Relatório - Tecnologias Escaláveis para a Análise de Dados

João Bragança (8190555)
Pedro Afonso (8090457)
José Fernandes (8190239)

Grupo 3

Escola Superior de Tecnologia e Gestão P.Porto

11 de junho de 2023

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Enquadramento

No âmbito da Unidade Curricular *Tecnologias Escaláveis para Análise de Dados* (TEAD), inserida no primeiro ano do mestrado em Engenharia Informática da Escola Superior de Tecnologia e Gestão, lecionada pelo professor Davide Carneiro, foi-nos proposto simular de forma realista o ciclo de vida de um projeto de análise de dados. Para isso será realizado um projeto em conjunto com uma empresa nacional do ramo da hotelaria, a e-GDS cujo âmbito é a previsão de reservas. Neste âmbito é pretendido o seguinte:

- Fazer a análise do dados fornecidos pela e-GDS;
- Identificar fontes de dados externas que possam ser relevantes para o problema;
- Das diferentes fontes de dados analisar de forma critica a sua relevância para o problema;
- Criar e validar novas variáveis que possam enriquecer o dataset fornecido;
- Treinar e avaliar modelos de Machine Learning com vista à resolução do problema em concreto.

É de notar que a partir da secção 6 o data set foi alterado pela e-GDS, podendo assim alguns dos dados não estarem completamente certos. No entanto, todos os valores apresentados estão bastante próximos da realidade.

1.2 Caso de Estudo

A e-GDS é uma empresa que oferece soluções tecnológicas para a indústria hoteleira, permitindo a automatização de operações de gestão de hotéis. Com o passar dos anos foram recolhendo vários dados de reservas feitas por hóspedes dos seus clientes. Com este trabalho pretende-se desenvolver um modelo de previsão de ocupação/afluência ao longo do tempo para os diferentes hotéis cujos dados foram disponibilizados, juntamente com dados de fontes externas que sejam consideradas relevantes para a resolução do problema.

1.3 Fases do trabalho

Identificação de fontes de dados externas

O trabalho será desenvolvido em várias fases, sendo a primeira a identificação de fontes de dados externas e públicas. O objetivo passa por recolher dados que possam ser úteis para a previsão de reservas dos hotéis. Essas fontes de dados podem incluir, informações sobre eventos locais (como concertos, feiras e conferências), condições meteorológicas, tendências de viagem, entre outras.

Integração de dados

Aos dados fornecidos pela e-GDS serão integrados dados obtidos mediante fontes públicas. Além disso, é necessário juntar os diferentes conjuntos de dados fornecidos entre si (uma vez que estão em tabelas diferentes), assim como os dados externos. Esta atividade envolverá a correspondência de colunas comuns entre os diferentes conjuntos de dados e a criação de novas colunas derivadas das fontes de dados obtidas.

Análise exploratória de dados

Após a integração dos dados, será feita uma análise exploratória de dados, dando uma melhor perspetiva sobre os valores que as variáveis podem tomar e como estas se relacionam entre si.

Aquisição e tratamento de dados

No final da integração, é necessário tratar os dados resultantes da junção dos diferentes *datasets*, uma vez que, não estão prontos a ser integrados numa solução de machine learning. Existem dados em falta, dados sem significado e dados com o mesmo significado representados de formas diferentes. Para resolver o problema técnicas de limpeza de dados serão aplicadas, como por exemplo, a eliminação, substituição, normalização, entre outras.

No final do tratamento, é possível enriquecer o dataset com novos atributos, atributos estes que podem ser derivados dos já existentes aplicando técnicas de *Feature Engineering*.

Desenvolvimento de modelo de previsão

Já com o dataset completo serão aplicadas técnicas de *machine learning* que permitirão obter previsões de reservas de preços e cancelamentos em unidades hoteleiras.

Por fim, serão treinados e avaliados vários modelos de forma a validar a sua utilizada e eficácia na resolução do problema proposto.

1.4 Descrição do Problema

O problema abordado neste trabalho é a previsão de ocupação/afluência ao longo do tempo em hotéis. Embora a empresa e-GDS disponibilize um dataset com dados de reservas efetuadas pelos seus clientes, o objetivo do trabalho vai

além da utilização desses dados. É necessário identificar fontes de dados externas e públicas que possam ser indicadores importantes para a previsão de afluência e integrá-las com os dados fornecidos pela e-GDS para análise conjunta. Assim, o desafio é adquirir, tratar e integrar essas fontes de dados com os dados existentes para melhorar a previsão de ocupação/afluência em hotéis. Além da previsão de ocupação/afluência ao longo do tempo, outro problema secundário é a previsão do cancelamento de reservas em hotéis e de preços. É importante que o hotel possa prever com antecedência quais reservas que poderão ser canceladas, permitindo que possam atuar sobre essa informação atempadamente. Dessa forma, o desafio é identificar os indicadores relevantes para a previsão de cancelamentos de reservas, bem como utilizar técnicas de análise de dados e modelação para fazer previsões. A resolução desse problema pode trazer benefícios para o hotel, como a possibilidade de otimizar a gestão de disponibilidade de quartos e maximizar a ocupação.

2 Revisão da Literatura

2.1 Influências nas reservas de hotéis

A influência nas reservas de hotéis é um tema com um impacto relevante para a indústria hoteleira, uma vez que, a ocupação é um fator-chave para o sucesso financeiro de um hotel. Diversos fatores podem influenciar a reserva de um hotel, tais como: a satisfação do cliente, as avaliações online, os eventos, feriados, fatores meteorológicos, períodos de férias, preços entre outros. Lee e Jeon (2012)[5] mostram que a satisfação dos clientes está diretamente relacionada à qualidade dos serviços e instalações do hotel, à interação com os funcionários, à limpeza e conforto dos quartos, e ao valor percebido relativamente ao preço. Além disso, eles destacam que a intenção de retorno é influenciada pela satisfação anterior e pelo compromisso com a marca do hotel. Para Xu e Ye (2015)[10] as avaliações online são um fator importante na decisão de reserva dos hotéis, isto é, avaliações positivas aumentam a probabilidade de reserva e avaliações negativas diminuem. Essas avaliações podem ser encontradas em diversas plataformas, como TripAdvisor, Booking.com, Google, entre outras. Segundo Zhang et al. (2018)[11], os hotéis devem monitorizar e responder às avaliações online, oferecer incentivos para os clientes deixarem avaliações, pois para eles avaliações positivas têm um efeito maior do que as negativas, e o número de avaliações também influencia a decisão de reserva. Além desses fatores, outros fatores externos também poderão ter um impacto significativo nas reservas dos hotéis, tais como:

- **Períodos de férias:** Os períodos de férias podem ter um impacto significativo nas reservas de hotéis, especialmente em destinos turísticos populares. Durante os períodos de férias, a procura por hotéis aumenta de forma significativa, o que pode levar a um aumento nos preços. Os períodos de férias podem ser divididos em diferentes categorias, como alta temporada, baixa temporada e média temporada. Por exemplo, a alta temporada geralmente apresenta maior procura e preços mais elevados, enquanto a baixa temporada apresenta menor procura e preços mais baixos. Esses fatores devem ser considerados pelos hotéis ao desenvolver sua estratégia de preços e distribuição Kimes et al. (1998)[4]. A ocupação dos hotéis pode variar também de acordo com a época do ano, sendo que a procura é geralmente maior nos meses de verão, nas épocas festivas (como o Natal, por exemplo), e em feriados prolongados.

- **Eventos:** Eventos como grandes eventos culturais, festivais de música, exposições de arte, feiras gastronómicas, e eventos desportivos, podem influenciar as reservas de hotéis, pois atraem visitantes de fora da cidade ou do país. A procura durante esses eventos pode ser alta e os hotéis podem ajustar os seus preços conforme a procura e a importância do evento. Os hotéis podem trabalhar em parceria com os organizadores de eventos para oferecer pacotes especiais de alojamento.

- **Fatores meteorológicos**: As condições meteorológicas, como temperatura, precipitação e neve, podem influenciar as reservas de hotéis, especialmente em destinos de lazer. Por exemplo, em destinos de praia, a procura por hotéis pode aumentar durante os meses de verão, quando as temperaturas são mais quentes. Por outro lado, em destinos de neve, quando as temperaturas são mais frias, a procura pode ser maior para quem procura, por exemplo, desportos de inverno. Kilic et al (2005)[3].

- **Preços:** O preço dos hotéis pode ser um fator determinante na escolha

de um local de alojamento. Preços mais altos podem desencorajar alguns turistas de reservar quartos de hotel, enquanto preços mais baixos podem atrair uma maior quantidade de visitantes.

- **Qualidade das instalações e Avaliações:** A qualidade das instalações de um hotel, incluindo o nível de conforto, a limpeza, o atendimento ao cliente, a localização e as comodidades oferecidas, pode influenciar a decisão dos turistas em reservar um quarto no hotel, assim como as avaliações e recomendações efetuadas pelos clientes anteriores.

2.2 Glossário do Negócio

Termo	**Descrição**
e-GDS	Empresa de tecnologia especializada em soluções de distribuição global para gestão do comércio hoteleiro
Comércio hoteleiro	Conjunto de atividades relacionadas à gestão e operação de hotéis e outros estabelecimentos de hospedagem
Automatização de operações	Processo de substituição de atividades manuais por processos automatizados, visando aumentar eficiência e reduzir erros
Previsão de ocupação	Estimativa do número de quartos que estarão ocupados num determinado período
Reservas	Solicitações feitas pelos clientes para reservar quartos em um hotel
Booking.com	Plataforma online de reservas de hotéis e outros estabelecimentos de hospedagem

Afluência	Número de clientes que frequentam o hotel num determinado período
Cadeia hoteleira	Grupo de hotéis que operam sob uma mesma marca ou empresa
Gestão de reservas	Processo de receber, confirmar e organizar as reservas dos clientes
Integrações	Conexões entre diferentes plataformas ou sistemas que permitem a troca de informações
Ocupação	Percentagem de quartos ocupados relativamente ao número total de quartos disponíveis num determinado período
Parceiro tecnológico	Empresa que fornece soluções tecnológicas para outras empresas

3 Dados e-GDS

O dataset descrito nesta secção foi fornecido pela e-GDS e contém 4 tabelas com dados relacionados a reservas hoteleiras:

- Quartos reservados - Contém informações acerca das reservas em si
- Hotel - Contém os dados dos Hotéis
- Tipologias - Contém informações acerca dos quartos dos hotéis
- Facilities - Contém os dados das *facilities* dos hotéis

Cada um dos *datasets* é constituído por diferentes dados, sendo estes descritos nas tabelas seguintes. Todas as tabelas especificam o nome da coluna (Atributo), o tipo de dados, os valores permitidos (quando aplicável), se aceita valores nulos e a descrição da coluna.

