas de "Cloud" da AWS (Amazon Web Services), o projeto também utiliza algumas. Além disso, exploram-se as bibliotecas utilizadas e teorias fundamentais sobre redes neurais e modelos de aprendizado de máquinas, que constituem a base teórica do projeto.

No capítulo 4, estão descritos os requisitos do projeto, abordando tanto os requisitos funcionais quanto os não-funcionais. Também é apresentado um diagrama que ilustra a solução proposta para enfrentar o desafio da previsão de faturamento.

O quinto capítulo descreve em detalhes o desenvolvimento do projeto, seguindo um processo de MLOps (Machine Learning Operations), começando desde a análise exploratória e indo até o ajuste de hiperparâmetros e implantação do modelo em produção, que, neste caso, terá infraestrutura na AWS.

No Capítulo 6, são realizadas análises dos resultados obtidos ao longo do projeto. Exploram-se as vantagens e desvantagens identificadas durante o processo, bem como os impactos que o projeto teve na empresa Seazone e em seus processos relacionados.

Por fim, na conclusão, é apresentado um resumo abrangente das atividades e realizações do projeto. Este capítulo fornece uma visão geral das contribuições do projeto para a resolução do problema da previsão de faturamento no contexto do Airbnb e destaca os principais aprendizados e conclusões obtidos ao longo do trabalho.

# 2 MOTIVAÇÃO

## 2.1 EMPRESA SEAZONE

A Seazone é uma empresa inovadora que começou em Florianópolis, mas já se expandiu para diversas cidades do Brasil. Fundada como uma "startup", a empresa concentra seus esforços no segmento de aluguel de temporada, oferecendo uma gama abrangente de serviços que abordam todas as necessidades dos proprietários de imóveis e dos viajantes em busca de acomodações excepcionais.

O principal foco de atuação da Seazone é a gestão completa de imóveis destinados ao aluguel de temporada. Isso inclui desde a criação e otimização de anúncios em plataformas renomadas como Airbnb e Booking, até a coordenação de tarefas essenciais, como a limpeza física e a manutenção dos imóveis. A empresa adota uma abordagem abrangente, simplificando o processo para proprietários de imóveis e garantindo que cada estadia seja memorável para os viajantes.

A Seazone se destaca em sua capacidade de adaptação e de sua constante busca por desenvolver novas tecnologias de auxilio interno. Por meio de avançadas estratégias de marketing, tecnologia de ponta e uma equipe altamente capacitada, a empresa é capaz de maximizar o potencial de cada imóvel sob sua gestão. Neste contexto, o lucro da empresa vêm por comissão, então quanto mais reservas geradas, mais a empresa e os proprietários lucram.

Dentro da empresa, existe o time de dados. Esta equipe é responsável pelo adquirimento e tratamento de dados, utilizando "scrappers" e serviços da AWS (Amazon Web Services). Essa abordagem baseada em dados permite à empresa obter "insights" valiosos sobre o mercado de aluguel de temporada, identificando tendências, preços competitivos e oportunidades para otimizar as estratégias de listagem e preços. Essa ênfase na análise de dados reflete o compromisso da Seazone em aprimorar sua posição no mercado de aluguel de temporada.

## 2.2 AIRBNB E SCRAPPERS

Os "web scrappers" permitem que a empresa adquira informações de forma consistente da internet. Atualmente, a Seazone mantém uma variedade de scrappers dedicados a diferentes plataformas, incluindo Vivareal, OLX, e até mesmo um em fase de testes voltado para o Booking. No entanto, neste contexto do projeto, entra em destaque o "scrapper" desenvolvido especificamente para o Airbnb, uma das maiores e mais influentes plataformas de aluguel de temporada do mundo.

O Airbnb é conhecido por permitir que indivíduos anunciem e reservem uma ampla gama de imóveis, tornando-se uma fonte rica de dados valiosos para a Seazone. Os "scrappers" desenvolvidos pela empresa têm a capacidade de verificar a

disponibilidade de todos os imóveis ativos em todo o Brasil. Além disso, eles coletam informações essenciais relacionadas aos preços, características e localizações dos imóveis. Esses dados são, então, processados e utilizados para gerar previsões de faturamento do Brasil inteiro.

Depois de geradas, a Seazone também consegue medir o erro e se as predições fazem sentido. Isso é possível devido ao fato dela administrar cerca de 1000 imóveis e conhecer o faturamento real deles. A ideia é comparar o faturamento real com o faturamento previsto das predições de faturamento do Airbnb e medir possíveis erros e encontrar lugares de melhorias.

O "scrapper" do Airbnb da Seazone analisa diariamente os 450 mil imóveis ativos no Airbnb no Brasil, monitorando a disponibilidade e os preços para os próximos 365 dias. Esses dados são então armazenados em tabelas internas, gerando cerca de 164 milhões de novas linhas de informações a cada atualização. A Tabela 1 apresenta um exemplo com dados para exemplicar o processo. Todo esse volume de dados em constante expansão serve como a fundação para as análises subsequentes de faturamento e comportamento do mercado.

| Imóvel | Data de aquisição | Data do calendário | Preço | Disponibilidade |
|---|---|---|---|---|
| 1111111 | 2023-01-01 | 2023-02-01 | 500 | Disponível |
| 1111111 | 2023-01-01 | 2023-02-02 | 550 | Disponível |
| 2222222 | 2023-01-01 | 2023-02-01 | 500 | Disponível |
| 1111111 | 2023-01-02 | 2023-02-01 | 550 | Indisponível |
| 2222222 | 2023-01-02 | 2023-02-03 | 600 | Disponível |

Tabela 1 – Exemplo de dados do "scrapper" do Airbnb.

## 2.3 PREDIÇÃO DE FATURAMENTO PELO AIRBNB

Após a coleta inicial de dados, o processo de enriquecimento entra em cena. Nessa fase, o sistema avalia quais datas passaram a estar ocupadas com base na disponibilidade e nos preços registrados anteriormente. Essa análise permite que a Seazone determine se houve ou não reservas durante um período específico. Por exemplo, se as diárias de um determinado imóvel custavam em média R$ 100,00 do dia 1 ao dia 10, e essas datas permanecem disponíveis, isso indica que não houve reservas e, portanto, o faturamento previsto para esse período é zero. No entanto, se em uma atualização subsequente essas datas se tornarem indisponíveis, isso sinaliza a ocorrência de uma reserva, e o faturamento estimado para esse período é recalculado, podendo ser, por exemplo, R$ 1.000,00.

No exemplo da Tabela 1, o imóvel "1111111" na data 2023-02-01 passou de disponível (aquisição 2023-01-01) para indisponível (aquisição 2023-01-02). Esse evento implicaria que este imóvel faturou de forma bruta R$ 550.

### 2.3.1 Problema dos bloqueios

O problema dos bloqueios representa um desafio significativo na lógica de predição de faturamento do Airbnb, afetando a precisão das estimativas e exigindo uma abordagem cuidadosa para lidar com essa complexidade inerente ao mercado de aluguel de temporada. Os bloqueios, que permitem aos proprietários indisponibilizar datas arbitrariamente, introduzem uma variável adicional na equação que pode distorcer as previsões de faturamento.

A razão por trás desses bloqueios pode variar amplamente. Em alguns casos, um imóvel pode exigir manutenção, forçando o proprietário a bloquear datas para garantir que nenhum hóspede seja afetado por condições inadequadas. Em outros casos, o proprietário pode simplesmente desejar utilizar o imóvel para uso pessoal durante um determinado período, tornando as datas indisponíveis para reservas externas. Esses bloqueios podem ser temporários ou recorrentes, e a sua gestão requer uma solução eficaz para evitar distorções nos cálculos de faturamento.

A complexidade surge quando reservas e bloqueios são tratados de maneira similar nos scrappers de disponibilidade. Quando um imóvel é reservado ou bloqueado, as datas previamente disponíveis se tornam indisponíveis. No entanto, enquanto as reservas refletem o verdadeiro faturamento gerado pelos imóveis, os bloqueios não têm esse impacto financeiro direto, uma vez que não representam renda real para os proprietários. Isso pode levar a erros na previsão de faturamento, uma vez que as datas bloqueadas são tratadas da mesma forma que as datas reservadas.

Para mitigar esse problema, é essencial desenvolver uma lógica sofisticada que permita identificar e distinguir bloqueios de reservas. Isso pode envolver a análise de padrões de bloqueio, histórico de uso do imóvel pelo proprietário e outras variáveis relevantes. Uma vez identificados, os bloqueios podem ser excluídos ou tratados de forma apropriada nas previsões de faturamento, garantindo que apenas as reservas efetivas sejam contabilizadas na estimativa de receita.

### 2.3.2 Lógica de detecção de bloqueios atual e proposta de melhoria

A lógica utilizada para realizar essa diferenciação é nomeada "heurística”. Ela é baseada em diversas regras de negócios, muitas delas empregadas heuristicamente, dado o nome do método. Embora essas regras tenham provado ser eficazes em muitos casos, elas também apresentam desafios e limitações.

Entre as principais regras de negócios usadas para distinguir bloqueios de reservas na Seazone, algumas merecem destaque:

1. **Tamanho da Reserva:** Uma das regras estabelece que reservas com duração muito grande, como 30, 60, 90 dias, são consideradas bloqueios. Essa abordagem se baseia na suposição de que estadias longas geralmente indicam que o

proprietário está usando o imóvel para propósitos pessoais ou que o imóvel está indisponível para aluguel durante esse período. Isso é especialmente verdade no contexto de aluguel de temporada.

2. **Imóveis Desativados:** Quando um imóvel é desativado, ou seja, não está mais disponível para aluguel no Airbnb, todas as datas posteriores são tratadas como bloqueios. Isso ocorre porque a interrupção da listagem do imóvel geralmente indica que ele não está disponível para reservas futuras.

3. **Preço por Quarto Elevado:** Se o preço por quarto de um imóvel estiver acima de um determinado limite (por exemplo, R$ 2.000,00), ele é classificado como bloqueio. Essa regra se baseia na ideia de que preços excepcionalmente altos podem indicar que o proprietário não deseja alugar o imóvel durante esse período e que o período ocupado na verdade é um bloqueio.

4. **Datas Indisponíveis Desde a Primeira Aquisição:** Caso o imóvel tenha datas indisponíveis desde sua primeira aquisição, essas datas são consideradas bloqueios. Essa regra parte do princípio de que se o imóvel já estava indisponível para aluguel desde o início, essas datas provavelmente não serão uma reserva.

É importante reconhecer que, embora essas regras tenham sido desenvolvidas com base na experiência e no conhecimento disponíveis, elas podem não ser infalíveis. O mercado de aluguel de temporada é dinâmico e complexo, e as ações dos proprietários podem variar amplamente. Portanto, há casos em que essas regras podem não capturar com precisão a natureza das datas indisponíveis. Por exemplo, pode acontecer de haver um período ocupado com tamanho de estadia de um mês que é de fato uma reserva.

Para melhorar a precisão da lógica de diferenciação entre bloqueios e reservas, a Seazone reconhece a necessidade de continuar refinando suas regras e explorando abordagens mais avançadas. Essa evolução contínua é essencial para garantir que a empresa permaneça na vanguarda do setor de aluguel de temporada.

É nesse contexto que surgiu a ideia de incorporar o aprendizado de máquina para diferenciar os bloqueios. Em vez de depender estritamente de regras de negócios estáticas e simplificadas, o aprendizado de máquina permite que a Seazone treine algoritmos para reconhecer padrões e relações complexas nos dados. Isso significa que o sistema pode aprender com os próprios dados, adaptando-se dinamicamente às mudanças e nuances do mercado de aluguel de temporada.

Para explorar essa abordagem, a Seazone realizou uma série de testes em paralelo, envolvendo diversos membros de sua equipe de dados. Cada membro conduziu testes com algoritmos diferentes, visando identificar qual ou quais algoritmos de aprendizado de máquina são mais adequado para resolver o desafio de distinguir bloqueios

de reservas com precisão. Entre os algoritmos testados, destacam-se o RandomForest, regressão logística e o XGBClassifier, que são abordagens amplamente utilizadas em problemas de classificação.

Além desses métodos, também se destaca o uso de algoritmos baseados em redes neurais, que será o tema para este projeto. As redes neurais são particularmente adequados para lidar com problemas complexos de classificação e aprendizado em dados altamente dimensionais. Elas têm a capacidade de identificar padrões sutis e relações não lineares nos dados, o que pode ser crucial para diferenciar com precisão bloqueios de reservas.

# 3 TÉCNOLOGIAS E CONCEITOS

## 3.1 AMAZON WEB SERVICES (AWS)

A Amazon Web Services (AWS) é uma plataforma líder em serviços de computação em nuvem que oferece uma ampla gama de recursos para empresas e organizações. A infraestrutura da área de dados da Seazone, conforme mencionado, está hospedada na AWS, aproveitando os diversos serviços e recursos oferecidos por essa plataforma. A seguir serão explorados alguns dos serviços da AWS que desempenharam um papel importante no desenvolvimento do projeto da Seazone.

### 3.1.1 Amazon S3 (Simple Storage Service)

O Amazon S3 (AMAZON WEB SERVICES, 2023b) é um serviço de armazenamento de objetos altamente escalável e durável da AWS. Ele oferece capacidade de armazenamento virtualmente ilimitada para dados em formato de objeto, tornando-o uma escolha ideal para empresas que precisam armazenar grandes volumes de dados, como a Seazone. Um dos principais benefícios do Amazon S3 é a sua integração com outros serviços da AWS, permitindo que os dados armazenados no S3 sejam facilmente acessados e processados por outras partes do sistema. No caso da Seazone, todas as tabelas de dados são armazenadas no S3 no formato Parquet. O Parquet é um formato colunar otimizado para armazenamento de dados, o que o torna mais eficiente em termos de espaço e de consulta em comparação com formatos como CSV ou JSON.

### 3.1.2 Amazon Athena

O Amazon Athena (AMAZON WEB SERVICES, 2023a) é um serviço que desempenha um papel fundamental na análise e consulta de dados na Seazone. Uma de suas principais vantagens é permitir que os usuários executem consultas SQL em cima dos dados armazenados no Amazon S3, onde ele se beneficia do formato colunas Parquet para realizar as consultas de forma eficiente e rápida.preparação dos dados. Esse serviço foi utilizado na integração dos resultados do modelo com o "pipeline” da Seazone.

### 3.1.3 Amazon Lambda

O Lambda (AMAZON WEB SERVICES, 2023d) é um serviço de computação em nuvem da AWS que foi utilizado para executar parte do processamento dos dados. Toda a equipe é muito familiarizada com esse serviço, por esse motivo ele é sempre a primeira opção para execução de "scripts” simples.

### 3.1.4 Amazon SageMaker

O Amazon SageMaker (AMAZON WEB SERVICES, 2023c) é um serviço da AWS projetado especificamente para desenvolver, treinar e implantar modelos de aprendizado de máquina. Ele fornece uma ampla variedade de ferramentas e recursos que facilitam o ciclo de vida completo de desenvolvimento de modelos de machine learning. O SageMaker oferece suporte para uma ampla gama de algoritmos populares de aprendizado de máquina e frameworks, permitindo que os desenvolvedores criem modelos personalizados e os treinem usando grandes conjuntos de dados. Além disso, o SageMaker simplifica a implantação desses modelos em produção, fornecendo uma infraestrutura escalável e gerenciada. No contexto do projeto da Seazone, o SageMaker foi utilizado na criação e treinamento dos modelos de aprendizado de máquina que aprimoram a lógica de diferenciação entre bloqueios e reservas.

