Análise de visitas à hotel: Abordagem usando ciência de dados para prever e aumentar as visitas de hóspedes

[1]Wellington C. Santos, [1]Davi V. L. Correia

[1]Instituto de Computação - Universidade Federal de Alagoas (UFAL)

Wcs@ic.ufal.br, dvlc@ic.ufal.br

Abstract. *This article was developed as part of the evaluation of the data science discipline, of the Computer Science course, taught by professor Bruno Almeida Pimentel. The main objective of this article is to predict, analyze and adopt measures to increase guest visits to a hotel, thus increasing its revenue, and to use the techniques developed in the classroom, in data science area.*

Resumo. *Este artigo foi desenvolvido como parte da avaliação da disciplina Ciência de Dados, do curso de Ciência da Computação, ministrada pelo professor Bruno Almeida Pimentel. O objetivo principal deste artigo é prever, analisar e adotar medidas para aumentar as visitas de hóspedes à um hotel, aumentando assim a sua receita, utilizando as técnicas desenvolvidas em sala de aula, na área de ciência de dados.*

1. Motivação

Em primeira instância, partimos do ponto de que, como uma área relativamente nova, a ciência de dados possui ferramentas poderosas que podem contribuir e impactar nos mais variados setores e áreas, e a hotelaria é uma delas, pois, poder usar probabilidade e os modelos preditivos de machine learning para prever se um cliente vai ou não cancelar a sua reserva, ou se ao aplicar determinada estratégia, aumentarão os números de clientes em épocas de baixa temporada, são apenas alguns dos benefícios que a ciência de dados pode proporcionar à hotelaria. O nosso principal objetivo é apresentar ideias ao gerente do hotel, para que o lucro seja maximizado.

2. Obtenção dos dados

Para poder trabalhar nesse sofisticado problema, usamos a base de dados “Hotel booking demand”, disponibilizada no site Kaggle, esse conjunto de dados possuí informações sobre a reserva e estadia de clientes em hotéis e resorts de Portugal, entre 2015 e meados de 2017.

3. Base de dados

```
print(Hotel_df.info(), Hotel_df.describe())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   hotel                           119390 non-null  object
 1   is_canceled                     119390 non-null  int64
 2   lead_time                       119390 non-null  int64
 3   arrival_date_year               119390 non-null  int64
 4   arrival_date_month              119390 non-null  object
 5   arrival_date_week_number        119390 non-null  int64
 6   arrival_date_day_of_month       119390 non-null  int64
 7   stays_in_weekend_nights         119390 non-null  int64
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
```

Após uma primeira visualização das colunas, tipo dos dados e ver uma descrição geral do conjunto de dados, fizemos um pré-processamento nos mesmos

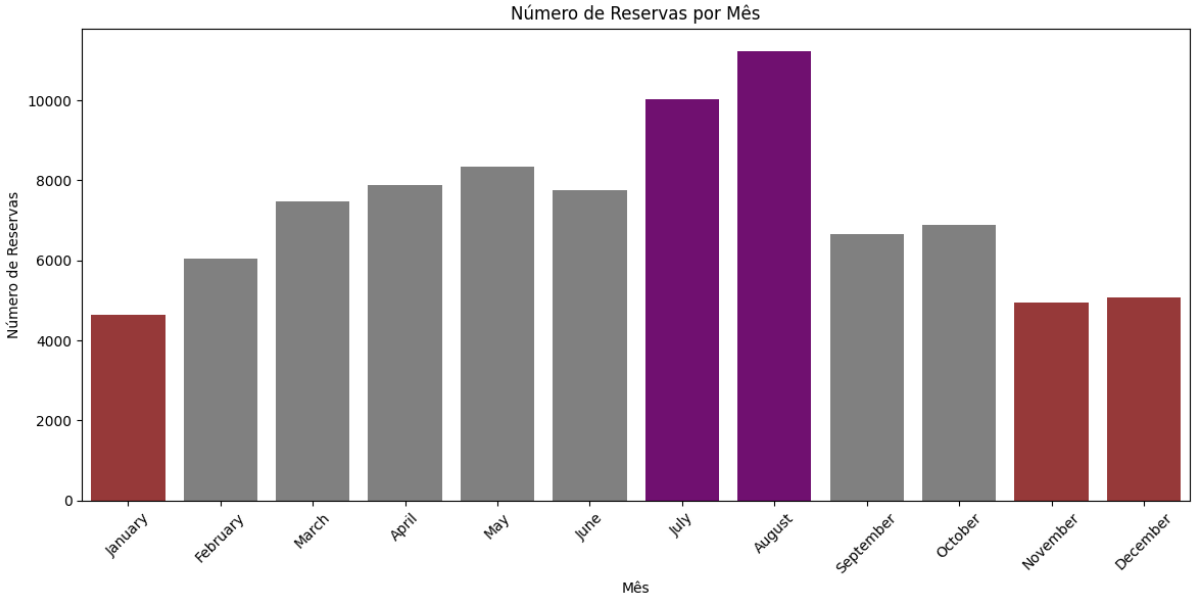
```
# verificando a quantidade de valores null
print(Hotel_df.isnull().sum())

# precisamos entender e nos perguntar sobre o que fazer com os valores null
# se os removemos ou os mantemos, se esses valores existerem no dataset é
# algo normal, se sim, avaliaremos se será melhor para a análise manter ou
# remover esses dados

Hotel_df.dropna(subset=['country'], inplace=True)

Hotel_df.fillna(0, inplace = True)

# Removendo valores duplicados, clientes repetidos em datas diferentes continuarão
Hotel_df = Hotel_df.drop_duplicates()
```