Tabela 3.1: Dicionário de dados: Quartos Reservados

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
Hotel ID	int	>= 1	Não	Id do hotel
Reserve ID	int	>= 1	Não	Id da reserva
País	string	N/A	Não	País de orgiem da pessoa que reservou
Estado da reserva	string	Cancelado, Confirmado, CourtesyHold, Modificada, Não Registado, Pendente, Registado	Não	Estado em que a reserva se encontra
Room ID	int	>= 1	Não	ID do quarto reservado
Tipo de Quarto	string	N/A	Não	Tipo do quarto reservado
RatePlan	string	N/A	Não	
Data da reserva	string	YYYY-MM-DD HH:MM:SS.sss	Não	Data em que a reserva foi realizada
Data chegada	string	DD/MM/YYYY	Não	Data de chegada do cliente
Data de partida	string	DD/MM/YYYY	Não	Data de partida do cliente
Número de noites	int	>= 1	Não	Número de noites reservadas
Ocupação	int	>= 1	Não	Quantidade de pessoas da reserva
Adultos	int	>= 1	Não	Quantidade de adultos da reserva
Crianças	int	>= 0	Não	Quantidade de crianças da reserva
Bebés	int	>= 0	Não	Quantidade de bebés da reserva
Preço (€)	float	>= 0	Não	Custo da reserva em euros

Tabela 3.2: Dicionário de dados: Hotéis

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
Hotel ID	int	$>= 0$	Não	Id do hotel
Localização	string	N/A	Não	Localização do hotel
Estrelas	int	0, 1, 2, 3, 4, 5	Não	Estrelas do hotel
Idade Máxima de Crianças	int	$>= 0$	Não	Idade limite em que uma pessoa é considerada criança em anos
Idade Máxima de Bebés	int	$>= 0$	Não	Idade limite em que uma pessoa é considerada bebé em meses
Hora máxima de check-in	string	Hora com minutos e segundos	Não	Hora máxima em que as pessoas podem dar check in
Quantidade de quartos	int	$>= 1$	Não	Quantidade de quartos que o hotel possui

Tabela 3.3: Dicionário de dados: Tipologias

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
Hotel ID	int	$>= 0$	Não	Id do hotel
Room ID	int	$>= 0$	Não	Id do quarto
Tipo de quarto	string	Qualquer string	Não	Tipo do quarto
Quantidade	int	$>= 0$	Não	Quantidade de quartos existentes
Capacidade máxima	int	$>= 0$	Não	Capacidade máxima em termos de ocupantes
Capacidade mínima	int	$>= 0$	Não	Capacidade mínima em termos de ocupantes
Capacidade máxima de adultos	int	$>= 0$	Não	Capacidade máxima de adultos
Capacidade mínima de adultos	int	$>= 0$	Não	Capacidade mínima de adultos
Capacidade máxima de crianças	int	$>= 0$	Não	Capacidade máxima de crianças
Capacidade mínima de crianças	int	$>= 0$	Não	Capacidade mínima de crianças
Capacidade máxima de bebés	int	$>= 0$	Não	Capacidade máxima de bebés
Capacidade máxima de camas extra	int	$>= 0$	Não	Capacidade máxima de camas extra
Capacidade máxima de camas extra (crianças)	int	$>= 0$	Não	Capacidade máxima de camas extra para crianças
Capacidade máxima de berços extra	int	$>= 0$	Não	Capacidade máxima de berços extra

Tabela 3.4: Dicionário de dados: Facilities

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
Hotel ID	int	>= 0	Não	Id do hotel
Facility ID	int	>= 0	Não	Id da facility
Nome	string	Qualquer string	Não	Nome da facility

4 Fontes de dados externas

4.1 Feriados

dataset criado através da utilização do serviço sapo [7]. Este serviço fornece os feriados existentes para um determinado ano, desta forma foi criado um ficheiro em formato csv com todos os feriados desde 2022 até 2023. Este dataset será relevante uma vez que a afluência dos hotéis aumenta nos dias de feriados ou fins de semana.

Tabela 4.1: Dicionário de dados: Feriados

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
Date	date	dd/MM/aaaa	Não	Data
day	int	1 - 31	Não	Dia do mês
dayOfWeek	int	1 - 7	Não	Dia da semana
month	int	1 - 12	Não	Mês
trimester	int	1, 2, 3 ou 4	Não	Trimestre
year	int	2011 - 2023	Não	Ano
isHoliday	boolean	0 ou 1	Não	Se é feriado

4.2 Meteorologia

Este dataset foi criado com recurso ao website Meteostat [6] que contêm uma enorme quantidade de dados sobre a meteorologia global. Desta forma, procuramos pela cidade que corresponde à localização mais próxima do hotel e nas datas compreendidas entre *01-01-2022* e *23-04-2023* (para que seja possível coincidir com as datas das reservas), posteriormente fizemos download em formato .csv do dataset disponibilizado pelo website para aquela zona e unimos num único dataset com as alterações mencionadas abaixo. Este dataset é importante, uma vez que a afluência dos hotéis poderá aumentar nos dias de calor ou de maior temperatura e diminuir nos dias de temperatura mais baixas.

Tabela 4.2: Dicionário de dados: Meteorologia

Atributo	Tipo	Valores Permitidos	Valores Nulos	Descrição
date	date	aaaa-mm-dd	Não	Data
tavg	float	Qualquer valor	Sim	Temperatura média
tmin	float	Qualquer valor	Sim	Temperatura mínima
tmax	float	Qualquer valor	Sim	Temperatura máxima
prcp	float	>= 0	Sim	Total de precipitação
wdir	int	>= 0 e <= 360	Sim	Direção do vento
wspd	float	>= 0	Sim	Velocidade do vento
wpgt	float	>= 0	Sim	Pico de rajada
pres	float	>= 0	Sim	Pressão do ar
city	string	Qualquer string	Não	Cidade correspondente à localização do hotel

4.2.1 Alterações e Limpeza

- Foram eliminadas as linhas que apenas continham a data e a cidade e não continham dados meteorológicos.

- Foi adicionado uma nova coluna *city* ao dataset para permitir integrar os dados meteorológicos com as reservas.
- Foi eliminado a coluna *snow* pois a profundidade da neve não é relevante para o contexto do problema.

4.3 Eventos

Dataset de eventos que foi elaborado com pesquisa e com recurso também do ChatGPT [1] que indicou eventos relevantes e úteis para o problema. Os eventos serão talvez se não o dataset com maior relevância e importância na afluência dos hotéis. Isto porque, um grande evento pode levar ao alojamento de milhares de pessoas naquela localização. É importante mencionar que uma vez que o ChatGPT apenas contêm dados até 2021 as datas dos eventos podem estar erradas e por isso pode levar a previsões ou conclusões menos precisas.

Tabela 4.3: Dicionário de dados: Eventos

Atributo	**Tipo**	**Valores Permitidos**	**Valores Nulos**	**Descrição**
Location	string	Qualquer string	Não	Cidade/Localização do evento
Event	string	Qualquer string	Não	Evento
Start_date	date	dd/mm/aaaa	Não	Data de ínicio do evento
End_date	date	dd/mm/aaaa	Não	Data de fim do evento

5 Análise dos dados

5.1 E-GDS

5.1.1 Análise Inicial

De forma a analisar os dados eles foram lidos em Spark, como mencionado na secção anterior e algumas modificações foram feitas. Nomes de colunas foram alterados e os tipos de dados foram transformados no tipo correto.

Tabela 5.1: Descrição da tabela Hotel

summary	hotel_ID	localizacao	estrelas	idade_max_criancas	idade_max_bebes	qtd_quartos	area_localizacao
count	145	145	145	145	145	145	145
mean	N/A	N/A	1,79	7,32	20,02	30,02	N/A
stddev	N/A	N/A	1,78	6,15	20,01	33,80	N/A
min	20	N/A	0	0	0	1	N/A
max	561	N/A	5	36	168	192	N/A

O conjunto de dados apresentado na tabela 5.1 é composto por informações de 145 hotéis e 7 colunas distintas. As colunas incluem o ID do hotel, sua localização, número de estrelas, idade máxima para crianças e bebés, número de quartos e área de localização. A média de estrelas dos hotéis é de 1,79. Não foram identificados valores nulos na tabela. Porém, é importante ressaltar que foi identificado um valor errado na idade máxima para crianças, que foi de 36 anos. Além disso, alguns hotéis foram registados com localizações anormais, como ”.”e ”teste”. Ao verificar se esses hotéis possuíam reservas, constatou-se que 57 hotéis não contavam com nenhuma reserva registada.

Tabela 5.2: Descrição da tabela Tipologia 1/2

summary	hotel_ID	room_ID	tipo_quarto	quantidade	capacidade_maxima	capacidade_minima	capacidade_max_adultos
count	741	741	741,00	741	741	741	741
mean	N/A	N/A	N/A	5,22	3	1,09	2,75
stddev	N/A	N/A	N/A	8,71	1,43	0,58	1,34
min	20	81	N/A	0	1	1	1
max	561	2905	N/A	90	10	10	10

Tabela 5.3: Descrição da tabela Tipologias 2/2

summary	capacidade_max_criancas	Capacidade_mx _bebés	capacidade_max_camas_extra	capacidade_mx_camas_criancas	capacidade_max_bercos
count	741	741	741,00	741	741
mean	1,08	0,59	0,14	0,15	0,361
stddev	1,38	0,64	0,38	0,4	0,55
min	0	0	0,00	0	0
max	8	3	2,00	2	2

Este dataset que se encontra na tabela 5.2 e 5.3 refere-se à tabela tipologia e contém como principais atributos as colunas tipo de quarto, quantidade, capacidade máxima e mínima, e capacidade máxima de adultos e crianças. Existem 741 tipos de quartos diferentes. A capacidade máxima média é de 3 pessoas, enquanto a capacidade mínima média é de 1 pessoa. Os valores mínimo e máximo variam em função das características de cada tipo de quarto, com um mínimo de 0 e um máximo de 90 para a quantidade de quartos. A capacidade máxima varia de 1 a 10 pessoas, enquanto a capacidade máxima de adultos varia de 1 a 10 e a capacidade máxima de crianças varia de 0 a 8. De notar que a tabela não contém valores nulos. Numa primeira observação verificamos que poderão existir campos irrelevantes, sendo eles os seguintes:

- "Capacidade mínima": irrelevante mais de 90% tem o valor de "1";
- "Capacidade mínima de adultos": irrelevante mais de 90% o valor de "1"
- "Capacidade mínima de crianças": irrelevante mais de 95% tem o valor de "0"
- "Capacidade máxima de camas extra": irrelevante, é um valor subinten-

dido na capacidade máxima de adultos, bebés e crianças;

- "Capacidade máxima de camas extra (crianças)": irrelevante, é um valor subintendido na capacidade máxima de adultos, bebés e crianças;
- "Capacidade máxima de berços extra": irrelevante, é um valor subintendido na capacidade máxima de adultos, bebés e crianças;

Tabela 5.4: Descrição da tabela Quartos Reservados 1/2

summary	hotel_ID	Reserve_ID	pais	estado_reserva	room_ID	tipo_quarto	rate_plan	num_noites
count	25105	25105	25105	25105	25105	25105	25105	25105
mean	N/A	N/A	N/A	N/A	1944,56	N/A	N/A	2,28
stddev	N/A	N/A	N/A	N/A	621,15	N/A	N/A	4,68
min	20	1418210	N/A	N/A	81	N/A	N/A	1
max	561	1723815	N/A	N/A	2897	N/A	N/A	481

Tabela 5.5: Descrição da tabela Quartos Reservados 2/2

summary	ocupacao	adultos	criancas	bebes	preco_euros	data_reserva	data_chegada	data_partida
count	25105	25105	25105	25105	25105	730	730	730
mean	1,87	2,04	0,06	1,87	243,33	N/A	N/A	N/A
stddev	0,66	1,02	0,30	0,66	373,54	N/A	N/A	N/A
min	1	1	0	1	0	01/01/2022	01/01/2022	02/01/2022
max	9	30	4	9	20683	31/12/2023	05/04/2024	31/05/2024

O conjunto de dados apresentados na tabela 5.4 e 5.5 é o nosso dataset principal. Este contém informações sobre reservas de hotéis, incluindo detalhes como país, estado da reserva, tipo de quarto, plano de tarifas, número de noites, ocupação, número de adultos, crianças e bebés, preço em euros, e datas de reserva, chegada e partida. Ao analisar o conjunto de dados, podemos ver que há 25.105 entradas e 13 atributos. A média de noites reservadas é de 2,28, com um desvio padrão de 4,68, indicando uma grande variação nos períodos de estadia. Além disso, a média de ocupação é de 1,87, o que sig-

nifica que, em média, cada quarto é ocupado entre uma a duas pessoas. No entanto, não conseguimos perceber se essa ocupação está correta, uma vez que não é coerente com a soma de adultos e crianças. Vamos optar por assumir a ocupação como dado correto[1]. Verifica-se que os dados referentes a bebés deverá estar errado por quase todas as reservas terem incluídos bebés, já com as crianças essa situação não se verifica. Relativamente aos preços em euros, a média é de 243,33 com um desvio padrão de 373,54, mostrando que há uma grande variação de preços nas reservas de hotéis. O número máximo de noites reservadas é de 481, o que pode indicar que algumas pessoas reservam hotéis por longos períodos de tempo, como para estadias prolongadas ou negócios, no entanto, como a média de reservas é 2,28 podemos verificar que o número 481 é um outlier. As datas de reserva variam de 01/01/2022 a 31/12/2023, com datas de chegada e partida que vão até 05/04/2024 e 31/05/2024, respetivamente. Isso sugere que as reservas foram feitas com bastante antecedência e que há uma ampla janela de tempo para reservar um quarto de hotel. Como análise complementar fizemos uma verificação direcionada para os seguintes campos:

1. Datas: Para verificar se as datas do dataset se encontravam coerentes, foi analisado se as datas de saída do hotel eram sempre superiores às datas de entrada e se as datas de reserva eram inferiores às datas de chegada. Também foi calculado o número de noites atráves da data de chegada e de saída e todos os valores se encontravam corretos.