## 3.2 MACHINE LEARNING

### 3.2.1 Resumo

O aprendizado de máquina (Machine Learning, em inglês) é uma área fundamental da ciência da computação que revolucionou a forma como os sistemas e as máquinas podem aprender e melhorar com dados. Em sua essência, o aprendizado de máquina é um subcampo da inteligência artificial (IA) que se concentra no desenvolvimento de algoritmos capazes de aprender a partir de dados e tomar decisões ou fazer previsões com base nesse aprendizado.

O princípio básico do aprendizado de máquina é permitir que um sistema ou algoritmo reconheça padrões nos dados e generalize esses padrões para fazer previsões ou tomar decisões em novos conjuntos de dados. Em outras palavras, o aprendizado de máquina se concentra em capacitar as máquinas a aprenderem com exemplos passados e, em seguida, aplicar esse conhecimento para tarefas futuras.

Existem três principais tipos de aprendizado de máquina:

1. Aprendizado Supervisionado: Neste tipo, os algoritmos são treinados em um conjunto de dados que contém exemplos rotulados, ou seja, os dados de treinamento incluem pares de entrada e saída desejada. O objetivo é aprender uma função que mapeie as entradas para as saídas corretas. Isso é frequentemente usado em tarefas de classificação e regressão.

2. Aprendizado Não Supervisionado: No aprendizado não supervisionado, os algoritmos são treinados em dados que não possuem rótulos ou categorias predefinidas. O objetivo é encontrar estruturas ocultas nos dados, como grupos de itens semelhantes (agrupamento) ou redução de dimensionalidade.

3. Aprendizado por Reforço: Este tipo de aprendizado envolve um agente que toma decisões em um ambiente e aprende com as consequências dessas decisões. O agente é recompensado ou penalizado com base em suas ações, incentivando-o a aprender ações que levem a recompensas mais altas ao longo do tempo. Isso é frequentemente usado em jogos e sistemas de controle.

No contexto da Seazone, por ela administrar diversos imóveis para aluguel, ela já possuí os dados verdadeiros de seus imóveis, ou seja, ela sabe quando períodos ocupados são de fato reservas ou se são eles bloqueios. Por causa disso, o tipo de aprendizado que mais faz sentido é o supervisionado, onde os dados de treino podem ser rotulados.

No decorrer desse projeto, o desenvolvimento de modelos de aprendizado de máquina seguiram um processo típico de MLOps (Machine Learning Operations), ou seja, envolveu os seguintes passos:

1. Coleta e análise dos Dados: O primeiro passo é reunir dados relevantes para a tarefa em questão. Os dados são essenciais para treinar e avaliar o desempenho do modelo. Para o projeto, esses dados vem de tabelas que tratam os dados dos "scrappers" do Airbnb.

2. Pré-processamento de Dados: Os dados coletados geralmente precisam ser limpos, normalizados e transformados para serem usados efetivamente por redes neurais. Isso inclui a remoção de valores ausentes, codificação de recursos categóricos e muito mais.

3. Divisão dos Dados: Os dados são divididos em conjuntos de treinamento, validação e teste. O conjunto de treinamento é usado para treinar o modelo, o conjunto de validação é usado para ajustar hiperparâmetros e o conjunto de teste é usado para avaliar o desempenho final do modelo.

4. Seleção do Modelo: Com base na tarefa e nos dados, um tipo apropriado de modelo de aprendizado de máquina é selecionado. Isso pode variar de regressões lineares a redes neurais profundas, entre outros.

5. Treinamento do Modelo: O modelo é treinado usando o conjunto de treinamento, ajustando seus parâmetros para minimizar uma função de perda, que mede o quão bem o modelo está performando.

6. Avaliação do Modelo: O desempenho do modelo é avaliado usando o conjunto de validação e métricas apropriadas, como precisão, recall, erro médio quadrático, entre outras.

7. Ajuste de Hiperparâmetros: Com base nos resultados da avaliação, os hiperparâmetros do modelo podem ser ajustados para melhorar o desempenho.

8. Teste e Implantação: Finalmente, o modelo é testado no conjunto de teste para avaliar seu desempenho final. Se satisfatório, o modelo é implantado em um ambiente de produção para uso real.

Também é importante salientar que esse processo é continuo, iterativo e, idealmente, deve-se ser automatizado para os modelos sempre possuírem os dados mais recentes e não perderem relevância com o tempo.

#### 3.2.1.1 Métricas de validação

Métricas de validação desempenham um papel crucial na avaliação do desempenho de modelos de aprendizado de máquina. No contexto deste projeto de classificação binária, são utilizadas quatro métricas principais: precisão, recall, F1-score e acurácia. Essas métricas ajudam a entender como o modelo está se comportando em relação às previsões e aos resultados reais (HOSSIN; SULAIMAN, 2015). A Tabela 2 apresenta uma matriz de confusão que ajuda a entender elas.

| | Previsão: Sim | Previsão: Não |
|---|---|---|
| Real: Sim | Verdadeiro Positivo | Falso Positivo |
| Real: Não | Falso Positivo | Verdadeiro Negativo |

Tabela 2 – Exemplo de matrix de confusão em classificação binária.

- Acurácia (Accuracy): A acurácia mede a proporção de previsões corretas em relação ao número total de amostras. Em outras palavras, é a capacidade do modelo de classificar corretamente as instâncias. A fórmula da acurácia é:

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos} + \text{Verdadeiros Negativos}}{\text{Total de Amostras}} \tag{1}$$

  A acurácia fornece uma visão geral do quão bem o modelo está realizando a classificação. No entanto, ela pode ser enganosa em casos de conjuntos de dados desequilibrados, nos quais uma classe é muito mais frequente do que a outra.

- Precisão (Precision): A precisão mede a proporção de verdadeiros positivos (instâncias classificadas corretamente como positivas) em relação a todas as previsões positivas. Em outras palavras, é a capacidade do modelo de não fazer previsões positivas errôneas. A fórmula da precisão é:

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \tag{2}$$

  A precisão é especialmente útil quando os falsos positivos têm custos significativos.

- Recall (Recall): O recall, também conhecido como sensibilidade ou taxa de verdadeiros positivos, mede a proporção de verdadeiros positivos em relação a todas as instâncias que realmente pertencem à classe positiva. Em outras palavras, é a capacidade do modelo de identificar todas as instâncias positivas. A fórmula do recall é:

$$\text{Recall} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \tag{3}$$

  O recall é particularmente importante quando é fundamental evitar falsos negativos, ou seja, quando a perda de uma previsão positiva pode ter sérias consequências.

- F1-Score (F1-Score): O F1-score é uma métrica que combina precisão e recall em uma única medida, fornecendo um equilíbrio entre as duas métricas. É a média harmônica da precisão e do recall e é particularmente útil quando se deseja equilibrar o compromisso entre essas duas métricas. A fórmula do F1-score é:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{4}$$

  O F1-score é uma métrica útil quando as classes estão desequilibradas e quando se deseja um compromisso entre a minimização de falsos positivos e falsos negativos.

Dentre todas essas métricas, a mais importante para o projeto é a maximização do F1 Score, visto que essa é a mais equilibrada e não há preferência entre evitar falsos positivos ou falsos negativos.

#### 3.2.1.2 Importância do MAPE para a Seazone

As métricas comentadas acima são muito importantes para a classificação binária, entretanto, elas não são as únicas métricas utilizadas no projeto. O Erro Percentual Absoluto Médio (MAPE) é uma métrica que fornece uma medida robusta e intuitiva da precisão das previsões em relação ao faturamento real. A Seazone utiliza o MAPE como principal método de validação, aplicando-o na granularidade mensal para medir o erro no faturamento. Normalmente, essa análise é realizada apenas em cima dos dados de faturamento obtidos através da detecção de bloqueios heurística, mas no final do projeto o modelo obtido também será medido por essa métrica.

A fórmula do MAPE é expressa por:

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{F_i - A_i}{A_i} \right| \times 100, \tag{5}$$

onde $F_i$ representa o faturamento previsto pelo modelo analisado, $A_i$ é o faturamento real, e $N$ é o número total de observações, sendo que cada observação é o faturamento mensal de um imóvel. O MAPE calcula a média das porcentagens absolutas dos erros para cada observação, proporcionando uma visão clara da precisão das previsões em termos percentuais.

O MAPE destaca não apenas a precisão média das previsões, mas também a magnitude dos erros relativos. Essa abordagem é relevante no contexto de gestão de faturamento da Seazone, onde pequenas discrepâncias podem ter impactos de previsões significativos.

Entretanto, o cálculo do MAPE não é perfeito. Em circunstâncias em que $A_i = 0$ (faturamento real igual a zero) e $F_i \neq 0$ (faturamento previsto diferente de zero), a divisão na fórmula do MAPE se tende ao infinito, resultando em um cálculo que perde sua interpretação prática. Para lidar com essa situação, a observação é desconsiderada, evitando distorções no MAPE. No entanto, a contagem dessas ocorrências é registrada numa métrica criada denominada "n_inf", refletindo o número de casos em que a divisão indefinida ocorreu.

A Seazone também se preocupa em minimizar os "n_inf", então essa métrica é importante de ser usada na validação do modelo. Também é importante notar que, quando tanto o faturamento real quanto o previsto são zero, a observação é considerada sem erro, resultando em um MAPE de 0% para esta obervação. Essa consideração reflete a intenção de reconhecer acertos mesmo em casos em que não há faturamento.

#### 3.2.1.3 Cross Validation

A validação cruzada ajuda a avaliar o desempenho e a lidar com problemas como "overfitting" , que ocorre quando um modelo se ajusta muito bem ao conjunto de treinamento, capturando até mesmo o ruído nos dados. Como resultado, o modelo tem um desempenho excepcional no conjunto de treinamento, mas seu desempenho em novos dados (conjunto de teste) é ruim. Isso acontece porque o modelo aprendeu relações específicas dos dados de treinamento e não consegue generalizar bem para novos exemplos.

Em vez de simplesmente dividir os dados em um conjunto de treinamento e um conjunto de teste, a validação cruzada divide os dados em várias partes (chamadas "folds"). O modelo é treinado em uma parte dos dados e testado nas outras partes. Esse processo é repetido várias vezes, permitindo uma avaliação mais robusta do desempenho do modelo.

A validação cruzada ajuda a mitigar os problemas de "overfitting" , fornecendo uma avaliação mais confiável do desempenho do modelo em diferentes conjuntos de dados. Também é útil para escolher os hiperparâmetros do modelo, ajustando-os de

acordo com o desempenho médio nas dobras de validação. Por esses motivos, ela foi amplamente utilizada no descorrer deste projeto.

#### 3.2.1.4 Hiperparametros

A busca por hiperparâmetros é uma parte fundamental do processo de desenvolvimento de modelos de aprendizado de máquina, incluindo redes neurais. Hiperparâmetros são configurações do modelo que não são aprendidas pelos algoritmos de treinamento, mas precisam ser definidas antes de iniciar o treinamento do modelo.

Alguns exemplos de hiperparâmetros em redes neurais incluem a taxa de aprendizado (learning rate), o número de camadas ocultas, o número de neurônios em cada camada, as funções de ativação, os algoritmos de otimização, as taxas de dropout, a regularização, entre outros.

A busca por hiperparâmetros geralmente segue um processo iterativo, onde o primeiro passo é definir um conjunto de hiperparâmetros iniciais, que podem ser escolhidos com base em conhecimento prévio ou em valores padrão. Em seguida, os dados são divididos em três conjuntos: treinamento, validação e teste.

- Conjunto de Treinamento: É usado para treinar o modelo com um conjunto específico de hiperparâmetros.
- Conjunto de Validação: Após o treinamento, o modelo é avaliado no conjunto de validação para medir seu desempenho. Com base nas métricas de desempenho, como precisão ou perda, os hiperparâmetros podem ser ajustados.
- Conjunto de Teste: Este conjunto é usado para avaliar o desempenho final do modelo depois que todos os ajustes nos hiperparâmetros foram feitos.

Existem várias técnicas que automatizam a busca de hiperparâmetros. Os principais métodos incluem:

- Busca em Grade (Grid Search): Essa abordagem consiste em definir um conjunto de valores possíveis para cada hiperparâmetro e combinar todas as combinações possíveis. É um método simples, mas pode ser computacionalmente caro.
- Otimização Bayesiana: A otimização bayesiana é um método mais avançado que usa probabilidades para selecionar os hiperparâmetros mais promissores. Ele modela a função de desempenho do modelo e faz escolhas informadas com base em observações anteriores.

Como redes neurais são mais lentas para realizar o treinamento, uma busca em grade não é muito viável, visto que a quantidade de parâmetros a serem testados é

muito grande. Por esse motivo, foi utilizado a Otimização Bayesiana na procura dos hiperparâmetros.

Por último, a validação cruzada também importante na busca por hiperparâmetros. Idealmente, os modelos são avaliados em várias divisões de treinamento/validação para obter uma estimativa mais confiável do desempenho do modelo. Isso ajuda a evitar o "overfitting" dos hiperparâmetros para um único conjunto de validação.

### 3.2.2 Redes neurais

As redes neurais são um subconjunto poderoso e sofisticado do campo do aprendizado de máquina, inspirado pelo funcionamento do cérebro humano. Elas têm a capacidade de resolver tarefas complexas de aprendizado de máquina, como classificação, regressão, processamento de linguagem natural, visão computacional e muito mais.

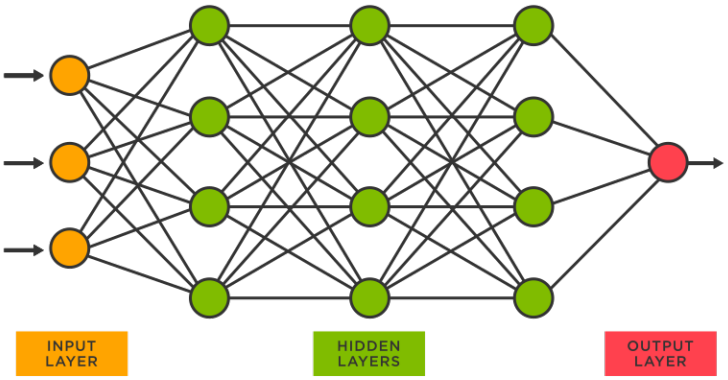
#### 3.2.2.1 Principais Componentes e Arquiteturas das Redes Neurais

Para compreender melhor as redes neurais, é fundamental conhecer seus componentes (SINGH, 2023), arquiteturas e como elas funcionam. Os principais componentes das redes neurais são:

- Neurônios: Os neurônios são a unidade fundamental de uma rede neural. Eles recebem entradas, realizam cálculos e geram saídas. Cada neurônio aplica uma função de ativação às entradas ponderadas e produz uma saída.
- Camadas: As redes neurais são compostas por camadas de neurônios. A Figura 1 apresenta uma imagem contendo as difenatos camadas, sendo que existem três tipos principais de camadas em uma rede neural:
    1. Camada de Entrada: Recebe os dados de entrada e repassa para a rede.
    2. Camadas Ocultas: Realizam cálculos intermediários e transformações dos dados.
    3. Camada de Saída: Produz as saídas finais da rede neural, que podem ser valores contínuos ou probabilidades de classes em problemas de classificação.
- Pesos e Bias: Cada conexão entre neurônios é associada a um peso que controla a influência dessa conexão na saída. Além disso, um valor chamado de bias é adicionado ao cálculo da saída do neurônio. Os pesos e bias são ajustados durante o treinamento da rede neural.

Além dos componentes citados anteriormente, as redes neurais também possuem diferentes tipos de arquitetura (PAI, 2023). As principais arquiteturas são:

Figura 1 – Rede neural simples.




Fonte: (CLOUD SOFTWARE GROUP, 2023)

1. Redes Neurais Feedforward (FNN): Essa é a arquitetura mais simples de rede neural. A informação flui em uma única direção, da camada de entrada para a camada de saída. Não há ciclos nas conexões, o que torna o processo de treinamento relativamente direto.