Após verificar a existência de valores nulos nos dados, resolvemos remover os dados com colunas críticas nula e então, para o restante dos dados com valores nulos, os substituímos por valor numérico, também removemos os dados duplicados, após ponderar se os mesmos seriam normais ou uma anomalia nos dados e concordamos sobre ser uma anomalia, pois se um mesmo cliente volta ao hotel, pelo menos a data será diferente, pensamos na possibilidade de uma pessoa ter reserva em mais de um hotel na mesma data e, nesse caso em um deles haveria o status de cancelado, ou pessoas com os mesmos dados e na mesma data em um determinado hotel, contudo, consideramos isso um outlier, pela baixa probabilidade.

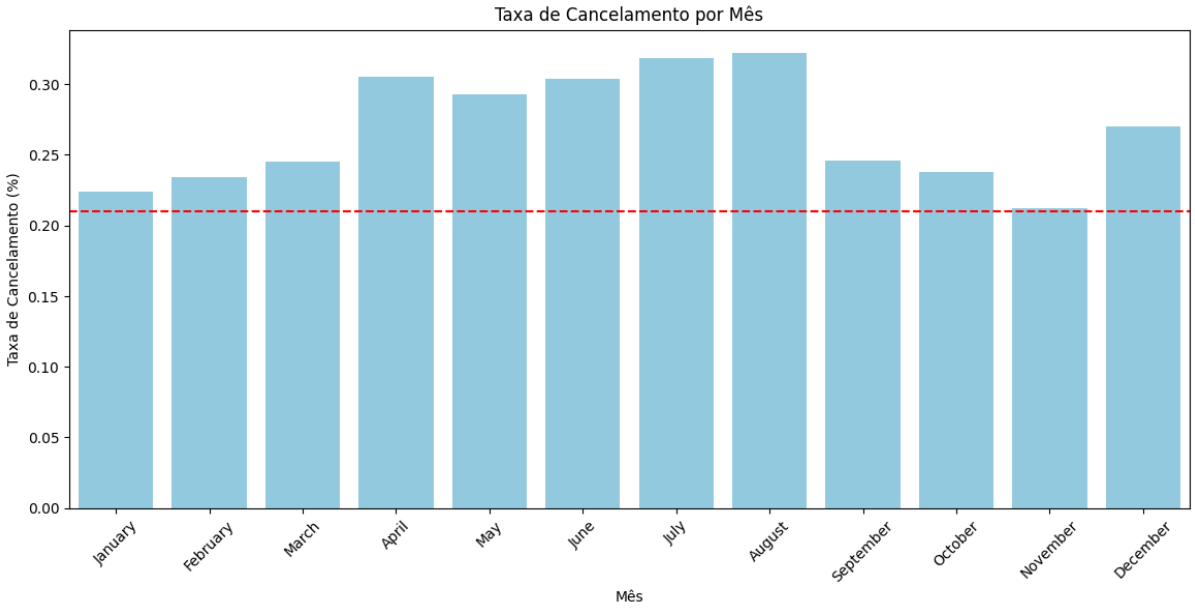
```
# encoding categorical variables

cat_df['hotel'] = cat_df['hotel'].map({'Resort Hotel' : 0, 'City Hotel' : 1})

cat_df['meal'] = cat_df['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC': 3, 'Undefined': 4})

cat_df['market_segment'] = cat_df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/TO': 3,
                                                         'Complementary': 4, 'Groups': 5, 'Undefined': 6, 'Aviation': 7})

cat_df['distribution_channel'] = cat_df['distribution_channel'].map({'Direct': 0, 'Corporate': 1, 'TA/TO': 2, 'Undefined': 3,
                                                                     'GDS': 4})

cat_df['reserved_room_type'] = cat_df['reserved_room_type'].map({'C': 0, 'A': 1, 'D': 2, 'E': 3, 'G': 4, 'F': 5, 'H': 6,
                                                                 'L': 7, 'B': 8})

cat_df['deposit_type'] = cat_df['deposit_type'].map({'No Deposit': 0, 'Refundable': 1, 'Non Refund': 3})

cat_df['customer_type'] = cat_df['customer_type'].map({'Transient': 0, 'Contract': 1, 'Transient-Party': 2, 'Group': 3})

cat_df['year'] = cat_df['year'].map({2015: 0, 2014: 1, 2016: 2, 2017: 3})
```

