In [71]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings("ignore")
```

In [72]:
```python
df=pd.read_csv(r"C:\Users\MARK 1.0\OneDrive\Desktop\Ml Practice\hotel_booking_data (1).csv")#load dataset
df
```

Out[72]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	...	customer_type	adr
0	Resort Hotel	0	342	2015	July	27	1	0	0	2	...	Transient	0.00
1	Resort Hotel	0	737	2015	July	27	1	0	0	2	...	Transient	0.00
2	Resort Hotel	0	7	2015	July	27	1	0	1	1	...	Transient	75.00
3	Resort Hotel	0	13	2015	July	27	1	0	1	1	...	Transient	75.00
4	Resort Hotel	0	14	2015	July	27	1	0	2	2	...	Transient	98.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
119385	City Hotel	0	23	2017	August	35	30	2	5	2	...	Transient	96.14
119386	City Hotel	0	102	2017	August	35	31	2	5	3	...	Transient	225.43
119387	City Hotel	0	34	2017	August	35	31	2	5	2	...	Transient	157.71
119388	City Hotel	0	109	2017	August	35	31	2	5	2	...	Transient	104.40
119389	City Hotel	0	205	2017	August	35	29	2	7	2	...	Transient	151.20

119390 rows × 36 columns

In [73]:
```python
df.shape
```

Out[73]:
```
(119390, 36)
```

In [74]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 36 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
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  object
 32  name                            119390 non-null  object
 33  email                           119390 non-null  object
 34  phone-number                    119390 non-null  object
 35  credit_card                     119390 non-null  object
dtypes: float64(4), int64(16), object(16)
memory usage: 32.8+ MB
```

In [75]:
```python
df.describe()
```

Out[75]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeated_gues
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119386.000000	119390.000000	119390.00000
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	2.500302	1.856403	0.103890	0.007949	0.03191
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	1.908286	0.579261	0.398561	0.097436	0.17576
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000	2.000000	0.000000	0.000000	0.00000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000	2.000000	0.000000	0.000000	0.00000
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	3.000000	2.000000	0.000000	0.000000	0.00000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000	55.000000	10.000000	10.000000	1.00000

In [76]:
```python
df.isnull().sum()
```

Out[76]:
```
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
name                                 0
email                                0
phone-number                         0
credit_card                          0
dtype: int64
```

In [77]:
```python
df.drop("company",axis=1,inplace=True)
```

In [78]:
```python
df.drop("agent",axis=1,inplace=True)
```

In [79]:
```python
df.dropna(subset=["children","country"],inplace=True)
```

In [80]:
```python
df.duplicated().sum()
```

Out[80]:
```
0
```

In [81]:
```python
df.isnull().sum()
```

Out[81]:
```
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                          0
babies                            0
meal                              0
country                           0
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
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
name                              0
email                             0
phone-number                      0
credit_card                       0
dtype: int64
```

In [82]:
```python
df.drop("is_repeated_guest",axis=1,inplace=True)
```

In [83]:
```python
df.drop("babies",axis=1,inplace=True)
```

In [84]:
```python
df.drop("required_car_parking_spaces",axis=1,inplace=True)
```

In [85]:
```python
cat=df.select_dtypes(object).columns
cat
```

Out[85]:
```
Index(['hotel', 'arrival_date_month', 'meal', 'country', 'market_segment',
       'distribution_channel', 'reserved_room_type', 'assigned_room_type',
       'deposit_type', 'customer_type', 'reservation_status',
       'reservation_status_date', 'name', 'email', 'phone-number',
       'credit_card'],
      dtype='object')
```

In [86]:
```python
from sklearn.preprocessing import OrdinalEncoder
oe=OrdinalEncoder()
df[cat]=oe.fit_transform(df[cat])
```

