# Study of Hotel Booking cancellation around the globe

CAPSTONE PROJECT-INTERIM REPORT

*Mentored By,*

**Shashank Prakash Shirude**

*Submitted By,*

Arjith Babu

Priyanka C

Sanjana S

Srinivasagopalan

## Table of Contents

# 1. Introduction:

Hospitality industry is one of the most diverse industries among the service line sector and also one of the most competitive industries around the world. Hospitality industry is basically an industry which is categorized as a service line sector where the main goal is to provide accommodation with different service to the guest. The modern world which is fully influenced by technology has also helped the industry to grow with it. The people are the blood for this business and they are the clients for the industry. The people who plan to travel have many preference and thoughts about their stay and expect more. The preference differs for each and every person and also based on the type of travel whether its leisure or official. According to a research released by Dohop website some 19% of hotels that are booked online are cancelled before the guest arrives at the hotel. By the day of the week, Tuesday shows the least amount of cancellations at an average of 14.3 per hotel, and Friday the most with an average of 20.1 cancelled rooms per hotel. This refundable cancellation trend has led to an increase in industry-wide cancellations which has not led to more revenue from cancellations for hotels. As a percentage of operating income, attrition and cancellation fees fell each year from 6.8% in 2009 to 2.1% in 2014, according to CBRE. High cancellation rates can lead to consequent loss of revenue due to empty rooms. With last-minute cancellations and "no-shows", the capacity allocation is no longer optimal because hotels do not succeed in attracting guests on such short notice.

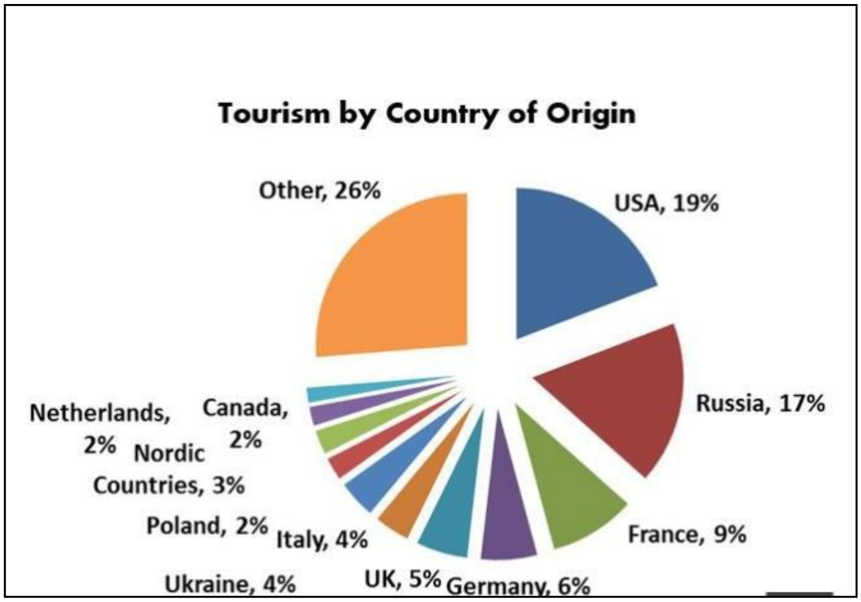


Figure 1: Hospitality trend analysis across the globe

### 1.1 Need of Study:

Several studies were previously done to study of Hotel Booking cancellation data around the globe, as it a people oriented industry due to increase in the growth of population and increase in the number of hotels the trend varies. This Study helps us to analyze the pattern of cancellations in hotel bookings across the globe which helps us to derive some business solutions in predicting the cancellation and reduce the loss faced by the hospitality industry respectively.

### 1.2 Study Objective:

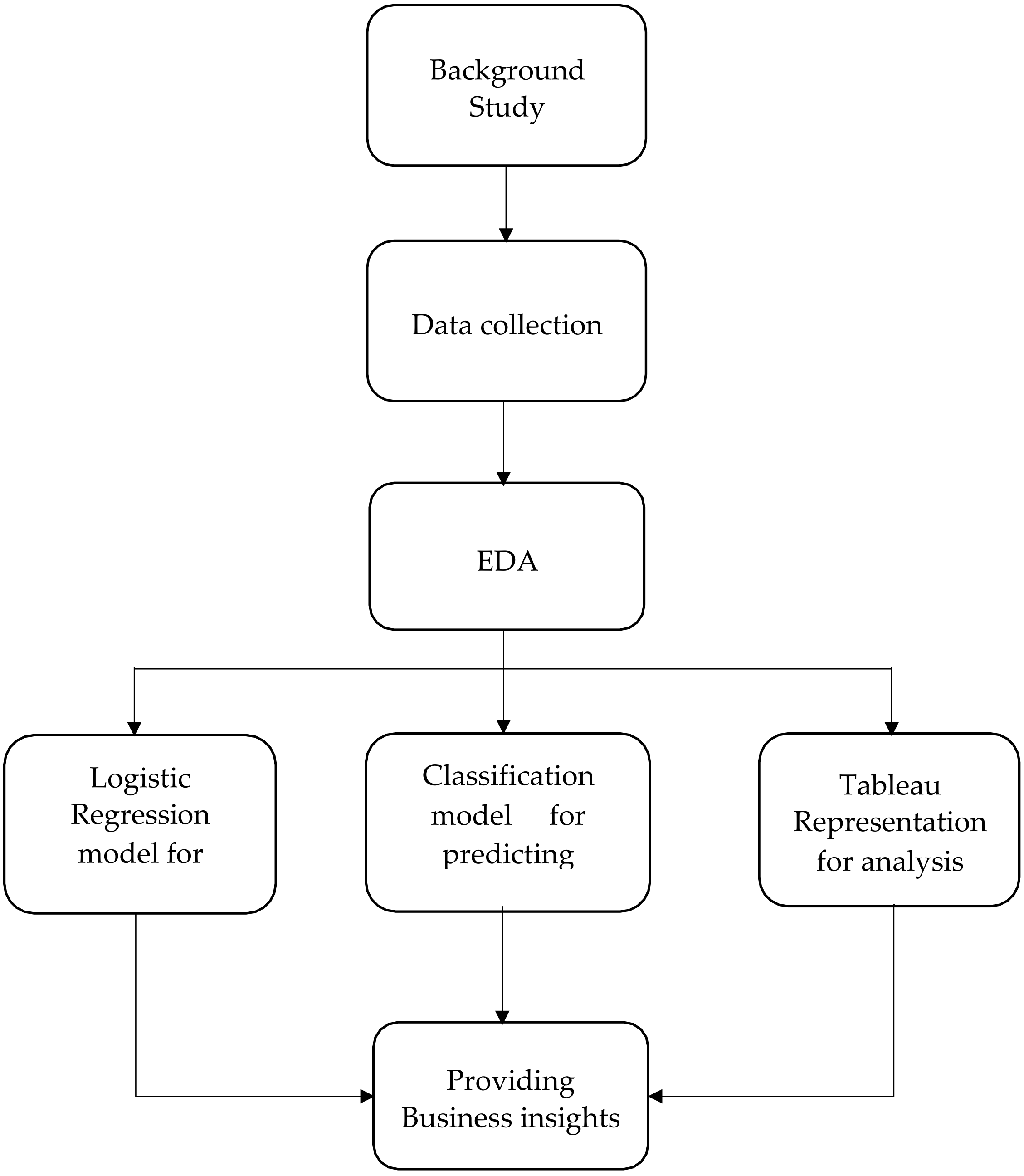
- Study of hotel booking cancellations across various countries to predict the probability of cancellation of the bookings.
- Foreseeing the probability of cancellation depending on lead time, ADR and deposit type.
- Comparing the probability of cancellation depending on the day/week of the month.
- Identifying the preferences of the customers in selecting hotel type during weekend and weekdays.
- Predicting the most frequently used marketing segment for booking hotels.

### 1.3 Scope of the Study:

Using uncensored data from four resort hotel's Property Management Systems (PMS) (region of Algarve, Portugal) representing this tendency for hotels to have increasingly higher booking cancellations rates ,this paper aims to demonstrate how data science can be applied in the context of hotel revenue management to predict bookings cancellations. Moreover, show that booking cancellations do not necessarily mean uncertainty in forecasting room occupation and forecasting revenue. This is achievable by:

- Identifying which features in hotel PMS's databases contribute to predict a booking cancellation probability.
- Building a model to classify bookings with high cancellation probability and using this information to forecast cancellations by date.
- Understanding if one prediction model fits all hotels or if a specific model has to be built for each hotel

## 1.4 Methodology:


Background Study
Data collection
EDA
Logistic Regression model for
Classification model for predicting
Tableau Representation for analysis
Providing Business insights

### 1.5 Data Source:

The Hotel booking demand dataset is taken from Kaggle which has originally been described in Antonio et al. (2019): Hotel booking demand datasets. This data article describes two datasets with hotel demand data. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). Both datasets share the same structure, with 31 variables describing the 40,060 observations of H1 and 79,330 observations of H2. Each observation represents a hotel booking. Both datasets comprehend bookings due to arrive between the 1st of July of 2015 and the 31st of August 2017, including bookings that effectively arrived and bookings that were canceled. Since this is hotel real data, all data elements pertaining hotel or costumer identification were deleted.

