Theodore Koby-Hercsky

12/01/2022

DSC680-T301 2233-1

Professor Catie Williams

Milestone 2 – Draft White Paper

Github: Applied-Data-Science/Hotel Booking Prediction & EDA at main · TheodoreKoby-Hercsky/Applied-Data-Science (github.com)

Business Problem

Hotels now a days are all competing to have you as guest at their hotel and are even offering free cancellations up to 24 hours before your stay. Problem hotels are running into is the ever-growing rate of cancellations which is making it difficult for hotels to provide accurate occupancy rates. The business problem I am aiming to solve is to provide hotels with a hassle-free way to predict cancellations to allow hotels to accurately forecast future occupancy rates. In an article I learned that the duration of the stay has an impact on cancellations as a "room is booked for only 1 or 2 nights, it is more likely to be cancelled then if the length of stay is 3 nights or longer." (Eric) Throughout this report I will investigate hotel reservations and see what drive cancellations to help determine the most accurate way to predict the rate of cancellation.

Background/History

The data I pulled from Kaggle was originally pulled from Science Direct which has access to over 1.4 million article that are open access that allows users to peer-review and available for free to all users. While that data that is provided is from two hotels which consist of a city hotel and a resort hotel from July of 2015 to august of 2017. Which include 31 variables such as the average daily rate, cancellations, lead time, country, guest type such as adult, child, and baby. Because

this is actual real hotel data all data variables that include information that identifies the customer or hotel has been deleted. While the hotels are in “Portugal with hotel one at the resort region of Algarve and hotel two at the city of Lisbon”(Antonioa) Overall these are two great hotel types to review and further understand the rate of cancellation and average daily rate throughout the years.

Data Explanation (Data Prep/Data Dictionary/etc)

The data set I went with is on hotel booking demand from Kaggle that focuses on two hotels which one is a city hotel and the other one is a resort hotel. The data set includes data on cancellations, lead time, arrival time, guests, market segment, previous cancellations, booking changes, and more. Below is a full dictionary of all the variables from the hotel booking demand data set. While Kaggle has indicated that all personal identifiable information was removed from the dataset and that the data is originally from an article titled Hotel Booking Display Dataset that can be accessed by clicking on the link.

In regard to data preparation, I used to describe which is a “method returns description of the data in the DataFrame” that was very useful as it indicated that mean cancelation rate for the data set is 37.04% that can be seen in the figures section as figure one. (W3Schools) I also used the function isnull() to “detect missing values for an array-like object” in the DataFrame that indicated that the variables had missing values such as children with four, country with 488, agent with 16,340, and company with 112,593 missing seen below in figure 2.(Pandas) While another interesting function I used was unique that showed all the different values within a variable as the variable children had six values which included nan meaning nothing was in that cell. Once the missing values were determine I decided to use several functions to fix these such as fillna(), drop(), and drop() all of these functions helped fill the missing children values with 0. While the drop() function was utilized to drop unneeded variables such as agent and company as these had the highest missing values. Last the dropna() method helped as it “allows the user to analyze and drop Rows/Columns with Null values” such as the 488 values missing from the

country variable. (bhutani) Below in figure four you can see a screenshot of the different methods that have been used.

Methods

When it came to my exploratory data analysis I used the value_counts() and sort_index() to find the amount of arrivals each month. When the sort_index() is used it "returns a new DataFrame sorted by label if *in place* argument is False, otherwise updates the original DataFrame and returns None." (Pandas.dataframe) In figure five below we see that month with the most arrivals is August at 8609 and the month with the least arrivals is January with 4061.

While I went on to create visualizations to help answer questions, I had regarding the data such as what types of guests checked into the hotel. Which I found with the use of a histogram by using hist() which " plot pre-computed bins and counts using hist() by treating each bin as a single point with a weight equal to its count." (Hunter) Below in figure six you can see that the histogram visualization that was created with the variable customer type showed that the highest number of guests are Transient guest and transient parties while groups is seen to be the lowest. One of the most interesting questions to me was what the average daily rate for first time guest and repeat guest which showed us that repeat guests waited for deals and paid on average $60 a night while the first-time guest paid around $100.00 and above. The bar plot below in figure seven shows that the hotels are paying not just in advertisement but also discounted rates to bring back guests. Another interesting histplot that calculated the rate of bookings for both hotels. Which the city hotel showed almost double rate of booking at just under 80,0000 while the resort hotel only had around 40,000 bookings that can be seen below in figure eight.

Analysis

In regard to analysis I created a seaborn heatmap that is an "Axes-level function and will draw the heatmap into the currently-active Axes that will be taken and used to plot a colormap." (Waskom)

The heat map from seaborn indicated that there was a correlation of 33% between the variables children and ADR which is average daily rate that can be seen below in figure nine. The percent of cancellation can be seen below in figure ten that shows a rate of 37.20% which is a little higher than average for hotels. Next decided to prep for modeling by removing unneeded variables such as reservation status, arrival date year, assigned room type, country, and more that can be seen below in figure eleven. While also replacing infinite values, dropping missing values, and resetting indexes. While analyzing the data I decided to encode the categorical variables which changed the year values from 2015 through 2017 to zero through three to allow for ease when modeling.