2. RatePlan: A partir dos dados apresentados na tabela 5.6, podemos concluir que os “Rateplans” mais frequentes nas reservas são ”MR - Main Rate”e ”Normal”, ambos com mais de 2400 reservas. Esses resultados sugerem que existe uma grande quantidade de reservas feitas com os mes-

[1]Depois de fazer grande parte do trabalho foi-nos informado que a ocupação era relativa ao quarto, no entanto não foi possível tomar isto em consideração uma vez que o projeto já tinha avançado sem este pressuposto

Tabela 5.6: Tabela com o 10 RatePlan mais Relevantes

Rate Plan	**Contagem**
MR - Main Rate	2545
Normal	2417
BAR	1827
WEB (Best Available Rate)	1707
Bar	1693
WebSite	1681
Main Rate BB	1057
Standard	557
(WEB) Best Available Rate	556
Não Reembolsável	401

mos rate plans, o que pode ter implicações na estratégia de preços e de previsão de reservas com base no rate plan

3. Preço:

hotel ID	**preco_medio_por_noite**
284	51.18
328	52.85
522	56.74
560	57.54
283	58.35

(a) Hoteís baratos

hotel_ID	**preco_medio_por_noite**
358	263.39
445	252.59
513	250.0
443	237.41
225	235.01

(b) Hoteís caros

Figura 5.1: Comparação entre preço por noite em hoteís caros e baratos

O preço é um fator importante a ser considerado ao reservar um hotel, pois pode impactar diretamente o número de reservas feitas. Com isso

em mente, realizamos uma agregação de preços divididos pelo número de noites em cada hotel, permitindo-nos obter o preço médio por noite durante o período de estudo. As tabelas 5.1a e 5.1b apresentam os hotéis com os preços médios mais baixos e mais altos por noite, respetivamente. É possível observar que o hotel com o preço mais baixo é o de ID 284, com um valor médio de 51,18 euros por noite. Em contrapartida, o hotel com o preço mais alto é o de ID 358, com um valor médio de 263,39 euros por noite.

4. País: Para termos uma compreensão mais abrangente das reservas de

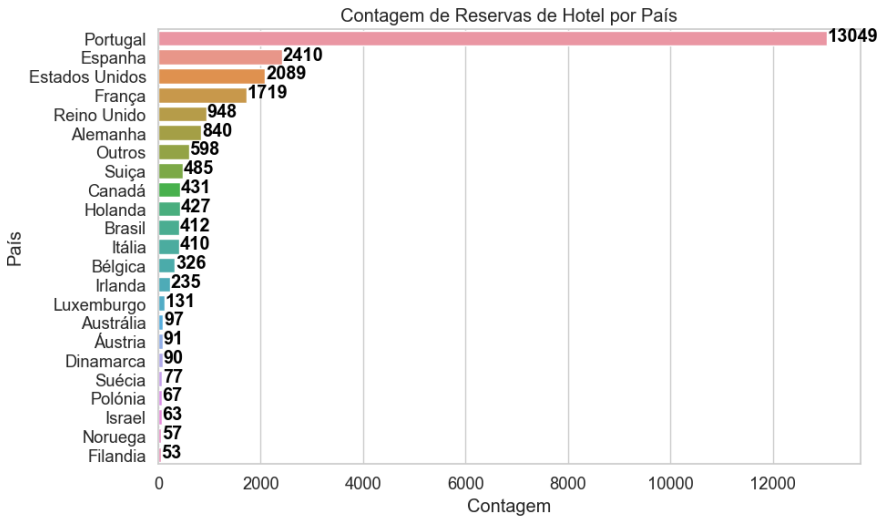


Figura 5.2: Países com mais reservas

hotéis e das nacionalidades dos hóspedes, a agregação dos dados por país foi uma etapa essencial. Inicialmente, o conjunto de dados possuía informações de várias nacionalidades, mas para melhor visualização dos resultados, foi necessário agrupá-las por país e ordená-las por número de reservas. A contagem de reservas por país foi realizada e utilizada para criar gráficos que destacam os países com maior número de reservas. Para melhorar a visualização dos dados como mostra a figura 5.2, os países com menos de 50 reservas foram agrupados em uma categoria

denominada ”Outros”, permitindo uma melhor compreensão dos dados e identificação dos países com maior e menor número de reservas. Com isso, descobrimos que Portugal é o país com maior número de reservas de hotéis, com 13049 reservas, e a Bulgária é o país com o menor número de reservas, com apenas 53 reservas. O agrupamento dos países menos frequentes numa única categoria ”Outros”, resultou em 598 reservas, com 43 países com apenas 1 ou 2 reservas. Esses países podem ser considerados outliers em futuras análises de previsão de reservas, mas neste momento com essa agregação outros, chegamos à conclusão que não vamos remover qualquer país sem uma análise mais profunda. Em resumo, a agregação por país na nossa perspetiva é uma etapa crucial para a compreensão dos dados e identificação dos países com maior e menor número de reservas, permitindo a tomada de decisões mais informadas relativamente a uma futura previsão de reservas.

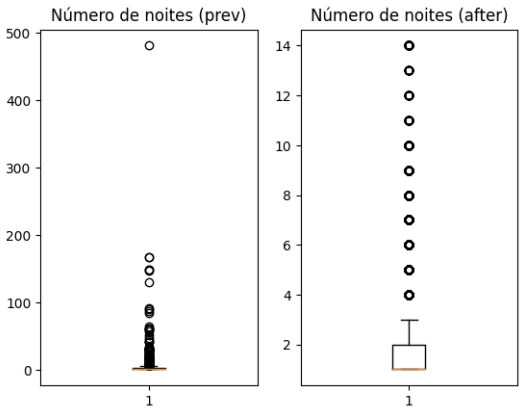
5.1.2 Análise de Outliers

A remoção de outliers é uma técnica comumente utilizada para eliminar valores discrepantes que podem distorcer a análise estatística dos dados e afetar a precisão das conclusões obtidas a partir deles. A existência destes valores pode também influenciar negativamente a qualidade de um modelo de machine learning.

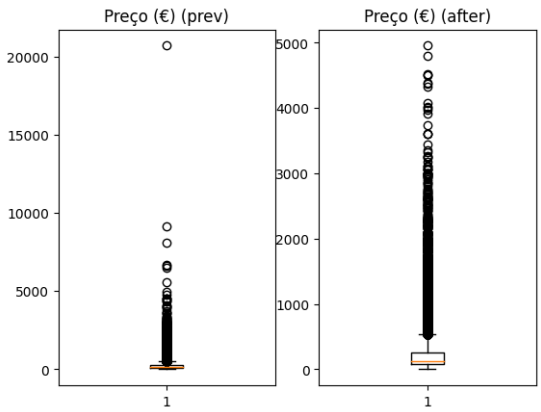
Número de Noites e Preço

Identificado que a coluna ”Número de Noites”continha valores muito divergentes, que eram notavelmente outliers. Para verificar os outliers no número de noites, utilizou-se o Three Sigma, uma técnica estatística que identifica valores discrepantes, para isso, aplicou-se a seguinte fórmula:

- (Limite Superior = média + 3 * desvio padrão)



(a) Análise Outliers: Número de noites



(b) Análise Outliers: Preço

- Média Número noites: 2,28;
- Desvio padrão número de Noites: 4,68
- Limite superior = 2.28 + 3 * 4.68
- Limite superior = 14,32

Este resultado foi arredondado para 14. No limite inferior, consideramos o valor 0, uma vez que, não existem reservas negativas. Assim, aplicando o filtro ao número de noites, temos o total de 115 reservas. Para efetuar a análise dos outliers foi criada uma biblioteca em python de suporte [2].

Como mostrado nas figuras 5.3a e 5.3b. Para o número de noites existem 115 reservas com um valor superior a 14, que por sua vez dá para verificar que esse número de noites anormal afeta o preço. Com esta análise, verifica-se que existem 7 reservas com valores superiores a 5000 Euros.

Número de Reservas

Com base na análise das reservas por hotel, podemos observar uma grande variação nos números de reservas, com alguns hotéis apresentando um número

[2]Em anexo "Analises.py"

significativamente menor de reservas do que outros. Para garantir uma análise mais robusta, decidimos considerar como outliers os hotéis com um número de reservas inferior a 13, com base no critério de que esses hotéis apresentam uma frequência muito baixa. Dessa forma, identificamos cinco hotéis como outliers, o que resultou na remoção de 20 reservas da análise. A tabela 5.8 apresenta os hotéis identificados como outliers, juntamente com suas localizações e o número de reservas. A escolha do número 13 foi feita com base na análise dos

Quartil	**Valor**
0.00	1.00
0.25	45.75
0.50	130.00
0.75	229.00
1.00	3325.00

Tabela 5.7: Tabela de Quartis - Reservas por hotel

quartis como mostra a tabela 5.7 de quantidade de reservas por hotel, verificamos que o primeiro quartil tem um valor de 45,75 reservas por hotel. No entanto, consideramos que remover todos os hotéis abaixo desse valor teria um impacto significativo na amostra, resultando na eliminação de um número considerável de hotéis. Dessa forma, escolhemos um valor intermédio de 13 reservas. Vale ressaltar que a escolha desse valor foi feita com base numa avaliação cuidadosa das implicações práticas dessa decisão, procurando minimizar o impacto na amostra e preservar a representatividade dos dados.

Tabela 5.8: Hoteis Com menos reservas

hotel_ID	localizacao	count
302	Tomar	5
491	Vilamoura	5
390	Alijó	5
442	Funchal	4
513	Tavira	1

5.2 Feriados

O dataset Feriados que se encontra na tabela 5.9 apresenta informações sobre as datas presentes no conjunto de dados, tais como dia, dia da semana, mês, trimestre, ano, indicação de feriado e nome da semana em português. Em resumo, a tabela fornece informações sobre as datas presentes no conjunto de dados, com destaque para os dias de fim de semana e feriados. A informação deste dataset poderá ser muito útil para perceber se existe mais reservas durante os períodos com feriados ou fim de semana.

