Beyond Predictive Models: The Causal Story Behind Hotel Booking Cancellations

Understanding why Hotel Bookings are cancelled using Microsoft DoWhy in Python.

Siddharth Dixit Sep 30 · 12 min read

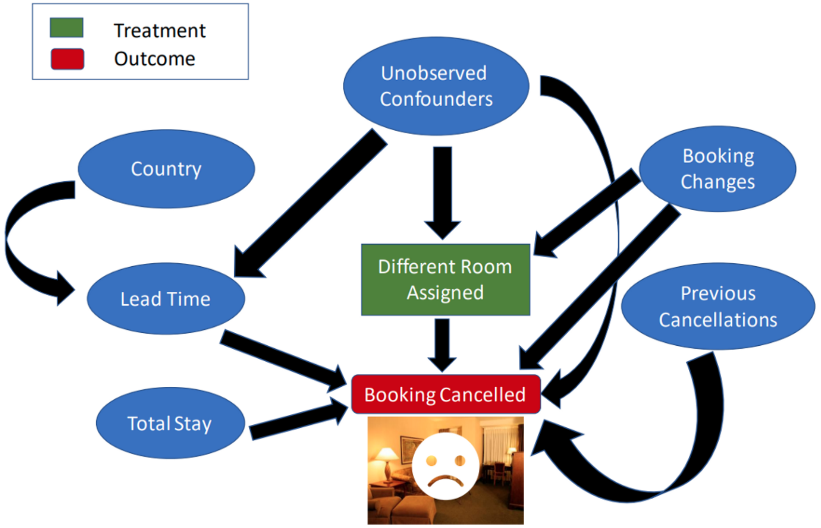


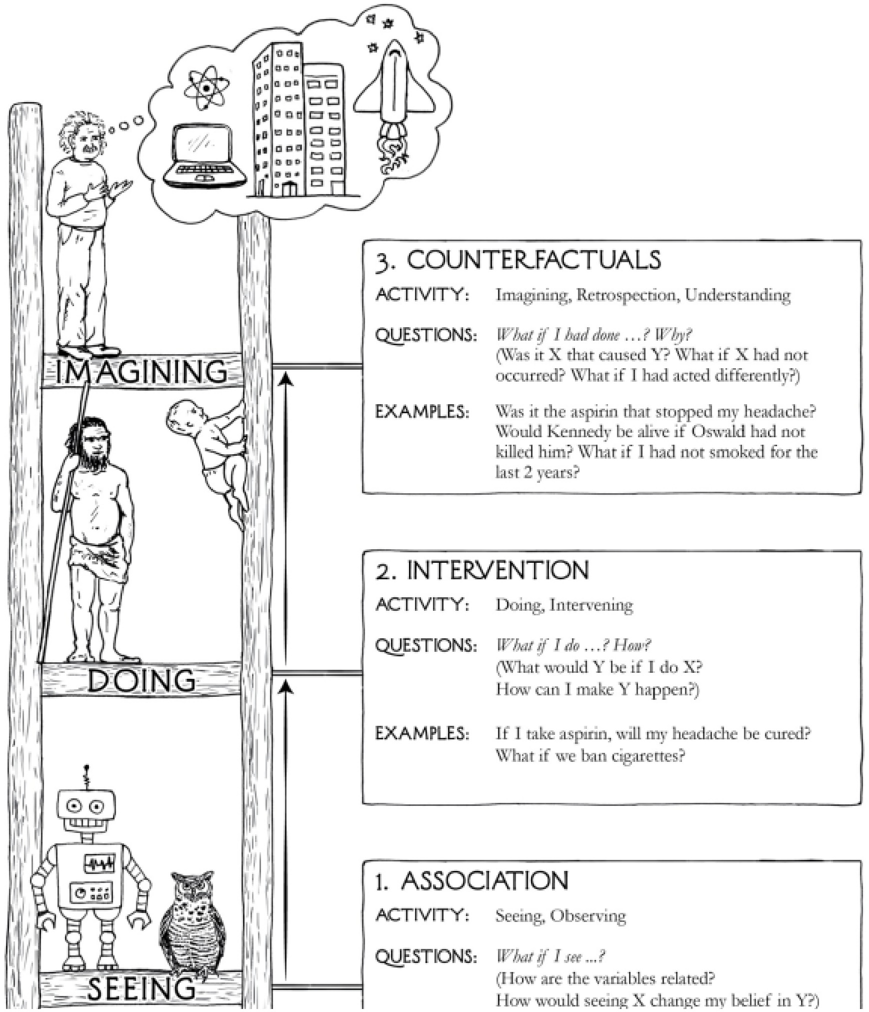
Image by author

Why did Medium/Linkedin or any other platform recommend this post to you? More importantly, what piqued your interest and made you click on this post? Was it the intriguing title or the cool thumbnail attached alongside? Maybe you just clicked because you knew me personally and as a good friend wanted to help me out by increasing the outreach of this article ;-)

As a popular use case of Machine Learning, the platform which recommended you this article must be having some complex Machine Learning or Deep Learning based Classifier running internally whose role would be to predict whether you should be shown this post or not. That predictive model would be forming its decision solely on the fact that the probability of you clicking that article is maximized (because most of the targeted online advertizing companies pay only when a user clicks on their content). All of these predictions would be made using observed values of certain features that the underlying predictive model would have been trained on.

Now that you have already clicked on this article, what would happen if I change its title or thumbnail, would you still have clicked on it?
I can continue asking such thought provoking questions one after the other and there is a good chance that even you might be unaware of the true cause!
Imagining scenarios which would have been true under different circumstances, are known as *counterfactuals*. Broadly speaking Causal Inference boils down to estimating the *counterfactual*.

Naturally, one can only observe either the *factual* (Ex- You clicking/not-clicking on the post when its title was the given title) or the counterfactual (Ex- You clicking/not-clicking on the post when its title was something different). In terms of data, it means that you can either have a data point corresponding to the factual scenario or the counterfactual scenario but never both. It is because of this fact that you can never ask that predictive model to explain its decision on why you would have clicked on this post.



3. COUNTERFACTUALS

ACTIVITY: Imagining, Retrospection, Understanding

QUESTIONS: *What if I had done …? Why?*
(Was it X that caused Y? What if X had not occurred? What if I had acted differently?)