Features in the dataset:

| **Variable** | **Type** | **Description** |
|---|---|---|
| ADR | Numeric | Average Daily Rate |
| Adults | Integer | Number of adults |
| Agent | Categorical | ID of the travel agency |
| ArrivalDateDayOfMonth | Integer | Day of the month of the arrival date |
| ArrivalDateMonth | Categorical | Month of arrival date. |
| ArrivalDateWeekNumber | Integer | Week number of the arrival date |
| ArrivalDateYear | Integer | Year of arrival date |
| AssignedRoomType | Categorical | Assigned room code. |
| Babies | Integer | Number of babies |
| BookingChanges | Integer | No. of changes made since the booking was made. |
| Children | Integer | Number of children |
| Company | Categorical | ID of the company that made the booking. |
| Country | Categorical | Country of origin. |
| CustomerType | Categorical | Type of booking : Contract, Group, Transient and Transient party |
| DaysInWaitingList | Integer | No.of days the booking was in the waiting list. |
| DepositType | Categorical | Type of deposit i.e. No Deposit, Non refund and Refundable. |
| DistributionChannel | Categorical | Booking distribution channel. |

| | | |
|---|---|---|
| | | TA and TO |
| IsCanceled | Categorical | If the booking is canceled (1) or not(0) |
| IsRepeatedGuest | Categorical | If the guest is repeated (1) or not(0). |
| LeadTime | Integer | No. of days between booking and the arrival date. |
| MarketSegment | Categorical | Market segment designation :TA and TO |
| Meal | Categorical | Type of meal booked. |
| PreviousBookingsNotCanceled | Integer | No. of previous bookings not cancelled by the customer prior to the current booking. |
| PreviousCancellations | Integer | No. of previous bookings that were cancelled by the customer prior to the current booking. |
| RequiredCardParkingSpaces | Integer | No. of car parking spaces required. |
| ReservationStatus | Categorical | Canceled, Check out and No show |
| ReservationStatusDate | Date | Date at which the last status was set. |
| ReservedRoomType | Categorical | Code of room type reserved. |
| StaysInWeekendNights | Integer | No. of weekend nights (Saturday or Sunday). |
| StaysInWeekNights | Integer | No. of week nights (Monday to Friday) |
| TotalOfSpecialRequests | Integer | No. of special requests(e.g. twin bed or high floor) |

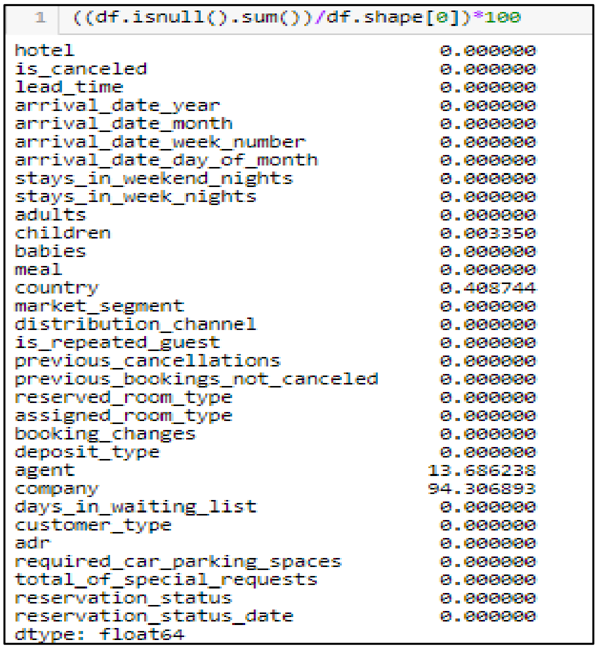
# 2. Exploratory Data Analysis:

In our EDA, we have performed the following:

- Treating null values
- Removal of unused or repeated columns
- Outlier Treatment
- Analysis between various features with respect to our target variable
- Data type conversion

## 2.1 Treating Null Values:

Below is the output which shows the percentile of null values present in our data set :

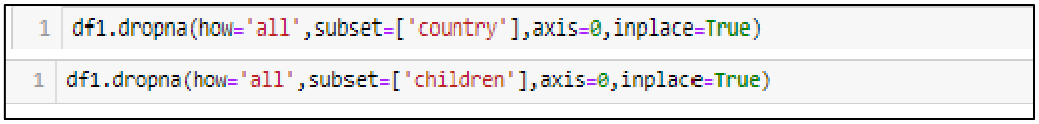


```
((df.isnull().sum())/df.shape[0])*100
```

```
hotel                               0.000000
is_canceled                         0.000000
lead_time                           0.000000
arrival_date_year                   0.000000
arrival_date_month                  0.000000
arrival_date_week_number            0.000000
arrival_date_day_of_month           0.000000
stays_in_weekend_nights             0.000000
stays_in_week_nights                0.000000
adults                              0.000000
children                            0.003350
babies                              0.000000
meal                                0.000000
country                             0.408744
market_segment                      0.000000
distribution_channel                0.000000
is_repeated_guest                   0.000000
previous_cancellations              0.000000
previous_bookings_not_canceled      0.000000
reserved_room_type                  0.000000
assigned_room_type                  0.000000
booking_changes                     0.000000
deposit_type                        0.000000
agent                              13.686238
company                            94.306893
days_in_waiting_list                0.000000
customer_type                       0.000000
adr                                 0.000000
required_car_parking_spaces         0.000000
total_of_special_requests           0.000000
reservation_status                  0.000000
reservation_status_date             0.000000
dtype: float64
```

.

Since our dataset contains 3.3% of null values out of the total data set, we decide to drop all the null/missing values

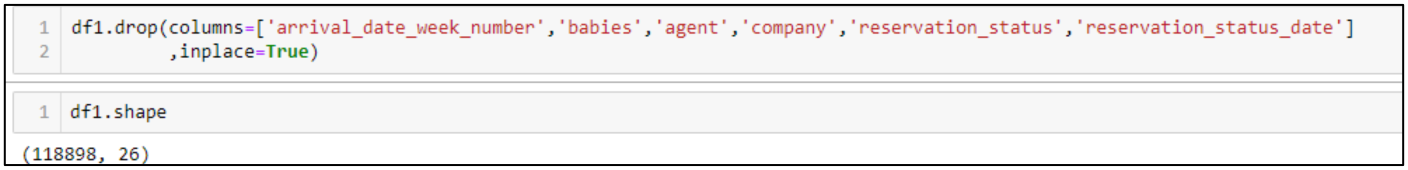


```
df1.dropna(how='all',subset=['country'],axis=0,inplace=True)
```

```
df1.dropna(how='all',subset=['children'],axis=0,inplace=True)
```

### 2.2 Removal of unused or repeated columns:

Our Dataset contains of total 32 features in total. Amongst these 32 features, we have removed few features as they either contain redundant data or contribute to noise in the data. Hence we drop these features.



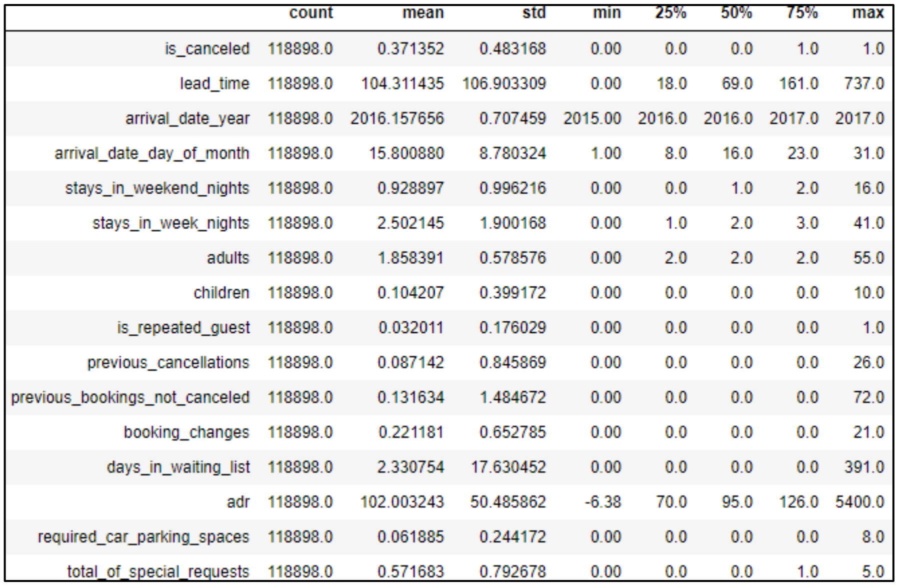
```
df1.drop(columns=['arrival_date_week_number','babies','agent','company','reservation_status','reservation_status_date']
         ,inplace=True)
```

```
df1.shape
```

```
(118898, 26)
```

Post removal of unused or repeated features, from 32 features we have obtained 26 features.