When it came to modeling, I created my X and Y by setting my X value to all the remaining edited variables except the canceled variable which was used in my Y value that has a shape of 118,732 values for 24 variables seen below in figure thirteen. The first model I created is the K-Nearest Neighbors which is an "one of the most famous machine learning algorithms that is unsupervised and Nonlinear." (Real Python) The model produced a recall of 98% and an F1-score of 97% that can be seen below in figure fourteen. While the second model I created is the Random Forest Classifier Random Forest Classifier which is a "supervised learning algorithm that can be used both for classification and regression and is flexible and easy to use." (Navlani) Showing that the array had 26,070 and 15,487 for the support for zero and one which is seen below in figure fifteen. https://www.msn.com/en-us/feedThe last model I went with is the Gradient Boosting Classifier which is a "group of machine learning algorithms that combine many weak learning models together to create a strong predictive model." (Nelson) Which showed that the precision, recall, and f1-score came back at 100% which is something I have never seen before which you can view in figure sixteen.

Conclusion

The three models had interesting findings such as the K-Nearest Neighbors which had a accuracy score of 96.31%. While the other models Random Forest and Gradient Boosting Classifier had the

most interesting finding as they both had a 100% which is very odd as I have never had this happen before and even tried and retested them and received the same results. Below you can see the classifications for all three models and in figure seventeen we can see that the accuracy for the KNN came back as 96.31% while the Random Forest and Gradient Boosting Classifier accuracy came back with 100% for both. Overall, all three models are great choices for predicting accuracy for hotel cancellations.

Assumptions

Regarding assumptions it can be assumed that both hotels are losing money on repeat guests as the average daily rate of a return guest is only around $60 while first time guests are spending upwards of $100 a night. While in figure eighteen below we see the changing average daily rate over each month for both hotels which it can be assumed that the city hotel is mostly used for business due to the average consistent rate that stays between $120 and $80 while the Resort hotel is assumed to focus on vacationers due to the ADR fluctuating between over $180 during the summer and lows $50 in the winter months.

Limitations

Limitations with the data frame is that there is know time range with when discounts and special promotions took place which doesn't allow users to see what cause the increase in bookings. While another limitation I see regarding predicting cancellations is that guests can choose to cancel their stay up to 24 hours in advance which is problematic as the hotel could be running a promotion and guests that have already booked could just cancel and rebook to receive a cheaper price which is not documented well within the data frame.

Challenges

Challenges I see with predicting cancellations is that it is constantly going to get harder to predict due to the ever-improving technology and competition between hotels. Which is causing

hotels to offer better rates and the internet and travel agents are making it easier and cost free to cancel and or change reservations making it harder for hotel staff to schedule workers for expected occupancy. While the introduction of Covid-19 has even made this more difficult as hotels are starting to even allow free cancellation to guests that pop a positive for Covid-19 which makes sense as you do not want the infected guest to still come and risk getting other guests and staff sick. Overall hotels will see consistent challenges with cancellations but if they put into consideration the fact that they will have to adjust for a 30% to 35% cancellation rate they will be able to proceed and go on with business as usual.

Future Uses/Additional Applications

These models can be used by the hotel for years to come in predicting future cancellations and on how to track the hotels occupancy throughout the year. While this report can also help hotel staff and management understand their slow months and allow them to determine the best course of action to improve their occupancy during those months. While an additional model that can be created to see if there is a change in accuracy could be the use of the Decision Tree Classifier. While another use that this model could be used for is to determine if a guest is going to have children with them or not. I feel as if this would be another great opportunity that hotel management could take advantage of as it would allow them to see if the hotel would be hosting families or business guests. This would allow them to target additional sales at the hotel for the guest's arrival such as activities for children or a happy hour for guests that are staying for business.

Recommendations

Regarding recommendations the hotel could work on a new campaign to attract new guests for the months of December and January as these are the month with the lowest occupancy. The hotel could offer lower daily rates for guests staying three nights or more as this

will increase occupancy throughout the month and would boost sales from restaurants and other sales, they may have such as bars. Seen in figure seven we can see that repeat guests are only willing to to pay an average daily rate of $60 while first time guests are willing to spend upwards of $100 a night. Therefore, the hotel needs to focus on attracting new guests and seeing what can be done around the hotel to improve it to have more guests come back.

Implementation Plan

The hotel will need to implement new promotions to gain new and returning guests to boost the occupancy within the hotel. As we have seen that in figure five January has the lowest rate that showed only 4,061 while in the month of august the hotels saw guest of 8,609 which is double then what the hotels are used to in the winter months. Due to these astounding rates the hotel will needed to offer promotions and even dinner packages if able to draw in guests during the holiday season. While the hotels have seen a cancellation rate of upwards of 37% which is high due to this hotel management will need to further examine the hotel and their policies to see what can be done to reduce the cancelation rate.