2. Redes Neurais Convolucionais (CNN): Essas redes são projetadas para processar dados de grade, como imagens. Elas contêm camadas de convolução que aprendem a detectar recursos locais em imagens, tornando-as ideais para tarefas de visão computacional.

3. Redes Neurais Recorrentes (RNN): As RNNs são usadas para dados sequenciais, como séries temporais e processamento de linguagem natural. Elas têm conexões ciclícas que permitem que informações anteriores afetem as previsões futuras.

4. Redes Neurais Convolucionais Recorrentes (CRNN): Essas redes combinam características de CNNs e RNNs, sendo úteis em tarefas que envolvem análise de dados de sequência em uma estrutura espacial, como vídeos.

#### 3.2.2.2 Retropropagação do erro (*Backpropagation*)

O funcionamento de uma rede neural envolve a propagação de informações da camada de entrada para a camada de saída por meio das camadas ocultas (caso houverem). Cada neurônio realiza um cálculo que envolve a soma ponderada das entradas, e a aplicação dos pesos e bias ajustados para minimizar uma função de perda , que quantifica o quão bem a rede está realizando a tarefa desejada. Esse processo de ajuste é feito usando algoritmos de otimização, como o gradiente descendente e é chamado de backpropagation.

O processo de backpropagation compara a saída prevista com a saída real para calcular o erro. Esse erro é propagado de volta através da rede, e os gradientes são calculados para cada neurônio em relação ao erro. Os gradientes indicam o quanto os pesos e bias devem ser ajustados para reduzir o erro. É importante mencionar que a taxa de aprendizado (learning rate) desempenha um papel fundamental nesse processo, pois controla o tamanho dos ajustes dos parâmetros e é um hiperparâmetro da rede.

Dentro do contexto deste projeto, dois algoritmos de otimização que utilizam o backpropagation foram utilizados: Adam e RMSprop (SANGHVIRAJIT, 2021). A principal diferença entre os dois é que o Adam usa o momento para acelerar o treinamento. Ele calcula médias móveis dos gradientes de primeira ordem (primeiro momento) e dos gradientes de segunda ordem (segundo momento), que ajudam a ajustar a taxa de aprendizado para cada parâmetro individualmente.

#### 3.2.2.3 Funções de Ativação

A função de ativação introduz não linearidade na rede, permitindo que ela capture relações complexas nos dados.

Existem diferentes tipos de funções de ativação no contexto de redes neurais (SHARMA, Sagar; SHARMA, Simone; ATHAIYA, 2017), sendo que três foram amplamente utilizadas no decorrer deste projeto e por isso são descritas abaixo.

1. A função de ativação ReLU (Rectified Linear Activation) é uma das funções mais utilizadas em redes neurais profundas. Ela é simples e eficaz, introduzindo não linearidade nas camadas da rede. A ReLU atribui zero a todos os valores negativos e deixa os valores positivos inalterados, como demostra a Equação (6).

$$f(x) = \begin{cases} x, & \text{se } x > 0 \\ 0, & \text{se } x \leq 0 \end{cases} \tag{6}$$

2. A função de ativação Swish é uma função mais recente que tem ganhado popularidade. Ela é uma variação suave da função ReLU. A principal diferença é que a Swish introduz uma curva suave, em vez de ser estritamente linear para valores positivos.

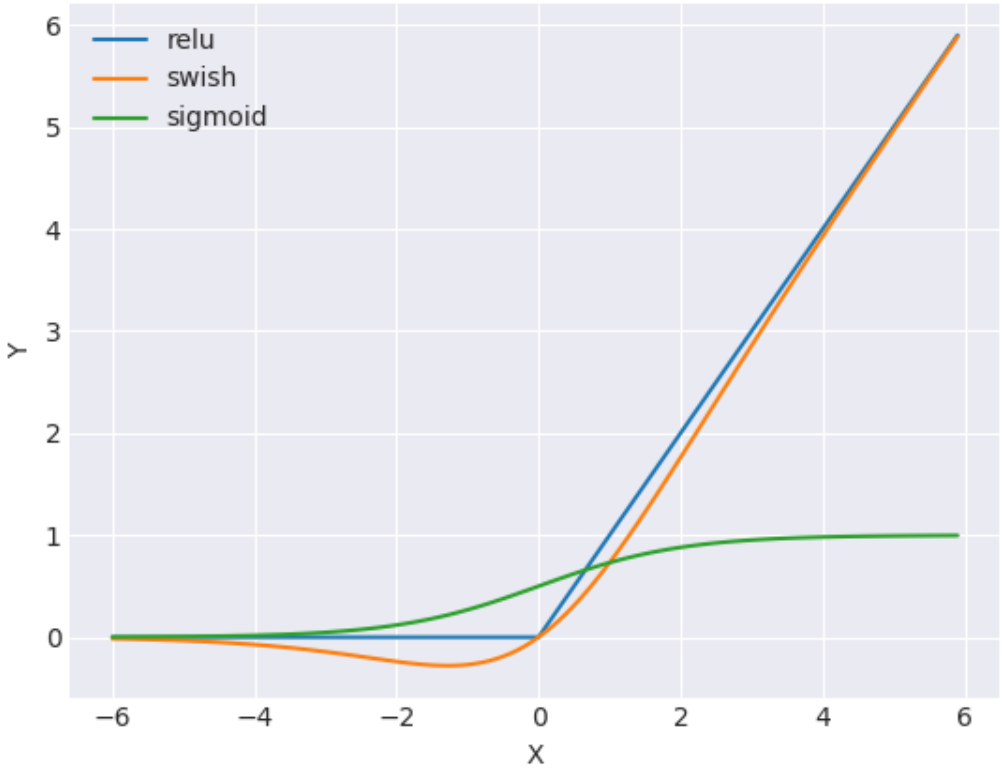
$$f(x) = \frac{x}{1 + e^{-x}} \tag{7}$$

3. A função de ativação Sigmoid é amplamente utilizada em redes neurais, especialmente em tarefas de classificação binária. Ela foi usada neste projeto como função de ativação da camada de saída. Ela mapeia os valores de entrada para o intervalo entre 0 e 1. A Sigmoid é uma função logística que possui uma curva em forma de "S".

$$f(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Essas são algumas das funções de ativação comumente usadas em redes neurais. Cada uma delas tem suas características e a escolha adequada pode impactar significativamente o desempenho do modelo de rede neural. A Figura 2 apresenta uma imagem contendo os 3 tipos de funções descritas previamente.

Figura 2 – Funções de Ativação.




Fonte: Autor.

#### 3.2.2.4 Funções de Callback

Funções de callback são componentes importantes no treinamento de redes neurais que permitem monitorar e controlar o processo de treinamento. Elas são acionadas em momentos específicos do treinamento para realizar uma tarefa específica. Neste projeto, duas funções de callback foram utilizadas: EarlyStopping e ReduceLROnPlateau.

A função de callback **EarlyStopping** é uma técnica utilizada para prevenir o "overfitting". O EarlyStopping monitora uma métrica, como o f1_score, no conjunto de validação durante o treinamento. Se a métrica no conjunto de validação parar de melhorar ou piorar, o treinamento é interrompido mais cedo para evitar o excesso de ajuste. Isso ajuda a economizar tempo e recursos computacionais, impedindo que o modelo treine além do ponto ideal, além de diminuir o "overfitting".

A função de callback **ReduceLROnPlateau** é usada para ajustar dinamicamente a taxa de aprendizado durante o treinamento. Ela monitora uma métrica específica, como o loss no conjunto de validação, e reduz a taxa de aprendizado se a métrica parar de melhorar. Isso é particularmente útil para evitar que o treinamento fique estagnado em uma fase de convergência lenta. Reduzir a taxa de aprendizado permite que o modelo faça ajustes mais finos e encontre uma solução melhor.

É importante comentar que, durante a busca dos hiperparâmetros ou durante a implantação do modelo final, o conjunto de dados utilizado para a validação da função de callback, idealmente, tem que ser diferente do conjunto de validação usado para calcular as métricas finais. Isso é para evitar o "overfitting" durante o treinamndo do modelo.

#### 3.2.2.5 Técnicas de regularização

Os regularizadores (DEVELOPERS, 2023) são utilizados no controle do "overfitting" em modelos de redes neurais, uma vez que oferecem maneiras eficazes de penalizar a complexidade do modelo. Existem diferentes tipos de regularizadores, e dois dos mais comuns são os regularizadores L1 e L2. O regularizador L1 penaliza a função de custo adicionando a soma dos valores absolutos dos pesos, o que pode resultar em alguns pesos zerados. Por outro lado, o regularizador L2 penaliza a função de custo com a soma dos quadrados dos pesos, forçando os pesos a se aproximarem de zero, mas não se tornando exatamente zero.

Outra técnica utilizada é a adição de "camadas de dropout". Durante o treinamento, essas camadas selecionam aleatoriamente uma porcentagem das conexões para serem desativadas em cada iteração. Isso impede que o modelo se torne excessivamente dependente de certas conexões, promovendo uma maior generalização dos dados. O "dropout" atua como um mecanismo de regularização ao introduzir um componente de aleatoriedade no treinamento da rede neural, o que contribui para a robustez do modelo em relação aos dados de entrada.

### 3.2.3 Python e bibliotecas

A escolha da linguagem de programação e das bibliotecas é uma parte crucial de qualquer projeto de aprendizado de máquina, e no caso do projeto da Seazone, Python foi a escolha acertada. Python é amplamente reconhecida e adotada na comunidade de aprendizado de máquina por várias razões, incluindo sua simplicidade, vasta quantidade de bibliotecas disponíveis e suporte ativo da comunidade de desenvolvedores.

O Pandas é uma das bibliotecas mais amplamente utilizadas para manipulação e análise de dados em Python. Ele oferece estruturas de dados flexíveis, como DataFrames, que são ideais para armazenar e processar dados tabulares. o Pandas

desempenhou um papel fundamental na limpeza e tratamento de dados, permitindo a preparação dos dados para treinamento de modelos de forma eficiente.

O Matplotlib e o Seaborn são bibliotecas de visualização de dados poderosas para Python. Eles permitem criar uma ampla variedade de gráficos e visualizações de dados, o que é fundamental para a analise exploratória e comunicar resultados de maneira eficaz. No projeto, essas bibliotecas ajudaram em criam visualizações informativas para compreender melhor os padrões nos dados.

O Scikit-Learn (PEDREGOSA *et al.*, 2011) é uma biblioteca de aprendizado de máquina de código aberto que oferece uma ampla variedade de algoritmos, métricas de avaliação e ferramentas de pré-processamento de dados. Ele foi usado para dividir os dados em conjuntos de treinamento, validação e teste, bem como para normalizar os dados, garantindo que os modelos de aprendizado de máquina funcionassem efetivamente.

Existem diversas bibliotecas e "frameworks" utilizados na criação e desenvolvimento de modelos com redes neurais. Dentre ela, a escolhida foi o Keras (CHOLLET, Francois *et al.*, 2015) , que é uma API de alto nível para desenvolvimento de redes neurais que roda sobre o TensorFlow, uma das principais bibliotecas de aprendizado de máquina de código aberto. Essa combinação oferece uma maneira rápida e eficaz de projetar, treinar e implantar redes neurais. Além dele ser prático para uso, o seu uso também está alinhado com o fato de os outros integrantes da equipe serem mais familiarizados com ele.

Existem diversas bibliotecas que podem ser utilizadas em conjunto com Keras, uma delas é a keras-tuner (O'MALLEY *et al.*, 2019). Essa é uma ferramenta utilizada para facilitar a procura por hiperparametros em redes neurais. Com várias estratégias de otimização, incluindo a otimização bayesiana, o Keras-Tuner automatiza a seleção de hiperparâmetros, economizando tempo e esforço no processo de ajuste fino dos modelos.

# 4 PROPOSTA E REQUISITOS DO PROJETO

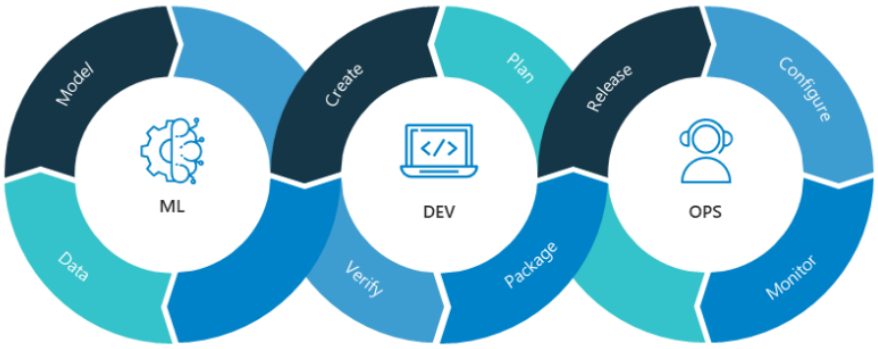
## 4.1 PROPOSTA

A proposta do projeto é desenvolver um modelo de aprendizado de máquina baseado em redes neurais para classificar e diferenciar reservas (criadas por hóspedes) de bloqueios (feitos pelos proprietários).

Com o melhor modelo treinado e avaliado, ele deve ser implantado em ambiente de produção, onde ele será usado para melhorar a diferenciação de bloqueios que indiretamente melhora a previsão de faturamento do Airbnb. Também é importante que o modelo em produção seja constantemente re-treinado com novos dados e esteja sempre atualizado.

O processo de aquisição de dados, criação de features, treinamento de modelos, avaliação e assim por diante, é um processo contínuo de desenvolvimento. A Figura 3 apresenta um esquema que resume bem os principais passos de MLOps esperado se encontrar neste projeto.

Figura 3 – Ciclo do MLOps.




Fonte: (MERRITT, 2020)

## 4.2 REQUISITOS

Para levantar os requisitos do projeto, foram realizadas discussões com a equipe de dados para entender melhor as necessidades e o esperado para o projeto.

Os requisitos funcionais descrevem as funcionalidades específicas que o sistema deve fornecer para atender às necessidades do usuário. Eles são declarativos e detalham o que o sistema deve fazer. Os definidos para este projeto são:

- **Análise e preparação de dados:** Em todo o projeto de aprendizado de máquina é necessário realizar um estudo dos dados e levantamento de "features". Mesmo

se o modelo final não apresentar resultados satisfatórios, as "features" e análises geradas podem ser utilizadas novamente em outros projetos internos da empresa.

- **Obtenção de um modelo otimizado:** No final do projeto é esperado a obtenção de pelo menos um modelo otimizado capaz de diferenciar reservas de bloqueios do Airbnb.
- **Validação com dados do Airbnb:** É importante realizar testes e validações em cima dos dados do Airbnb. O intuito é garantir se há limitações e, caso haja, definir quais são. Também é importante medir o desempenho comparando o resultado do modelo com dados reais da Seazone.

Os requisitos não funcionais referem-se a atributos ou características do sistema que não estão relacionados diretamente às funcionalidades específicas, mas que afetam seu desempenho, escalabilidade, entre outros aspectos. Eles definem as características de qualidade e restrições sob as quais o sistema deve operar. Os requisitos não funcionais do projeto são:

- **Desempenho:** O modelo deve ser capaz de lidar eficientemente com grandes volumes de dados, garantindo tempos de resposta aceitáveis. Além disso, idealmente, a qualidade das predições também precisam ser superiores ao do método heurístico já utilizado na Seazone.
- **Custo:** O sistema deve ser projetado considerando a otimização de custos, utilizando recursos de nuvem de maneira eficiente.
- **Modelo funcionando em produção automaticamente:** É importante que, se o modelo final encontrado for capaz de melhorar a predição de bloqueios do Airbnb, o modelo seja implantado em produção. Isso incluí a automatização da criação de "features", retreino do modelo e predição de dados. Além disso, também é importante que o modelo alimente as tabelas de faturamento da Seazone.