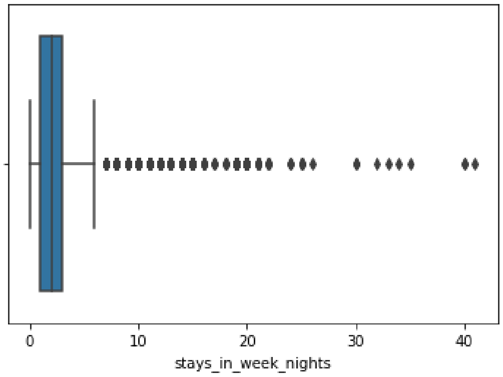
### 2.3 Outlier Treatment:



| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| is_canceled | 118898.0 | 0.371352 | 0.483168 | 0.00 | 0.0 | 0.0 | 1.0 | 1.0 |
| lead_time | 118898.0 | 104.311435 | 106.903309 | 0.00 | 18.0 | 69.0 | 161.0 | 737.0 |
| arrival_date_year | 118898.0 | 2016.157656 | 0.707459 | 2015.00 | 2016.0 | 2016.0 | 2017.0 | 2017.0 |
| arrival_date_day_of_month | 118898.0 | 15.800880 | 8.780324 | 1.00 | 8.0 | 16.0 | 23.0 | 31.0 |
| stays_in_weekend_nights | 118898.0 | 0.928897 | 0.996216 | 0.00 | 0.0 | 1.0 | 2.0 | 16.0 |
| stays_in_week_nights | 118898.0 | 2.502145 | 1.900168 | 0.00 | 1.0 | 2.0 | 3.0 | 41.0 |
| adults | 118898.0 | 1.858391 | 0.578576 | 0.00 | 2.0 | 2.0 | 2.0 | 55.0 |
| children | 118898.0 | 0.104207 | 0.399172 | 0.00 | 0.0 | 0.0 | 0.0 | 10.0 |
| is_repeated_guest | 118898.0 | 0.032011 | 0.176029 | 0.00 | 0.0 | 0.0 | 0.0 | 1.0 |
| previous_cancellations | 118898.0 | 0.087142 | 0.845869 | 0.00 | 0.0 | 0.0 | 0.0 | 26.0 |
| previous_bookings_not_canceled | 118898.0 | 0.131634 | 1.484672 | 0.00 | 0.0 | 0.0 | 0.0 | 72.0 |
| booking_changes | 118898.0 | 0.221181 | 0.652785 | 0.00 | 0.0 | 0.0 | 0.0 | 21.0 |
| days_in_waiting_list | 118898.0 | 2.330754 | 17.630452 | 0.00 | 0.0 | 0.0 | 0.0 | 391.0 |
| adr | 118898.0 | 102.003243 | 50.485862 | -6.38 | 70.0 | 95.0 | 126.0 | 5400.0 |
| required_car_parking_spaces | 118898.0 | 0.061885 | 0.244172 | 0.00 | 0.0 | 0.0 | 0.0 | 8.0 |
| total_of_special_requests | 118898.0 | 0.571683 | 0.792678 | 0.00 | 0.0 | 0.0 | 1.0 | 5.0 |

From the above figure it can be seen that there were various improbable values and outliers existing in the data.

Outliers were treated based on the box plot method. For example:



The inter quartile range is a measure of where the “middle fifty” is in a data set. Where a range is a measure of where the beginning and end are in a set, an inter quartile range is a measure of where the bulk of the values lie. That’s why it’s preferred over many other measures of spread

The inter quartile range formula is the first quartile subtracted from the third quartile:

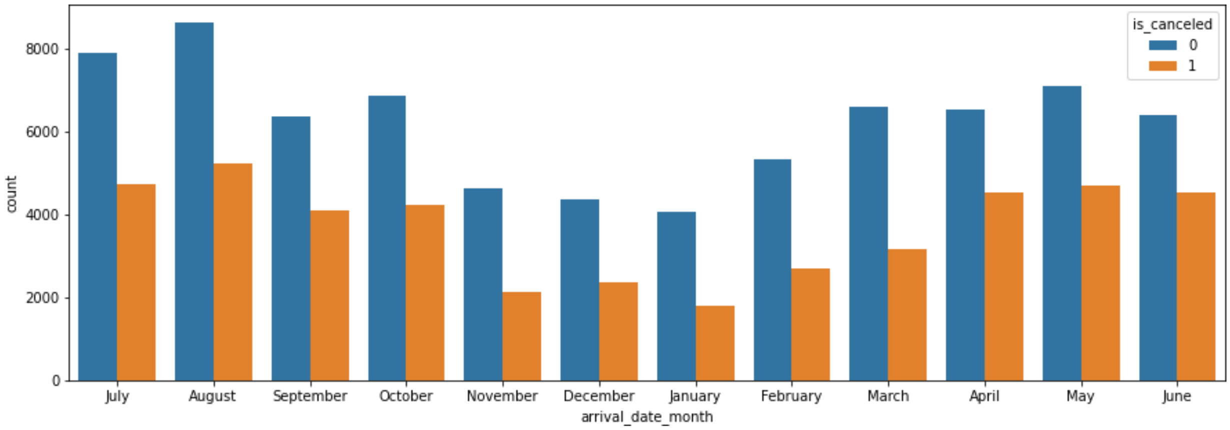
IQR = Q3– Q1.

Upper limit=Q3+1.5*IQR Upper limit=Q1-1.5*IQR

Generally in outlier treatment, any values below the lower limit or above the upper limit were capped or removed based on each feature’s variation. But in case of our dataset, we can note that the removal of outliers may affect the data, as for the booking and cancellations these outliers may contain few valid information which will be necessary for our analysis.
Hence, we proceed forward without removing the outliers.

## 2.4 Analysis between various features with respect to our target variable

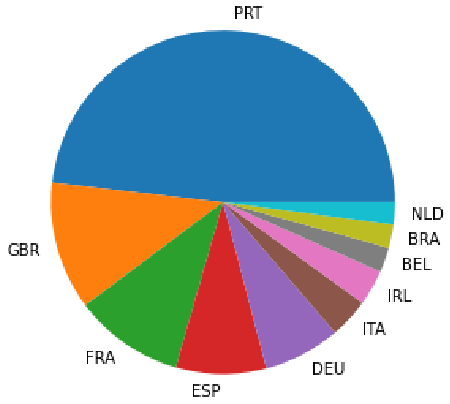
1. most no. of booking/cancellation month wise:



**Inference**: The month which has maximum no. of bookings also has the max no. of cancellation. In our case, August has the maximum no. of bookings and cancellations too, and January being the least in booking and cancellation.

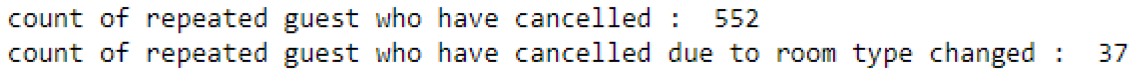
Also, during the months april, may, jun, july and august which are generally the Vacation period which may also be the reason for maximum bookings and cancellation. It can also be noted that during the months Nov - Jan, the no .of bookings is considered very less than other months which may be the result of extreme weather conditions.

2. Top ten countries with max bookings :



**Inference**: From the above data we can notice that the most visited places are Portugal, Great Britain, France, Spain and German y. We can assume that European countries are most frequently visited destination.

3. checking if repeated guests have an effect on cancellation:

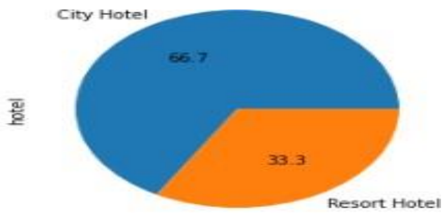


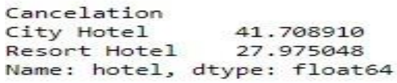
```
count of repeated guest who have cancelled :  552
count of repeated guest who have cancelled due to room type changed :  37
```

**Inference**: Repeated guest may have cancelled their booking either due to business meetings would have got postponed or cancelled or might be due to some personal issues.

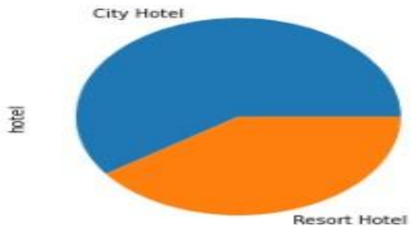
Also we can see that 37 guests have cancelled their booking who's reserved room type and assigned room type are different, which can be the reason for their cancellation.