Tabela 5.9: Descrição da tabela Feriados

summary	day	dayOfWeek	month	trimester	year	is_holiday	portugueseWeekName	date
count	730	730	730	730	730	730	730	730
mean	15,7	4	6,53	2,51	2022,50	0,04	N/A	N/A
stddev	8,8	2	3,45	1,12	0,50	0,19	N/A	N/A
min	1	1	1	1	2022	0	N/A	01/01/2022
max	31	7	12	4	2023	1	N/A	31/12/2023

5.3 Meteorologia

O dataset Meteorologia contêm dados meteorológicos de 28 localizações diferentes compreendidos entre as datas *2022-01-01* e *2023-04-23*. Estas localizações

correspondem com as zonas onde os hóteis se encontram. Como se segue na tabela 5.10, este dataset fornece-nos informações como temperatura média, máxima, mínima, precipitação total daquele dia, informações do vento e a cidade onde os valores foram registados. Este dataset apresenta também poucos valores nulos e permite-nos retirar informações como:

- Maior temperatura registada: 44.3
- Menor temperatura registada: -4.7
- Temperatura média de todos os registos: 15.59

Tabela 5.10: Descrição da tabela Meteorologia

summary	tavg	tmin	tmax	prcp	wdir	wspd	wpgt	pres	city	date
count	13293	13285	13289	9935	13275	13275	8352	13275	13293	13275
mean	15,59	11,65	20,13	2,57	193,56	11,67	31,49	1019,3	N/A	N/A
stddev	5,13	5,19	6,01	6,98	112,44	6,09	9,95	6,4	N/A	N/A
min	-0,3	-4,7	3,1	0	0	1,4	7,4	992,9	N/A	01/01/2022
max	35,1	28,6	44,3	118	360	49,1	87	1040,7	N/A	23/04/2023

5.4 Eventos

Este dataset contêm eventos relevantes compreendidos entre as datas de *01/06/2022* e *05/11/2023* e a respetiva localização para que seja possível relacionar com as reservas e meteorologia. Da descricão que se segue 5.11 apenas conseguimos retirar que contamos com 59 eventos diferentes compreendidos nas datas mencionadas acima. Neste dataset não existem valroes nulos.

Ï

Tabela 5.11: Descrição da tabela Eventos

summary	Location	Event	start_Date	end_date
count	59	59	59	59
mean	N/A	N/A	N/A	N/A
stddev	N/A	N/A	N/A	N/A
min	N/A	N/A	01/06/2022	11/06/2022
max	N/A	N/A	05/11/2023	05/11/2023

6 Tratamento de dados

Neste capítulo, apresentamos um resumo de tratamento de dados para cada um dos conjuntos de dados analisados nos capítulos anteriores: Hotél, Tipologias, Facilities e Quartos Reservados. Foram realizadas diversas ações, tais como remoção de colunas irrelevantes, remoção de outliers e adição de novas colunas. O objetivo é obter um conjunto de dados mais limpo e coerente para uma melhor análise do dataset principal.

6.1 Dataset Hotel

- Remoção de 57 linhas de hotéis para os quais não existem reservas
- Adição de coluna "area_localizacao"para permitir fazer a integração dataset meteorologia e eventos;

6.2 Dataset Tipologias

- Remoção da coluna "Capacidade mínima": irrelevante mais de 90% tem o valor de "1";
- Remoção da coluna "Capacidade mínima de adultos": irrelevante mais de 90% tem o valor de "1"
- Remoção da coluna "Capacidade mínima de crianças": irrelevante mais de 95% tem o valor de "0"

6.3 Dataset Facilities

Não será feita nenhuma limpeza uma vez que não se conseguiu identificar como poderão ser usados os dados. Alguns hotéis têm muitas facilites, outros poucas. Além disso, existe uma grande quantidade de variedade dos dados, sendo que são poucos os hotéis que possuem facilites similares.

6.4 Dataset Quartos Reservados

- Remoção de 4 linhas com estado de reserva "CourtesyHold". (25105-4 = 25101)
- Remoção de 115 linhas onde o número de noites é superior a 14. (outliers) (25101-115 = 24986)
- Remoção de cinco hotéis com ids(491, 302, 390, 442 e 513), que equivale a 20 linhas de reservas, onde o número de reservas por hotel é inferior a 13 (outliers). (24986-20=24966)
- Remoção de 1 linha onde o preço por noite por ocupação é maior ou igual a 1000€ (24966-1=24965)
- Remoção de 10 linhas duplicadas (24965-10=24955)
- Adicionada uma nova coluna chamada "dif_data_chegada_data_reserva". Essa coluna representa a diferença em dias entre a data de chegada (coluna "data_chegada") e a data da reserva (coluna "data_reserva").
- Normalização da coluna "rate_plan"é feita usando a função withColumn para adicionar uma nova coluna chamada "rate_plan_normalized". As expressões regulares são usadas para encontrar padrões específicos nos valores da coluna "rate_plan"e atribuir valores correspondentes na nova coluna. Após a normalização, o campo "rate_plan"foi agrupado e contado,

resultando em 44 categorias distintas a partir de 230 valores iniciais. Categorias com uma contagem inferior a 20 foram substituídas por ”other”. Para melhor exemplificar o tipo de normalização na imagem 6.1 mostramos o tipo de normalização que é feita.

Figura 6.1: Exemplo Normalização

```
//Remover espaços em branco extras
val dfTrimmed = result_after_cleaning.withColumn("tipo_quarto", trim(col("tipo_quarto")))
//Converter para letras minúsculas
val dfLower = dfTrimmed.withColumn("tipo_quarto", lower(col("tipo_quarto")))
//Remover caracteres especiais e pontuações
val dfCleaned = dfLower.withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "[^a-zA-Z0-9 ]", ""))
//aplicar regexp_replace
val dfReplace = dfCleaned.withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)quarto|room", ""))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)standard|standart", "standard"))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)triple|triplo", "triplo"))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)double|duplo|db|dbt", "duplo"))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)twin1", "twin"))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)comfort", "confort"))
                         .withColumn("tipo_quarto", regexp_replace(col("tipo_quarto"), "(?i)econmico", "economico"))

val dfNormalized = dfReplace.withColumn("tipo_quarto_normalized", normalizeString(col("tipo_quarto")))
 //Substituir os valores correspondentes e manter os demais valores como estão
val dfCategorized = dfNormalized.withColumn("tipo_quarto_normalized",
                  when(lower(col("tipo_quarto")).rlike("duplo standard|double standard|dpl "), "duplo standard")
                  .when(lower(col("tipo_quarto")).rlike("twin standard"), "twin standard")
                  .when(lower(col("tipo_quarto")).rlike("suite"), "suite")
                  .when(lower(col("tipo_quarto")).rlike("superior"), "superior")
                  .when(lower(col("tipo_quarto")).rlike("familiar|family"), "familiar")
                  .when(lower(col("tipo_quarto")).rlike("estudio|studio|estdio"), "studio")
                  .when(lower(col("tipo_quarto")).rlike("casal com wc privativa"), "casal com wc privativa")
                  .when(lower(col("tipo_quarto")).rlike("triplo standard"), "triplo standard")
                  .when(lower(col("tipo_quarto")).rlike("duplo economico"), "duplo economico")
                  .when(lower(col("tipo_quarto")).rlike("executivo|executive"), "executivo")
                  .when(lower(col("tipo_quarto")).rlike("t1"), "t1")
                  .when(lower(col("tipo_quarto")).rlike("t2"), "t2")
                  .when(lower(col("tipo_quarto")).rlike("t3"), "t3")
                .otherwise(col("tipo_quarto_normalized"))).withColumn("tipo_quarto_normalized", trim(col("tipo_quarto_normalized"

val groupedData = dfCategorized.groupBy("tipo_quarto_normalized").count()
// Substituir valores com count inferior a 20 por "other"
val dfUpdated = dfCategorized.join(groupedData, Seq("tipo_quarto_normalized"))
  .withColumn("tipo_quarto_normalized", when(col("count") < 20, "other").otherwise(col("tipo_quarto_normalized")))
  .drop("count")
```

- Normalização do Tipo de Quarto foi feita em várias etapas. Inicialmente removemos números no início, caracteres especiais e pontuações, e convertemos tudo para letras minúsculas. Em seguida, espaços em branco extras no início e no final das strings são removidos, caracteres especiais e pontuações são eliminados, deixamos apenas letras e números. Valores correspondentes nas strings são substituídos usando expressões regulares. Uma nova coluna é adicionada ao DataFrame, contendo as strings normalizadas.Após agrupar e contar os diferentes tipos de quartos, optamos por todos os valores inferior a 20 são substituídas por ”other”. No final desse processo, observamos que inicialmente tínhamos 344 valores diferentes, porém, após a normalização e agrupamento, chegamos a um

resultado com 82 categorias distintas. Esta normalização está exemplificada na imagem 6.1.

7 Integração dos dados

Com o objetivo de obter um único dataset final, foi seguido um processo de integração de todos os *datasets*. Destes *datasets*, 4 foram fornecidos pela e-GDS, dados de reservas, de hotéis, de tipologias e facilities. Decidimos descartar o dataset das facilities uma vez que apenas foi dado o nome das facilites por hotel e a grande maioria, se não todos, os nomes são diferentes uns dos outros, impossibilitando a sua compreensão e tratamento. Adicionalmente foram obtidos 3 *datasets* externos de fontes públicas sobre feriados, meteorologia e eventos.

Primeiramente começamos por juntar numa só tabela os dados da e-GDS. Exportamos as tabelas fornecidas em excel para um ficheiro csv e com recurso ao spark lemos estes ficheiros e a partir deles criamos 4 *datasets* em memória. [1] Seguidamente à operação de leitura, foram feitas algumas alterações, sendo estas a renomeação de colunas para remover caracteres especiais e a transformação dessas mesmas colunas no seu tipo de dados correto. De forma similar, este processo de leitura foi aplicado aos *datasets* externos.

Ainda durante a leitura dos dados dos hotéis foi adicionada uma nova coluna (area_localizacao), baseada na localização do hotel, que permite a integração com o dataset da meteorologia. Esta coluna mapeia a localização do hotel para a localização para o qual o dataset de meteorologia têm dados.

A segunda etapa consistiu em juntar estes *datasets* com base nas colunas idênticas [2]. Como no capitulo anterior foi realizada a primeira limpeza do da-

[1] Em anexo "LeituraDatasets.zpln"
[2] Em anexo "DatasetsJoin.zpln"

taset principal "QuartosReservados", onde existiam $25,090$ o número de linhas restantes após essa remoção é de 24.955. De seguida passamos a descrever a junção de dados da e-GDS com o dataset "QuartosReservados":

- "Tipologias": Colunas "QuartosReservados:hotel_ID,room_ID", "Tipologias: hotel_ID,room_ID". Ao fazer Join verificamos que existem 8 reservas no hotel ID 309 com o "room_ID"2802 que não possuem correspondência na tabela "Tipologias". Portanto, o novo número de linhas passou a ser 24.947 após a realização desse Join.

- "Hotel": Colunas "QuartosReservados:hotel_ID", "Hotel:hotel_ID", Com esta junção o número de entradas manteve-se. Existia um hotel com ID 7556 que está presente na tabela de reservas mas não existe na tabela dos hotéis. Como já foi feito um join pelas tipologias este hotel já foi removido.

Desta forma, finalizamos a integração dos dados fornecidos pela e-GDS. Faltando juntar os dados das fontes externas. Abaixo descrevemos a integração com o dataSet principal:

- "Feriados": Fizemos um left join da tabela resultante do passo anterior com a dos feriados onde a data de partida seja maior ou igual à data do feriado e a data de chegada seja menor ou igual à data do feriado. Com esta junção o número de linhas aumentou para $25,213$ uma vez que existem reservas com mais que um feriado entre a data de chegada e de partida. Esta integração permitirá saber quais as reservas agendadas durante feriados.

- "Eventos":Fizemos um left Join com os eventos, para tal a coluna criada na tabela dos hotéis "area_localizacao"foi usada com a coluna "Localização"do dataset dos eventos, comparamos as datas de chegada e partida de cada reserva com as datas de início e fim dos eventos correspondentes. Após a intersecção, o número de linhas aumentou para $25,239$, pois existem

reservas que ocorrem durante um ou mais eventos durante a estadia. Esta integração permitirá saber que eventos existiam na altura da reserva.