# 5 DESENVOLVIMENTO

## 5.1 ANALISE EXPLORATÓRIA

A análise exploratória de dados é uma etapa fundamental em projetos de aprendizado de máquina, pois permite compreender melhor os dados disponíveis, identificar padrões, anomalias e insights que podem ser relevantes para o desenvolvimento do modelo.

### 5.1.1 Tabelas

Esta seção irá comentar de algumas das principais tabelas que a Seazone possuí de dados referentes ao Airbnb. Apenas as colunas e atributos mais importantes e relevantes para este projeto são descritas.

#### 5.1.1.1 reservations (Airbnb)

As tabelas de dados são a base da análise exploratória. Uma das tabelas centrais do projeto é a "reservations", que contém informações sobre os períodos ocupados resultantes dos scrappers comentados na Seção 2.2. No entanto, é importante notar que, nesta fase, ainda não se sabe se esses períodos são de fato reservas confirmadas ou bloqueios feitos pelos proprietários dos imóveis.

A Tabela 3 apresenta um exemplo de dados dessa tabela, que inclui informações como o ID do imóvel no Airbnb (airbnb_listing_id), a data em que o período passou a ser ocupado (formatted_booked_on), as datas de check-in e check-out, a duração da estadia (length_of_stay), a antecedência (advance) e o preço médio das diárias (night_price). Esses dados são essenciais para a análise e treinamento do modelo.

| airbnb_listing_id | formatted_booked_on | checkin | checkout | length_of_stay | advance | night_price |
|---|---|---|---|---|---|---|
| 826761560179331990 | 2023-05-16 06:32:14.000 | 2023-05-16 | 2023-05-17 | 1 | 0 | 201.0 |
| 826763335369774245 | 2023-08-04 19:53:47.000 | 2023-08-04 | 2023-08-05 | 1 | 0 | 490.0 |
| 826764680344819629 | 2023-08-21 21:26:06.000 | 2023-08-21 | 2023-08-22 | 1 | 0 | 5500.0 |
| 828207590326833836 | 2023-08-31 20:34:59.000 | 2023-09-07 | 2023-09-09 | 2 | 7 | 90.0 |
| 829803451234232635 | 2023-05-16 07:49:00.000 | 2023-05-19 | 2023-05-21 | 2 | 3 | 235.0 |

Tabela 3 – Exemplo de dados da tabela reservations (Airbnb).

#### 5.1.1.2 reservations (Seazone)

Uma das características mais importantes da tabela de reservas da Seazone é a sua confiabilidade. Os dados presentes nesta tabela são gerados a partir das informações internas dos imóveis que a empresa administra. Isso significa que a Seazone possui conhecimento direto e preciso sobre cada período ocupado em seus imóveis, permitindo a diferenciação eficaz entre bloqueios e reservas reais. Diferentemente da

tabela de reservas do Airbnb, que pode conter informações ambíguas, a tabela da Seazone oferece uma base sólida para o treinamento supervisionado do modelo.

Além da confiabilidade dos dados, ela também possuí mais duas diferenças em relação a tabela de reservations (Airbnb):

1. Coluna "blocked”: Esta coluna é do tipo booleano (Verdadeiro/Falso) e serve para indicar se um determinado período é um bloqueio (Verdadeiro) ou uma reserva (Falso). Essa distinção é fundamental para o desenvolvimento do modelo, pois fornece rótulos precisos para os dados.

2. Coluna "price”: Essa coluna representa o preço médio associado a uma reserva. No entanto, ela se diferencia da coluna "night_price” presente na tabela de reservas do Airbnb. Enquanto a "night_price” do Airbnb fornece o preço médio das diárias na data de ocupação, a coluna "price” na tabela da Seazone é nula (sem valor) quando se trata de um bloqueio. Isso ocorre porque, em casos de bloqueio, as informações de preço tornam-se irrelevantes e não são armazenadas de forma direta nas tabelas da empresa. É possível, no entanto, a partir da data de criação desses bloqueios e do preço sendo ofertado nos imóveis, de se obter um preço médio para esses bloqueios, o problema é que os preços ofertados apenas começaram a ser armazenado de forma interna a partir de 2022, então todas os bloqueios anteriores a esse ano continuariam com "price” nulo.

A informação sobre o preço é crucial a ser destacada, pois os modelos de aprendizado de máquina serão treinados principalmente com base nos dados da tabela de reservas da Seazone. Esta tabela fornece o rótulo para distinguir entre reservas e bloqueios, mas é importante notar que as informações de preço podem estar ausentes quando se trata de bloqueios dificultando a criação de "features” com base neles.

#### 5.1.1.3 details

A "details” é uma tabela que traz características gerais dos anúncios do Airbnb, sendo que ela é atualizada semanalmente através de um "web scrapper”. Os dados presentes nela incluem o ID do imóvel no Airbnb, o número de banheiros, quartos e camas, a latitude e a longitude da localização, a avaliação dos hóspedes, o número máximo de hóspedes permitidos, o número de avaliações recebidas, a taxa de limpeza, a data de aquisição dos dados, entre outros. Cada um desses atributos podem ser importantes para entender as características de um anúncio e sua influência nas reservas. A Tabela 4 apresenta um exemplo dos dados presentes nela, vale ressaltar que algumas das informações variam no decorrer do tempo, então é importante ter isso em mente ao montar o conjunto de dados de treino do modelo.

Os atributos presentes nela podem fazer sentido num algoritmo de aprendizado de máquina, visto que é de conhecimento do negócio que o comportamento dos

| airbnb_listing_id | number_of_bathrooms | number_of_bedrooms | number_of_beds | latitude | longitude | star_rating | number_of_guests | number_of_reviews | cleaning_fee | acquisition_date | listing_type |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 41435891 | 1 | 1 | 6 | -20.72196 | -48.91867 | 0.0 | 6 | 0 | 60 | 2020-12-30 | hotel |
| 16610915 | 3.0 | 3 | 5 | -26.66720 | -48.68630 | 5.0 | 7 | 32 | 80.0 | 2023-10-16 | casa |
| 21923594 | 2.0 | 2 | 4 | -26.63080 | -48.69120 | 0.0 | 5 | 0 | 0.0 | 2022-12-12 | casa |
| 47307951 | 1.0 | 1 | 8 | -23.03022 | -44.16601 | 4.6 | 8 | 5 | 0.0 | 2023-08-28 | apartamento |
| 46139666 | 1.0 | 0 | 1 | -26.89167 | -48.64798 | 0.0 | 2 | 0 | 0.0 | 2021-10-07 | casa |

Tabela 4 – Exemplo de dados da tabela details.

anúncios tende a variar dependo de suas características físicas e comportamentais. Clientes que buscam alugar um hotel normalmente não estão procurando por uma casa e vice-versa. Além disso, imóveis com um maior número de avaliação implicam que ele é mais reservado. Esses são apenas alguns exemplos de diferenças comportamentais dentre as presentes características da "details".

#### 5.1.1.4 dates

A tabela "dates" é usada para auxiliar a realização de algumas consultas nos bancos de dados da Seazone. Ela atua como um recurso auxiliar, fornecendo informações úteis sobre as datas em questão. Essa tabela contém informações sobre feriados nacionais, a classificação de datas como dias da semana ou finais de semana e a identificação de datas que estavam dentro do período da pandemia de COVID-19.

Esses dados podem ajudar o modelo, visto que os feriados nacionais são eventos relevantes que podem influenciar significativamente a demanda por aluguel de temporada. Além disso, a tabela "dates" também identifica os fins de semana, e é de conhecimento do negócio que eles também tendem a possuir mais ocupação do que nos dias de semana. Portanto, a inclusão dessas informações no modelo deve aumentar sua qualidade de fazer previsões.

Por último, os dados da pandemia de COVID-19 também podem trazer "insights" importantes para o modelo, já que esse evento teve um impacto significativo nas tendências de aluguel de temporada. A Tabela 5 apresenta algumas linhas dessa tabela.

| date | holiday | weekend | pandemic |
|---|---|---|---|
| 2017-10-03 | nenhum | False | False |
| 2020-03-14 | nenhum | True | True |
| 2023-02-17 | Carnaval | True | False |
| 2023-09-07 | Independência do Brasil | False | False |
| 2023-12-27 | nenhum | False | False |

Tabela 5 – Exemplo de dados da tabela dates.

### 5.1.2 Visualização dos Dados

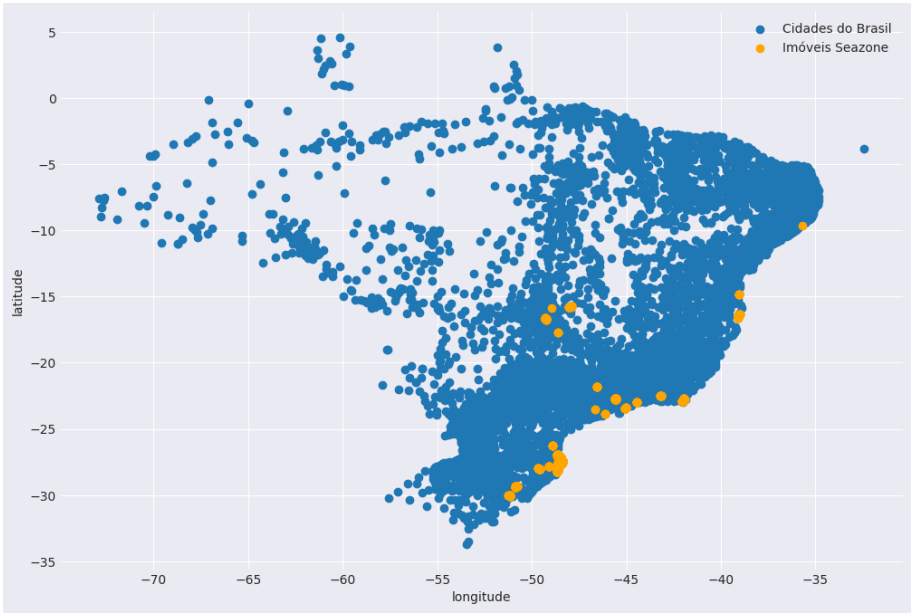
A análise exploratória e visualização dos dados é uma etapa crucial em qualquer projeto de aprendizado de máquina, pois ajuda a compreender o conjunto de dados,

identificar padrões e problemas, bem como orientar as próximas fases do projeto. Num primeiro momento, foi realizado uma analise focando principalmente nas features das tabelas descritas na Seção 5.1.1.

**Distribuição Geográfica dos Anúncios**

O primeiro aspecto analisado foi a distribuição geográfica dos anúncios da Seazone em relação às cidades brasileiras. O gráfico da Figura 4 mostra a latitude e longitude das cidades do Brasil, com os anúncios da Seazone destacados em laranja. É notado que os imóveis da Seazone estão concentrados em um número limitado de cidades brasileiras. Isso sugere que o uso direto da latitude e longitude como características pode não ser representativo de todo o conjunto de dados, uma vez que a maioria das cidades não possui anúncios da Seazone.

Figura 4 – Imóveis do Brasil e localização dos anúncios da Seazone.




Fonte: Autor.

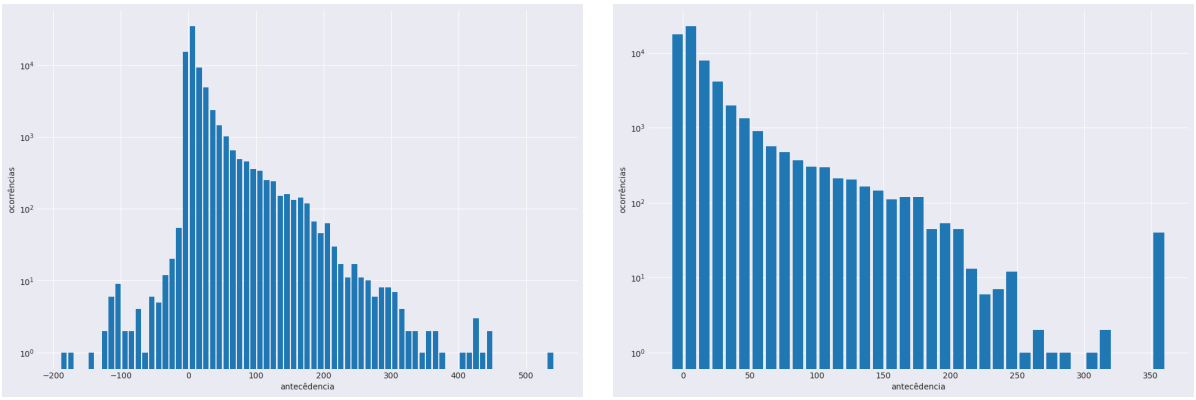
**Problema de Antecedência nas Reservas (Advance)**

Outro ponto crítico observado envolveu a coluna "advance" (antecedência) na tabela "reservations (Seazone)". Foi descoberto que alguns registros apresentavam datas de criação de reserva posteriores à data de check-in. Isso provavelmente ocorreu devido a problemas internos na hora de salva-las nas tabelas internas da empresa, onde elas foram refeitas posteriormente o período da reserva/bloqueio. Os gráficos da 5 comparam a antecedência das reservas na tabela da Seazone com as reservas do Airbnb, percebe-se que as provenientes de dados internos da Seazone possuem antecedências de -200 dias, enquanto que as do Airbnb sempre estão entre 0 e 365 dias. Dado que o treinamento do modelo será baseado nos dados da "Reservations

(Seazone)", é imperativo realizar uma limpeza nessa coluna para garantir a qualidade dos dados de treinamento e também para não corromper futuras normalizações.

Figura 5 – Gráficos da antecedência pelo número de ocorrências.

(a) Reservation (Seazone). (b) Reservation (Airbnb).




Fonte: Autor.

.

**Tamanho da estadia**

O tamanho da estadia é uma "feature" que possui bastante correlação com o fato de ser bloqueio ou reserva. A Figura 6 demonstra que períodos pequenos de tamanho 1 ou períodos grandes tendem a ser bloqueios.

Figura 6 – Relação do tamanho da estadia e número de bloqueios.

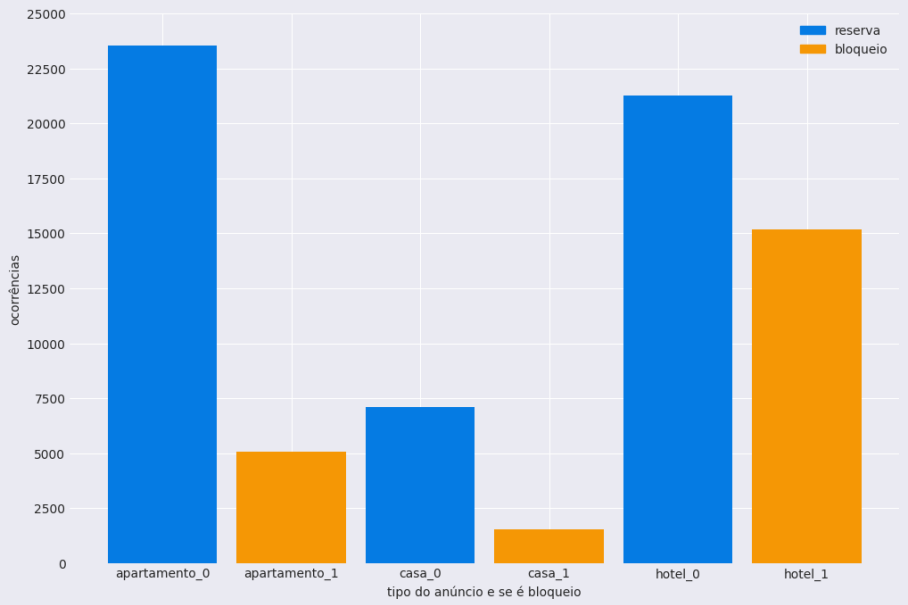

Fonte: Autor.