4. which type of hotel has maximum booking and cancellation :





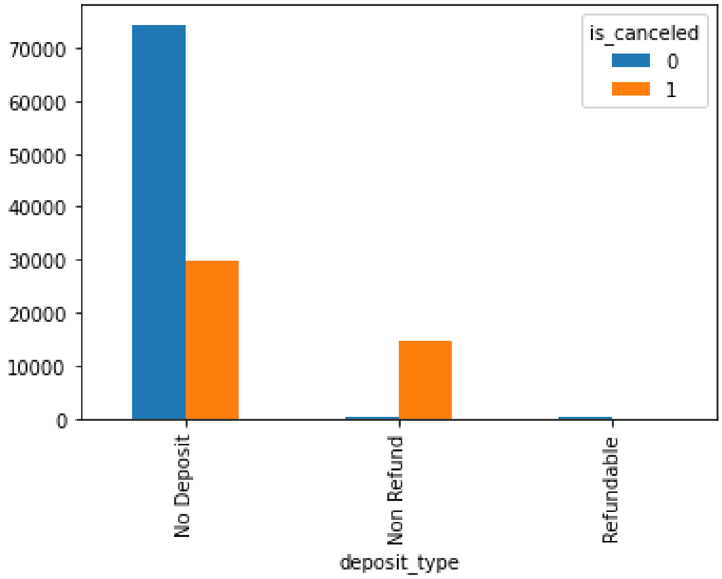
```
Cancelation
City Hotel      41.708910
Resort Hotel    27.975048
Name: hotel, dtype: float64
```



**Inference**: We can see city hotel has the max no.of bookings i.e.66.7% and resort hotel has around 33.3 bookings.

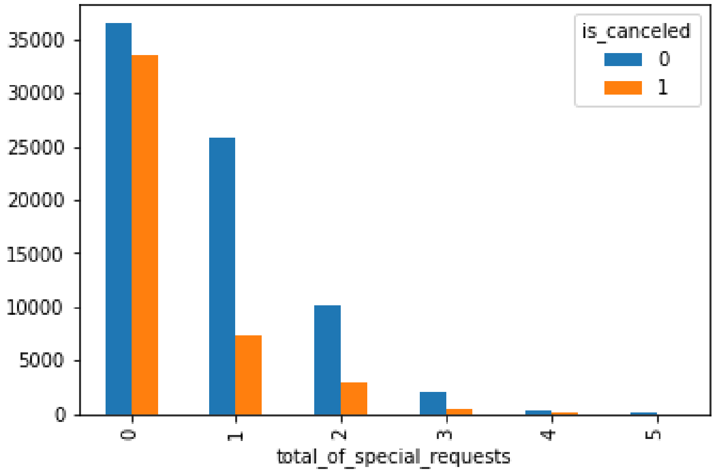
In the case of cancellations, city hotels have 41.7% of cancellation and resort hotels have 28% of cancellation.

5. Analysis between deposit and our target variable is_cancelled:



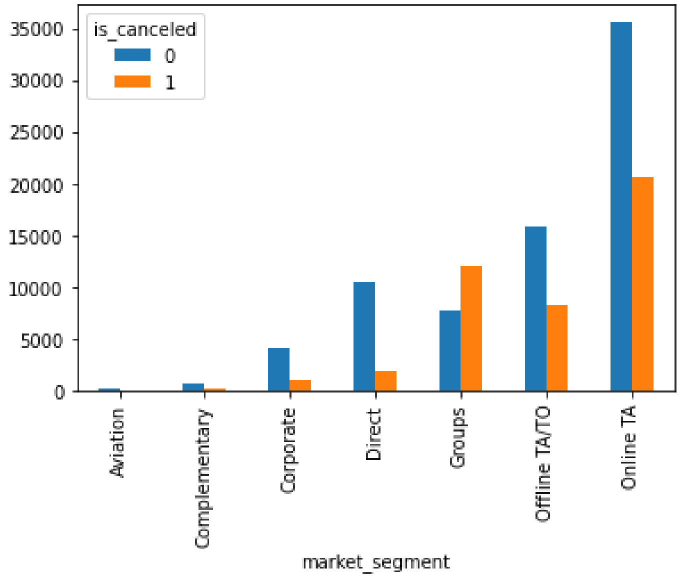
**Inference**: We can find more no. of bookings when deposit is not asked. And the ratio of cancellation is low with respect to booking that are not canceled in - no deposit.

6. Analysis between the no. of special request and cancellation :



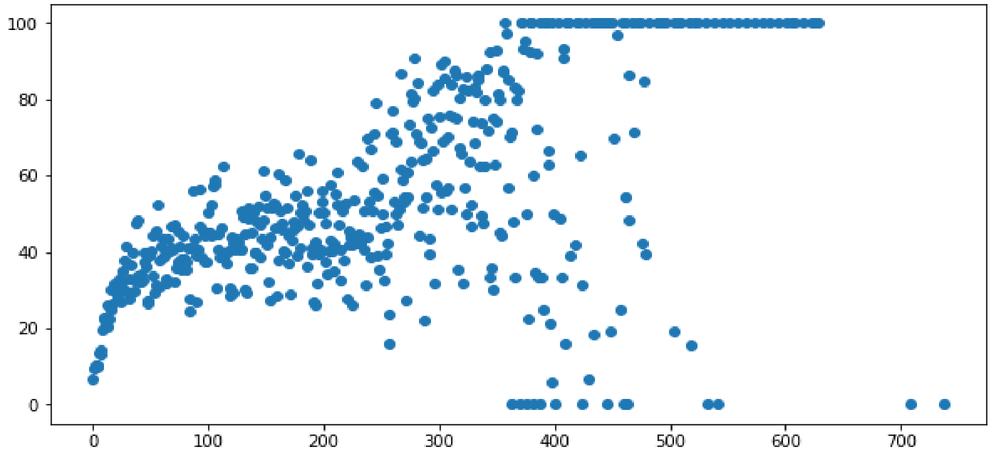
**Inference**: Guest who have requested most no. of special request have least cancellation count, and the those haven't placed any special request have cancelled the maximum.

7. Analysis for which market segment has max no. of cancellation:



**Inference**: Online TA has maximum no of booking compared to other marketing segments and also cancellations. But, in case of Groups, we can notice that cancellations are more than compared to booking which may because each group will have more people and each being considered as an individual booking.

8. Analysis of relation between lead time cancellation rate



**Inference**: Bookings made a few days before the arrival date are rarely canceled, whereas bookings made over one year in advance are canceled very often.

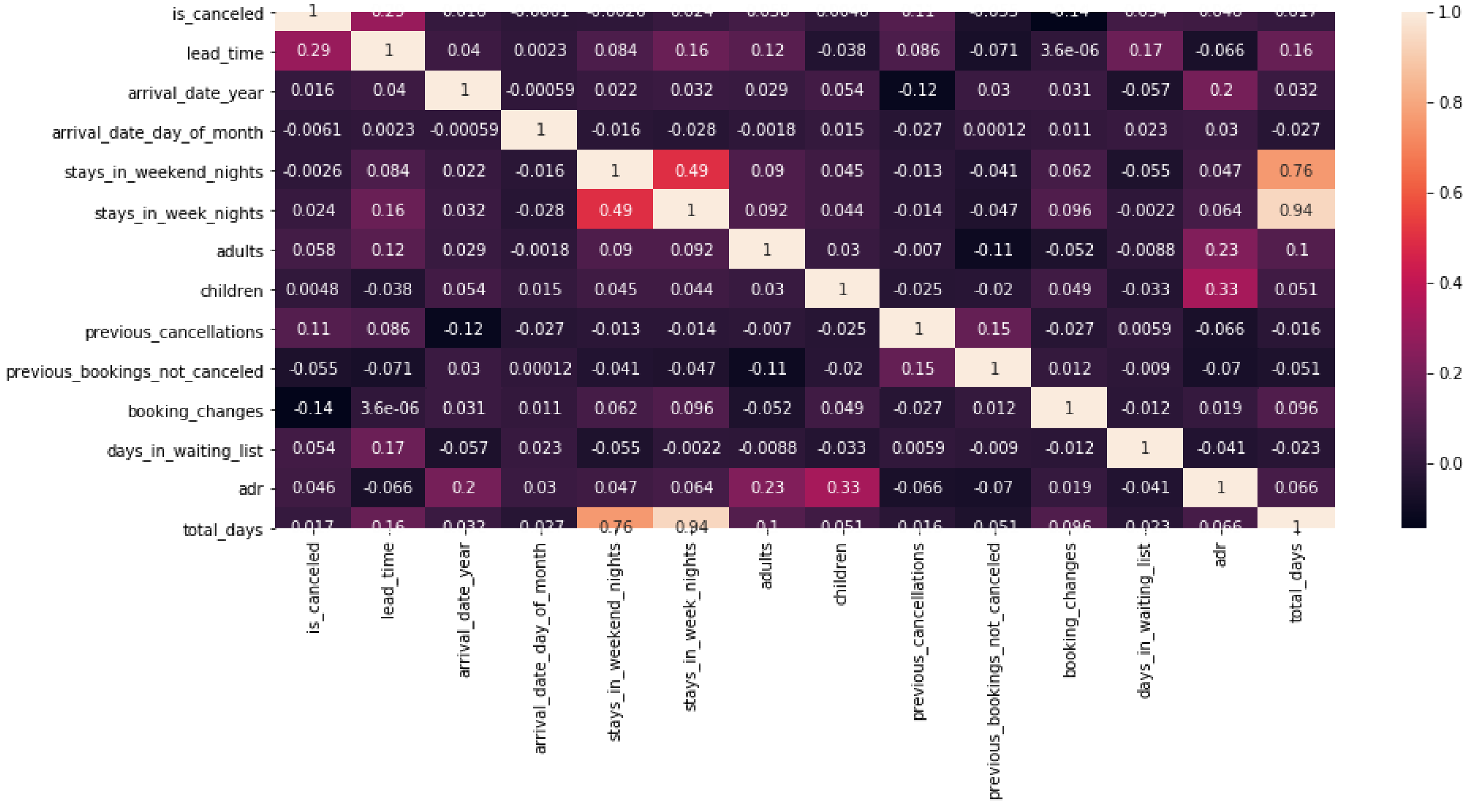
### 2.5 Data Type Conversion:

Before modeling, the data types of the Categorical Columns must be converted into integer/float data types and vice versa in order to make the machine understand. In our dataset we have converted 3 numerical features into categorical.

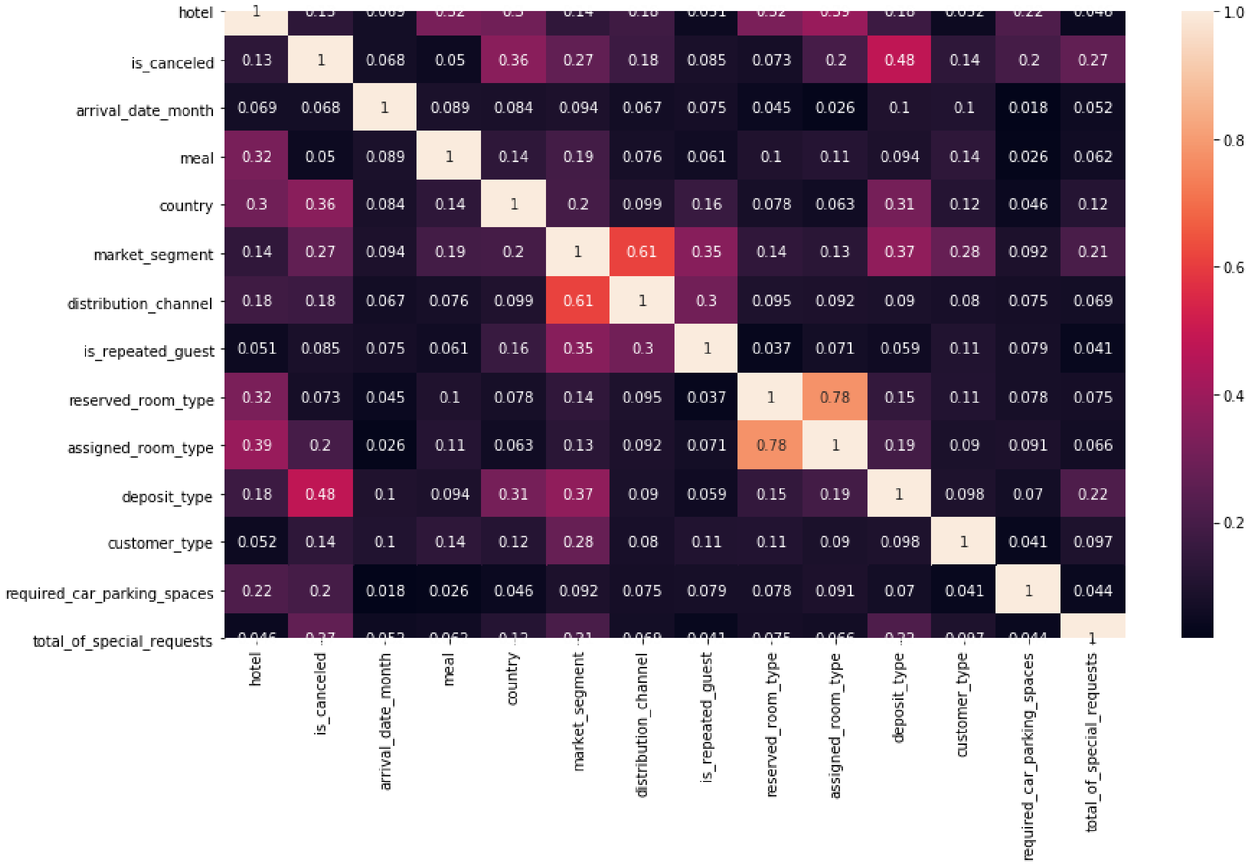
# 3. Correlation Analysis:

We have used Heat Map to find the correlation between the target variable and other features.
We have performed two different correlation analysis, i.e between target variables and all the numerical features and the other analysis is between target variable and the categorical features, for which we made use of **CRAMER'S V RULE.**

**Correlation Analysis between target variable and Numerical features:**



### Correlation Analysis between target variable and Categorical features:

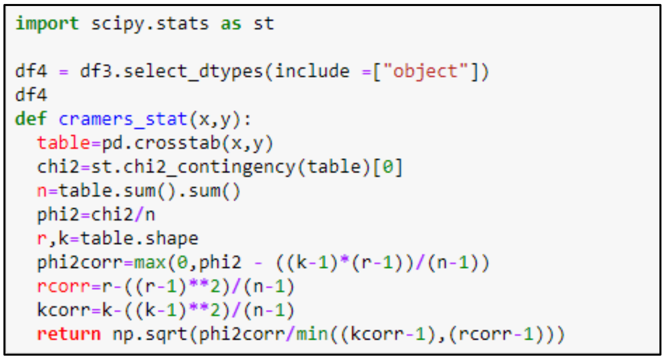


### Cramer’s v Test:

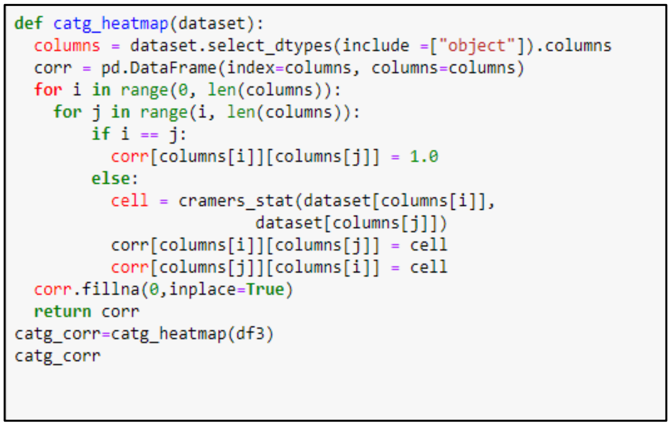
Cramer's V is a way of calculating correlation in tables which have more than 2x2 rows and columns. It is used as post- test to determine strengths of association after chi-square has determined significance. V is calculated by first calculating chi-square, then using the following calculation:

V = SQRT( c2 / (n (k - 1)) )