EXAMPLES: Was it the aspirin that stopped my headache? Would Kennedy be alive if Oswald had not killed him? What if I had not smoked for the last 2 years?

2. INTERVENTION

ACTIVITY: Doing, Intervening

QUESTIONS: *What if I do …? How?*
(What would Y be if I do X?
How can I make Y happen?)

EXAMPLES: If I take aspirin, will my headache be cured?
What if we ban cigarettes?

1. ASSOCIATION

ACTIVITY: Seeing, Observing

QUESTIONS: *What if I see ...?*
(How are the variables related?
How would seeing X change my belief in Y?)

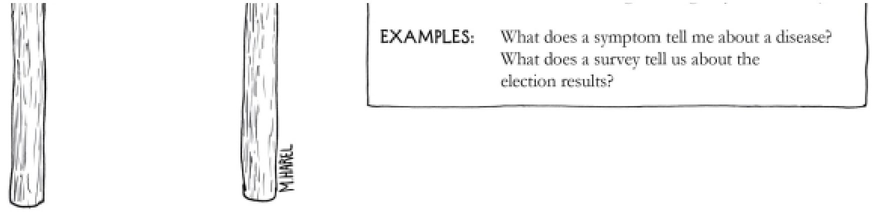

EXAMPLES:
What does a symptom tell me about a disease?
What does a survey tell us about the election results?
M.HAREL

Fig-1: Image source: “The Book of Why” by Judea Pearl

The image above describes the Ladder of Causation as stated by Prof. Judea Pearl in “*The Book of Why*”.

Most Machine Learning and complex Deep Learning models lie at the bottomost rung of this ladder because they make predictions solely by exploiting associations or correlations amongst different variables. However that does not provide us with a framework to answer *“What if?”* and *“Why”* questions which are described by the 2nd and 3rd rungs of this ladder and are key to ***decision making*** if these models are to be used in any societally critical domains such as:-

- **Healthcare**:- Ex- *If I take this vaccine, will I be cured of COVID-19? What if I don’t take this vaccine right now. Will it put me at more risk of getting infected with COVID’19 compared to the case if I had taken this vaccine?*
- **Finance:-** *Ex- If the price of Stock-A drops sharply, will it cause the price of Stock-B drop as well? What if the Price of Stock-A had not dropped, would the Price of Stock-B stayed the same as well?*
- **Governance:**- *Do countries with higher GDPs, have happier citizens in general? What if there is a sharp decrement in the GDP for the coming year, will it affect the happiness of its citizens?*

- **Education:-** *A professor changes his grading scheme for a particular course, does it affect how much you're going to learn out of this course. What if he had not changed the grading scheme and continued with the previous one, would his students in general had learned more?*

Case Study:- Hotel Booking Cancellations

We consider the problem of estimating the impact of assigning a different room as compared to what the customer had reserved on *Booking Cancellations*.