Também fizemos *encoding label* nas variáveis categóricas

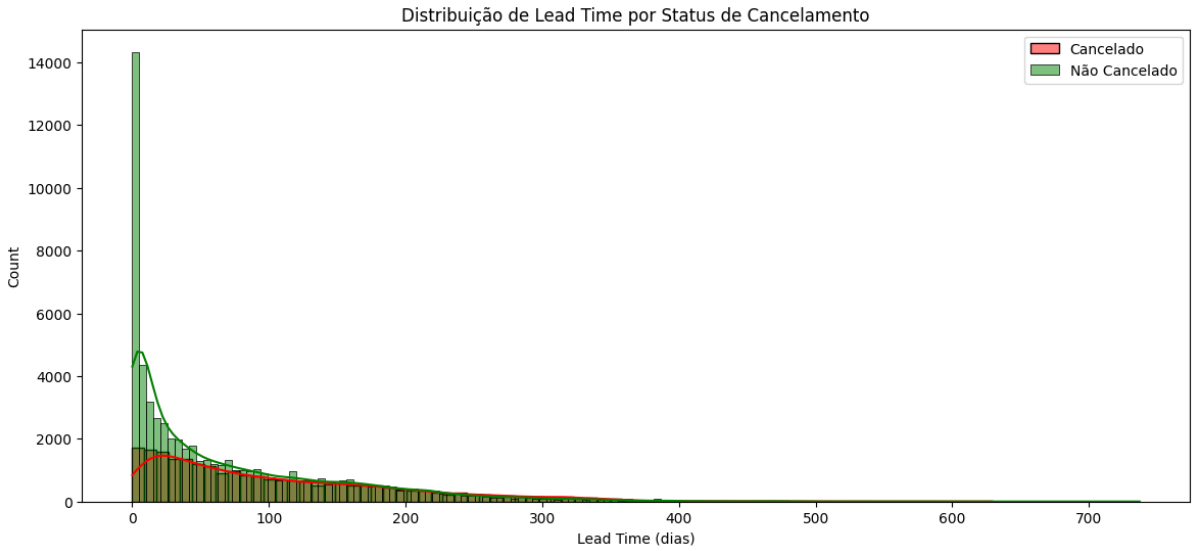
4. Estatística descritiva e inferência

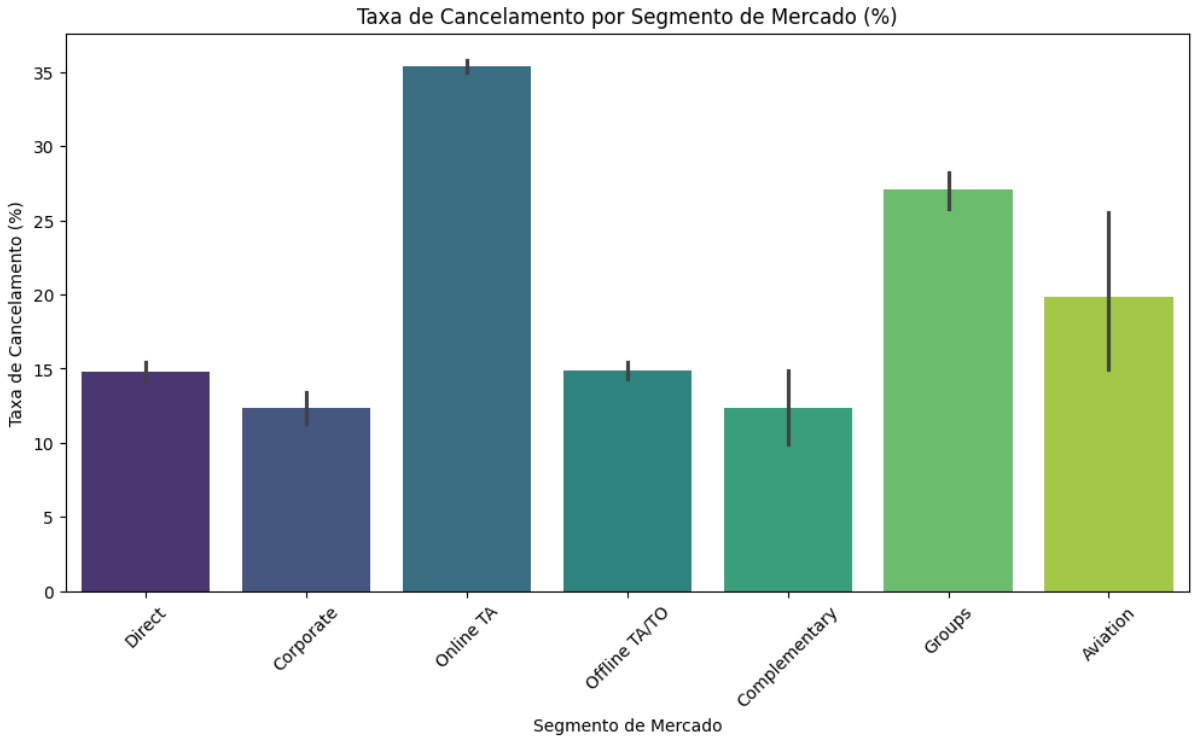


Para uma primeira análise, decidimos procurar por sazonalidades no conjunto de dados, buscamos, então, observar a relação do número de reservas ao longo ano, e percebemos que há um pico de visitas nos meses de julho e agosto e um número reduzido nos meses de janeiro, novembro e dezembro.



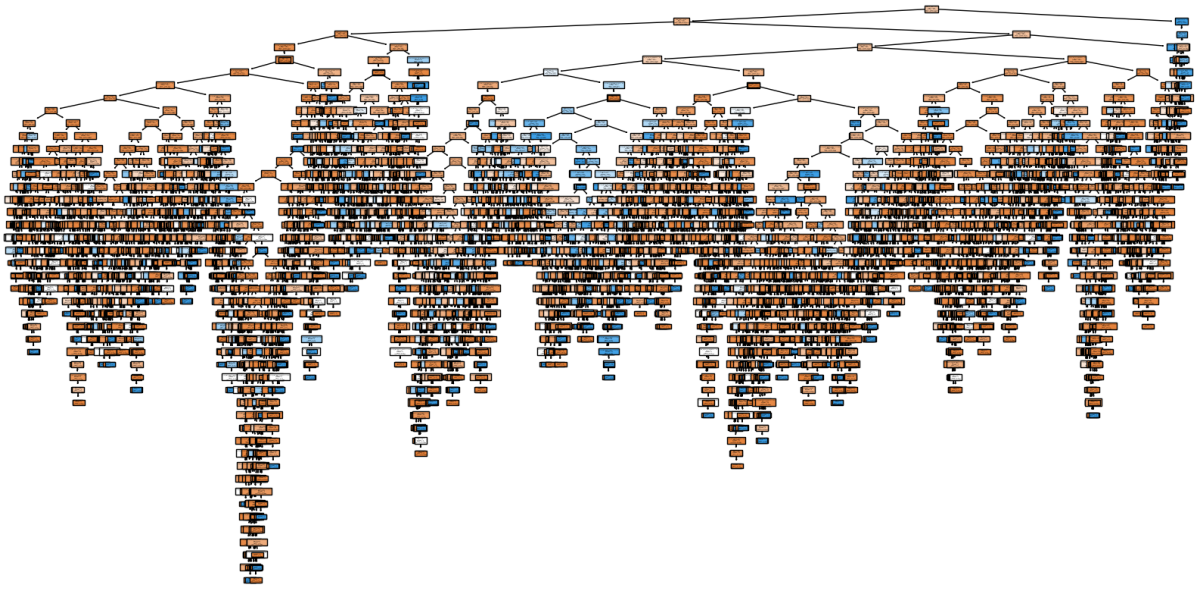
Achamos importante verificar a relação de cancelamento durante os meses do ano, buscando sazonalidades e uma relação com os meses mais e menos visitados, para a nossa surpresa, a taxa de cancelamento mensal é alta em todos os meses, mas ainda mais alta em alta temporada, e os meses em que os clientes menos desistem das reservas, janeiro e novembro, meses em que há menos reservas, chegamos à conclusão que muitos clientes optam por reservar com antecedência os quartos, para garantir a vaga e que, por imprevistos ao longo do ano, as pessoas precisam cancelar a reserva.