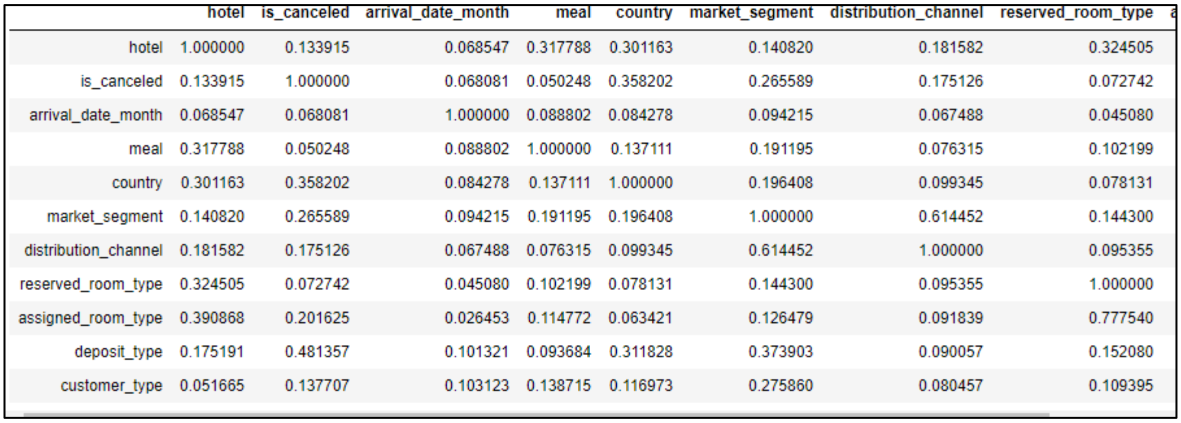
where c2 is chi-square and k is the number of rows or columns in the table



```
import scipy.stats as st

df4 = df3.select_dtypes(include =["object"])
df4
def cramers_stat(x,y):
  table=pd.crosstab(x,y)
  chi2=st.chi2_contingency(table)[0]
  n=table.sum().sum()
  phi2=chi2/n
  r,k=table.shape
  phi2corr=max(0,phi2 - ((k-1)*(r-1))/(n-1))
  rcorr=r-((r-1)**2)/(n-1)
  kcorr=k-((k-1)**2)/(n-1)
  return np.sqrt(phi2corr/min((kcorr-1),(rcorr-1)))
```



```
def catg_heatmap(dataset):
  columns = dataset.select_dtypes(include =["object"]).columns
  corr = pd.DataFrame(index=columns, columns=columns)
  for i in range(0, len(columns)):
    for j in range(i, len(columns)):
        if i == j:
          corr[columns[i]][columns[j]] = 1.0
        else:
          cell = cramers_stat(dataset[columns[i]],
                         dataset[columns[j]])
          corr[columns[i]][columns[j]] = cell
          corr[columns[j]][columns[i]] = cell
  corr.fillna(0,inplace=True)
  return corr
catg_corr=catg_heatmap(df3)
catg_corr
```



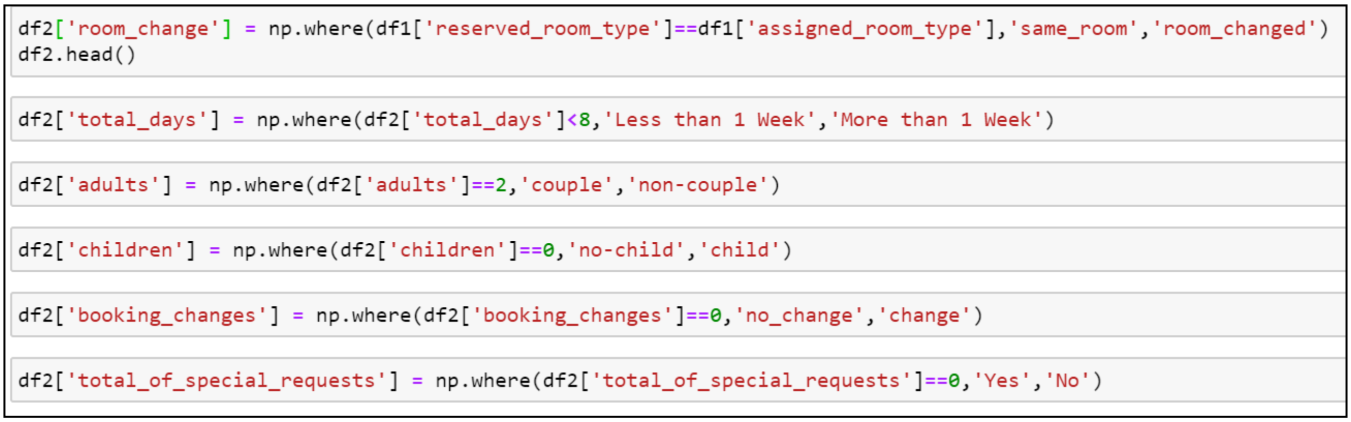
| | hotel | is_canceled | arrival_date_month | meal | country | market_segment | distribution_channel | reserved_room_type | a |
|---|---|---|---|---|---|---|---|---|---|
| hotel | 1.000000 | 0.133915 | 0.068547 | 0.317788 | 0.301163 | 0.140820 | 0.181582 | 0.324505 | |
| is_canceled | 0.133915 | 1.000000 | 0.068081 | 0.050248 | 0.358202 | 0.265589 | 0.175126 | 0.072742 | |
| arrival_date_month | 0.068547 | 0.068081 | 1.000000 | 0.088802 | 0.084278 | 0.094215 | 0.067488 | 0.045080 | |
| meal | 0.317788 | 0.050248 | 0.088802 | 1.000000 | 0.137111 | 0.191195 | 0.076315 | 0.102199 | |
| country | 0.301163 | 0.358202 | 0.084278 | 0.137111 | 1.000000 | 0.196408 | 0.099345 | 0.078131 | |
| market_segment | 0.140820 | 0.265589 | 0.094215 | 0.191195 | 0.196408 | 1.000000 | 0.614452 | 0.144300 | |
| distribution_channel | 0.181582 | 0.175126 | 0.067488 | 0.076315 | 0.099345 | 0.614452 | 1.000000 | 0.095355 | |
| reserved_room_type | 0.324505 | 0.072742 | 0.045080 | 0.102199 | 0.078131 | 0.144300 | 0.095355 | 1.000000 | |
| assigned_room_type | 0.390868 | 0.201625 | 0.026453 | 0.114772 | 0.063421 | 0.126479 | 0.091839 | 0.777540 | |
| deposit_type | 0.175191 | 0.481357 | 0.101321 | 0.093684 | 0.311828 | 0.373903 | 0.090057 | 0.152080 | |
| customer_type | 0.051665 | 0.137707 | 0.103123 | 0.138715 | 0.116973 | 0.275860 | 0.080457 | 0.109395 | |

From the cramer's v test performed we were able to find the correlation between target variable (is_cancelled) and other categorical variables, and plot a heat map respectively. From the heat map we can notice that deposit type, country and market segment have better positive correlation with the target variable and also that there are no negative correlations.

## 4. Feature Engineering:

We have modified our existing data set with respect to the domain knowledge acquired. We have combined few features which were redundant. Since the dependent feature is binary data, we have tried categorizing the independent features depending on the average or the mode, which makes it much easier for better understanding.

For example:



```
df2['room_change'] = np.where(df1['reserved_room_type']==df1['assigned_room_type'],'same_room','room_changed')
df2.head()
```

```
df2['total_days'] = np.where(df2['total_days']<8,'Less than 1 Week','More than 1 Week')
```

```
df2['adults'] = np.where(df2['adults']==2,'couple','non-couple')
```

```
df2['children'] = np.where(df2['children']==0,'no-child','child')
```

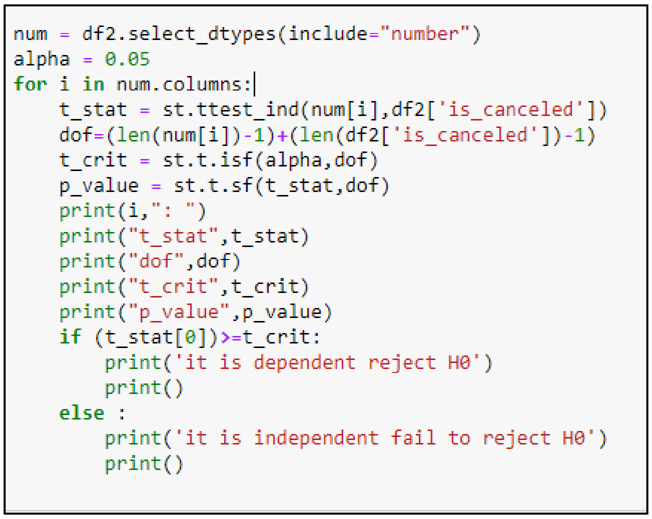
```
df2['booking_changes'] = np.where(df2['booking_changes']==0,'no_change','change')
```

```
df2['total_of_special_requests'] = np.where(df2['total_of_special_requests']==0,'Yes','No')
```

## 5. Statistical Test:

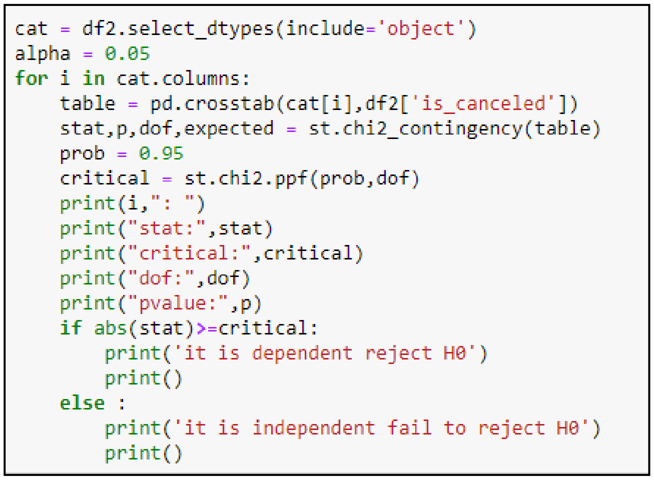
We have performed two samples T-Test between the target variable and the independent numerical features; similarly we have performed Chi-Square test between target variable and independent categorical variables. We have considered confidence interval of 95%.