- ”meteorologia”: Neste data set também foi feito um left Join com o dataset da meteorologia com base nas colunas area_localizacao da tabela anterior e city da tabela da meteorologia e também com base nas datas de partida e chegada das reservas. Com esta integração o dataset passou a ter $70,958$ entradas, criando para cada reserva o mesmo número de entradas quanto o número de dias. Esta integração adiciona dados meteorológicos às reservas.

Com a integração completa, o próximo passo foi agrupar as linhas conforme a expansão provocada pela integração dos *datasets* externos. O agrupamento desses dados foi feito da seguinte forma: para ”Feriados”, mantivemos apenas o primeiro feriado encontrado em cada reserva (”first(is_holiday)”); para ”Eventos”, contabilizamos o número de eventos por reserva (”count(Event)”); e para ”Meteorologia”, calculamos a média de temperatura e precipitação, além de adicionar duas colunas para a temperatura máxima e mínima dos dias de cada reserva. No fim desse processo de tratamento ficamos com as 24947 linhas, ou seja, o mesmo número com que iniciamos esta integração.

7.1 Tratamentos finais

- Agregação do dataset resultante pelas colunas do dataset das reservas com o seguinte tratamento:
  - Remoção de todas as colunas relativas aos feriados, exceto a coluna is_holiday.
  - Remoção das colunas dos eventos e adição da coluna event_count que retrata a quantidade de eventos durante a reserva.
  - Quanto à meteorologia, acrescentámos colunas relativas à tempe-

ratura mínima da reserva, média e máxima. Para a precipitação também teremos a mínima, média e máxima. É de notar que para algumas datas não temos informação sobre precipitação, portanto algumas reservas não têm essa informação.

8 Análise Exploratória

8.1 Correlação de Variáveis

A análise de correlação desempenha um papel crucial na exploração e compreensão dos relacionamentos entre variáveis em conjuntos de dados. Ela permite identificar padrões, associações e dependências entre as variáveis, fornecendo *insights* sobre como elas se comportam em conjunto. Silva (2013)[2] destaca a relevância da matriz de correlação como uma ferramenta essencial na análise de dados de reservas de hotéis. Ela permite identificar relações fortes, moderadas ou fracas entre as variáveis, contribuindo para a compreensão dos fatores que influenciam as reservas e possibilitando a seleção adequada de variáveis para análises posteriores.

8.1.1 Matriz de Correlação

Nesta etapa, realizamos a construção da matriz de correlação para as variáveis selecionadas, a seguir mostramos um pequeno exemplo como isso foi feito:

```
correlation_matrix = data_subset.corr()
plt.figure(figsize=(20, 15))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.
plt.title('Matriz de Correlação')
plt.show()
```

Essa matriz fornece uma visão geral das correlações existentes, permitindo identificar as relações mais fortes entre as variáveis. A matriz de correlação

foi calculada com base nos coeficientes de correlação, que mede a relação linear entre as variáveis. Na figura 8.1 mostramos a primeira análise antes de qualquer alteração.

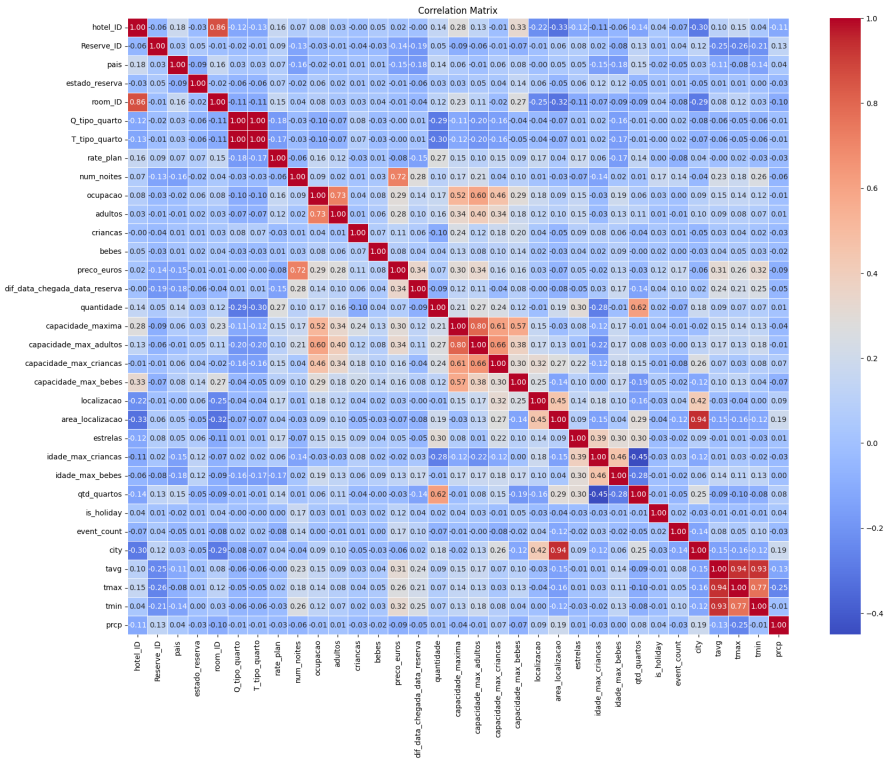


Figura 8.1: Correlação de Variáveis: Primeira análise

8.1.2 Variáveis com uma Correlação Alta

Durante a análise da matriz de correlação, identificamos variáveis com uma correlação significativa. Essas correlações foram identificadas a seguir:

- "T_tipo_quarto" apresentou correlação perfeita com a variável "Q_tipo_quarto" (correlação de 1.0).
- "Area_localizacao" e "city" apresentaram alta correlação com a variável "localizacao" (correlação de 0.94).

- ”temperature_max” e ”temperature_min” apresentaram alta correlação com a variável ”temperature_avg” (correlação de 0.93).
- ”capacidade_max_adultos” apresentou uma correlação positiva forte com a variável ”capacidade_maxima” (correlação de 0.80).
- ”ocupacao” apresentou uma correlação positiva significativa com a variável ”adultos” (correlação de 0.73).

Com base na informação acima, decidimos remover essas variáveis do conjunto de dados, a fim de evitar problemas de multicolinearidade, reduzir a complexidade do modelo e evitar redundância.

8.1.3 Variáveis com Baixa Correlação

Durante a análise, também identificamos variáveis com uma baixa correlação entre si. Essas correlações mais fracas indicam uma relação limitada ou inexistente entre as variáveis, no entanto, é fundamental considerar que a baixa correlação não implica necessariamente em falta de importância ou irrelevância das variáveis. A seguir identificamos as variáveis com baixa correlação e que achamos que serão menos relevantes para o estudo:

- ”idade_max_bebe” apresentou uma correlação negativa moderada de -0,28 com a variável ”quantidade”.
- ”idade_max_criancas” apresentou uma correlação negativa moderada de -0,45 com a variável ”quantidade”.

Com base na informação acima, decidimos remover essas variáveis do conjunto de dados

8.1.4 Matriz de Correlação Após tratamento de dados

Após o tratamento e correção dos dados, foi gerada uma nova matriz de correlação para as variáveis selecionadas. Essa matriz reflete as correlações atualizadas

entre as variáveis, levando em consideração as alterações realizadas. A figura 8.2 apresenta a matriz de correlação atualizada:

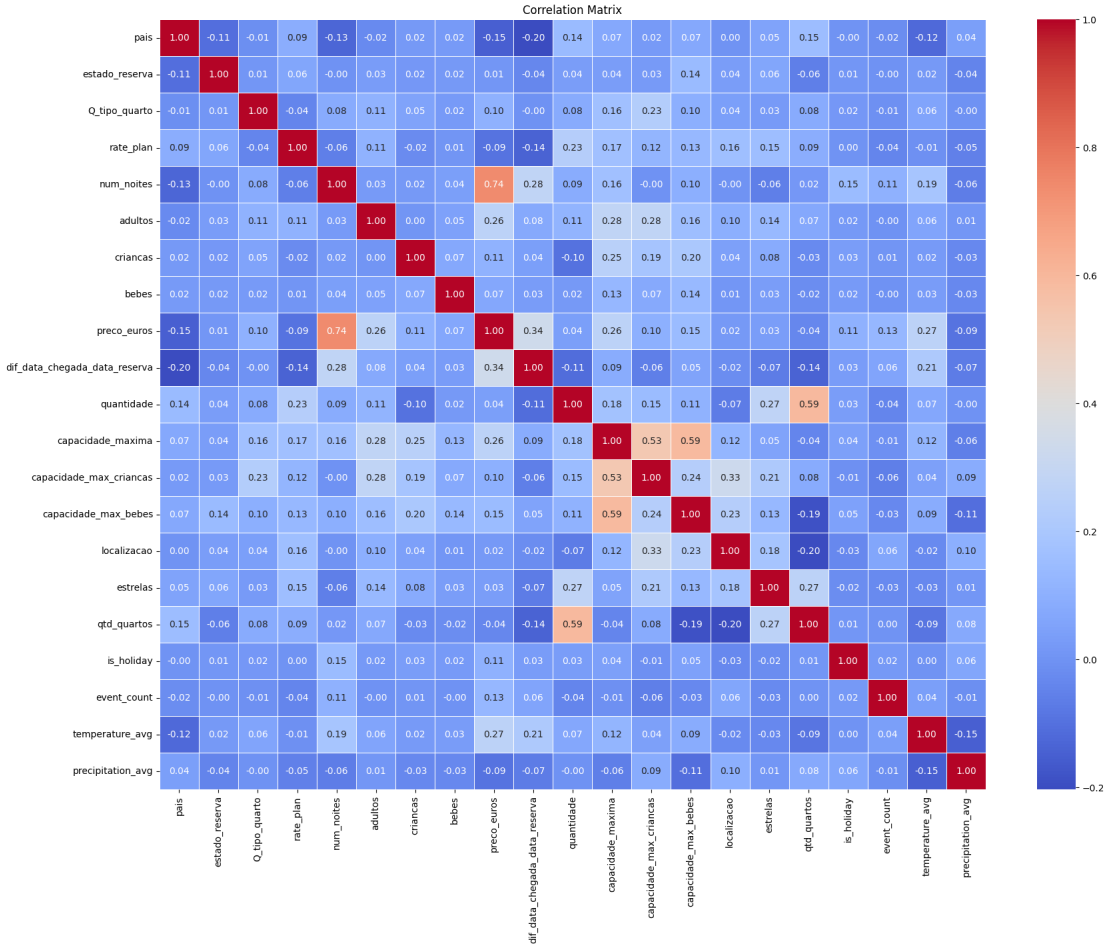


Figura 8.2: Matriz de Correlação após Tratamento de Dados

Através da análise da matriz de correlação, é possível observar as relações existentes entre as variáveis após as alterações realizadas. Após examinar a matriz de correlação atualizada, deixamos algumas correlações elevadas, como a correlação entre o preço e o número de noites. Embora essas variáveis estejam correlacionadas, acreditamos que elas ainda possam desempenhar um papel importante na próxima etapa da análise. Na próxima etapa da análise, iremos explorar ainda mais o impacto e a influência dessas variáveis correlaci-

onadas, procurando uma compreensão mais completa dos fatores que afetam os resultados do estudo.

Em suma, a análise de correlação permitiu-nos identificar as relações mais fortes e significativas entre as variáveis selecionadas. Essas informações foram fundamentais para a seleção adequada das variáveis para a previsão de reservas de hotéis.

8.2 Map Reduce

Neste trabalho, cada membro do grupo ficou responsável por desenvolver uma aplicação em Map Reduce com o objetivo de realizar uma análise relevante no dataset fornecido pela E-GDS.