Em seguida, analisamos também a relação entre as datas em que as reservas foram feitas e o *status* de cancelamento ou não dessa reserva ao longo do tempo. Juntamente com a taxa de cancelamento de reserva de acordo com o meio pelo qual a reserva foi feita, isso nos ajudou a entender os fatores que contribuem com os cancelamentos e em quais meios de marketing será importante investir.

5. Métodos avaliados



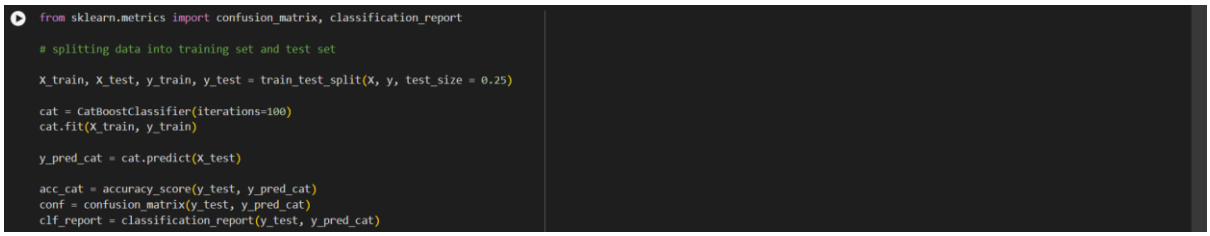
dada a natureza do problema, em se tratando de classificação, onde temos interesses em ver o relacionamento de features na classificação de grupos, modelos hierárquicos brilham os olhos, nesse sentido, começamos com o mais básico deles, a árvore de decisão, *decision tree*, que contou com uma excelente acurácia

```
[ ] #implementando o random forest

    from sklearn.ensemble import RandomForestClassifier


    X = pd.concat([cat_df, num_df], axis = 1)
    y = Hotel_df['is_canceled']
    x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    forest_model = RandomForestClassifier(random_state=42, n_estimators=300, bootstrap=False) #manual tunning, (grid_search demorou muito), em média, 200, 300 e 500 estimadores ficaram e
    forest_model.fit(x_train, y_train)
    y_pred = forest_model.predict(x_test)

    #acurácia simples

    accuracy = accuracy_score(y_test, y_pred)
    print("Acurácia:", accuracy)

    #cross validation

    scores = cross_val_score(forest_model, X, y, cv=kf)

    print("Pontuações em cada fold:", scores)
    print("Pontuação média:", scores.mean())
```

Se uma árvore foi boa, o que dizer de 300? Aí então decidimos testar o random forest, um modelo ensemble que utiliza de várias *decision trees* em sua modelagem

```
from sklearn.experimental import enable_halving_search_cv  # Necessário para habilitar o HalvingGridSearchCV
from sklearn.model_selection import HalvingGridSearchCV


# Definir os parâmetros para o grid search
param_grid = {
    'n_estimators': [200],
    'max_depth': [10, 20, 30, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 5],
    'max_features': ['sqrt', 'log2'],
    'bootstrap': [False]
}


halving_Grid = HalvingGridSearchCV(forest_model, param_grid, cv=5, factor=2, verbose=1, n_jobs=1)
halving_Grid.fit(X, y)

# Imprimir os melhores parâmetros encontrados
print("Melhores parâmetros encontrados:")
print(halving_Grid.best_params_)
```

onde testamos diferentes técnicas de *tunning* de hiperparâmetros, como grid_search, *halving_grid_search* e manual *tunning*, onde testamos diferentes técnicas de *tunning* de hiperparâmetros, como *grid_search*, *halving_grid_search* e manual *tunning*, sendo a última a mais eficaz no processo, tendo em vista o alto custo computacional de um *grid_search*



```
from sklearn.metrics import confusion_matrix, classification_report

# splitting data into training set and test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)

cat = CatBoostClassifier(iterations=100)
cat.fit(X_train, y_train)

y_pred_cat = cat.predict(X_test)

acc_cat = accuracy_score(y_test, y_pred_cat)
conf = confusion_matrix(y_test, y_pred_cat)
clf_report = classification_report(y_test, y_pred_cat)
```

ademais, testamos mais um modelo, o *cat boost* que, em síntese, é um *random forest* mais potente, ele tem como sua vantagem uma necessidade menor de pré-processamento ao lidar com valores categóricos, isso que dá o seu nome, além de ter a vantagem de ser um algoritmo baseado em árvores e ser um *gradient boost* ao mesmo tempo, para a nossa surpresa a sua acurácia de 99.4%.