**Impacto do Tipo de Imóvel**

Outra investigação importante foi referente ao impacto do tipo de imóvel (apartamento, casa ou hotel) no número de bloqueios. O gráfico da Figura 7 revela que os hotéis tendem a apresentar mais bloqueios em comparação com casas ou apartamentos. Essa observação sugere uma possível correlação entre o tipo de imóvel e a

variável dependente, que é a previsão de bloqueios. Essa informação é crucial, uma vez que diferentes tipos de imóveis podem ter comportamentos distintos em relação às reservas e bloqueios.

Figura 7 – Gráfico do tipo do imóvel e se é bloqueio pelo número de ocorrências.




Fonte: Autor.

**Influência de Número de Hóspedes, Banheiros, Quartos e Camas**

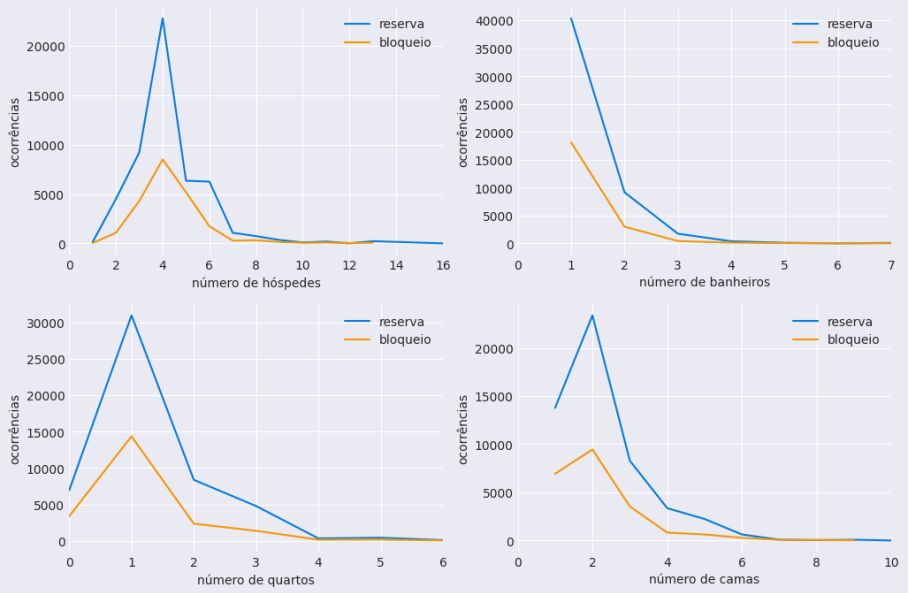
Além disso, foi investigado se o número de hóspedes, banheiros, quartos e camas influencia no número de bloqueios. O gráfico da Figura 8 mostra a análise dessas características em relação aos bloqueios. Inicialmente, não parece haver uma correlação clara entre essas "features" e o número de bloqueios. A proporção de bloqueios em relação às reservas parece relativamente constante em diferentes configurações de hóspedes, banheiros, quartos e camas.

## 5.2 PREPARAÇÃO DOS DADOS

### 5.2.1 Limpeza dos Dados

A limpeza de dados consiste em principalmente preencher valores nulos ou errados com outros valores, ou até mesmo eliminar completamente as linhas erradas. Nesse projeto, uma das operações mais significativas envolveu a coluna de "antecedência" na tabela "reservations (Seazone)". Como comentado anteriormente na Seção 5.1.2, foi constatado a presença de valores negativos nessa coluna, o que claramente não fazia sentido em termos de antecedência de reserva. Para abordar esse problema, decidiu-se limitar os valores dessa coluna a um intervalo mais plausível,

Figura 8 – Gráficos dos números de cômodos do imóvel e se é bloqueio pelo número de ocorrências.




Fonte: Autor.

entre 0 e 365 dias. Isso significa que qualquer valor negativo foi ajustado para zero, enquanto os valores que excediam 365 dias foram reduzidos para 365 dias, mantendo a antecedência dentro de um intervalo válido para análise. Essa medida contribuiu para melhorar a integridade dos dados e evitar distorções nos resultados futuros.

Além disso, também foi realizado a formatação de certas colunas para reduzir gastos computacionais de memória. Nisso se enquadra a conversão das colunas numéricas de 64 bits para 32 ou 16 dependendo do número de casas decimais ou grandeza representada, transformação de colunas strings em categórica, entre outros.

### 5.2.2 Construção dos Dados

A construção de dados envolve a criação de novas features com base nas informações disponíveis no conjunto de dados original. Isso é importante porque as vezes os dados da forma que se encontram não são otimizados para funcionarem num algoritimo de aprendizado de máquina. Um bom exemplo disso são as colunas de tempo, como a data de criação das reservas, ela sozinha não consegue trazer muita utilidade nos algorítimos, mas se combina-la com as datas de checkin é obitido a antecedência que, por ser um número inteiro, os algorítimos conseguem extrair mais valor dela.

#### 5.2.2.1 Features de Check-in/Check-out:

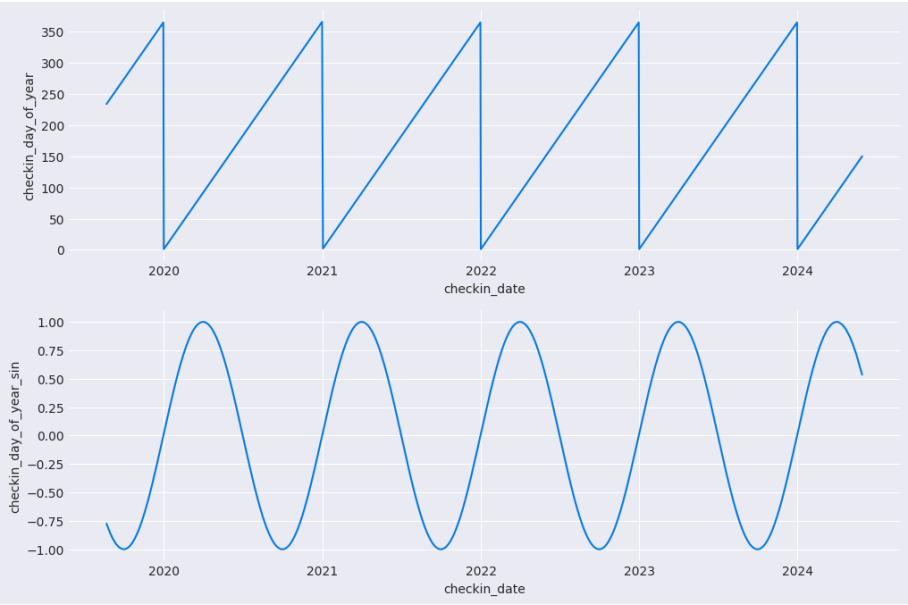
As datas de check-in e check-out são informações essenciais para entender o comportamento das reservas, mas, por si só, podem não ser facilmente utilizadas por algoritmos de aprendizado de máquina. Para tornar essas informações mais relevantes, foram criadas novas features que extraem dados úteis das datas.

Uma abordagem adotada consiste em extrair o ano, o dia do ano e o dia da semana das datas de check-in e check-out. Essas features ajudam a capturar padrões sazonais e comportamentais que podem variar com o tempo. Uma característica importante a ser considerada é a natureza cíclica de algumas dessas informações, como o dia do ano, que se repete a cada ciclo de 365 dias.

Para lidar com essa ciclicidade, uma técnica foi aplicada para converter essas informações em valores de seno e cosseno, conforme ilustrado na Equação (9), onde "DDAS" representa o dia do ano convertido em seno e "DDA" apenas o dia do ano. Isso permite que o modelo de aprendizado de máquina compreenda a natureza cíclica dos dados, onde o valor máximo de uma característica está vinculado ao valor mínimo do ciclo subsequente, representado na Figura 9.

$$\text{DDAS} = \sin\left(\text{DDA} \cdot (2\frac{\pi}{365})\right) \tag{9}$$

Figura 9 – Gráficos da data do dia do ano comparada com transformação seno.




Fonte: Autor.

Essas novas features criadas a partir das datas de check-in e check-out fornecem informações adicionais e mais relevantes para o modelo, permitindo que ele capture os padrões sazonais e de comportamento dos dados com maior precisão.

#### 5.2.2.2 Features one-hot-encoding:

Uma técnica comum aplicada no processo de engenharia de "features" é o "one-hot encoding". Essa prática envolve a conversão de uma coluna categórica em várias colunas booleanas, onde cada coluna representa uma categoria específica. Essa abordagem é particularmente útil, pois permite que algoritmos de aprendizado de máquina lidem com variáveis categóricas de forma eficaz, uma vez que podem relacioná-las diretamente em equações matemáticas simples.

Um exemplo disso é a coluna "listing_type", que representa o tipo de imóvel (hotel, apartamento, casa). Para capturar as diferentes categorias e seu impacto no modelo, essa coluna foi convertida em três novas colunas: "is_hotel", "is_apartment", e "is_house". Cada uma dessas colunas booleanas indica a presença ou ausência de um determinado tipo de imóvel. Esse processo permite ao modelo levar em consideração o tipo de imóvel como um fator relevante na predição e, ao mesmo tempo, preserva a natureza categórica dos dados, pois, como visto anteriormente, os hotéis parecem apresentar um fator maior de bloqueios do que os outros tipos de imóveis.

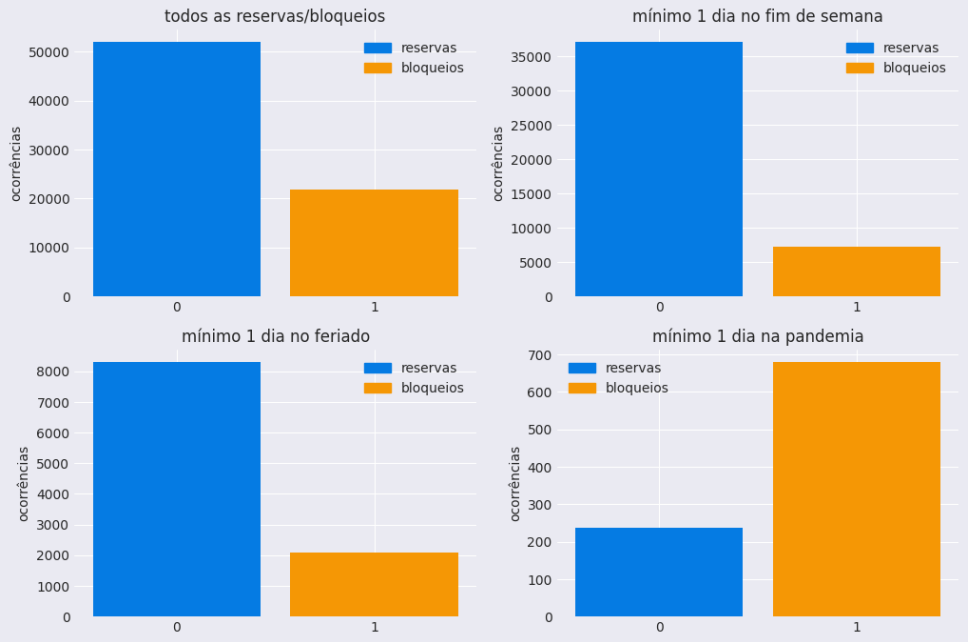
#### 5.2.2.3 Número de dias de reserva em feriado, pandemia ou fim de semana

A sazonalidade, como já comentada anteriormente, é um fator significativo no mercado de aluguel de temporada, mas vários outros elementos temporais também podem influenciar a demanda por imóveis, como feriados, fins de semana e eventos especiais. Além disso, eventos como a pandemia de COVID-19 tiveram um impacto notável nas tendências de aluguel, reduzindo a procura por propriedades temporárias. Para capturar esses efeitos, foram criadas novas "features" que representam a contagem de dias em que um período de reserva ou bloqueio inclui feriados, fins de semana ou o período de pandemia.

A análise da Figura 10 mostra que períodos que incluem pelo menos um dia de final de semana ou feriado têm uma proporção maior de reservas em relação a bloqueios. Enquanto isso, períodos ocupadas durante a pandemia tendem a ser o oposto, visto que a proporção de bloqueios chega a ser quase 3 vezes acima da de reservas.

Portanto, as features "n_days_pandemic", "n_days_holiday" e "n_days_weekend" foram criadas para quantificar os dias incluídos em cada um desses períodos, com base nos dados da tabelas de datas, presente na Tabela 5. Essas novas features ajudarão o modelo a levar em consideração a influência dos eventos sazonais e situações excepcionais na predição, tornando-o mais robusto e capaz de identificar padrões associados a esses fatores.

Figura 10 – Gráficos da diferença do número de reservas/bloqueios com base nos fins de semana, feriados e pandemia.




Fonte: Autor.

#### 5.2.2.4 Outras features

Além das features discutidas anteriormente, foram incorporadas ao modelo três novos atributos destinadas a informar a frequência recente de ocupação de cada imóvel.

- "days_from_last_checkout”: Representa o intervalo de tempo, em dias, desde o último checkout registrado, assumindo o valor mínimo de 0 para situações em que há dois períodos ocupados seguidos.
- "last_10_n_checkin” ou "last_60_n_checkin”: Representam o número de checkins registrados nos últimos 10 ou 60 dias daquele período. Essas informações podem variar de 0 e irem até 10 ou 60, dependendo de qual das duas "features” está sendo analisada.

#### 5.2.2.5 Normalização

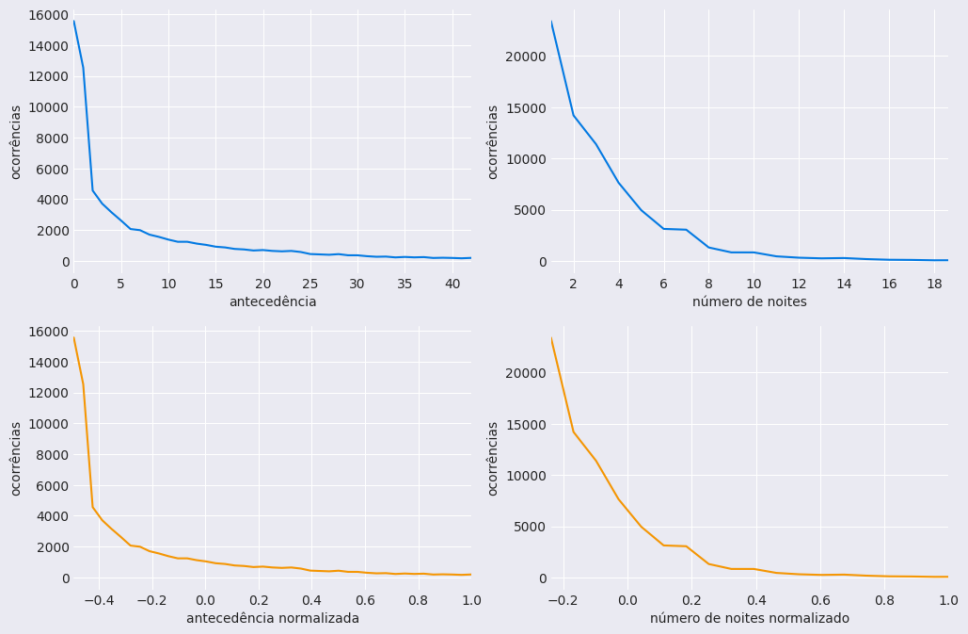
A normalização é importante no contexto de redes neurais, especialmente em situações em que diferentes features possuem escalas distintas. No contexto deste projeto, a técnica de normalização adotada foi a "StandardScaler” do "scikit-learn”. Essa abordagem normaliza cada "feature” calculando a média e o desvio padrão, transformando os valores para um intervalo próximo de -1 a 1, conforme exemplificado

na Equação 10. Nessa equação, "x" representa o valor original da "feature", "u" é a média da "feature" e "s" é o seu desvio padrão.

$$z = \frac{x - u}{s} \tag{10}$$

O resultado da normalização pode ser observado na Figura 11. Percebe-se que esse método consegue manter a estrutura original das "features", sendo que a maioria dos valores estão entre -1 e 1.