T-Test:



```
num = df2.select_dtypes(include="number")
alpha = 0.05
for i in num.columns:
    t_stat = st.ttest_ind(num[i],df2['is_canceled'])
    dof=(len(num[i])-1)+(len(df2['is_canceled'])-1)
    t_crit = st.t.isf(alpha,dof)
    p_value = st.t.sf(t_stat,dof)
    print(i,": ")
    print("t_stat",t_stat)
    print("dof",dof)
    print("t_crit",t_crit)
    print("p_value",p_value)
    if (t_stat[0])>=t_crit:
        print('it is dependent reject H0')
        print()
    else :
        print('it is independent fail to reject H0')
        print()
```

Chi-Square Test :



```
cat = df2.select_dtypes(include='object')
alpha = 0.05
for i in cat.columns:
    table = pd.crosstab(cat[i],df2['is_canceled'])
    stat,p,dof,expected = st.chi2_contingency(table)
    prob = 0.95
    critical = st.chi2.ppf(prob,dof)
    print(i,": ")
    print("stat:",stat)
    print("critical:",critical)
    print("dof:",dof)
    print("pvalue:",p)
    if abs(stat)>=critical:
        print('it is dependent reject H0')
        print()
    else :
        print('it is independent fail to reject H0')
        print()
```

# 6. Model Building:

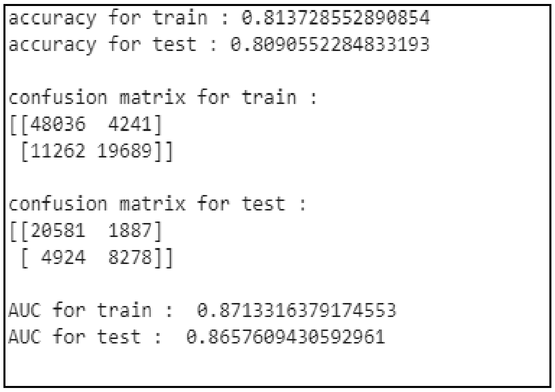
The following are the steps taken before model building:.

1. The country feature was converted as continents by using country converter package.
2. Created dummies for the categorical features

## 6.1 Logistic Regression:

Logistic Regression Models works best in classification conditions (basically 0 and 1) where there is a presence of two categories.

As per our dataset (Capstone project), it possesses a binary classification problem i.e. target variable having 0 and 1 are the values, where 0 represents not cancelled and 1 represents cancelled. It is categorical in nature due to which Performing logistic regression model would be necessary to provide us with good accuracy and prediction. The below is the output of Logistic regression model for our dataset

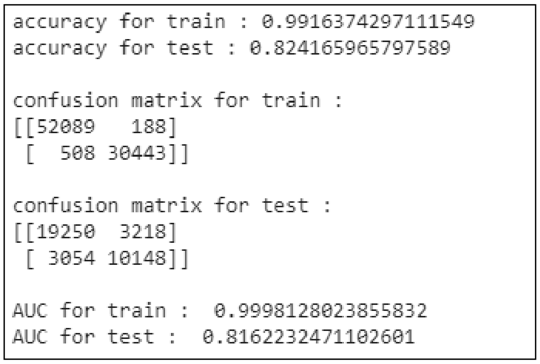


```
accuracy for train : 0.813728552890854
accuracy for test : 0.8090552284833193

confusion matrix for train :
[[48036  4241]
 [11262 19689]]

confusion matrix for test :
[[20581  1887]
 [ 4924  8278]]

AUC for train :  0.8713316379174553
AUC for test :  0.8657609430592961
```

### 6.2 Decision Tree:

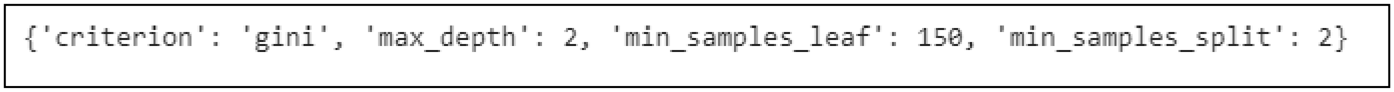
It is the most powerful and popular tool for classification and prediction. A Decision Tree is a flowchart like a tree structure, where the internal node denotes a test on an attribute, each branch represents an outcome of the test and each leaf node (terminal node) holds a class label. At the beginning, the whole dataset is considered as the root. Feature values are preferred to be categorical. If the values are continuous then they are discredited prior to building the model. Records are distributed recursively on the basis of attribute values. Order of placing attributes as root or internal node of the tree is done by using some statistical approach.

The below is the output for our decision tree model:



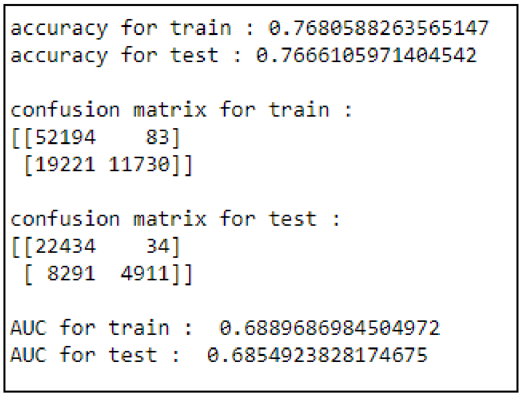
```
accuracy for train : 0.9916374297111549
accuracy for test : 0.824165965797589

confusion matrix for train :
[[52089   188]
 [  508 30443]]

confusion matrix for test :
[[19250  3218]
 [ 3054 10148]]

AUC for train :  0.9998128023855832
AUC for test :  0.8162232471102601
```

Since, we can notice the AUC for train and test are not efficient, we go for Hyper-parameter tuning. Below are the best parameters obtained:



```
{'criterion': 'gini', 'max_depth': 2, 'min_samples_leaf': 150, 'min_samples_split': 2}
```

After performing Hyper parameter tuning, below is the output of our decision tree model :

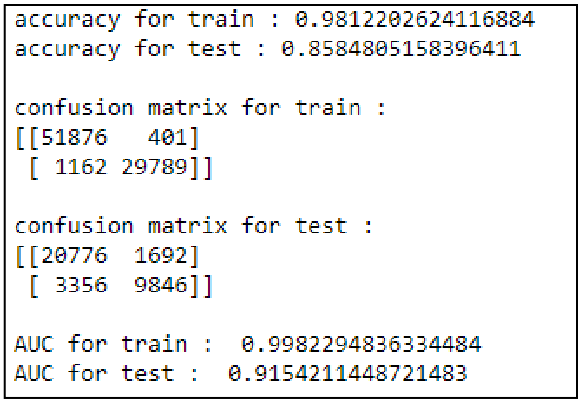


```
accuracy for train : 0.7680588263565147
accuracy for test : 0.7666105971404542

confusion matrix for train :
[[52194    83]
 [19221 11730]]

confusion matrix for test :
[[22434    34]
 [ 8291  4911]]

AUC for train :  0.6889686984504972
AUC for test :  0.6854923828174675
```

### 6.3 Random Forest:

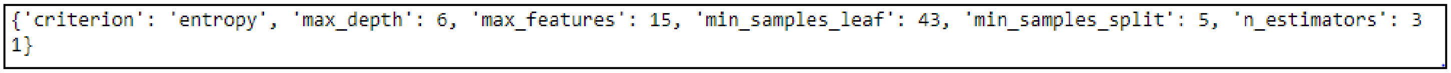
Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble.
Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model prediction.

The below is the output for our random forest model:



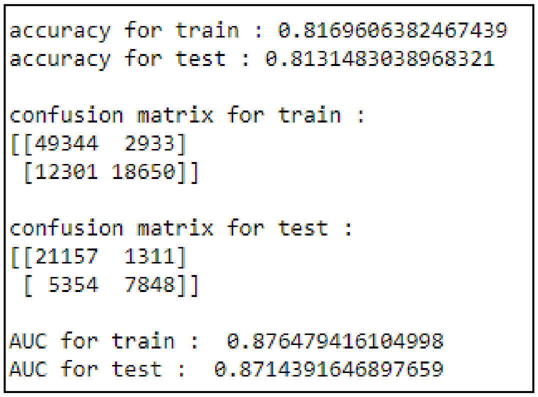
```
accuracy for train : 0.9812202624116884
accuracy for test : 0.8584805158396411

confusion matrix for train :
[[51876   401]
 [ 1162 29789]]

confusion matrix for test :
[[20776  1692]
 [ 3356  9846]]

AUC for train :  0.9982294836334484
AUC for test :  0.9154211448721483
```

Since, we can notice the AUC for train and test are not efficient, we go for Hyper-parameter tuning. Below are the best parameters obtained:



```
{'criterion': 'entropy', 'max_depth': 6, 'max_features': 15, 'min_samples_leaf': 43, 'min_samples_split': 5, 'n_estimators': 3
1}
```

After performing Hyper parameter tuning, below is the output of our random forest model:



```
accuracy for train : 0.8169606382467439
accuracy for test : 0.8131483038968321

confusion matrix for train :
[[49344  2933]
 [12301 18650]]

confusion matrix for test :
[[21157  1311]
 [ 5354  7848]]

AUC for train :  0.876479416104998
AUC for test :  0.8714391646897659
```

### 6.4 Boosting Algorithms:

The following boosting algorithms were taken into consideration for model building :

#### 6.4.1 ADA Boost:

Adaboost helps you **combine multiple “weak classifiers” into a single “strong classifier”.** AdaBoost works by putting more weight on difficult to classify instances and less on those already handled well.