Ethical Assessment

An ethical assessment regarding the hotel business and the problems they can be handled as followed as the hotel manager will need to make sure to communicate with their staff and make sure everyone is on the same page in regard to the hotel assessment. The hotel manager can conduct audits and determine the best course of action for each department within the hotel. This will help strengthen the team and help improve the hotel over and hopefully find the reason behind the high rate of cancellations.

Data Dictionary

- **Is Canceled** – Indicates if the booking has been canceled or not with a (1) for cancelled and (0) if not.
- **Leadtime** – Indicates the number of days that elapsed between the entering date of the booking into the PMS and arrival date.
- **Arrival Date Year** – Indicates the arrival date
- **Arrival Date Month** – Indicates the month of the arrival date.
- **Arrival Date Week Month** – Indicates the week number of year for the arrival date
- **Arrival Date Day of Month** – Indicates the day of arrival
- **Stays In Weekend Nights** – Indicates the number of weekend nights such as Saturday and Sunday the guest will be staying or booked to stay at the hotel
- **Stays In Weeknights** – Indicates the number of week nights such as Monday through Friday the guests will be staying or have booked to stay at the hotel.
- **Adults** – Indicates the number of adults
- **Children -** Indicates the number of children
- **Babies -** Indicates the number of babies
- **Meal** – Indicates the type of meal booked. These categories include standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)
- **Country** – Indicates the country of origin

- **Market Segment** – Indicates the market segment designation such as the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- **Distribution Channel** – Indicates the booking distribution channel such as the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- **Is Repeated Guest** – Indicates the value indicating if the booking name was from a repeated guest is (1) and (0) if not
- **Previous Cancellations** – Indicates the number of previous bookings that were cancelled by the customer prior to the current booking.
- **Previous Bookings Not Cancelled** – Indicates the Number of previous bookings not cancelled by the customer prior to the current booking.
- **Reserved Room Type** – Indicates the Code of room type reserved. Code is presented instead of designation for anonymity reasons.
- **Assigned Room Type** – Indicates the Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (such as overbooking) or by customer request. Code is presented instead of designation for anonymity reasons.
- **Booking Changes** – Indicates the Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation.
- **Deposit Type** – Indicates the Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories:
    - **No Deposit** – no deposit was made

    - **Non-Refund** – a deposit was made in the value of the total stay cost
    - **Refundable** – a deposit was made with a value under the total cost of stay

- **Agent** – Indicates the ID of the travel agency that made the booking.
- **Company** – Indicates the ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
- **Days In Waiting List** – Indicates the Number of days the booking was in the waiting list before it was confirmed to the customer.
- **Customer Type** – Indicates the Type of booking, assuming one of four categories:
    - **Contract -** when the booking has an allotment or other type of contract associated to it
    - **Group** – when the booking is associated to a group
    - **Transient** – when the booking is not part of a group or contract, and is not associated to other transient booking
    - **Transient-party** – when the booking is transient, but is associated to at least other transient booking
- **ADR** – Indicates the Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
- **Required Car Parking Spaces** – Indicates the Number of car parking spaces required by the customer
- **Total Of Special Requests** – Indicates the Number of special requests made by the customer (such as twin bed or high floor)

- **Reservation Status** – Indicates the Reservation last status, assuming one of three categories:
    - **Canceled** – booking was canceled by the customer
    - **Check-Out** – customer has checked in but already departed
    - **No-Show** – customer did not check-in and did inform the hotel of the reason why
- **Reservation Status Date** – Indicates the Date at which the last status was set. This variable can be used in conjunction with the Reservation Status to understand when the booking was canceled or when did the customer checked-out of the hotel.

Figures

- Figure 1: Describe method showed a 37.04% for mean cancelations

```
# Use describe to see different statistics for the data set
Hotel_Bookings.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_wee
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	11
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	
std	0.482918	106.863097	0.707476	13.605138	8.780829	
min	0.000000	0.000000	2015.000000	1.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	
max	1.000000	737.000000	2017.000000	53.000000	31.000000	

- Figure 2: Isnull() and sum() showed the number of missing values for each variable

```
# I will now use isnull and sum to check for missing data
Hotel_Bookings.isnull().sum()
```

```
hotel                               0
is_canceled                         0
lead_time                           0
arrival_date_year                   0
arrival_date_month                  0
arrival_date_week_number            0
arrival_date_day_of_month           0
stays_in_weekend_nights             0
stays_in_week_nights                0
adults                              0
children                            4
babies                              0
meal                                0
country                           488
market_segment                      0
distribution_channel                0
is_repeated_guest                   0
previous_cancellations              0
previous_bookings_not_canceled      0
reserved_room_type                  0
assigned_room_type                  0
booking_changes                     0
deposit_type                        0
agent                           16340
company                        112593
days_in_waiting_list                0
customer_type                       0
adr                                 0
required_car_parking_spaces         0
total_of_special_requests           0
reservation_status                  0
reservation_status_date             0
dtype: int64
```

- Figure 3: Unique function showed that the variable children had six unique values including nan

```
# View how many unique values the children variable has
Hotel_Bookings['children'].unique()
```

```
array([ 0.,  1.,  2., 10.,  3., nan])
```

- Figure 4: The methods I used to fix the missing vules consist of fillna(), drop(), and dropna().

I will first take care of the children variable that has four missing values. That will be filled with 0 as if they did not indicate they had children then we will go with no children being that it is only four missing values.