Figura 11 – Gráficos das "features" de antecedência e número de noites normalizadas.




Fonte: Autor.

### 5.2.3 Separação dos Dados

A estratégia de separação dos dados em conjuntos de treino, validação e teste foi planejada para abordar desafios associados ao leve desbalanceamento dos dados, considerando tanto o tipo de imóvel quanto a natureza da reserva ou bloqueio. A proposta foi manter a proporcionalidade desses grupos, visando evitar viés e garantir que o modelo seja treinado e avaliado de maneira equilibrada em relação às diferentes categorias presentes nos dados.

Ao observar a Tabela 6, é possível verificar o tamanho de cada grupo resultante dessa abordagem. Essa segmentação é suficientemente grande para treinar e validar o modelo em diferentes contextos.

Quanto à separação do conjunto de teste, ela foi realizada apenas uma vez e servirá como uma avaliação final do desempenho do modelo. Já os conjuntos de treino

| Tipo do imóvel | Reserva | Bloqueio |
|---|---|---|
| Apartamento | 7083 | 1553 |
| Casa | 235228 | 5073 |
| Hotel | 21273 | 15183 |

Tabela 6 – Tamanho de cada grupo.

e validação foram divididos em seis partes distintas, adotando a técnica de validação cruzada. Essa abordagem, além de contribuir para uma avaliação mais robusta do modelo, ajuda a reduzir a aleatoriedade dos dados e mitigar eventuais viés nos cálculos dos hiperparâmetros. Essas separação de treino, teste e validação também mantém a proporção dos grupos apontados na Tabela 6.

É relevante ressaltar que, durante o treinamento, uma pequena porcentagem do conjunto de treino é reservada para as funções de "callback", uma prática adotada para evitar que as métricas de validação sejam afetadas por "overfitting".

## 5.3 MODELAGEM

### 5.3.1 Primeira versão do modelo

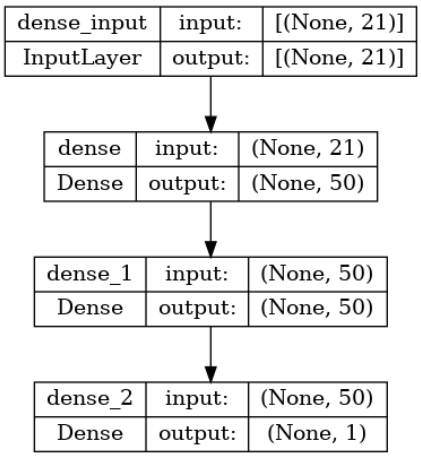
O primeiro modelo desenvolvido representou um experimento inicial, visando estabelecer uma base de referência para o projeto. Este modelo adotou uma abordagem simples, com um número fixo de 50 neurônios por camada e a escolha arbitrária de parâmetros, incluindo o número de camadas ocultas. A decisão de utilizar duas camadas ocultas com 50 neurônios cada foi tomada após algumas iterações empíricas simples. A Figura 12 apresenta uma representação gráfica deste primeiro modelo gerada pela própria biblioteca "keras"

Na arquitetura do modelo, a camada de entrada foi projetada com o mesmo número de neurônios correspondente aos atributos selecionados, sendo que esses atributos são as "features" construídas ou visualizadas nas Seção 5.1.2 e Seção 5.2.2. A camada de saída, por sua vez, consistiu em um único neurônio com uma função de ativação sigmoid, adequada para realizar a classificação binária. A função de ativação ReLU foi escolhida para as conexões entre as camadas. O modelo incorporou uma regularização L2 com um fator de 1e-5 para evitar "overfitting", e a função de "callback" "EarlyStopping" foi implementada para interromper o treinamento quando necessário.

#### 5.3.1.1 Avaliação do Modelo

Os resultados da validação cruzada, apresentados nas tabelas Tabela 7 e Tabela 8, revelam um desempenho notável no conjunto de treino, evidenciado por um "F1 Score" de 0.809 e um baixo desvio padrão. No entanto, a performance no conjunto de validação não acompanhou esse sucesso, mostrando sinais claros de "overfitting". O

Figura 12 – Representação do primeiro modelo.




Fonte: Autor.

"F1 Score" variou de 0.724 a 0.781, indicando que o modelo não generalizou tão bem quanto o desejado para dados não vistos. Este cenário serviu como ponto de partida para ajustes subsequentes e refinamento do modelo.

| Iteração | Loss | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|---|
| 1 | 0.288243 | 0.894268 | 0.857111 | 0.771347 | 0.81197 |
| 2 | 0.296019 | 0.889443 | 0.834518 | 0.781401 | 0.807086 |
| 3 | 0.295757 | 0.89037 | 0.853686 | 0.759799 | 0.804011 |
| 4 | 0.284021 | 0.895034 | 0.861998 | 0.768374 | 0.812498 |
| 5 | 0.288589 | 0.893446 | 0.87682 | 0.7446 | 0.805319 |
| 6 | 0.278737 | 0.89811 | 0.877936 | 0.761649 | 0.815669 |
| Média | 0.288561 | 0.893445 | 0.860345 | 0.764528 | 0.809426 |
| Desvio Padrão | 0.006703 | 0.003176 | 0.016158 | 0.01244 | 0.004616 |

Tabela 7 – Métricas do conjunto de treino da primeira versão do modelo.

| Iteração | Loss | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|---|
| 1 | 0.373934 | 0.860684 | 0.853469 | 0.639059 | 0.730863 |
| 2 | 0.345369 | 0.869097 | 0.798495 | 0.746027 | 0.77137 |
| 3 | 0.360408 | 0.864019 | 0.788393 | 0.738998 | 0.762896 |
| 4 | 0.372407 | 0.859495 | 0.86306 | 0.624274 | 0.724499 |
| 5 | 0.409499 | 0.852619 | 0.745514 | 0.762152 | 0.753741 |
| 6 | 0.349879 | 0.874242 | 0.804665 | 0.759401 | 0.781378 |
| Média | 0.368582 | 0.863359 | 0.808933 | 0.711652 | 0.754125 |
| Desvio Padrão | 0.023121 | 0.007606 | 0.043553 | 0.062713 | 0.022521 |

Tabela 8 – Métricas do conjunto de validação da primeira versão do modelo.

### 5.3.2 Otimizando o modelo calculando os hiperparâmetros

A busca pelos hiperparâmetros foi realizada por meio da biblioteca "keras_tuner" e sua implementação de busca bayesiana. A escolha apropriada de hiperparâmetros é

essencial para a eficácia do modelo, e, para isso, foi necessário definir quais parâmetros seriam otimizados, bem como estabelecer os espaços de busca correspondentes.

Os espaços de busca podem ser uma lista ou um intervalo numérico de mínimo e máximo. A lista é comumente utilizada para otimizar parâmetros categóricos ou booleanos, enquanto que os intervalos podem variar com um passo fixo de forma linear ou logarítmica. Depois de diversos testes com a biblioteca, os seguintes parâmetros e espaços foram definidos:

- optimizer: Otimizador utilizado, variando entre Adam e RMSprop;
- learning_rate: Taxa de aprendizado do otimizador, variando de 0.001 a 0.1 em uma escala logarítmica com passo 2;
- activation: Função de ativação utilizada entre as camadas, variando entre ReLU e Swish;
- neurons1, neurons2, neurons3: Número de neurônios em cada uma dessas 3 possíveis camadas ocultas. Variação de 0 a 300 de forma linear, permitindo camadas com 0 neurônios (ausência da camada);
- neurons4, neurons5, neurons6: Esses são o número de neurônios para mais 3 possíveis camadas ocultas. Eles são semelhantes aos apresentado acima, mas com variação de 0 a 150;
- dropout: Taxa de dropout utilizada, variando de 0 a 0.3 de forma linear. Adiciona uma camada de dropout para cada conexão entre camadas, se a taxa for maior que 0;
- regularizer_L1: Taxa de regularização do tipo L1, variando de 1e-7 a 1e-2 em uma escala logarítmica com passo 5;
- regularizer_L2: Taxa de regularização do tipo L2, com a mesma variação que regularizer_L1;
- batch_size: Tamanho dos batches utilizados no treinamento, variando de 128 a 8192 em uma escala logarítmica com passo 2.

O código desenvolvido para realizar a busca bayesiana utilizando a biblioteca "keras_tuner” se resume na criação da classe "MyHyperModel”. Existem diversas formas de utilizar essa biblioteca, mas a abordagem de personalizar uma classe é a mais flexível para a definição e otimização dos hiperparâmetros do modelo.

Ao criar uma instância da classe "MyHyperModel”, é necessário fornecer os tamanhos das camadas de entrada e saída como argumentos, o que automatiza o

processo de ajuste das características do modelo conforme necessário, através dos parâmetros "input_size" e "output_size" atribuídos no construtor da classe.

Dois métodos essenciais de serem definidos na classe são "build" e "fit". O método "build" permite a definição dos parâmetros e a especificação de intervalos para cada uma delas, utilizando a variável hp (hiperparâmetro). Por meio desse método, o espaço de busca para otimização é configurado, permitindo a experimentação sistemática de diferentes configurações de hiperparâmetros. O método "fit" não é obrigatório de ser definido ao criar uma classe com o "keras_tuner", o motivo disso é que a biblioteca já possuí uma implementação padrão para esse método, entretanto, sua implantação foi feita para adicionar uma lógica personalizada de validação cruzada. Dentro de "fit" é chamado uma função de validação cruzada desenvolvida para esse projeto, que faz toda a validação comentada na Seção 3.2.1.3, utilizando a proporção de grupos descritas na Seção 5.2.3. A função retorna um dicionário de métricas, sendo a busca bayesiana orientada a maximizar uma delas durante o processo de otimização.

Após toda essa definição dos espaços de busca e criação da classe, é crucial especificar a métrica que o algoritmo deve maximizar durante a busca de hiperparâmetros. Neste caso, por ser a mais balanceada, a métrica escolhida foi o "F1 Score" do conjunto de validação, ou seja, a busca irá retornar o conjunto de parâmetros que obteve o melhor "F1 Score" na validação cruzada.

Depois de realizar diversas buscas com múltiplas combinações de "features" e depois de decido o padrão para o intervalo dos parâmetros na busca bayesiana, foi realizado 150 iterações de busca sob essas condições e o melhor modelo obtido no projeto foi encontrado.

#### 5.3.2.1 Avaliação do Modelo

O modelo otimizado demonstrou uma melhoria, alcançando um aumento de 1.5% no "F1 Score" em relação ao conjunto de validação e uma redução de 0.7% no desvio padrão, conforme evidenciado na Tabela 10 que exibe as métricas de validação. Embora essa diferença possa parecer modesta, é crucial observar que as métricas do conjunto de treino agora estão mais alinhadas com as de validação, conforme visualizado na Tabela 9, indicando também uma redução no "overfitting" do modelo.

A Tabela 11 apresenta as métricas geradas para cada modelo treinado durante a validação cruzada, utilizando o conjunto de teste. Com o melhor modelo encontrado, é de grande importância compará-lo e testá-lo nos dados de testes que nunca foram vistos para medir mais uma vez sua capacidade de generalização.

Além disso, como cada um dos 6 modelos foi treinado sem a utilização dos dados de validação, um modelo final foi treinado com todos os dados de treinamento (treino e validação) e testado no conjunto de teste, com os resultados destacados na Tabela 12. Todos esses resultados evidenciam que o modelo conseguiu se generalizar

bem no conjunto de teste e que está com um desempenho satisfatório.

| Iteração | Loss | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|---|
| 1 | 0.376864 | 0.866362 | 0.791775 | 0.744175 | 0.767237 |
| 2 | 0.308734 | 0.893946 | 0.880274 | 0.742681 | 0.805645 |
| 3 | 0.334296 | 0.882308 | 0.883222 | 0.69411 | 0.77733 |
| 4 | 0.369515 | 0.868577 | 0.793601 | 0.751392 | 0.77192 |
| 5 | 0.314817 | 0.892682 | 0.876867 | 0.741543 | 0.803548 |
| 6 | 0.315883 | 0.891295 | 0.881731 | 0.730743 | 0.799168 |
| Média | 0.336685 | 0.882528 | 0.851245 | 0.734107 | 0.787475 |
| Desvio Padrão | 0.029631 | 0.01238 | 0.045411 | 0.020688 | 0.017203 |

Tabela 9 – Métricas do conjunto de treino do modelo otimizado.

| Iteração | Loss | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|---|
| 1 | 0.398551 | 0.860684 | 0.806875 | 0.695905 | 0.747292 |
| 2 | 0.339328 | 0.886105 | 0.868816 | 0.724633 | 0.790202 |
| 3 | 0.358184 | 0.87569 | 0.837722 | 0.719438 | 0.774087 |
| 4 | 0.371164 | 0.870714 | 0.843145 | 0.691837 | 0.760034 |
| 5 | 0.385467 | 0.862662 | 0.770775 | 0.762764 | 0.766749 |
| 6 | 0.3508 | 0.877861 | 0.829051 | 0.739835 | 0.781906 |
| Média | 0.367249 | 0.872286 | 0.826064 | 0.722402 | 0.770045 |
| Desvio Padrão | 0.022178 | 0.009626 | 0.033733 | 0.02677 | 0.015441 |

Tabela 10 – Métricas do conjunto de validação do modelo otimizado.

| Iteração | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|
| 1 | 0.859547 | 0.781265 | 0.730064 | 0.754797 |
| 2 | 0.882752 | 0.856216 | 0.72594 | 0.785714 |
| 3 | 0.875153 | 0.858931 | 0.692026 | 0.766497 |
| 4 | 0.864432 | 0.790378 | 0.737855 | 0.763214 |
| 5 | 0.880309 | 0.840671 | 0.735105 | 0.784352 |
| 6 | 0.880852 | 0.853579 | 0.721357 | 0.781917 |
| Média | 0.873841 | 0.830173 | 0.723724 | 0.772749 |
| Desvio Padrão | 0.008803 | 0.031987 | 0.015192 | 0.011825 |

Tabela 11 – Métricas do conjunto de teste de cada modelo treinado durante a validação cruzada.

| Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|
| 0.883024 | 0.848101 | 0.736939 | 0.788622 |

Tabela 12 – Métricas do conjunto de teste do modelo otimizado.

## 5.4 TESTANDO O MELHOR COM DADOS DO AIRBNB

Foram realizados testes do modelo otimizado utilizando os dados da tabela de reservas do Airbnb. Inicialmente, o modelo foi treinado utilizando 90% dos dados,

enquanto que os 10% restantes foram reservados para funções de "callback". Essa divisão dos dados foi projetada para manter a proporção de dados nos grupos definidos previamente, conforme detalhado na Tabela 6.

### 5.4.1 Validação diária onde a "reservations" do Airbnb identificou um período

Ao contrário das avaliações anteriores que utilizavam exclusivamente a tabela de "reservations" da Seazone, agora o modelo foi desafiado a prever os dados da tabela de reservas do Airbnb. Essa mudança introduz uma complexidade adicional, uma vez que as tabelas podem não ser idênticas devido a possíveis variações nos "scrappers" ou na lógica de agrupamento de períodos. Essas divergências dificultam a validação na granularidade de períodos, visto que apenas a "reservations" da Seazone possuí o rótulo verdadeiro.