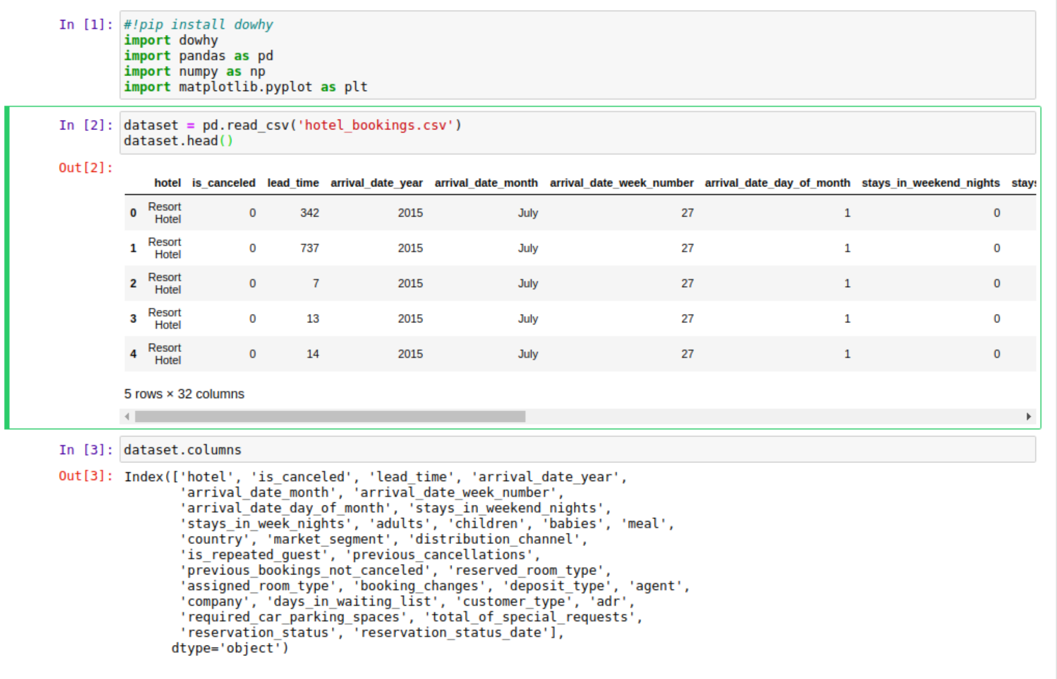
The gold standard of finding this out would be to use experiments such as R*andomized Controlled Trials* wherein each customer is randomly assigned to one of the two categories i.e. each customer is either assigned a different room or the same room as he had booked before. But what if we cannot intervene or its too costly too perform such an experiment (Ex- The Hotel would start losing its reputation if people learn that its randomly assigning people to different rooms). Can we somehow answer our query using only observational data or data that has been collected in the past?

Description of the Dataset

This dataset contains booking information for a city hotel and a resort hotel taken from a real hotel in Portugal, and includes information such as when the booking was made, length of stay, the number of adults, children, and/or babies, and the number of available parking spaces, among other things. All personally identifying information has been removed from the data.

For additional details, the reader is referred to article *Hotel Booking Demand Datasets*, written by *Antonio et. al for Data in Brief, Volume 22, February 2019* from which this data has been originally sourced. If you're in a hurry then you can take a quick glance here.

P.S.- If you are interested to know more about building predictive models on this dataset to predict whether a booking would be cancelled or not, please have a look at this notebook created by me earlier. The focus is more on *Exploratory Data Analysis* and *Model Building* in that notebook.