In [87]:
```python
df
```

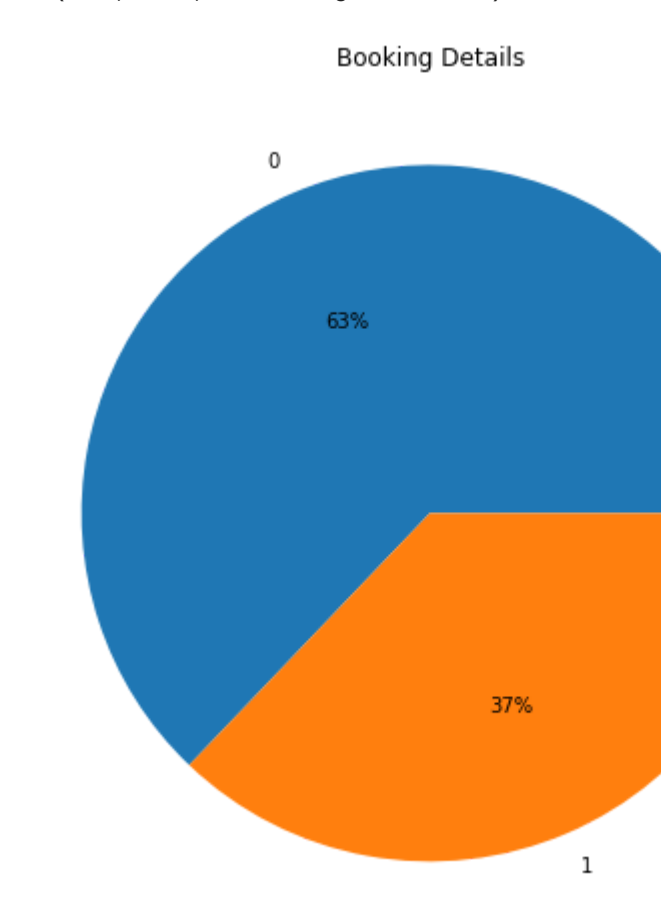
Out[87]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	...	days_in_waiting_list
0	1.0	0	342	2015	5.0	27	1	0	0	2	...	0
1	1.0	0	737	2015	5.0	27	1	0	0	2	...	0
2	1.0	0	7	2015	5.0	27	1	0	1	1	...	0
3	1.0	0	13	2015	5.0	27	1	0	1	1	...	0
4	1.0	0	14	2015	5.0	27	1	0	2	2	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
119385	0.0	0	23	2017	1.0	35	30	2	5	2	...	0
119386	0.0	0	102	2017	1.0	35	31	2	5	3	...	0
119387	0.0	0	34	2017	1.0	35	31	2	5	2	...	0
119388	0.0	0	109	2017	1.0	35	31	2	5	2	...	0
119389	0.0	0	205	2017	1.0	35	29	2	7	2	...	0

118898 rows × 31 columns

In [88]:
```python
d = df['is_canceled'].value_counts()
plt.figure(figsize=(8,8))
p = plt.pie(d, labels=d.index, autopct="%.0f%%")
plt.title("Booking Details")
```

Out[88]:
```
Text(0.5, 1.0, 'Booking Details')
```



In [89]:
```python
from scipy.stats import zscore
z=np.abs(zscore(df))
z
```

Out[89]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	...	days_in_waiting_list
0	1.415195	0.76858	2.223407	1.636365	0.138739	0.012256	1.685694	0.932429	1.316808	0.244755	...	0.132201
1	1.415195	0.76858	5.918350	1.636365	0.138739	0.012256	1.685694	0.932429	1.316808	0.244755	...	0.132201
2	1.415195	0.76858	0.910279	1.636365	0.138739	0.012256	1.685694	0.932429	0.790536	1.483635	...	0.132201
3	1.415195	0.76858	0.854153	1.636365	0.138739	0.012256	1.685694	0.932429	0.790536	1.483635	...	0.132201
4	1.415195	0.76858	0.844799	1.636365	0.138739	0.012256	1.685694	0.932429	0.264264	0.244755	...	0.132201
...	...	...	...	...	...	...	...	...	...	...	...	...
119385	0.706616	0.76858	0.760610	1.190666	1.270292	0.576416	1.617159	1.075176	1.314550	0.244755	...	0.132201
119386	0.706616	0.76858	0.021622	1.190666	1.270292	0.576416	1.731050	1.075176	1.314550	1.973145	...	0.132201
119387	0.706616	0.76858	0.657713	1.190666	1.270292	0.576416	1.731050	1.075176	1.314550	0.244755	...	0.132201
119388	0.706616	0.76858	0.043858	1.190666	1.270292	0.576416	1.731050	1.075176	1.314550	0.244755	...	0.132201
119389	0.706616	0.76858	0.941870	1.190666	1.270292	0.576416	1.503267	1.075176	2.367093	0.244755	...	0.132201