#### 6.4.2 XG Boost:

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. As opposed to the bagging where trees are built parallely, in boosting, the trees are built Sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors.

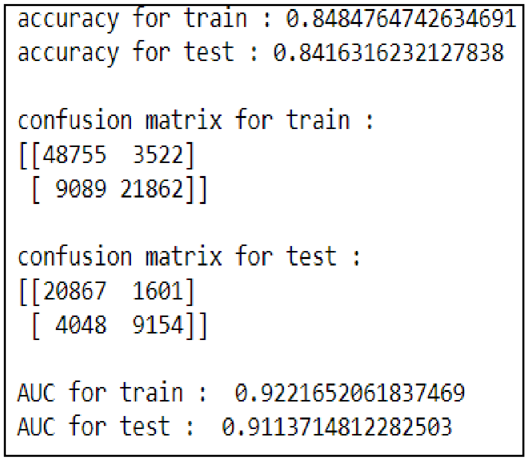
#### 6.4.3 CAT Boost:

CatBoost can be used without any explicit pre-processing to convert categories into numbers. CatBoost converts categorical values into numbers using various statistics on combinations of categorical features and combinations of categorical and numerical features.

### 6.4.4 Light GBM:

**Light GBM grows tree vertically** while other algorithm grows trees horizontally meaning that Light GBM grows tree **leaf-wise** while other algorithm grows level-wise.
It will choose the leaf with max delta loss to grow. When growing the same leaf, Leaf-wise Algorithm can reduce more loss than a level-wise algorithm.

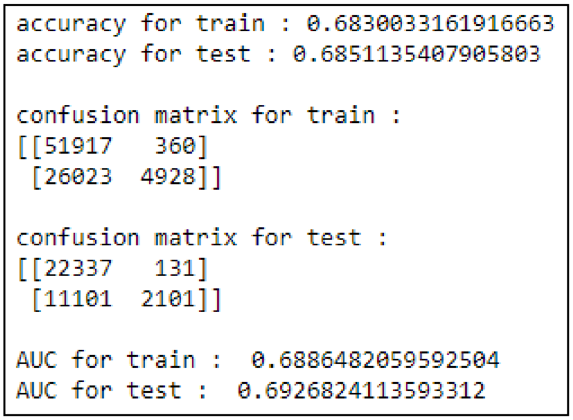
Out of all the above boosting algorithms, Light GBM gave us the better result.



```
accuracy for train : 0.8484764742634691
accuracy for test : 0.8416316232127838

confusion matrix for train :
[[48755  3522]
 [ 9089 21862]]

confusion matrix for test :
[[20867  1601]
 [ 4048  9154]]

AUC for train :  0.9221652061837469
AUC for test :  0.911371481228250З
```

## 6.5 Support Vector Machine (SVM):

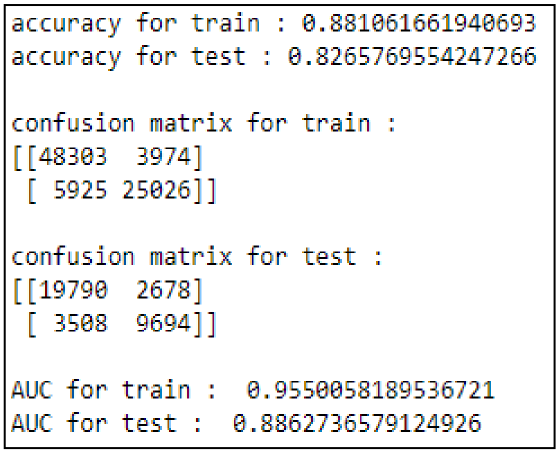
"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. It is mostly used in classification problems.
In SVM we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.



```
accuracy for train : 0.6830033161916663
accuracy for test : 0.6851135407905803

confusion matrix for train :
[[51917   360]
 [26023  4928]]

confusion matrix for test :
[[22337   131]
 [11101  2101]]

AUC for train :  0.6886482059592504
AUC for test :  0.6926824113593312
```

### 6.6 K-NN Algorithm:

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set.



```
accuracy for train : 0.881061661940693
accuracy for test : 0.8265769554247266

confusion matrix for train :
[[48303  3974]
 [ 5925 25026]]

confusion matrix for test :
[[19790  2678]
 [ 3508  9694]]

AUC for train :  0.9550058189536721
AUC for test :  0.8862736579124926
```

## 7. Conclusion:

Analysis on background study, exploratory data analysis, feature engineering were performed and built Logistics Regression model along with other base models such as Decision Tree, Random Forest, Boosting algorithms, SVM and K-NN are done to predict the accuracy of each model in accordance to the target variable.

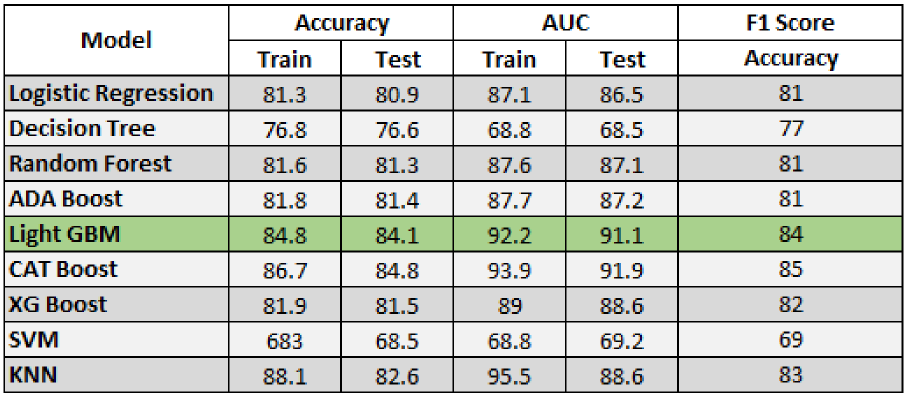
Important findings:

- Europe has the maximum number of bookings as well as cancelations.
- City hotel has the max no.of bookings i.e.66.7% and resort hotel has around 33.3 bookings. In the case of cancellations, city hotels have 41.7% of cancellation and resort hotels have 28% of cancellation.
- August has the maximum no.of bookings and cancellations too, and January being the least in booking and cancellation.
- Bookings made a few days before the arrival date are rarely canceled, where as bookings made over one year in advance are canceled very often.
- Customers availing special request have less probability of cancelling the hotel booking.

Various classification models were performed in order to find the best result in terms of accuracy and prediction. Light GBM classification model provides better result in comparison to other models.

In order to verify Light GBM as our best model, we used other methods as well to evaluate the performance of the model with classification models which are as follows:

- Confusion Matrix
- ROC-AUC Curve
- F1-Score



| Model | Accuracy | | AUC | | F1 Score |
|---|---|---|---|---|---|
| | Train | Test | Train | Test | Accuracy |
| Logistic Regression | 81.3 | 80.9 | 87.1 | 86.5 | 81 |
| Decision Tree | 76.8 | 76.6 | 68.8 | 68.5 | 77 |
| Random Forest | 81.6 | 81.3 | 87.6 | 87.1 | 81 |
| ADA Boost | 81.8 | 81.4 | 87.7 | 87.2 | 81 |
| Light GBM | 84.8 | 84.1 | 92.2 | 91.1 | 84 |
| CAT Boost | 86.7 | 84.8 | 93.9 | 91.9 | 85 |
| XG Boost | 81.9 | 81.5 | 89 | 88.6 | 82 |
| SVM | 683 | 68.5 | 68.8 | 69.2 | 69 |
| KNN | 88.1 | 82.6 | 95.5 | 88.6 | 83 |

## 8. Business Solutions:

The following can be advised to the hotel management as an precautionary step to avoid loss due to cancellations :

- Endorse customers to avail special requests provided by the hotel which may reduce the chances of cancellation.
- Management can make sure the repeated Guests are given the same room reserved while booking.
- Provide better offers for direct booking, as it sees considerably less cancellations.
- The hotel management can come up with confirmation check calls with the customer prior to the check in date.
- Reduce the price of the hotel room during the months where least no. of bookings have been made.
- The management can have an eye for the bookings made under non - refundable type, as they have high chances of cancellation.