Dessa forma, a avaliação inicial do modelo foi conduzida em uma granularidade diária, focando apenas nas datas em que a tabela de reservas do Airbnb identificou um período. Essa abordagem proporciona uma validação mais eficiente e direta da predição do modelo, considerando as nuances nas tabelas de dados. Esse processo revela a capacidade do modelo em prever a ocupação diária de imóveis do Airbnb com base nas informações disponíveis.

Agora que a motivação para a nova forma de validação foi explicada, antes de serem gerados previsões e métricas em cima dos dados do Airbnb, a Tabela 13 mostra o resultado do modelo nessas condições em cima dos dados da "reservations" da Seazone na granulação diária. Observa-se um aumento significativo no "F1 Score" quando comparado à granularidade por período ocupado, alcançando uma melhora de aproximadamente 10%. Essa diferença é atribuída à capacidade do modelo de realizar previsões mais precisas em períodos longos, onde acertos em períodos estendidos se traduzem em múltiplos acertos na granularidade diária, contribuindo para um aumento global nas métricas de avaliação.

| Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|
| 0.907802 | 0.941030 | 0.806357 | 0.868504 |

Tabela 13 – Métricas do modelo na granularidade diária em datas que possuem dados da "reservation"em cima dos dados reais.

Entretanto, ao realizar as previsões em cima dos dados do Airbnb, esse aumento não se repete. A Tabela 14 apresenta as métricas apenas para os dias em que reservas foram detectadas na tabela do Airbnb, percebe-se uma queda significativa no desempenho do modelo, visto que o "F1 Score" atinge 0.64. Isso acontece principalmente por causa das inconsistências dos dados nos períodos da "reservations" do Airbnb. Segundo as tabelas de verdade da Seazone, existem 550698 diárias ocupadas ou bloqueadas que poderiam ser detectadas pelos "scrappers" e tabelas do Airbnb. No

entanto, apenas 525261 desses dias se encontram indisponíveis nas tabelas diárias que a Seazone possuí do Airbnb e, dessas, apenas 319661 são realmente detectadas como períodos ocupados e incorporadas à tabela "reservations" Airbnb. Essa discrepância evidencia desafios inerentes à obtenção e consistência dos dados do Airbnb, impactando diretamente a qualidade da predição do modelo.

| Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|
| 0.865060 | 0.762025 | 0.552909 | 0.640839 |

Tabela 14 – Métricas do modelo em dados do Airbnb na granularidade diária em datas da "reservations".

Para efeitos de comparação, a lógica já existente de detecção de bloqueios (comumente chamada de heurística) também foi avaliada seguindo o mesmo método utilizado nas análises anteriores, onde é considerado a granulação diária e apenas datas contidas nos períodos da "reservations". A Tabela 15 apresenta as métricas de validação. É visível que o problema não acontece apenas no modelo, mas ele se repete na heurística.

| Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|
| 0.844969 | 0.662834 | 0.586092 | 0.622106 |

Tabela 15 – Métricas da lógica heuristica na granularidade diária em datas da "reservations".

### 5.4.2 Validação diária para todas as datas indisponíveis

Como comentado anteriormente, das 525261 datas indisponíveis, apenas 319 mil são detectadas na "reservation". A análise dos dados referentes aos períodos que não foram incluídos nesta tabela, totalizando 205600 diárias, revela que a maioria consiste em bloqueios. Essa constatação proporciona uma abordagem alternativa na implementação do modelo em cima dos dados do Airbnb, permitindo considerar que todas essas diárias indisponíveis, ausentes na tabela de "reservations", sejam interpretadas como bloqueios.

A Tabela 16 apresenta uma nova comparação entre o modelo e a heurística, utilizando essa nova lógica de validação baseada exclusivamente nas datas indisponíveis do Airbnb. As métricas revelam um "F1 Score" de 0.9038 pro modelo, demonstrando que assumir essas datas como bloqueio realmente garante uma taxa acertiva maior. Além disso, também é visível que o "F1 Score" do método da heurística, por ser 0.9090, ficou acima do modelo. Isso mostra que a lógica de detecteção de bloqueios da heurística em datas não presentes na "reservations" é melhor do que o utilizado pro modelo, onde se consideram-se todos bloqueios.

| Método | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|
| Modelo | 0.883852 | 0.882527 | 0.923408 | 0.902505 |
| Heurística | 0.895656 | 0.922115 | 0.896489 | 0.909121 |

Tabela 16 – Métricas na granularidade diária para todas datas indisponíveis da heurística e modelo.

## 5.5 COMBINANDO HEURÍSTICA E REDES NEURAIS

Com base nos resultados anteriores, torna-se evidente que o modelo apresenta bom desempenho para os períodos com dados na tabela de "reservations", visto que o "F1 Score" ficou 0.0187 maior, contudo, sua capacidade de generalização para outras datas indisponíveis é limitada e a heurística se torna melhor. Diante dessa constatação, propôs-se uma abordagem híbrida, combinando o modelo com o método heurístico.

A estratégia consiste em utilizar os valores de saída do modelo quando há dados na tabela de "reservations" do Airbnb e recorrer à heurística nos casos em que esses dados não estão disponíveis. A Tabela 17 apresenta as métricas dessa combinação, tanto para datas contidas na "reservations" quanto para todas as datas indisponíveis. Percebe-se que para datas contidas na "reservation", o resultado ficou igual ao obtido na Tabela 14. Observa-se ainda que, na comparação com todas as datas indisponíveis, houve um aumento no "F1 score" de 0.0055 em comparação com a heurística isolada, conforme evidenciado na Tabela 16.

| Comparação | Acurácia | Precisão | Recall | F1 Score |
|---|---|---|---|---|
| Apenas datas da "reservations" | 0.865060 | 0.762025 | 0.552909 | 0.640839 |
| Todas as datas indisponíveis | 0.903012 | 0.937928 | 0.892468 | 0.914633 |

Tabela 17 – Métricas na granularidade diária para todas datas indisponíveis da heurística e modelo.

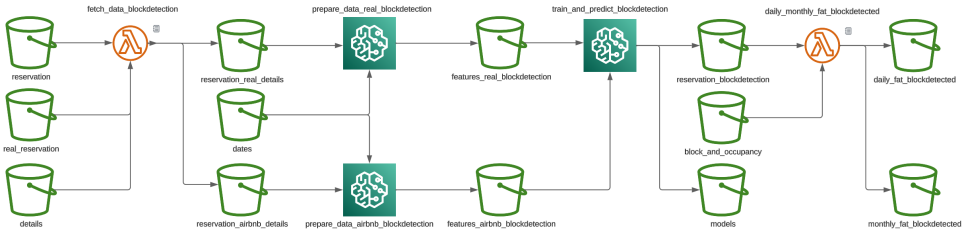
## 5.6 IMPLANTAÇÃO DO MELHOR MODELO EM AMBIENTE STAGING

Em um primeiro momento, o processo de implantação do melhor modelo foi realizado na "pipeline" de "staging" da Seazone. Essa "pipeline", configurada como um ambiente de testes, ela é composta exclusivamente pelos imóveis da Seazone, mas alimentada com dados provenientes do Airbnb. Essa abordagem permite realizar testes em um ambiente controlado, com um volume reduzido de dados em comparação ao ambiente de produção (que inclui todos os imóveis do Brasil disponíveis no Airbnb). A validação no ambiente de "staging" é essencial antes de promover o modelo para o ambiente de produção, pois não só garante seu correto funcionamento e eficácia, mas também acelera o tempo de desenvolvimento, devido a quantidade reduzida de dados.

O diagrama apresentado na Figura 13 ilustra a estrutura das tabelas e serviços da AWS utilizados no processo de implantação. Cada umas das etapas é descrita nas seções abaixo. Os baldes verdes representam tabelas salvas no serviço S3

(Seção 3.1.1) da AWS. O ícone laranja contendo o símbolo λ é o serviço Lambda (Seção 3.1.3). Por último, os quadrados verdes são o Sagemaker (Seção 3.1.4).

Figura 13 – Diagrama de implementação do modelo na pipeline da Seazone.




Fonte: Diagrama interno da Seazone.

### 5.6.1 Lambda "fetch_data_blockdetection”

O Lambda "fetch_data_blockdetection” realiza parte da preparação dos dados para o modelo. Sua principal responsabilidade é combinar informações das tabelas "details” com a "reservations” da Seazone, gerando assim a tabela "reservation_real_details” que contém os dados rotulados utilizados para o treinamento, e também combinar a "details” com a "reservation” Airbnb, construindo a tabela "reservation_airbnb_details” usada na predição. Após a consolidação, essas tabelas são armazenadas em tabelas num bucket S3, cujo acesso é concedido ao serviço Sagemaker.

O código do Lambda executa uma consulta nas tabelas mencionadas utilizando o serviço Athena (Seção 3.1.2). Essa consulta, essencialmente um join, é a responsável por agregar os dados necessários.

### 5.6.2 Sagemaker "prepare_data”

Os jobs do Sagemaker, denominados "prepare_data_real_blockdetection” e "prepare_data_airbnb_blockdetection”, compartilham o mesmo código, com a única distinção sendo que o primeiro executa a executa com um parâmetro "pipe” definido como falso, enquanto o segundo utiliza o valor verdadeiro. Um valor falso para o argumento implica que está sendo processado um dado das tabelas rotuladas da Seazone, enquanto que um valor verdadeira representa um processamento de dados do Airbnb. Ambos os funcionamentos realizam a parte restante da preparação dos dados, que é a criação de "features”, limpeza e pré-processamento dos dados.

Na prática, as principais distinções no tipo de processamento Seazone ou Airbnb se encontram no começo e no final da função presente no script. Nas primeiras linhas, a diferença é o caminho de onde os arquivos se encontram, visto que o nome das

tabelas não são apenas diferentes, mas também há uma diferença no nome das colunas, então no começo da função esses nomes são padronizados. Por último, o script salva o resultado do processamento em dois possíveis caminhos, que dependem novamente se o processamento está feito com os dados da Seazone ou Airbnb.

### 5.6.3 Sagemaker "train_and_predict_blockdetection”

Essa é a etapa do Sagemaker que abrange o treinamento e a predição do modelo. Inicialmente, foi adotada uma estratégia para garantir a separação adequada entre os dados de treinamento e predição, visando evitar potenciais problemas de "overfitting” e assegurar uma validação robusta no ambiente de "staging”. Como existem menos dados da Seazone em comparação com o Airbnb, essa separação é essencial para mostrar que o modelo consegue se generalizar e que funcionaria em prdução. A lógica implementada para essa separação baseou-se na data de criação da reserva, pois como comentado anteriormente, as tabelas de treinamento (Seazone) e predição (Airbnb) apresentam diferenças substanciais, dificultando uma conexão direta entre elas. Dessa forma, para prever períodos do Airbnb criados entre 2023-02-01 e 2023-02-28, foram utilizados dados de treinamento da Seazone com criação até 2023-01-31. Essa estratégia é executada interativamente no "script” para os últimos 12 meses, onde para diferenciar reservas de bloqueios no mês "n”, é usado dados de treinamento criados até "n-1”. Apesar de existir um recalculo dos dados para datas passadas, o modelo se atualiza automaticamente com novos dados incorporados à "pipeline”.

Ao final do processo, o Sagemaker "job” armazena os resultados em uma pasta chamada "models”. Nesta pasta, são mantidos os 12 modelos criados para cada um dos últimos 12 meses, preservando um registro histórico das iterações do modelo. Além disso, é gerada a tabela "reservation_blockdetection”, cujo formato é análogo à tabela "reservations” do Airbnb. No entanto, esta tabela inclui as colunas "prob” e "blocked”, que representam a porcentagem resultante do modelo e se é um bloqueio ou não, respectivamente.

### 5.6.4 Lambda "daily_monthly_fat_blockdetection”

O Lambda "daily_monthly_fat_blockdetection” desempenha um papel central ao unir a lógica de bloqueio do modelo com a heurística e salvar os resultados em tabelas de faturamento diárias e mensais. Este Lambda recebe como entrada a tabela "reservation_blockdetection”, criada no passo anterior, e a tabela "block_and_occupancy”. A última, até este ponto do documento, não havia sido discutida, mas é uma tabela fundamental na "pipeline” da Seazone. Essa tabela contém todas as datas do calendário e os preços quando ocorre uma reserva ou bloqueio, sendo necessária não somente para a geração de informações de faturamento diário e mensal, mas também por conter as saídas da heurística.

A primeira etapa do Lambda envolve uma consulta ao Athena para expandir a granularidade da tabela "reservation_blockdetection” para o nível diário. Em seguida, é realizado um join com a tabela "block_and_occupancy”. Com os dados provenientes do modelo e da heurística, é possível gerar a nova coluna "blocked_mix”, que aplica a lógica de combinação dos dois métodos. Para fins de comparação, a coluna "blocked”, que utiliza apenas a saída do modelo, é mantida, e também é criada a coluna "blocked_av”, que bloqueia todas as datas indisponíveis. Essas 3 colunas representam cada um dos métodos de predição que utilizam o modelo abordados na Seção 5.4 e Seção 5.5.

Ao final do processo, são geradas duas tabelas fundamentais: "daily_fat_ blockdetected”, que contém o faturamento diário, e "monthly_fat_blockdetected”, que abrange o faturamento mensal. Estas tabelas possuem dados derivados apenas do modelo (que consideram datas indisponíveis como faturamento), do faturamento do modelo considerando datas indisponíveis como bloqueio e do faturamento resultante da combinação do modelo com a heurística. Essa estrutura considera diferentes perspectivas de bloqueio e permite validar e testar todos os casos.

## 5.7 AVALIAÇÃO DO MELHOR MODELO EM STAGING

A avaliação do melhor modelo no ambiente de "staging” usa principalmente as métricas de MAPE e "n_inf” comentadas na Seção 3.2.1.2. Essa é uma etapa importante na validação do modelo, visto que o trabalho todo de melhorar a detecção de bloqueios tem o objetivo indireto de melhorar a predição de faturamento da empresa.

### 5.7.1 Comparação dos Resultados

Aqui será feito uma análise comparativa entre os quatro métodos: heurístico, modelo, modelo considerando datas indisponíveis como bloqueio e o modelo combinado com a heurística, utilizando a tabela "monthly_fat_blockdetected” e a tabela de faturamento contendo os dados da heurística mensal. Essa analise proporciona "insights” importantes para o desempenho de cada abordagem na predição do faturamento para a Seazone. A Tabela 18 apresenta uma visão abrangente dessas comparações, com ênfase no MAPE e no número de infinitos "n_inf”.

Ao observar os resultados, é evidente que o modelo combinado com a heurística se destaca em vários aspectos. Em praticamente todos os meses analisados, o MAPE desse método é menor ou muito próximo ao da heurística já existente. Quando o método combinado se desempenha pior, a diferença raramente ultrapassa 1%, entretanto, quando ele melhora a heurística, a diferença chega a 7% em alguns meses.