118898 rows × 31 columns

In [90]:
```python
newdf=df[(z<3).all(axis=1)]
newdf
```

Out[90]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	...	days_in_waiting_list
2	1.0	0	7	2015	5.0	27	1	0	1	1	...	0
3	1.0	0	13	2015	5.0	27	1	0	1	1	...	0
4	1.0	0	14	2015	5.0	27	1	0	2	2	...	0
5	1.0	0	14	2015	5.0	27	1	0	2	2	...	0
6	1.0	0	0	2015	5.0	27	1	0	2	2	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
119384	0.0	0	21	2017	1.0	35	30	2	5	2	...	0
119385	0.0	0	23	2017	1.0	35	30	2	5	2	...	0
119386	0.0	0	102	2017	1.0	35	31	2	5	3	...	0
119388	0.0	0	109	2017	1.0	35	31	2	5	2	...	0
119389	0.0	0	205	2017	1.0	35	29	2	7	2	...	0

97732 rows × 31 columns

In [91]:
```python
#sns.pairplot(newdf)
```

In [92]:
```python
x=newdf.drop(["is_canceled"],axis=1)
x
```

Out[92]:

	hotel	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	...	days_in_waiting_list
2	1.0	7	2015	5.0	27	1	0	1	1	0.0	...	0
3	1.0	13	2015	5.0	27	1	0	1	1	0.0	...	0
4	1.0	14	2015	5.0	27	1	0	2	2	0.0	...	0
5	1.0	14	2015	5.0	27	1	0	2	2	0.0	...	0
6	1.0	0	2015	5.0	27	1	0	2	2	0.0	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...
119384	0.0	21	2017	1.0	35	30	2	5	2	0.0	...	0
119385	0.0	23	2017	1.0	35	30	2	5	2	0.0	...	0
119386	0.0	102	2017	1.0	35	31	2	5	3	0.0	...	0
119388	0.0	109	2017	1.0	35	31	2	5	2	0.0	...	0
119389	0.0	205	2017	1.0	35	29	2	7	2	0.0	...	0