8.2.1 O que é o Map Reduce?

O Map Reduce é um modelo de programação e técnica de processamento distribuído que permite lidar com grandes volumes de dados de forma eficiente. Este consiste em duas etapas principais: *map* e *reduce*.

8.2.2 Funcionamento

Na etapa de *map*, os dados de entrada são divididos em partes menores e processados independentemente por tarefas de *map*. Cada tarefa aplica uma função de mapeamento aos elementos, resultando em pares chave-valor intermédios.

Na etapa de *reduce*, os pares chave-valor intermédios são agrupados com base na chave e processados por uma função de redução. Isto permite realizar operações como agregação, filtragem ou ordenação dos dados.

8.2.3 Vantagens

O Map Reduce oferece várias vantagens no processamento de grandes volumes de dados:

- **Escalabilidade**: Permite distribuir o processamento em um cluster de computadores, lidando com grandes volumes de dados de maneira eficiente.
- **Tolerância a falhas**: É projetado para lidar com falhas individuais nos nós de processamento, realocando tarefas para outros nós disponíveis, garantindo a confiabilidade do sistema.
- **Flexibilidade**: Pode ser aplicado a diversos problemas de análise de dados, permitindo diferentes tipos de análises, como contagem de palavras ou cálculos estatísticos.

8.2.4 Aplicações e Resultados

Cada membro do grupo desenvolveu uma aplicação em Map Reduce para realizar uma análise específica no dataset fornecido pela E-GDS. Os resultados obtidos através dessas aplicações foram analisados e os insights relevantes foram extraídos. Além disso, gráficos e visualizações foram gerados para facilitar a interpretação dos dados.

Reservas por hotel: Esta análise permitiu saber quais os hotéis com mais e menos reversas 8.3, e efetuar validações como: se o hotel com mais reversas realmente é o hotel com mais ocupação, entre outras.

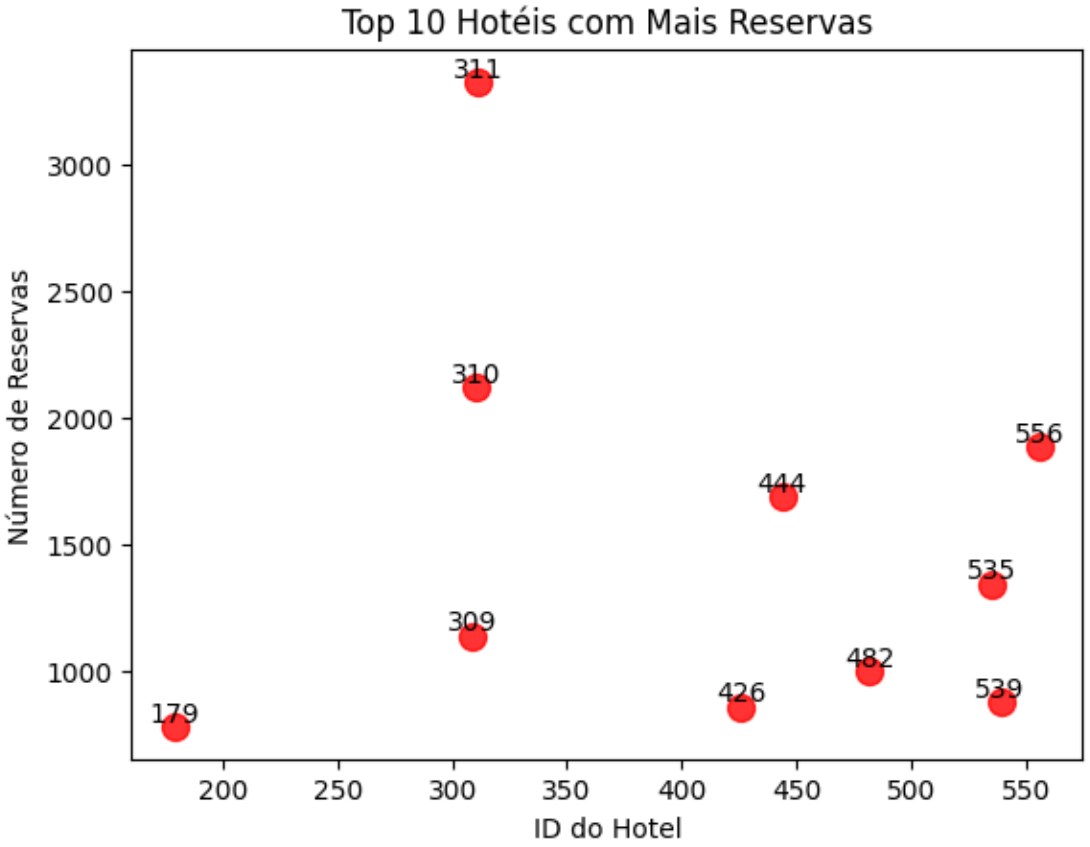


Figura 8.3: Top 10 Hotéis com mais reservas

Média de noites por origem: Permitiu identificar quais as origens dos clientes que mais reservas efetuam 8.4. Bielorússia foi o país que mais se destacou com cerca de 21.5 noites de média.

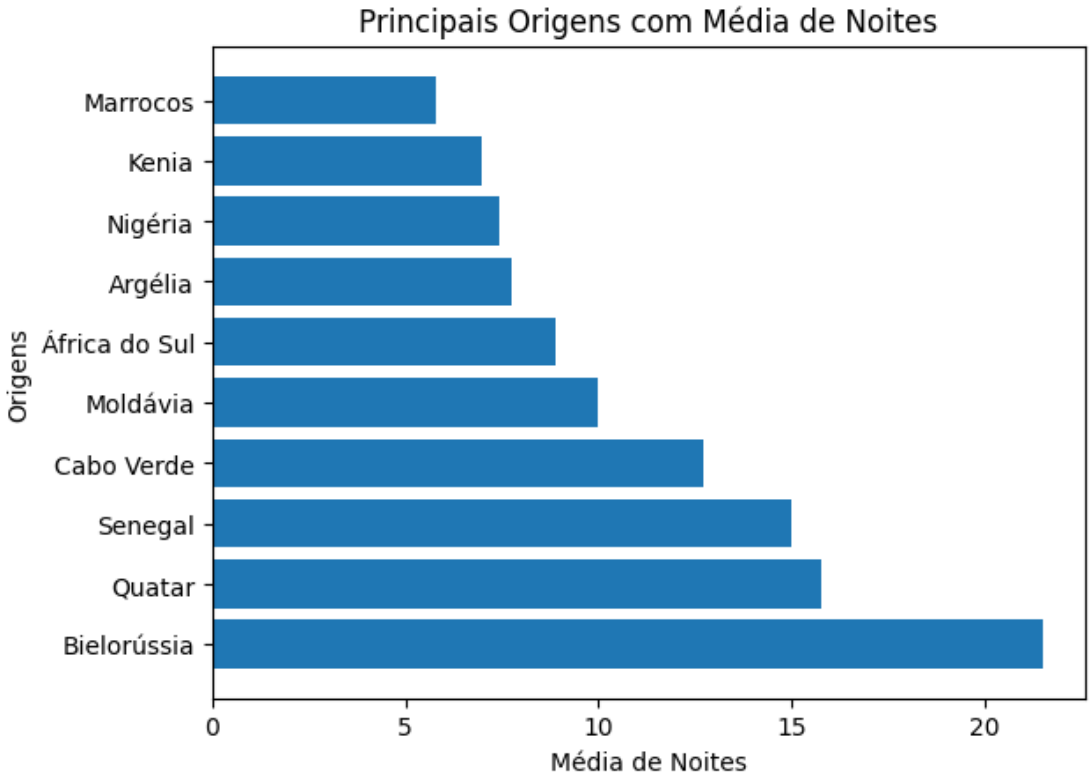


Figura 8.4: Média de noites por origem

Preço médio por noite e por hotel: Análise do preço médio por noite e por hotel. Permite identificar os hotéis com preços mais altos e mais baixos 8.5 e relacionar com o número de reservas para assim verificar qual o preço que os clientes mais optam para efetuar as reservas.

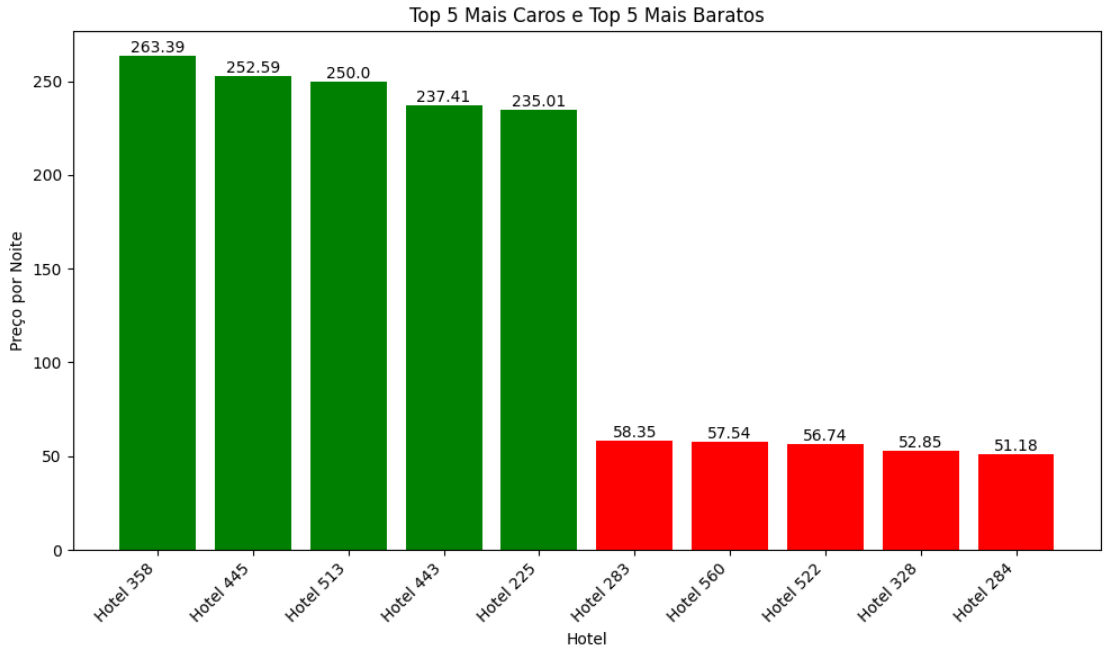


Figura 8.5: Top 5 hoteis mais caros e Top 5 mais baratos

8.2.5 Conclusão

Em resumo, o Map Reduce é uma ferramenta poderosa para processar e analisar grandes volumes de dados de forma distribuída. Com escalabilidade, tolerância a falhas e flexibilidade, tornou-se essencial para lidar com o processamento de big data e obter insights valiosos a partir dos dados. As aplicações desenvolvidas pelo grupo no contexto deste trabalho demonstraram a eficácia do Map Reduce na análise do dataset fornecido pela E-GDS, fornecendo resultados significativos e visualizações claras dos dados.

9 Machine Learning

Neste capítulo, abordaremos técnicas de Machine Learning para prever preços e cancelamentos de reservas em hotéis. Utilizaremos algoritmos de regressão, como *GBTRegressor* e *RandomForestRegressor*, para estimar os preços das reservas. Também usaremos algoritmos de Classificação, e para isso, empregaremos o algoritmo *RandomForestClassifier* para identificar se uma reserva está propensa a ser cancelada ou não. Ao longo do capítulo, exploraremos os diferentes algoritmos de regressão e classificação os impactos das variáveis e a criação de novas colunas relacionadas à temporada (Alta, baixa e Média) e a respetiva estação do ano entre outras explorações.