Essa melhora também se reflete na análise do "n_inf”, onde não existe mês onde o número de infinitos ficou superior ao da heurística. Esse resultado destaca a

| | heurística | | modelo | | modelo + indisponível | | modelo + heurística | |
|---|---|---|---|---|---|---|---|---|
| ano_mes | mape | n_inf | mape | n_inf | mape | n_inf | mape | n_inf |
| 2021-11-01 | 0.209743 | 1 | 0.240940 | 1 | 0.224380 | 1 | 0.219421 | 1 |
| 2021-12-01 | 0.414729 | 0 | 0.565937 | 0 | 0.492943 | 0 | 0.323921 | 0 |
| 2022-01-01 | 0.563356 | 0 | 0.723086 | 0 | 0.517273 | 0 | 0.566436 | 0 |
| 2022-02-01 | 0.298180 | 1 | 1.300847 | 1 | 0.239634 | 1 | 0.265881 | 1 |
| 2022-03-01 | 0.340599 | 0 | 0.497282 | 0 | 0.316440 | 0 | 0.298229 | 0 |
| 2022-04-01 | 0.170480 | 0 | 0.163659 | 0 | 0.153936 | 0 | 0.180600 | 0 |
| 2022-05-01 | 0.465922 | 10 | 0.646378 | 17 | 0.430118 | 5 | 0.414002 | 10 |
| 2022-06-01 | 0.535289 | 26 | 0.685351 | 33 | 0.608406 | 22 | 0.525052 | 25 |
| 2022-07-01 | 0.494658 | 9 | 0.619660 | 23 | 0.533602 | 9 | 0.478310 | 7 |
| 2022-08-01 | 0.336925 | 8 | 0.543271 | 23 | 0.358006 | 11 | 0.313283 | 8 |
| 2022-09-01 | 0.292600 | 11 | 0.445681 | 22 | 0.271120 | 13 | 0.275065 | 10 |
| 2022-10-01 | 0.379715 | 13 | 0.546887 | 26 | 0.473811 | 15 | 0.329488 | 12 |
| 2022-11-01 | 0.269706 | 6 | 0.356685 | 13 | 0.263137 | 4 | 0.245994 | 5 |
| 2022-12-01 | 0.261564 | 6 | 0.329514 | 10 | 0.255694 | 5 | 0.231724 | 3 |
| 2023-01-01 | 0.328460 | 6 | 0.472491 | 9 | 0.371912 | 5 | 0.322549 | 4 |
| 2023-02-01 | 0.295827 | 15 | 0.309913 | 16 | 0.235297 | 10 | 0.237699 | 10 |
| 2023-03-01 | 0.261257 | 17 | 0.261873 | 25 | 0.273143 | 15 | 0.263769 | 14 |
| 2023-04-01 | 0.250456 | 22 | 0.302361 | 37 | 0.276597 | 22 | 0.239123 | 21 |
| 2023-05-01 | 0.240337 | 25 | 0.350096 | 50 | 0.310597 | 43 | 0.229814 | 24 |
| 2023-06-01 | 0.188019 | 37 | 0.191035 | 61 | 0.179082 | 65 | 0.184551 | 37 |
| 2023-07-01 | 0.284208 | 33 | 0.346757 | 76 | 0.303674 | 45 | 0.283659 | 32 |
| 2023-08-01 | 0.236493 | 53 | 0.261443 | 90 | 0.231165 | 56 | 0.227408 | 51 |
| 2023-09-01 | 0.314233 | 40 | 0.353711 | 63 | 0.322357 | 45 | 0.284700 | 39 |
| 2023-10-01 | 0.320128 | 42 | 0.423708 | 102 | 0.312296 | 50 | 0.275462 | 41 |
| 2023-11-01 | 0.273552 | 69 | 0.380908 | 109 | 0.321869 | 81 | 0.265916 | 67 |

Tabela 18 – Métricas na granularidade diária para todas datas indisponíveis da heurística e modelo.

capacidade do modelo combinado de lidar melhor com situações em que o faturamento real é zero.

É interessante notar que o comportamento observado na comparação do MAPE segue uma tendência semelhante ao analisar o F1 Score na Seção 5.4. O modelo combinado com a heurística apresenta o melhor desempenho, seguido pela heurística isolada, o modelo considerando datas indisponíveis como bloqueio e, por último, o modelo sem considerar essas datas. Isso faz sentido, visto que o F1 Score reflete o quão bem o modelo consegue diferenciar bloqueios de reservas e esse é a principal causa de erro na previsão de faturamento da Seazone.

A consistência na melhoria do MAPE e a capacidade de lidar com casos especiais, representados por "n_inf", fortalecem a confiança na utilização desse modelo combinado para aprimorar as projeções de faturamento da Seazone.

## 5.8 TESTE EM PRODUÇÃO

O teste do modelo combinado com a heurística da fase de "staging" para produção envolveu uma série de adaptações e otimizações para garantir sua eficiência diante do volume massivo de dados provenientes do Airbnb. Esse modelo, embora

tenha demonstrado ser o mais promissor nos testes preliminares, ainda exige uma validação adicional para confirmar sua viabilidade em um ambiente de produção, considerando principalmente sua viabilidade e custo associado.

Um dos desafios iniciais encontrados durante a adaptação dos scripts foi lidar com a grande quantidade de dados do Airbnb. Otimizações significativas foram implementadas nos scripts para garantir a eficiência do processamento. A escolha de tipos de dados mais leves para os atributos dos "dataframes" foi uma estratégia adotada para evitar problemas de estouro de memória RAM. Além disso, técnicas de otimização foram aplicadas em cálculos demorados de determinadas features, acelerando assim todo o processo.

O fluxo de execução, delineado no diagrama apresentado na Figura 13, permaneceu como base, mas as tabelas que anteriormente eram subconjuntos com apenas imóveis da Seazone do Airbnb foram substituídas por conjuntos de dados completos do Airbnb. Essa alteração exigiu ajustes nas máquinas do Sagemaker, que, mesmo após as otimizações, precisaram ser dimensionadas para uma capacidade maior, passando para máquinas com 64 GB de RAM que têm um custo associado de $ 0.922 por hora.

O tempo de execução do modelo em um ambiente de produção envolveu principalmente duas partes: o preparo dos dados e o treinamento e predição. O processo de preparo de dados, realizado pelo Sagemaker "prepare_data", consumiu aproximadamente 50 minutos, enquanto o treinamento e a predição, conduzidos pelo Sagemaker "train_and_predict_blockdetection", demandaram cerca de 1 hora para processar todos os dados do Airbnb.

Os Lambdas, por utilizarem o Athena para processamento, são mais velozes e também não impactam muito no preço. Considerando todos esses aspectos, o custo total estimado para executar o modelo em produção é de até $2.00 por iteração, sendo que em produção essa "pipeline" rodaria de forma semanal, pois é nessa amostragem que o processo de atualização das tabelas nas quais o modelo se baseia são atualizadas. Esse custo se mostrou baixo no contexto dos gastos da área de dados da Seazone, solidificando ainda mais a viabilidade econômica dessa implementação em escala produtiva.

# 6 RESULTADOS FINAIS

Embora tenha sido alcançado uma melhora significativa nos resultados de predição e faturamento para os imóveis da Seazone, a implementação e validação do modelo para o Airbnb inteiro apresentou desafios substanciais. Antes de deixar o processo totalmente automático e antes de parar de usar apenas a lógica heurística, há a necessidade de validar extensivamente em todas as tabelas de predição de faturamento se os valores dos imóveis que não são da Seazone fazem sentido. Isso é um processo que demanda tempo e recursos consideráveis, então, por enquanto, o modelo foi apenas testado para o Airbnb inteiro e validado nos imóveis da Seazone.

## 6.1 REQUISITOS FUNCIONAIS

Dentre os requisitos funcionais, pode-se afirmar que todos eles foram cumpridos.

O requisito "Análise e preparação de dados”, foi o primeiro requisito cumprido. Uma análise numérica e gráfica das diversas "features” presentes nas tabelas foi realizada, proporcionando uma compreensão da estrutura e padrões dos dados. Além disso, também foi criadas novas "features”, e a automação desse procedimento foi implementada, garantindo eficiência e escalabilidade.

A "obtenção de um modelo otimizado”, segundo requisito funcional, foi efetuada com sucesso. Utilizando a biblioteca keras_tuner para busca bayesiana, foi possível identificar e salvar o conjunto ótimo de parâmetros para o modelo final. Esse processo assegurou a qualidade do modelo.

O último requisito, "Validação com dados do Airbnb”, foi realizado ao alcançar um F1 Score de 0.9146 na combinação do modelo com a heurística. Essa métrica robusta reflete a capacidade do modelo em lidar com a complexidade dos dados do Airbnb. Além disso, a comparação do MAPE e a gestão dos casos em que o faturamento real é zero proporcionam uma análise detalhada e abrangente da eficácia do modelo em cenários práticos.

## 6.2 REQUISITOS NÃO FUNCIONAIS

O projeto demonstrou sucesso na maioria dos requisitos não funcionais, entretanto, o requisito de "Modelo funcionando em produção automaticamente” foi o único não atendido completamente.

Em relação ao requisito de "desempenho”, o modelo apresentou uma melhoria substancial no "F1 Score”, um aumento de 0.0055 que, embora possa parecer modesto, é significativo quando consideramos a complexidade do problema abordado. Além disso, observou-se uma notável progressão nas métricas de faturamento, como

o MAPE e o número de infinitos, indicando que o modelo está alinhado com as preocupações práticas da Seazone. Por esses motivos esse requisito foi cumprido.

A eficiência no quesito "custos" também foi um ponto forte do projeto. O modelo foi implementado de maneira econômica, custando apenas $2.00 por iteração, sendo que num cenário ideal, o "pipeline" do modelo seria executado uma vez por semana. Esse baixo custo aponta mais um cumprimento dos requisitos não funcionais.

No entanto, o único requisito não funcional ainda não atendido foi o de "Modelo funcionando em produção automaticamente". Apesar de o modelo ter sido testado manualmente com dados do Airbnb, de já ter sido criada a "pipeline" para funcionamento em produção, e de já ter sido validado seus resultados com os imóveis da Seazone, sua implementação total permanece pendente. É necessário realizar uma validação mais a fundo do resultado do modelo para imóveis que não são da Seazone, visto que isso pode trazer resultados indesejados às tabelas de faturamento da Seazone. A substituição da lógica de heurística pela do modelo combinado com ela pode vim depois dos testes.

# 7 CONCLUSÃO

Este projeto se iniciou com a imersão profunda na análise de dados, conduzindo uma exploração meticulosa para compreender a complexidade e as nuances dos conjuntos fornecidos. A etapa inicial foi essencial para o entendimento dos dados servindo como "insights" durante o desenvolvimento do modelo.

A aplicação de testes iniciais de modelos ofereceu uma visão inicial das capacidades e desafios inerentes aos dados, além de maior entendimento da funcionalidade das bibliotecas utilizadas. Esse processo proporcionou uma compreensão mais clara das expectativas realistas em relação aos resultados finais. Cada iteração de teste contribuiu para refinar a abordagem e ajustar estratégias conforme necessário.

A busca por hiperparâmetros na rede neural destacou-se como um ponto crucial no desenvolvimento do projeto. A utilização da biblioteca keras_tuner permitiu uma exploração sistemática e eficiente do espaço de hiperparâmetros, através da busca bayesiana, resultando na obtenção de um modelo otimizado.

Ao usar o modelo para predizer dados do Airbnb, uma descoberta importante surgiu: a complementaridade do modelo com a heurística existente. Enquanto o modelo, por si só, não alcançou níveis satisfatórios de precisão, por conta principalmente de haverem muitas datas indisponíveis não presentes na tabela de reservas, a integração com a heurística resultou em um desempenho superior. Essa sinergia enfatiza a importância de considerar abordagens híbridas e a complementaridade entre métodos automatizados e heurísticas baseadas em regras.

O próximo passo, ainda a ser concluído, é validar a aplicabilidade do modelo em imóveis do Airbnb que não são da Seazone. Esta etapa final é essencial para garantir que o modelo seja capaz de generalizar e fornecer previsões significativas em diferentes contextos. A integração bem-sucedida do modelo em produção depende, em grande parte, dessa validação.

O processo de desenvolvimento de modelos de aprendizado de máquina é inerentemente iterativo, com espaço para contínuo aprimoramento e refinamento. À medida que novos dados se tornam disponíveis e novas técnicas emergem, há sempre a possibilidade de elevar ainda mais a eficácia dos modelos implementados.

Em suma, este projeto representa um marco significativo na aplicação prática de métodos de aprendizado de máquina de redes neurais no contexto específico da Seazone. A trajetória trilhada, desde a análise exploratória até a busca de hiperparâmetros e a integração com heurísticas, ilustra a complexidade e a riqueza desse processo. O comprometimento e a disposição para aprimoramento contínuo são elementos cruciais que delineiam o caminho futuro para a implementação bem-sucedida deste modelo.

# REFERÊNCIAS

AMAZON WEB SERVICES, Inc. **What is Amazon Athena?** [*S.l.*: *s.n.*], 2023a. https://docs.aws.amazon.com/athena/latest/ug/what-is.html. [Acesso em: 05/11/2023].

AMAZON WEB SERVICES, Inc. **What is Amazon S3?** [*S.l.*: *s.n.*], 2023b. https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html. [Acesso em: 05/11/2023].

AMAZON WEB SERVICES, Inc. **What is Amazon SageMaker?** [*S.l.*: *s.n.*], 2023c. https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html. [Acesso em: 05/11/2023].

AMAZON WEB SERVICES, Inc. **What is AWS Lambda?** [*S.l.*: *s.n.*], 2023d. https://docs.aws.amazon.com/lambda/latest/dg/welcome.html. [Acesso em: 05/11/2023].

CHOLLET, Francois *et al.* **Keras**. [Acesso em: 05/11/2023]. 2015. Disponível em: https://github.com/fchollet/keras.

CLOUD SOFTWARE GROUP, Inc. **What is a neural network?** [*S.l.*: *s.n.*], 2023. https://www.spotfire.com/glossary/what-is-a-neural-network. [Acesso em: 05/11/2023].

DEVELOPERS, scikit-learn. **An Overview of Regularization Techniques in Deep Learning (with Python code)**. [*S.l.*: *s.n.*], 2023. [Acesso em: 14/11/2023]. Disponível em: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/.

HOSSIN, Mohammad; SULAIMAN, Md Nasir. A review on evaluation metrics for data classification evaluations. **International journal of data mining & knowledge management process**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 2, p. 1, 2015.

MERRITT, Rick. **O que é MLOps?** Operações de machine learning, MLOps (Machine Learning Operations), são as melhores práticas para as empresas executarem AI com sucesso com a ajuda de uma ampla gama de produtos de software e serviços em

cloud. [*S.l.*: *s.n.*], 2020. [Acesso em: 05/10/2023]. Disponível em: https://blog.nvidia.com.br/2020/09/08/o-que-e-mlops/.

O'MALLEY, Tom; BURSZTEIN, Elie; LONG, James; CHOLLET, François; JIN, Haifeng; INVERNIZZI, Luca *et al.* **Keras Tuner**. [*S.l.*: *s.n.*], 2019. https://github.com/keras-team/keras-tuner. [Acesso em: 05/11/2023].

OXFORD ECONOMICS. **Impacto econômico do Airbnb no Brasil**. [*S.l.*: *s.n.*], 2022. https://news.airbnb.com/wp-content/uploads/sites/4/2022/10/Impacto-econo%CC%82mico-do-Airbnb-no-Brasil_Oxford-Economics.pdf. [Acesso em: 05/10/2023].

PAI, Aravindpai. **Analyzing Types of Neural Networks in Deep Learning**. [*S.l.*: *s.n.*], 2023. https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/. [Acesso em: 02/12/2023].

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. [Acesso em: 05/11/2023].

SANGHVIRAJIT. **A Complete Guide to Adam and RMSprop Optimizer**. [*S.l.*: *s.n.*], 2021. https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer-75f4502d83be. [Acesso em: 27/11/2023].

SHARMA, Sagar; SHARMA, Simone; ATHAIYA, Anidhya. Activation functions in neural networks. **Towards Data Sci**, v. 6, n. 12, p. 310–316, 2017.

SINGH, Himanshi. **Deep Learning 101: Beginners Guide to Neural Network**. [*S.l.*: *s.n.*], 2023. https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/. [Acesso em: 02/12/2023].