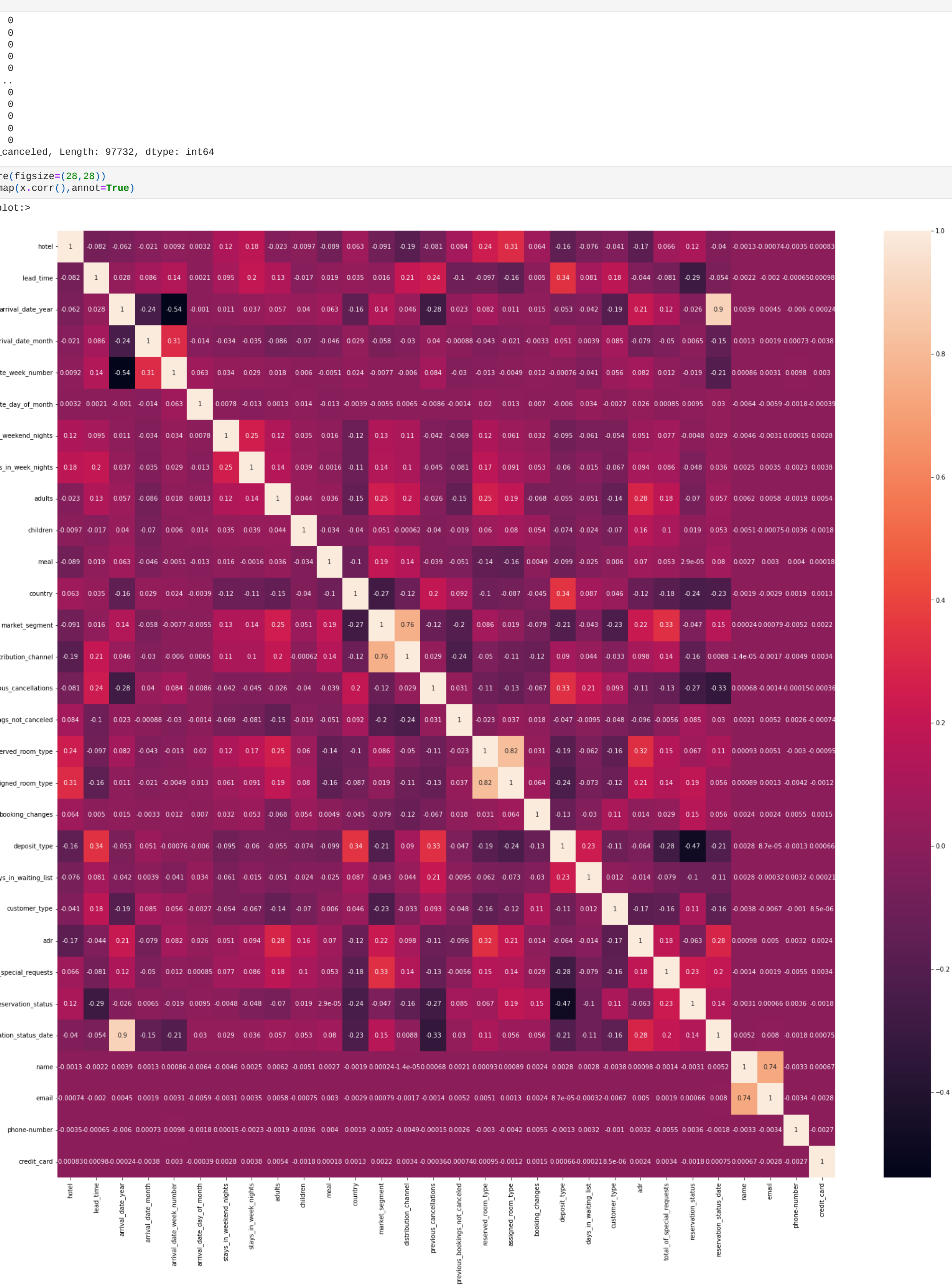
97732 rows × 30 columns

In [93]:
```python
y=newdf.iloc[:,1]
y
```

Out[93]:
```
2         0
3         0
4         0
5         0
6         0
         ..
119384    0
119385    0
119386    0
119388    0
119389    0
Name: is_canceled, Length: 97732, dtype: int64
```

In [94]:
```python
plt.figure(figsize=(28,28))
sns.heatmap(x.corr(),annot=True)
```

Out[94]:
```
<AxesSubplot:>
```



In [95]:
```python
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
from sklearn.naive_bayes import BernoulliNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
```

In [96]:
```python
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=1)
```

In [97]:
```python
#logistic
```

In [98]:
```python
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(xtrain,ytrain)
ypred=logreg.predict(xtest)
```

In [99]:
```python
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
ac=accuracy_score(ytest,ypred)
cr=classification_report(ytest,ypred)
print(f"Accuracy:{ac}\n{cr}")
```

```
Accuracy:0.7060027285129604
              precision    recall  f1-score   support

           0       0.72      0.86      0.78     18268
           1       0.66      0.45      0.54     11052

    accuracy                           0.71     29320
   macro avg       0.69      0.66      0.66     29320
weighted avg       0.70      0.71      0.69     29320
```

In [100…
```python
train=logreg.score(xtrain,ytrain)
test=logreg.score(xtest,ytest)
print(f"Training Accuracy:{train}\nTesting Accuracy{test}")
```

```
Training Accuracy:0.706045722972578
Testing Accuracy0.7060027285129604
```

In [101…
```python
#random forest
```

In [102…
```python
rf=RandomForestClassifier(n_estimators=61,max_depth=8)
rf.fit(xtrain,ytrain)
```

Out[102]:
```
RandomForestClassifier
RandomForestClassifier(max_depth=8, n_estimators=61)
```

In [103…
```python
ypred=rf.predict(xtest)
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     18268
           1       1.00      0.98      0.99     11052

    accuracy                           0.99     29320
   macro avg       0.99      0.99      0.99     29320
weighted avg       0.99      0.99      0.99     29320
```

