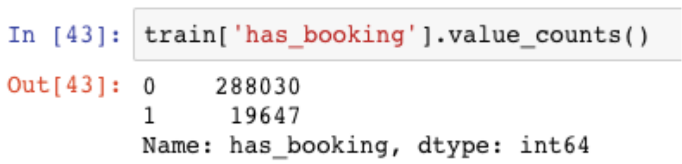
Additional Questions

Question 1, Part A: What makes the classification problem difficult in this task?
The main problem faced during solving the problem was:

Solution 1: Part A:

Our dependent variable i.e. **has_booking** was unevenly distributed as shown in Figure 1 and Figure 2

Figure 1: Frequency of "has_booking" (Yes/No)



```
In [43]: train['has_booking'].value_counts()

Out[43]: 0    288030
         1     19647
         Name: has_booking, dtype: int64
```

Figure 2: Absolute frequency of "has_booking"

If we observe, in order to build an efficient classifier, we need properly to build a proper training dataset out of the given training set.

Question 1, Part B: How do you handle that?
Solution 1, Part B:

In order to tackle the above problem, I followed the following steps:

1. Extracted the data with “has_booking” =1 from given “case_study_bookings_train.csv” and named the dataframe as *df1*.
2. Extracted the data with “has_booking” =0 from given “case_study_bookings_train.csv” and named the dataframe as *df0*.
    a. Decided the ratio in which df0 and df1 should be to prepare the final dataset. Using hit and trials I analyzed the following table:

Classifier used here is Random Forest

Df1 : Df0 (Df1 ratio Df2)	Accuracy on test dataset produced by combining Df1 and Df2	Accuracy on random dataset taken training data where : #rows = #rows_in_target_data
1 : 1	0.675	0.632
1 : 2	0.704	0.858
1 : 3	0.753	0.883
1 : 3.2	0.764	0.905
1 : 3.3	0.767	0.917
1 : 3.4	0.775	0.931
1 : 3.5	0.782	0.931
1 : 3.6	0.780	0.931
1 : 3.8	Not required	Not required

    b. In order to avoid over-fitting. I finalized to resume with ***df1 : df0 = 1 : 3.4.***

3. Combined both the dataframe i.e. df0 and df1 and named it as “*data_equal*”.
4. Randomized the rows in the “*data_equal*” and constructed the final dataset to train my classifier

Question 2: Evaluate and compare at least 3 classification algorithm for this task.
Solution 2:

Chosen classification algorithms:

1. Random Forest
2. Decision Tree
3. Naïve Bayes
4. Logistic Regression

Predefined parameters:
1. Test_size = 25%
2. Train_size = 75%
3. As decided above Df1: Df0 = 1 : 3.4

Table for classification algorithm comparison according to the dataset provided is as follows:

Parameters	**Random Forest**	**Decision Tree**	**Naive Bayes**	**Logistic Regression**
Accuracy on test_set ACC = (TP+TN)/(TP+FP+FN+TN)	0.774	0.772	0.596	0.773
Predicted accuracy on target set ACC = (TP+TN)/(TP+FP+FN+TN)	0.930	0.920	0.568	0.938
Sensitivity TPR = TP/(TP+FN)	0.986	0.972	0.553	1.0
Specificity TNR = TN/(TN+FP)	0.058	0.087	0.744	0.0
Precision or Positive predictive value PPV = TP/(TP+FP)	0.779	0.784	0.881	0.773
Negative predictive value NPV = TN/(TN+FN)	0.565	0.476	0.326	NAN
False positive rate FPR = FP/(FP+TN)	0.941	0.912	0.255	1
False negative rate FNR = FN/(TP+FN)	0.013	0.027	0.446	0
False discovery rate FDR = FP/(TP+FP)	0.220	0.215	0.118	0.226

On the basis of above data, I can say, there is close competition between Random Forest and Decision Tree but Random forest is little better.
Naïve Bayes has very low accuracy (59.6%) and other parameters reveals it is not performing good at classification task.
Logistic Regression is performing outstanding in case of accuracy on both the datasets. But, if we look at the confusion matrix below:

```
Confusion matrix:

array([[16713,     0],
       [ 4899,     0]])
```

Figure 3: Confusion matrix for Logistic Regression

It reveals that logistic regression, is not able to predict any True Negative (TN) nor any False Negative(FN) value. Sensitivity is 1. Specificity is 0. Hence, we cannot categorize it as a good classifier for this dataset.

Hence, **Random Forest** is best classification algorithm according to me for this dataset.