In [1]:
```
#!pip install dowhy
import dowhy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
dataset = pd.read_csv('hotel_bookings.csv')
dataset.head()
```

Out[2]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
2	Resort Hotel	0	7	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

5 rows × 32 columns

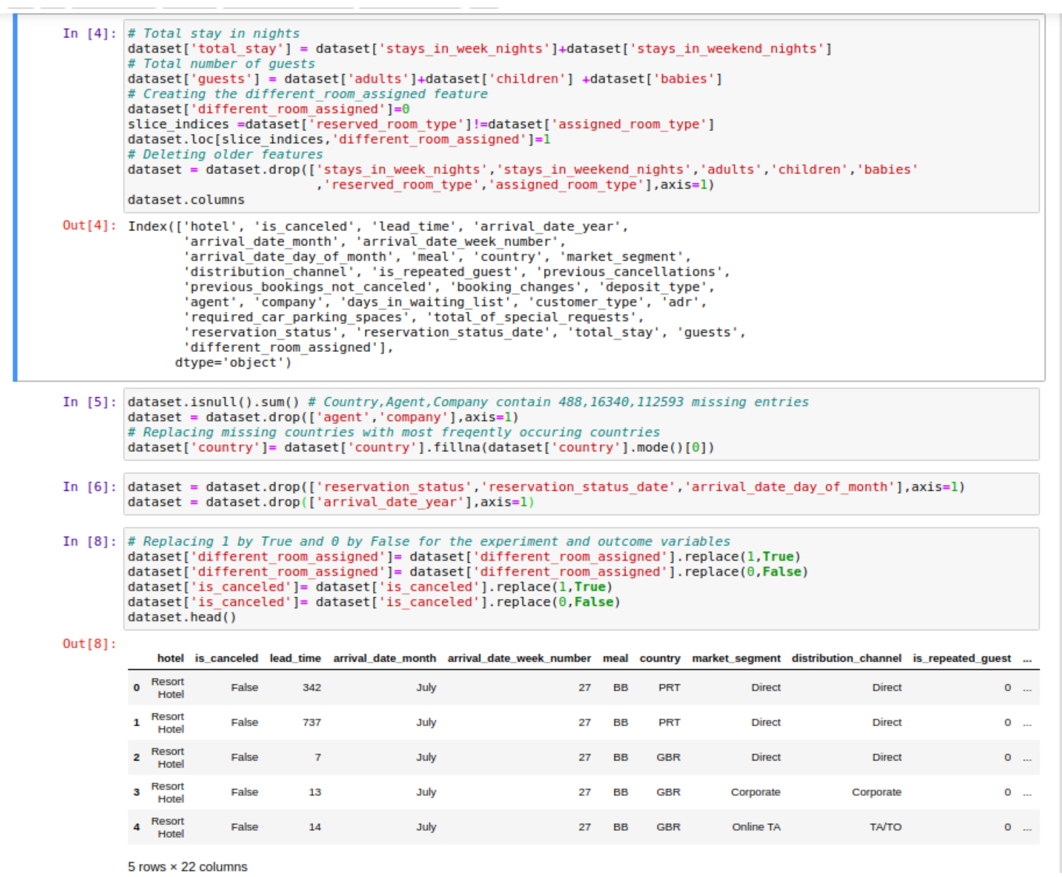
In [3]:
```
dataset.columns
```

Out[3]:
```
Index(['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', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

Feature Engineering & Pre-Processing

Lets create some new and meaningful features so as to reduce the dimensionality of the dataset. The following new features have been created:-

- **Total Stay** = *stays_in_weekend_nights* + *stays_in_week_nights*
- **Guests** = *adults* + *children* + *babies*
- **Different_room_assigned** = 1 if *reserved_room_type* & *assigned_room_type* are different, 0 otherwise.



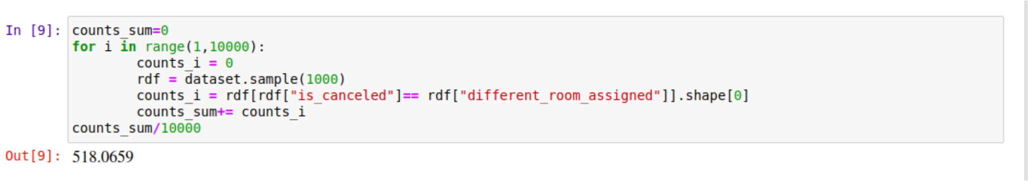
```
In [4]: # Total stay in nights
        dataset['total_stay'] = dataset['stays_in_week_nights']+dataset['stays_in_weekend_nights']
        # Total number of guests
        dataset['guests'] = dataset['adults']+dataset['children'] +dataset['babies']
        # Creating the different_room_assigned feature
        dataset['different_room_assigned']=0
        slice_indices =dataset['reserved_room_type']!=dataset['assigned_room_type']
        dataset.loc[slice_indices,'different_room_assigned']=1
        # Deleting older features
        dataset = dataset.drop(['stays_in_week_nights','stays_in_weekend_nights','adults','children','babies'
                               ,'reserved_room_type','assigned_room_type'],axis=1)
        dataset.columns