```
# I will use fillna to fill the missing values from the children variable with o
Hotel_Bookings['children'] = Hotel_Bookings['children'].fillna(0)
```

I will now drop agent and company as these values will not be needed for my analysis.

```
# I will use drop to remove my agent and company variables
Hotel_Bookings=Hotel_Bookings.drop(['agent','company'],axis=1)
```

Being that the data frame has 119,390 rows of values and the Country variable is only missing 488 values I will be dropping the Na values.

```
# I will use dropna to remove the remaining values na values
Hotel_Bookings = Hotel_Bookings.dropna(axis = 0)
```

- Figure 5:

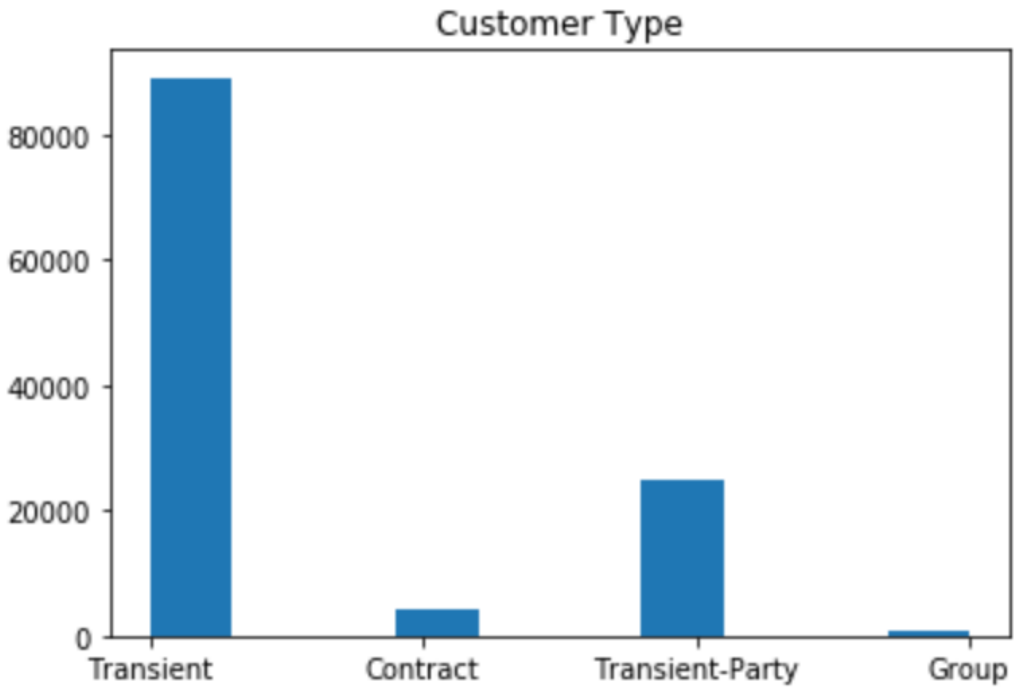
```
# I will now create a value that shows the bookings confirmed
bookings_confirm = Hotel_Bookings[Hotel_Bookings.is_canceled=='0']
```

```
# Next I will view the arrivals each month
import datetime as dt
bookings_confirm['ArrivingMonth'] = Hotel_Bookings['Arrival_Date'].dt.month
Monthly_Bookings=bookings_confirm['arrival_date_month'].value_counts().sort_index()
Monthly_Bookings
```

```
Out[22]: April        6527
         August       8609
         December     4353
         February     5308
         January      4061
         July         7880
         June         6387
         March        6566
         May          7087
         November     4617
         October      6854
         September    6356
         Name: arrival_date_month, dtype: int64
```

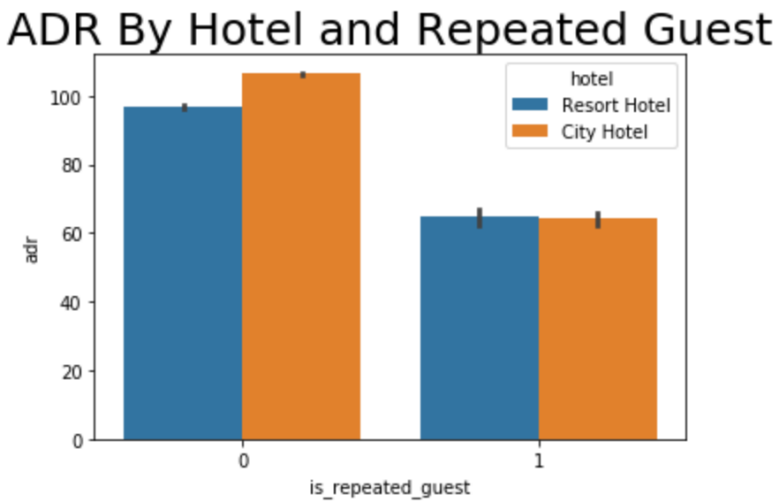
- Figure 6:

```
# Create a histogram that shows the customer type with plt
plt.hist(Hotel_Bookings['customer_type'])
plt.title("Customer Type")
plt.show()
```



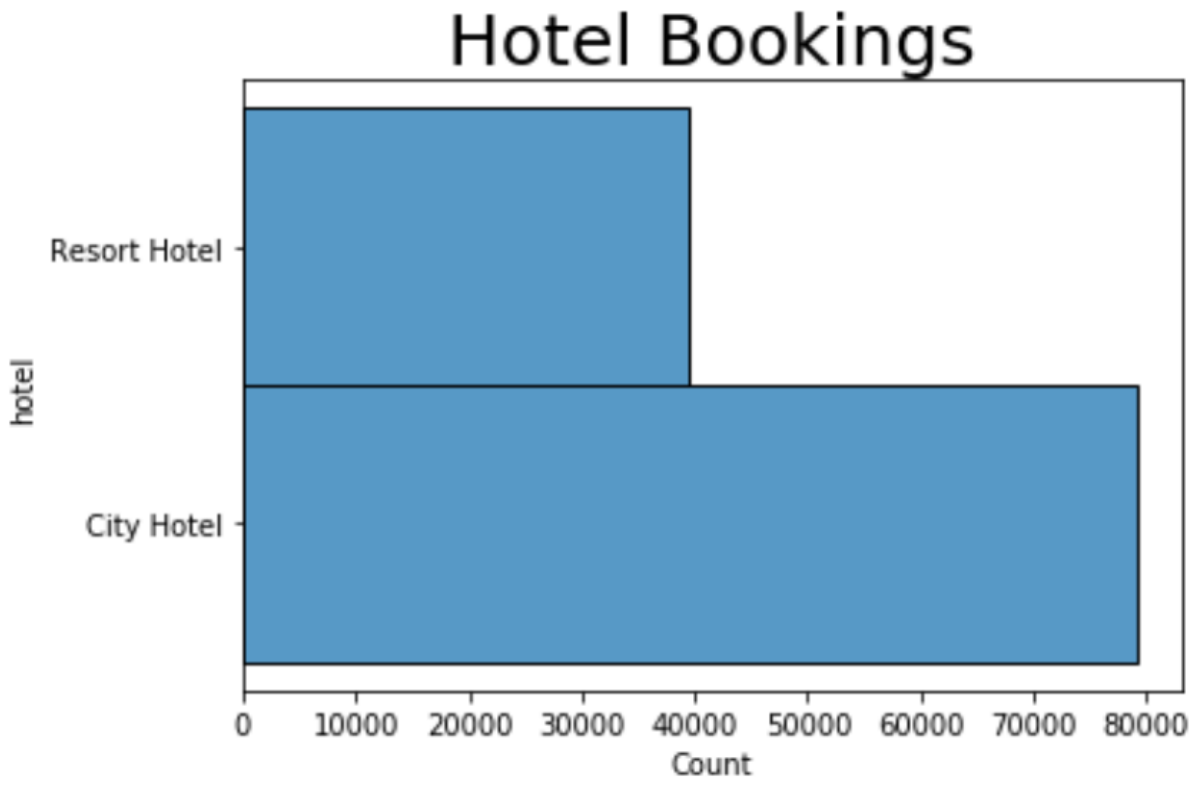
- Figure 7:

```
# I will create a bar plot that shows the adr for repeated guest and not and the two hotels
plt.title("ADR By Hotel and Repeated Guest", fontdict = {'fontsize': 25})
sns.barplot(x='is_repeated_guest',y='adr', data=Hotel_Bookings, hue='hotel')
plt.show()
```



- Figure 8:

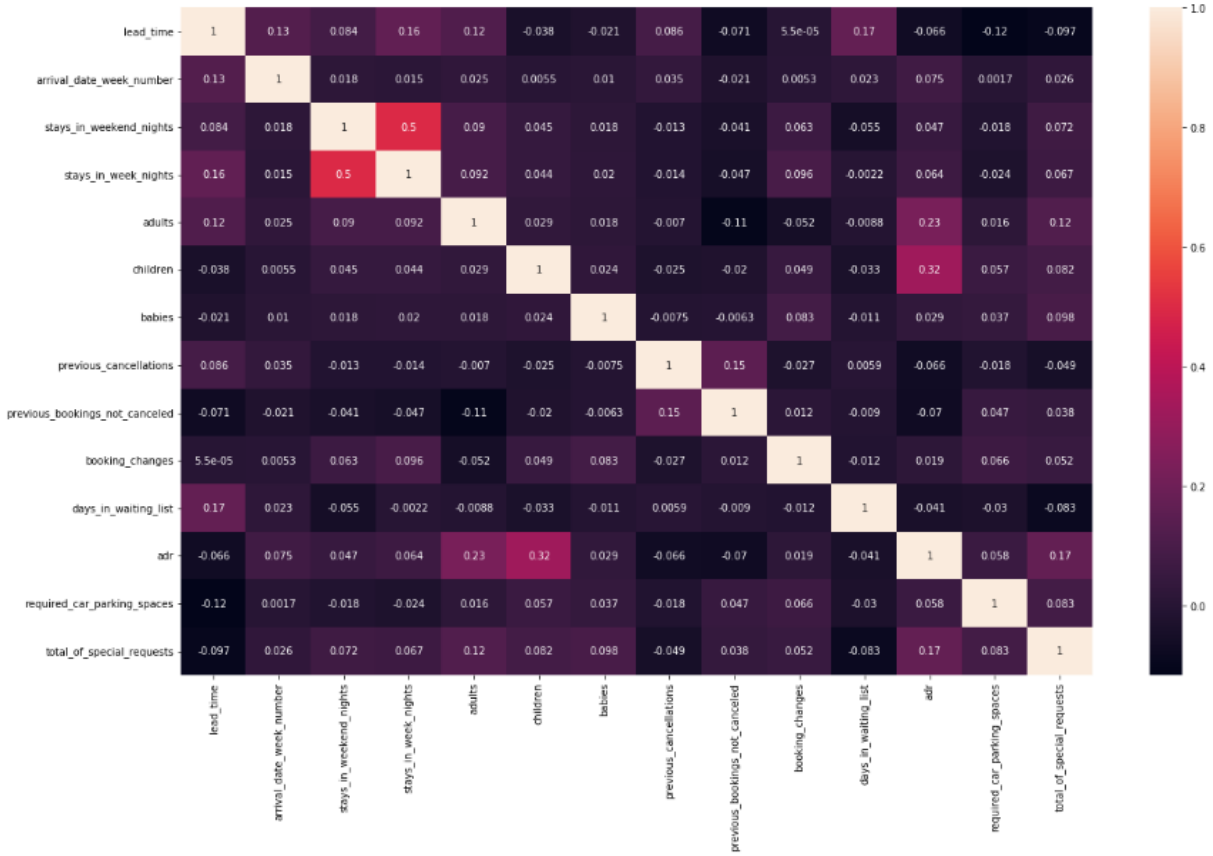
```
# Which hotel has the higher rate of reservations
plt.title("Hotel Bookings", fontdict = {'fontsize': 25})
sns.histplot(data=Hotel_Bookings, y="hotel")
plt.show()
```



- Figure 9:

```
# I will use plot to creat a heatmap to view
# the correltion between the data.
plt.figure(figsize=(20,12))
sns.heatmap(Hotel_Bookings.corr(),annot=True)
```

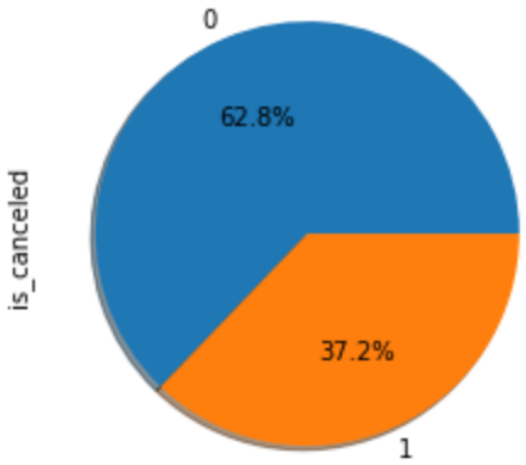
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d36380f9b0
>
```



- Figure 10:

```
# I will create a pie chart that shows the percent of cancelations
Hotel_Bookings['is_canceled'].value_counts().plot.pie(
    autopct='%1.1f%%',shadow=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d363139f28>
```



- Figure 11:

```
# I will fisrt drop some uneeded columns
drop_columns = ['reservation_status', 'arrival_date_year', 'assigned_room_type',
                'country', 'days_in_waiting_list', 'arrival_date_year', 'booking_changes']
Hotel_Bookings.drop(drop_columns, axis = 1, inplace = True)
```

```
# I will replace infinite numbers and drop NAs if located as I was having issues with my model
Hotel_Bookings = Hotel_Bookings.replace([np.inf, -np.inf], np.nan)
Hotel_Bookings = Hotel_Bookings.dropna()
Hotel_Bookings = Hotel_Bookings.reset_index()
```