9.1 Pré-processamento de colunas para testes dos Algoritmos

Durante os testes dos algoritmos de Machine Learning, foram identificadas necessidades de tratamento adicional das variáveis para tentar obter melhores resultados. Para isso, foram realizadas as seguintes etapas:

1. Foi feita a criação de novas colunas com informações sobre a temporada das reservas. A baixa temporada foi definida como os meses de dezembro a março em Portugal, enquanto a alta temporada incluiu os meses de junho, julho e agosto, juntamente com os feriados. A partir dessas condições, uma nova coluna chamada ”temporada”foi adicionada ao DataFrame, atribuindo valores numéricos para representar cada tipo de temporada. A seguir um exemplo:

```
val condBaixaTemporada = month(col("data_chegada")).isin(11, 12, 1,
    2, 3)
val condAltaTemporada = month(col("data_chegada")).isin(6, 7, 8) ||
    condFeriado
val condMediaTemporada = !condBaixaTemporada && !condAltaTemporada
val dfComTemporada = df_nc.withColumn("temporada",
    when(condBaixaTemporada, 0).when(condAltaTemporada,
    2).otherwise(1))
```

2. Uma nova coluna chamada ”cancelamento”foi criada com base na coluna ”estado_reserva”. As reservas canceladas receberam o valor 1, enquanto as não canceladas receberam o valor 0.

3. As colunas de data, ”data_reserva”e ”data_chegada”, passaram por um pré-processamento. Primeiro, elas foram formatadas para o tipo ”Date”. Em seguida, uma nova coluna chamada ”estacao_chegada”foi criada com base no mês de chegada da reserva, atribuindo a estação correspondente. Abaixo um exemplo desse tipo de tratamento:

```
val dfformatDate = df_com_cancelamento.withColumn("data_reserva",
    to_date(col("data_reserva"), "yyyy-MM-dd"))
                            .withColumn("data_chegada",
                                to_date(col("data_chegada"),
                                "yyyy-MM-dd"))

val processedDF = dfformatDate.withColumn("estacao_chegada",
    when(month(col("data_chegada")).isin(12, 1, 2), "Inverno")
    .when(month(col("data_chegada")).isin(3, 4, 5), "Primavera")
    .when(month(col("data_chegada")).isin(6, 7, 8), "Verao")
    .otherwise("Outono"))
```

Essas etapas de tratamento e pré-processamento das variáveis visam melhorar a qualidade dos dados e preparar o *DataFrame* para a aplicação dos algoritmos

de Machine Learning na previsão de preços e cancelamentos de reservas em hotéis.

9.2 Previsão de Preços de Reservas

Para a previsão dos preços de reservas, foram utilizados os algoritmos de regressão RandomForestRegressor e GBTRegressor. Esses algoritmos foram escolhidos devido ao seu desempenho e capacidade de lidar com conjuntos de dados complexos.

O RandomForestRegressor é um algoritmo baseado em árvores de decisão, que combina várias árvores para obter uma previsão final. Ele é capaz de lidar com características não lineares e interações entre variáveis, o que o torna adequado para problemas de previsão de preços. [9]

O *GBTRegressor*, por sua vez, é um algoritmo de *boosting* que também utiliza árvores de decisão. Ele treina iterativamente várias árvores, onde cada nova árvore é treinada para corrigir os erros das árvores anteriores. O *GBTRegressor* é capaz de capturar relacionamentos não lineares e interações complexas entre as variáveis, permitindo uma previsão mais precisa dos preços das reservas.

Foi utilizado o feature ranking do RandomForestRegressor e do GBTRegressor, assim foi possível identificar as variáveis mais relevantes na previsão dos preços das reservas. Essas informações são muito importantes para compreender quais as variaveis que têm maior impacto na privisão dos preços. [8]

9.2.1 RandomForestRegressor

Para a previsão de preços, foi desenvolvida uma função denominada trainAndEvaluateRFR usando o algoritmo RandomForestRegressor. Essa função permite treinar e avaliar um modelo de regressão utilizando o algoritmo Random Forest. A função trainAndEvaluateRFR realiza as seguintes etapas:

1. Define um pipeline que consiste no VectorAssembler e no algoritmo RandomForestRegressor:

```
val pipeline = new Pipeline()
  .setStages(Array(assembler, algorithm))
```

2. O dataset df é dividido em dois conjuntos: trainData e testData. A proporção utilizada é de 70% para treino e 30% para teste. A seed é definida como 42 para garantir que a divisão seja sempre a mesma e os resultados sejam reproduzíveis:

```
val Array(trainData, testData) = df.randomSplit(Array(0.7, 0.3),
    seed = 42)
```

3. O modelo é treinado utilizando o método *fit* do pipeline:

```
val model = pipeline.fit(trainData)
```

4. O modelo treinado é utilizado para fazer previsões nos dados de teste (testData). O método transform do modelo aplica as transformações do pipeline nos dados de teste e gera as previsões:

```
val predictions = model.transform(testData)
```

5. O desempenho do modelo é avaliado usando uma métrica de avaliação de regressão, neste caso, o *Root Mean Squared Error (RMSE)*. O *RegressionEvaluator* é criado e configurado com a coluna de rótulo *(labelCol)* e a coluna de previsão (”prediction”). O método *evaluate* é chamado com as previsões para calcular o *RMSE*.

```
val evaluator = new RegressionEvaluator()
  .setLabelCol(labelCol)
  .setPredictionCol("prediction")
  .setMetricName("rmse")

val rmse = evaluator.evaluate(predictions)
```

```
7
8 println("Root␣Mean␣Squared␣Error␣(RMSE):␣" + rmse)
```

6. Por fim, algumas previsões são selecionadas do DataFrame para ter noção das previsões que estão a ser feitas:

```
1   predictions.select("room_ID", labelCol, "prediction").show()
```

9.2.2 GBTRegressor

Para verificar e comparar os resultados, também foi utilizado o algoritmo GBTRegressor. A função utilizada é a mesma descrita anteriormente para o RandomForestRegressor, sendo necessário apenas passar o novo algoritmo como parâmetro. As etapas e o código apresentados na seção anterior são aplicáveis aqui, com a substituição do algoritmo RandomForestRegressor pelo algoritmo GBTRegressor.

Essa abordagem permite comparar o desempenho dos dois algoritmos e selecionar o mais adequado para a previsão de preços.

9.2.3 Função de Classificação de Atributos (Feature Ranking)

Como forma de lidar com um grande número de variáveis e visando uma melhor seleção delas, foi desenvolvida a função "featureRanking". Essa função visa classificar as variáveis nos dois algoritmos:

A função "featureRanking"desempenha um papel importante ao fornecer um ranking de importância relativa de cada variável. Essa medida é calculada com base nas características fornecidas pelos modelos RandomForestRegressor e GBTRegressor.

A seguir está o exemplo do código da função "featureRanking"para o algoritmo RandomForestRegressionModel:

```
1 def featureRanking(algorithm: RandomForestRegressionModel, featureCols:
      Array[String]): Unit = {
```

```
2   val featureImportances = algorithm.featureImportances.toArray
3   val rankedFeatures = featureCols.zip(featureImportances).sortBy(-_._2)
4
5   println("Feature Ranking:")
6   rankedFeatures.zipWithIndex.foreach { case ((feature, importance),
        rank) =>
7     println(s"Feature ${rank+1}: $feature - Importance $importance")
8   }
9 }
```

Um exemplo da importância de cada variável para o algoritmo é apresentado abaixo:

- Feature 1: room_ID (Importância: 0.2386)
- Feature 2: qtd_quartos (Importância: 0.1976)
- Feature 3: hotel_ID (Importância: 0.1535)
- Feature 4: temperature_avg (Importância: 0.1117)
- Feature 5: adultos (Importância: 0.0769)
- Feature 6: estrelas (Importância: 0.0706)
- Feature 7: num_noites (Importância: 0.0563)
- Feature 8: capacidade_maxima (Importância: 0.0385)
- Feature 9: criancas (Importância: 0.0267)
- Feature 10: event_count (Importância: 0.0118)
- Feature 11: temporada (Importância: 0.0116)
- Feature 12: is_holiday (Importância: 0.0061)

9.2.4 Resultados na Previsão de Preços

Para avaliar os resultados dos dois algoritmos, RandomForestRegressor e GBTRegressor, foram considerados os valores preço por noite por room_id.

O resultados dos Algoritmo com base no Root Mean Squared Error (RMSE):

- GBTRegressor: 17.54049738716371
- RandomForestRegressor: 21.68354392108588

Nas tabelas 9.1 e 9.2 está uma amostra dos valores reais e das previsões geradas pelo modelo

Tabela 9.1: Tabela de Resultados GBTRegressor

room_ID	**preco_noite_adulto**	**prediction**
81	28.5	31.29484738379246
81	29.5	29.08658871842678
81	31.25	28.345357587874094
81	28.5	25.498833181179428
81	27.0	26.84495168929202
85	62.5	67.9726062040213
85	67.0	74.11553050981094
190	57.0	67.6776023950932
190	45.5	53.25816632902684
85	72.5	67.9726062040213
85	107.0	84.1241782911615
190	100.0	122.15293974790283

Na análise dos resultados, é possível observar que as previsões do modelo podem variar em relação aos valores reais. O RMSE indica a diferença média entre as previsões e os valores reais, sendo desejável ter um valor de RMSE menor, o que indica uma melhor capacidade de previsão do modelo. Estes resultados

Tabela 9.2: Tabela de Resultados RandomForestRegressor

room_ID	**preco_noite_adulto**	**prediction**
81	28.5	58.546431042678556
81	29.5	52.58288062454881
81	31.25	58.546431042678556
81	28.5	43.03895199281975
81	44.0	82.93799806188757
85	62.5	62.55340415968901
85	67.0	58.693005417616746
190	57.0	60.23613863543294
190	45.5	51.436251808719476
85	72.5	60.636429933126806
85	107.0	93.97093037857026
190	100.0	97.01677646841392

foram os melhores durante alguns testes de diferentes variáveis apesar do uso da função “featureRanking” As variaveis usadas pera estes resultados foram as seguintes:

```
val featureCols = Array("hotel_ID","room_ID",
    "adultos", "estrelas", "qtd_quartos",
     "temperature_avg","temporada")
val labelCol = "preco_noite_adulto"
```

Comparando os resultados, podemos concluir que o modelo GBTRegressor apresentou um desempenho superior em relação ao modelo RandomForestRegressor. O RMSE mais baixo do GBTRegressor indica que suas previsões tiveram uma menor diferença média em relação aos valores reais. Isso sugere que o GBTRegressor foi capaz de capturar melhor os padrões e as relações nos dados, resultando em previsões mais precisas.

9.3 Previsão de Cancelamentos e reservas

Neste capítulo, vamos tratar da previsão de cancelamentos de reservas. Utilizamos o algoritmo RandomForestClassifier, um método de classificação, para identificar se uma reserva tem maior probabilidade de ser cancelada. Escolhemos esse algoritmo por já termos obtido bons resultados ao utilizá-lo em outra situação parecida.

Na etapa inicial antes de realizar a previsão de cancelamentos, realizamos uma avaliação mensal do número geral de cancelamentos para os anos de 2022 e 2023. No entanto, é importante ressaltar que o ano de 2023 ficam a faltar muitas reservas em virtude de só termos recebido o dataset em março de 2023, tornando-o menos relevante para a análise comparativa.

Ao examinar os resultados dessa análise, observamos que o mês com o maior número de cancelamentos, conforme evidenciado na imagem 9.1, é o mês 7 de 2022. Por outro lado, o mês com o menor número de cancelamentos é o mês 12, provavelmente por ser o mês com menor afluência de reservas.