Out[4]: Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
               'arrival_date_month', 'arrival_date_week_number',
               'arrival_date_day_of_month', 'meal', 'country', 'market_segment',
               'distribution_channel', 'is_repeated_guest', 'previous_cancellations',
               'previous_bookings_not_canceled', 'booking_changes', 'deposit_type',
               'agent', 'company', 'days_in_waiting_list', 'customer_type', 'adr',
               'required_car_parking_spaces', 'total_of_special_requests',
               'reservation_status', 'reservation_status_date', 'total_stay', 'guests',
               'different_room_assigned'],
              dtype='object')

In [5]: dataset.isnull().sum() # Country,Agent,Company contain 488,16340,112593 missing entries
        dataset = dataset.drop(['agent','company'],axis=1)
        # Replacing missing countries with most freqently occuring countries
        dataset['country']= dataset['country'].fillna(dataset['country'].mode()[0])

In [6]: dataset = dataset.drop(['reservation_status','reservation_status_date','arrival_date_day_of_month'],axis=1)
        dataset = dataset.drop(['arrival_date_year'],axis=1)

In [8]: # Replacing 1 by True and 0 by False for the experiment and outcome variables
        dataset['different_room_assigned']= dataset['different_room_assigned'].replace(1,True)
        dataset['different_room_assigned']= dataset['different_room_assigned'].replace(0,False)
        dataset['is_canceled']= dataset['is_canceled'].replace(1,True)
        dataset['is_canceled']= dataset['is_canceled'].replace(0,False)
        dataset.head()

Out[8]:
```

	hotel	is_canceled	lead_time	arrival_date_month	arrival_date_week_number	meal	country	market_segment	distribution_channel	is_repeated_guest	...
0	Resort Hotel	False	342	July	27	BB	PRT	Direct	Direct	0	...
1	Resort Hotel	False	737	July	27	BB	PRT	Direct	Direct	0	...
2	Resort Hotel	False	7	July	27	BB	GBR	Direct	Direct	0	...
3	Resort Hotel	False	13	July	27	BB	GBR	Corporate	Corporate	0	...
4	Resort Hotel	False	14	July	27	BB	GBR	Online TA	TA/TO	0	...

5 rows × 22 columns

Calculating Expected counts

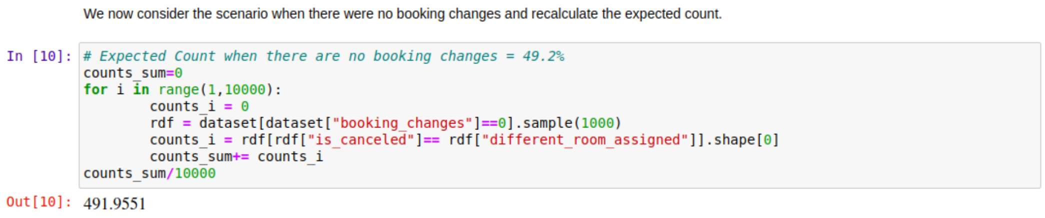
Since the number of number of cancellations and the number of times a different room was assigned is heavily imbalanced, we calculate expected counts.