- Figure 12:

```
# Next I will encode the categorical variables
categorical_data['distribution_channel'] = categorical_data['distribution_channel'].map({'Direct': 0, 'Corporate': 1,
                                           'TA/TO': 2, 'Undefined': 3, 'GDS': 4})
categorical_data['reserved_room_type'] = categorical_data['reserved_room_type'].map({'C': 0, 'A': 1,
                                           'D': 2, 'E': 3, 'G': 4, 'F': 5, 'H': 6, 'L': 7, 'B': 8, 'P':9})
categorical_data['hotel'] = categorical_data['hotel'].map({'Resort Hotel' : 0, 'City Hotel' : 1})
categorical_data['deposit_type'] = categorical_data['deposit_type'].map({'No Deposit': 0, 'Refundable': 1, 'Non Refund': 3})
categorical_data['customer_type'] = categorical_data['customer_type'].map({'Transient': 0, 'Contract': 1,
                                           'Transient-Party': 2, 'Group': 3})
categorical_data['meal'] = categorical_data['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC': 3, 'Undefined': 4})
categorical_data['market_segment'] = categorical_data['market_segment'].map({'Direct': 0, 'Corporate': 1,
                                           'Online TA': 2, 'Offline TA/TO': 3, 'Complementary': 4,
                                           'Groups': 5, 'Undefined': 6, 'Aviation': 7})
categorical_data['year'] = categorical_data['year'].map({2015: 0, 2014: 1, 2016: 2, 2017: 3})
```

- Figure 13:

```
# I will now define my X and Y
X = pd.concat([categorical_data, numerical_data], axis = 1)
y = Hotel_Bookings['is_canceled']
```

```
# Set Y to intagers as I was running into an
# error with my models
y=y.astype('int')
```

```
# Now I will view the shape of my X and Y
X.shape, y.shape
```

```
((118732, 24), (118732,))
```

- Figure 14:

```
# Next I set my prediction to the x_test and accuracy
knn_pred = knn.predict(X_test)
knn_accuracy = accuracy_score(y_test, knn_pred)
```

```
# I will create the confusion matrix and classification report
knn_Matrix = confusion_matrix(y_test, knn_pred)
knn_class = classification_report(y_test, knn_pred)
# Print Matrix
knn_Matrix
```

```
array([[25523,   547],
       [  985, 14502]], dtype=int64)
```

```
# I will now print my Classification Repor for KNN
print(knn_class)
```

```
              precision    recall  f1-score   support

           0       0.96      0.98      0.97     26070
           1       0.96      0.94      0.95     15487

    accuracy                           0.96     41557
   macro avg       0.96      0.96      0.96     41557
weighted avg       0.96      0.96      0.96     41557
```

- Figure 15:

```
# I will create the confusion matrix and classification report
rf_Matrix = confusion_matrix(y_test, rf_pred)
rf_class = classification_report(y_test, rf_pred)
# Print Matrix
rf_Matrix
```

```
array([[26070,     0],
       [    0, 15487]], dtype=int64)
```

```
# I will now print my Classification Repor for Random Forest
print(rf_class)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     26070
           1       1.00      1.00      1.00     15487

    accuracy                           1.00     41557
   macro avg       1.00      1.00      1.00     41557
weighted avg       1.00      1.00      1.00     41557
```

- Figure 16:

```
# I will create the confusion matrix and classification report
gb_Matrix = confusion_matrix(y_test, gb_pred)
gb_class = classification_report(y_test, gb_pred)
# Print Matrix
gb_Matrix
```

```
array([[26070,     0],
       [    0, 15487]], dtype=int64)
```

```
# I will now print my Classification Repor for Gradient Boostir
print(gb_class)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     26070
           1       1.00      1.00      1.00     15487

    accuracy                           1.00     41557
   macro avg       1.00      1.00      1.00     41557
weighted avg       1.00      1.00      1.00     41557
```