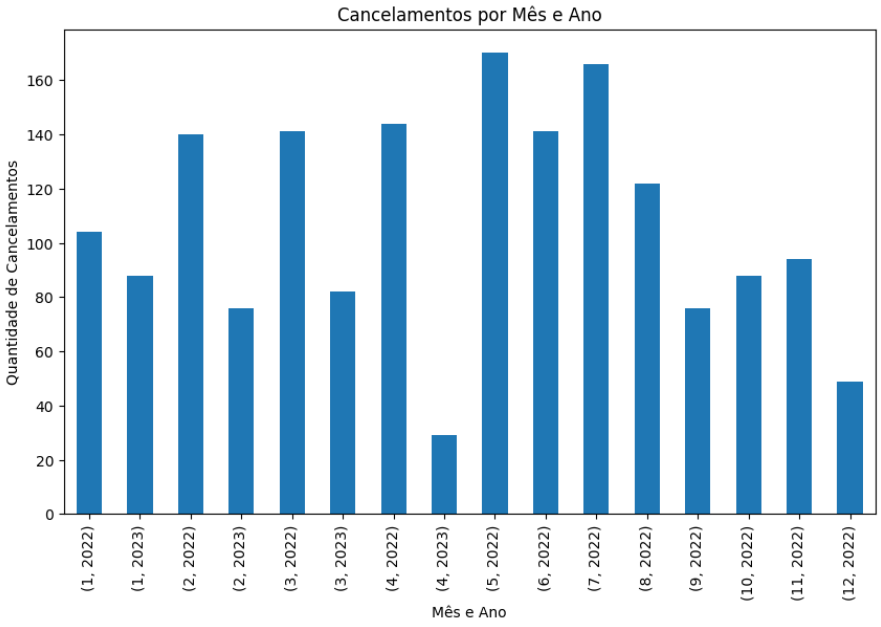


Figura 9.1: Cancelamentos Geral de Reservas

Passamos agora a um resumo do algoritmo de classificação e sua implementação como primeiro passo:

Os dados são divididos em conjuntos de treino e teste usando o método randomSplit. Nesse caso, 80% dos dados são destinados ao treino e 20% para teste.

```
val Array(trainData, testData) = assembledDF.randomSplit(Array(0.8,
    0.2), seed = 42)
```

É criado um objeto RandomForestClassifier, que será utilizado como modelo de classificação.

```
val classifier = new RandomForestClassifier() .setLabelCol(labelCol)
    .setFeaturesCol("features") .setMaxBins(5000)
```

As previsões são feitas utilizando o modelo treinado e o conjunto de teste (testData).

```
val model = classifier.fit(trainData)
val predictions = model.transform(testData)
```

O desempenho do modelo é avaliado utilizando métricas específicas. Neste caso, é utilizado o BinaryClassificationEvaluator para calcular a *accuracy* (*accuracy*) e o MulticlassClassificationEvaluator para calcular o F1-Score.

```
val evaluator = new BinaryClassificationEvaluator()
    .setLabelCol(labelCol) .setMetricName("areaUnderROC") val
    \textit{accuracy}  = evaluator.evaluate(predictions)
    println("\textit{accuracy} : " + \textit{accuracy} ) val
    evaluatorMulti = new MulticlassClassificationEvaluator()
    .setLabelCol(labelCol) .setPredictionCol("prediction")
    .setMetricName("f1") val f1Score =
    evaluatorMulti.evaluate(predictions) println("F1-Score: " + f1Score)
```

É gerada uma confusion matrix a partir das previsões realizadas.

```
val confusionMatrix = predictions .groupBy(labelCol, "prediction")
    .count() .orderBy(labelCol, "prediction") confusionMatrix.show()
```

Após a criação do algoritmo, o próximo passo foi testar as variáveis. Iniciamos os testes com as seguintes variáveis:

```
val featureCols = Array("hotel_ID",
"adultos", "capacidade_maxima", "qtd_quartos",
"is_holiday","pais", "preco_noite","temporada" "Q_tipo_quarto",
    "localizacao"
)
val labelCol = "cancelamento"
```

Durante a execução inicial do algoritmo, obtivemos uma *accuracy* de 0,51, o que foi considerado relativamente baixo. Diante disso, decidimos realizar testes com diferentes variáveis para identificar aquelas que têm maior influência nos cancelamentos de reservas. A seguir, apresentamos os resultados desses testes:

- Adicionada variável "estrelas": A *accuracy* foi de 0,5886.
- Adicionada variável "dif_data_chegada_data_reserva": A *accuracy* foi de 0,6153.
- Adicionada variável "rate_plan": A *accuracy* foi de 0,606, mas essa variável foi removida posteriormente por ter um efeito negativo no desempenho do modelo.
- Adicionada variável "num_noites": A *accuracy* foi de 0,6115, mas essa variável foi removida posteriormente.
- Adicionada variável "is_holiday": A *accuracy* foi de 0,6099, mas essa variável foi removida posteriormente.
- Adicionada variável "crianças": A *accuracy* foi de 0,6146, mas essa variável foi removida posteriormente.
- Removemos a variável "temporada": A *accuracy* foi de 0,6261
- Removemos a variável "pais": A *accuracy* foi de 0,62477 e essa variável foi mantida.

- Removemos a variável "preço_noite": A *accuracy* foi de 0,6358.
- Removemos a variável "localizacao": A *accuracy* foi de 0,61
- Removemos a variável "Q_tipo_quarto": A *accuracy* foi de 0,62771.
- Adicionada variável "eventos": A *accuracy* foi de 0,62825, mas essa variável foi removida posteriormente.
- Adicionada variável "quantidade": A *accuracy* foi de 0,63404, mas essa variável foi removida posteriormente.
- Adicionada variável "room_ID": A *accuracy* foi de 0,636
- Adicionada variável "estacao_chegada": A *accuracy* foi de 0,6186, mas essa variável foi removida posteriormente.

Após os testes das variáveis a melhor *accuracy* que conseguimos foi de 0.636. O próximo passo foi testar com diferentes valores de treino e teste, no entanto, os melhores resultados foram de 0,8 para treino e 0,2 para testes.

Para finalizar este processo, o algoritmo foi ajustado para realizar previsões de reservas e cancelamentos com base no mês e no ID do hotel. A seguir na tabela 9.3, apresentamos exemplos dos resultados obtidos, incluindo a *accuracy* , o número de cancelamentos e o número de reservas:

Tabela 9.3: Tabela de Resultados

Hotel	**Mês**	***accuracy***	**Cancelamentos**	**Reservas**
444	1	0.5348	3	115
444	7	0.6051	8	132
309	7	0.6690	7	60

Concluímos que os resultados obtidos até o momento mostram que a *accuracy* do modelo de previsão de reservas e cancelamentos não é muito alta. No entanto, é importante ressaltar que esses resultados foram alcançados com um

conjunto de dados limitado, abrangendo apenas um ano completo. Para melhorar a precisão do modelo, seria necessário ter um conjunto de dados mais amplo e abrangente, com informações de múltiplos anos. Isso permitiria ao algoritmo melhorar os padrões e tendências dos dados, resultando em previsões mais precisas. Portanto, apesar dos resultados atuais, há potencial para aprimorar o modelo com a inclusão de mais dados.

10 Conclusões e Trabalho Futuro

Durante o desenvolvimento deste trabalho, realizamos uma análise abrangente de dados de reservas de hotéis, com o objetivo de prever preços, cancelamentos e reservas futuras. Ao longo do processo, encontramos desafios e limitações que impactaram a qualidade do trabalho realizado, mas também identificamos áreas com potencial para trabalhos futuros.

Uma das dificuldades encontradas foi a procura por dados externos adicionais, como feriados, meteorologia e eventos, tendo sido uma tarefa difícil. Essas fontes de dados extras permitiram ter dados mais completos para atingir o objetivo. No entanto, encontrar e integrar esses dados foi um desafio, pois nem sempre estavam prontamente disponíveis ou eram de fácil acesso, nomeadamente os eventos. Além disso, a mudança frequente do conjunto de dados ao longo do projeto também trouxe desafios adicionais para a análise e integração dos dados.

A análise extensiva de todos os conjuntos de dados também foi um processo desafiante. Lidamos com a necessidade de corrigir dados incompletos, limpar strings e normalizar os dados. Além disso, tivemos de lidar com a presença de dados duplicados e a necessidade de adicionar novas colunas para enriquecer as informações disponíveis. A verificação de outliers também foi realizada para garantir a qualidade dos dados.

A integração dos diferentes conjuntos de dados e a criação de uma matriz de correlação foram etapas importantes para entender as relações entre as variáveis. A visualização gráfica desempenhou um papel crucial na exploração

e compreensão dos dados, permitindo identificar tendências e padrões relevantes.

No contexto de Machine Learning, aplicamos modelos como RandomForestRegressor e GBTRegressor para prever preços de reservas. Também utilizamos uma função de classificação de atributos para prever cancelamentos e reservas. No entanto, os resultados obtidos foram limitados devido à natureza restrita do conjunto de dados, que abrangia apenas um ano completo. Além disso, encontramos alguns desafios relacionados à qualidade dos dados disponíveis no conjunto de dados.

No que diz respeito a trabalhos futuros, a melhoria da qualidade dos dados é fundamental. A possibilidade de obter conjuntos de dados mais completos e confiáveis pode levar a resultados mais precisos e significativos. Além disso, a exploração de técnicas de Machine Learning e a incorporação de outras variáveis relevantes, como comentários de clientes e outros dados que permitam perceber o tipo de cliente, podem enriquecer ainda mais os modelos de previsão.

Em resumo, embora tenhamos enfrentado desafios ao longo deste trabalho, como a dificuldade em encontrar fontes de dados externos adicionais e a necessidade de correção e integração dos conjuntos de dados, conseguimos desenvolver um primeira versão de um sistema de previsão de preços, de cancelamentos e reservas de hotéis. No entanto, devido a restrições de tempo, não foi possível explorar plenamente o potencial dos dados disponíveis para técnicas de Machine Learning mais precisas. Acreditamos que uma análise mais aprofundada e uma maior diversidade de dados poderiam levar a resultados mais robustos e relevantes. Portanto, há oportunidades para realizar trabalhos futuros que explorem melhor as aprendizagem dos algoritmos e aproveitem conjuntos de dados mais abrangentes e precisos. Isso permitiria uma compreensão mais aprofundada dos padrões e tendências no setor hoteleiro, fornecendo perceções valiosos para tomadas de decisão estratégicas.

Bibliografia

[1] Chat.openai.com. https://chat.openai.com/, 2023.

[2] Janice Da Silva Jesus. A qualidade na prestação de serviços hoteleiros: o impacto da satisfação na fidelização de clientes. 2013.

[3] Huseyin Kilic and Fevzi Okumus. The effect of service quality on customer loyalty within the context of ski resorts. *Journal of Hospitality & Leisure Marketing*, 12(3):23–43, 2005.

[4] Sheryl E Kimes, Jochen Wirtz, and Betsy M Noone. Restaurants and hotels: Strategies for the hospitality industry. *Cornell Hotel and Restaurant Administration Quarterly*, 39(3):60–68, 1998.

[5] Jae-Hyeon Lee and Byong-Hun Jeon. Determinants of customer satisfaction and loyalty in the korean hotel industry. *Journal of Hospitality & Tourism Research*, 36(3):389–417, 2012.

[6] Meteostat.net. https://meteostat.net/en/, 2023.

[7] Sapo.pt. https://services.sapo.pt/Holiday/GetNationalHolidays?year=2023, 2023.

[8] Spark 3.0.0 ScalaDoc. GBTRegressor, 2023.

[9] scikit learn. RandomForestRegressor, 2023.

[10] Fang Xu and Qiang Ye. Effects of online reviews on hotel booking intention: The moderating role of hotel price. *Journal of Hospitality & Tourism Research*, 39(1):3–23, 2015.

[11] Xinyuan Zhang, Lijuan Huang, and Xin Zhao. The impact of online reviews on hotel booking intentions and perception of trust. *Tourism Management*, 67:287–308, 2018.