6. Métricas de avaliação

Como principal método utilizamos a acurácia, e sobre divisão de *samples* de teste utilizamos o *k-fold cross validation* fixado em 5.

7. Resultados

A análise mostrou picos de reservas durante os meses de verão (especialmente julho e agosto) e um menor volume de reservas em meses como novembro, dezembro e janeiro.

Recomendação:

Preços Dinâmicos: oferecer descontos ou pacotes promocionais nos meses de baixa ocupação.

Promoções de Baixa Temporada: Criar pacotes especiais que incluam refeições e atividades locais para atrair mais hóspedes nos meses de menor movimento.

Composição de Hóspedes

Insight: A maioria das reservas é composta por adultos viajando sozinhos ou em grupos pequenos, enquanto reservas com crianças e bebês são menos frequentes.

Recomendação:

Promoções para Famílias: Criar pacotes familiares que incentivem a estadia de grupos maiores, incluindo descontos em acomodações adicionais ou serviços como recreação infantil.

Marketing Direcionado: Direcionar campanhas de marketing específicas para famílias durante as férias escolares, destacando atividades e facilidades voltadas para crianças.

Canal de Reserva Mais Lucrativo

Insight: Alguns canais de distribuição são mais lucrativos do que outros, com taxas de comissão menores e clientes que tendem a cancelar menos.

Recomendação:

Focar em Canais Eficientes: Investir mais em marketing nos canais que trazem hóspedes com menor taxa de cancelamento e maior taxa de visitas

Programas de Fidelidade: Oferecer incentivos para reservas diretas por meio do site do hotel, como upgrades de quarto gratuitos ou check-in antecipado.

Insight: Reservas em um periodo muito longo são mais propensas a serem canceladas. Por outro lado, reservas feitas com antecedência moderada (30-60 dias) têm menor chance de cancelamento. sendo que, para reservas em periodos maiores de 100 dias, a probabilidade é maior que cancele do que continue com a reserva.

Recomendação:

Adiantamento: para reservas com antecedência maior que 30 dias, haverá uma cobrança de 25% (ou 10%) adiantada.

Ofertas para Reservas de Última Hora: Implementar promoções para reservas de última hora para preencher quartos não reservados

Justificativas para as Recomendações

Política de Depósito: Reduz a incerteza e protege a receita do hotel contra cancelamentos.

Preços Dinâmicos: Maximiza o lucro durante a alta temporada e minimiza a perda na baixa temporada.

Promoções Familiares: Atrai um público-alvo que pode aumentar a ocupação durante períodos de baixa demanda.

Programas de Fidelidade e reserva direta no site: Diminuem a dependência de intermediários, aumentam a margem de lucro e aumenta a visibilidade do hotel.

Além disso, recomendamos fortemente ao gerente do hotel, que use o nosso modelo preditivo com maior acurácia, e o integre em seu sistema, para que ele possa ser avisado, de clientes

com alta probabilidade de cancelamento e possa tomar medidas, como mandar email confirmando a reserva após algum tempo e que, com o uso do nosso modelo, implemente as medidas citadas anteriormente. Sugerimos também que, a cada ano, o modelo seja atualizado com dados do ano.

8. Conclusão

Finalizando a análise concluímos que as limitações do hotel são cruciais e ao mesmo solucionáveis, usando Ciência de Dados, durante esse artigo, sugerimos medidas que se aplicadas aumentarão de forma significativa o lucro do hotel.

9. Bibliografia

PIMENTEL, Bruno Almeida. Ciência de dados. Universidade Federal de Alagoas, 2º semestre de 2024. Apresentação de slides. Disponível em: <https://sites.google.com/ic.ufal.br/ciencia-de-dados/documentos?authuser=0>. Acesso em: 13 nov. 2024.

MOSTIPAK, Jesse . Hotel booking demand. 2019. Disponível em: <https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand/code?datasetId=511638&sortBy=voteCount>. Acesso em: 13 nov. 2024.