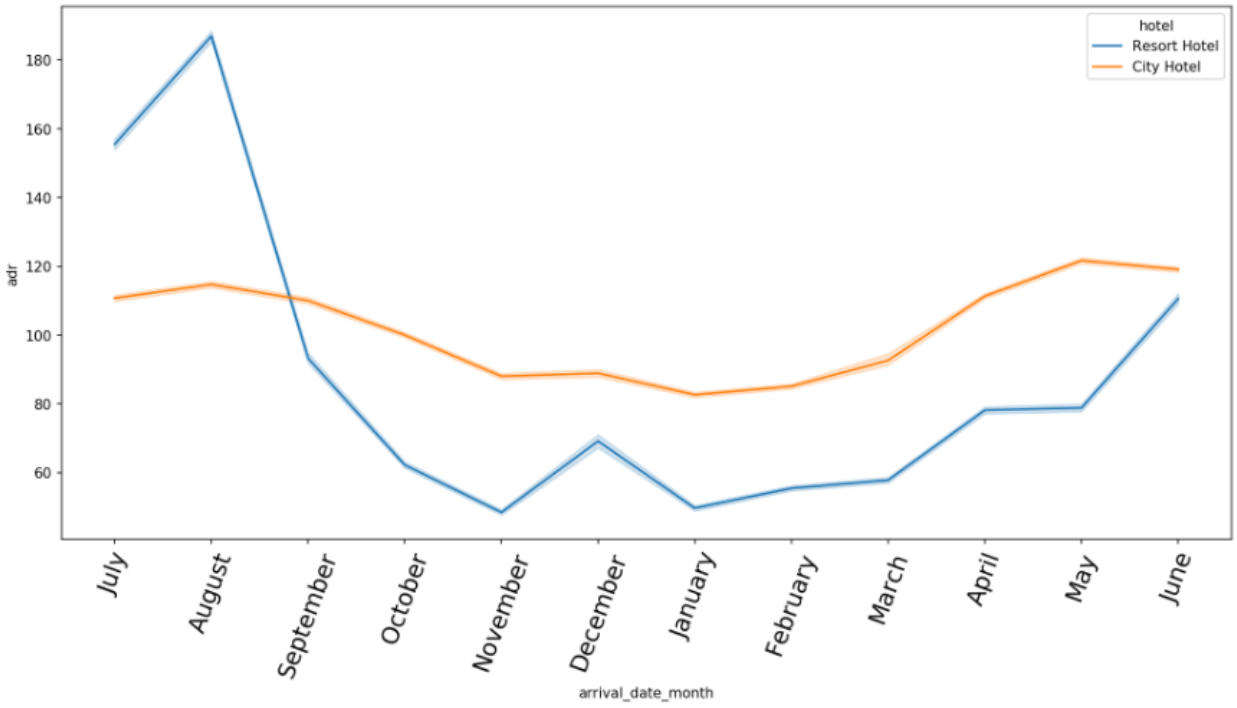
- Figure 17:

```
# I will now print the accuracy of each model
print(f"KNN Accuracy: {knn_accuracy}")
print(f"Random Forest Accuracy: {rf_accuracy}")
print(f"Gradient Boosting Classifier Accuracy: {gb_accura
```

```
KNN Accuracy: 0.963134971244315
Random Forest Accuracy: 1.0
Gradient Boosting Classifier Accuracy: 1.0
```

- Figure 18:

```
# I will use seaborn to see which hotel has the cheaper
# rate per month
fig = plt.figure(figsize=(15,7),dpi=150)
sns.lineplot(x='arrival_date_month', y='adr',
             data=Hotel_Bookings, hue='hotel')
plt.xticks(rotation=70,fontsize=17);
plt.show()
```



- Figure 19:

Citations

Eric. (n.d.). *Hotel cancellations pose a great challenge*. Revenue Management Software für Hotels jeder Größe. Retrieved December 6, 2022, from https://www.rateboard.io/en/blog/hotel-cancellations-pose-a-great-challenge

W3Schools. (n.d.). *Pandas DataFrame describe() Method*. Pandas dataframe describe() method. Retrieved December 6, 2022, from https://www.w3schools.com/python/pandas/ref_df_describe.asp

Pandas isnull() function. w3resource. (n.d.). Retrieved December 6, 2022, from https://www.w3resource.com/pandas/isnull.php#:~:text=The%20isnull()%20function%20is,arrays%2C%20NaT%20in%20datetimelike).&text=Object%20to%20check%20for%20null%20or%20missing%20values

bhutani, K. (2018, July 5). *Python: Pandas dataframe.dropna()*. GeeksforGeeks. Retrieved December 6, 2022, from https://www.geeksforgeeks.org/python-pandas-dataframe-dropna/

Pandas.dataframe.sort_index#. pandas.DataFrame.sort_index - pandas 1.5.2 documentation. (n.d.). Retrieved December 7, 2022, from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.sort_index.html

Hunter, J. (n.d.). *Matplotlib.pyplot.hist#*. matplotlib.pyplot.hist - Matplotlib 3.6.2 documentation. Retrieved December 7, 2022, from https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Waskom, M. (n.d.). *Seaborn.heatmap#*. seaborn.heatmap - seaborn 0.12.1 documentation. Retrieved December 7, 2022, from https://seaborn.pydata.org/generated/seaborn.heatmap.html

Real Python. (2022, September 1). *The K-nearest neighbors (knn) algorithm in Python*. Real Python. Retrieved December 7, 2022, from https://realpython.com/knn-python/

Navlani, A. (2018, May 16). *Sklearn Random Forest classifiers in python tutorial*. DataCamp. Retrieved December 7, 2022, from https://www.datacamp.com/tutorial/random-forests-classifier-python

Nelson, D. (2022, July 21). *Gradient boosting classifiers in python with scikit-learn*. Stack Abuse. Retrieved December 7, 2022, from https://stackabuse.com/gradient-boosting-classifiers-in-python-with-scikit-learn/

Antonioa, N., & Nunesabd, L. (2018, November 29). *Hotel booking demand datasets*. Data in Brief. Retrieved December 9, 2022, from https://www.sciencedirect.com/science/article/pii/S2352340918315191