In [104…
```python
train=rf.score(xtrain,ytrain)
test=rf.score(xtest,ytest)
print(f"Training accuracy:{train}\nTesting accuracy:{test}")
```

```
Training accuracy:0.9935976144536046
Testing accuracy:0.9927694406548431
```

In [105…
```python
#Decision tree
```

In [106…
```python
dtc2=DecisionTreeClassifier(max_depth=8,min_samples_leaf=29)
dtc2.fit(xtrain,ytrain)
ypred=dtc2.predict(xtest)
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18268
           1       1.00      1.00      1.00     11052

    accuracy                           1.00     29320
   macro avg       1.00      1.00      1.00     29320
weighted avg       1.00      1.00      1.00     29320
```

In [107…
```python
train=dtc2.score(xtrain,ytrain)
test=dtc2.score(xtest,ytest)
print(f"Training accuracy:{train}\nTesting accuracy:{test}")
```

```
Training accuracy:1.0
Testing accuracy:1.0
```

In [108…
```python
#gradient boosting
```

In [109…
```python
gb=GradientBoostingClassifier(n_estimators=90, learning_rate=0.3,max_depth=1, random_state=1)
```

In [110…
```python
gb.fit(xtrain,ytrain)
gb.score(xtest,ytest)
ypred=gb.predict(xtest)
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18268
           1       1.00      1.00      1.00     11052

    accuracy                           1.00     29320
   macro avg       1.00      1.00      1.00     29320
weighted avg       1.00      1.00      1.00     29320
```

In [111…
```python
train=gb.score(xtrain,ytrain)
test=gb.score(xtest,ytest)
print(f"Training accuracy:{train}\nTesting accuracy:{test}")
```

```
Training accuracy:1.0
Testing accuracy:1.0
```

In [112…
```python
#Naive bayes
```

In [113…
```python
gnb=BernoulliNB()
gnb.fit(xtrain,ytrain)
ypred=gnb.predict(xtest)
```

In [114…
```python
cr=classification_report(ytest,ypred)
acc=accuracy_score(ytest,ypred)
print(f"Accuracy score:{acc}\n{cr}")
```

```
Accuracy score:0.9889836289222373
              precision    recall  f1-score   support

           0       0.98      1.00      0.99     18268
           1       1.00      0.97      0.99     11052

    accuracy                           0.99     29320
   macro avg       0.99      0.99      0.99     29320
weighted avg       0.99      0.99      0.99     29320
```

In [115…
```python
train=gnb.score(xtrain,ytrain)
test=gnb.score(xtest,ytest)
print(f"Training score:{train}\nTesting score:{test}")
```

```
Training score:0.9894901479272642
Testing score:0.9889836289222373
```

In [116…
```python
# xgb
```

In [117…
```python
from xgboost import XGBClassifier
```

In [118…
```python
xgb=XGBClassifier(n_estimators=55,reg_alpha=2)
```

In [119…
```python
xgb.fit(xtrain,ytrain)
ypred=xgb.predict(xtest)
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18268
           1       1.00      1.00      1.00     11052

    accuracy                           1.00     29320
   macro avg       1.00      1.00      1.00     29320
weighted avg       1.00      1.00      1.00     29320
```

In [120…
```python
train=xgb.score(xtrain,ytrain)
test=xgb.score(xtest,ytest)
print(f"Training Accuracy:{train}\nTesting Accuracy{test}")
```

```
Training Accuracy:1.0
Testing Accuracy1.0
```

In [121…
```python
# AdaBooster
```

In [122…
```python
abc=AdaBoostClassifier(n_estimators=100,random_state=1)
abc.fit(xtrain,ytrain)
ypred=abc.predict(xtest)
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     18268
           1       1.00      1.00      1.00     11052

    accuracy                           1.00     29320
   macro avg       1.00      1.00      1.00     29320
weighted avg       1.00      1.00      1.00     29320
```

In [123…
```python
train=abc.score(xtrain,ytrain)
test=abc.score(xtest,ytest)
print(f"Training Accuracy:{train}\nTesting Accuracy{test}")
```

```
Training Accuracy:1.0
Testing Accuracy1.0
```