Question 3: Propose at least three 3 features that are significant to predict booking
Solution 3:

As both the dependent and independent variables are categorical so in order to select the significant features we will go for **Chi-Square Test**

Result of Chi-Square test are as follows:

```
Chi-squared Statistic between has_booking and referer_code : 4365.767623826621
Degrees of Freedom between has_booking and referer_code : 10

Chi-squared Statistic between has_booking and is_app : 216.89892149487767
Degrees of Freedom between has_booking and is_app : 1

Chi-squared Statistic between has_booking and agent_id : 2461.0311853484927
Degrees of Freedom between has_booking and agent_id : 14

Chi-squared Statistic between has_booking and traffic_type : 6506.874658406188
Degrees of Freedom between has_booking and traffic_type : 6
```

Figure 4: Chi-Square Statistics and Degree-of-Freedom

Now, we will look at Chi-Square Distribution table to check cut off for a p-value of 0.05.

	Probability less than the critical value				
ν	0.90	0.95	0.975	0.99	0.999
1	2.706	3.841	5.024	6.635	10.828
2	4.605	5.991	7.378	9.210	13.816
3	6.251	7.815	9.348	11.345	16.266
4	7.779	9.488	11.143	13.277	18.467
5	9.236	11.070	12.833	15.086	20.515
6	10.645	12.592	14.449	16.812	22.458
7	12.017	14.067	16.013	18.475	24.322
8	13.362	15.507	17.535	20.090	26.125
9	14.684	16.919	19.023	21.666	27.877
10	15.987	18.307	20.483	23.209	29.588
11	17.275	19.675	21.920	24.725	31.264
12	18.549	21.026	23.337	26.217	32.910
13	19.812	22.362	24.736	27.688	34.528
14	21.064	23.685	26.119	29.141	36.123
15	22.307	24.996	27.488	30.578	37.697
16	23.542	26.296	28.845	32.000	39.252
17	24.769	27.587	30.191	33.409	40.790
18	25.989	28.869	31.526	34.805	42.312
19	27.204	30.144	32.852	36.191	43.820
20	28.412	31.410	34.170	37.566	45.315
21	29.615	32.671	35.479	38.932	46.797
22	30.813	33.924	36.781	40.289	48.268
23	32.007	35.172	38.076	41.638	49.728
24	33.196	36.415	39.364	42.980	51.179
25	34.382	37.652	40.646	44.314	52.620
26	35.563	38.885	41.923	45.642	54.052
27	36.741	40.113	43.195	46.963	55.476
28	37.916	41.337	44.461	48.278	56.892
29	39.087	42.557	45.722	49.588	58.301
30	40.256	43.773	46.979	50.892	59.703
31	41.422	44.985	48.232	52.191	61.098
32	42.585	46.194	49.480	53.486	62.487
33	43.745	47.400	50.725	54.776	63.870
34	44.903	48.602	51.966	56.061	65.247

Figure 5: Chi-Square Distribution table with circled values according to degree of freedoms at 5% Significance

Hypothesis in case of Chi square test:

Null hypothesis: Assumes that there is no association between the two variables.
Alternative hypothesis: Assumes that there is an association between the two variables.

As we can see our Chi-Square statistics value is more than p-value at particular degree of freedom. So we can say **all the evidence are against null hypothesis.**

Hence, all the variables i.e. referer_code, is_app, agent_id, traffic_type **are significant to predict** has_booking**.**

By looking at Chi-Square statistics, here is the strength of association is decreasing order:
1. traffic_type (Most Significant)
2. referer_code
3. agent_id
4. is_app

Question 4: We can spot a very significant action type. What might this action refer to?
Solution 4:

The most significant action type must be the one which yield maximum amount of booking.

Steps taken:

1. Merge the training set of booking and action based on same ‘ymd','user_id' and 'session_id' and prepare a new dataframe.

2. Generate the contingency table from fields: ‘has_booking’ and ‘action_id’.

3. Find the ‘action_id’ against which there are maximum count of ‘has_booking’=1

Result: 2142

Extracting the 'action_id' with maximum #has_booking

```
In [43]: main_table[1].argmax()

/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: FutureWarning: 'argmax' is deprecated, use 'idxmax' i
nstead. The behavior of 'argmax'
will be corrected to return the positional maximum in the future.
Use 'series.values.argmax' to get the position of the maximum now.
  """Entry point for launching an IPython kernel.

Out[43]: 2142
```

Figure 6: Snapshot of solution from Significant_Action.ipynb

According to me action_id: 2142, should belong to **“HOTELS”.**