```
In [9]:  counts_sum=0
         for i in range(1,10000):
                 counts_i = 0
                 rdf = dataset.sample(1000)
                 counts_i = rdf[rdf["is_canceled"]== rdf["different_room_assigned"]].shape[0]
                 counts_sum+= counts_i
         counts_sum/10000

Out[9]:  518.0659
```

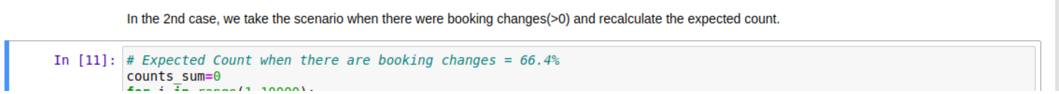
Since the expected count turns out to be ~50% (i.e. the probability of these two variables attaining the same value at random). So statistically speaking, we have no definite conclusion at this stage.

Let's see what happens when we re-calculate the expected count taking different values of *Booking Changes* into account.



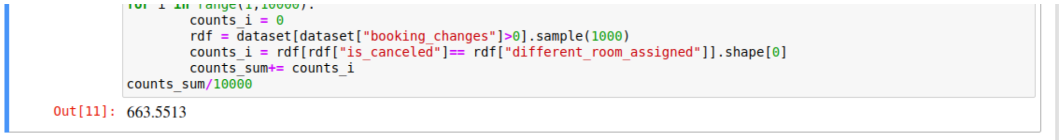
We now consider the scenario when there were no booking changes and recalculate the expected count.

```
In [10]: # Expected Count when there are no booking changes = 49.2%
         counts_sum=0
         for i in range(1,10000):
                 counts_i = 0
                 rdf = dataset[dataset["booking_changes"]==0].sample(1000)
                 counts_i = rdf[rdf["is_canceled"]== rdf["different_room_assigned"]].shape[0]
                 counts_sum+= counts_i
         counts_sum/10000

Out[10]: 491.9551
```



In the 2nd case, we take the scenario when there were booking changes(>0) and recalculate the expected count.

```
In [11]: # Expected Count when there are booking changes = 66.4%
         counts_sum=0
         for i in range(1,10000):
```



```
for i in range(1,10000):
        counts_i = 0
        rdf = dataset[dataset["booking_changes"]>0].sample(1000)
        counts_i = rdf[rdf["is_canceled"]== rdf["different_room_assigned"]].shape[0]
        counts_sum+= counts_i
counts_sum/10000
```

```
Out[11]: 663.5513
```

There is definitely some change happening when the number of booking changes are non-zero. So it gives us a hint that *Booking Changes* must be a confounding variable.

But is *Booking Changes* the only confounding variable? What if there were some unobserved confounders, regarding which we have no information(feature) present in our dataset. Would we still be able to make the same claims as before?

Enter DoWhy, the proverbial white knight

DoWhy focuses its attention on the assumptions required for causal inference and provides estimation methods(its relatively easier since its a statistical procedure) such as matching and IV, so that the user can focus more on identifying assumptions i.e. Model assumptions explicitly using *Causal Graphical Models*.

Input:- *Observational data* and *Causal Graph* wherein the *treatment* variable, *covariates*(all variables except outcome and treatment) and *outcome* have been clearly specified.

Output:- Causal Effect between desired variables, *"What-if"* analysis (Topmost rung of the *Ladder of Causation*).

Step-0. Fix your Goal and formulate the problem:

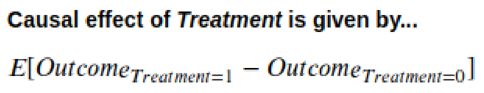
We consider the problem of estimating what impact does assigning a room different to what a customer had reserved has on the booking cancellation. The gold standard of finding this out would be to use *Randomized Controlled Trials* (or *A/B Tests*) each customer is either assigned a different room or the same room as he had booked.

> *A Randomized Controlled Trial (**RCT**) is an experimental form of impact evaluation in which the population receiving the programme or policy intervention is chosen at random from the eligible population, and a control group is also chosen at random from the same eligible population. ~UNICEF*

But this might not be the best practice as far as the hotels reputation is concerned. For Ex- Imagine a scenario wherein a family of 6 arrives at the hotel, but instead of their booked penthouse they are assigned a room with a capacity of only 2 people. So randomized experiments often turn out to be infeasible, costly or downright unethical in real world scenarios.

Formally…

We say that the *Treatment* causes *Outcome* iff changing *Treatment* leads to a change in *Outcome* keeping everything else constant. Causal effect is the magnitude by which *Outcome* is changed by a unit change in *Treatment*.



Causal effect of *Treatment* is given by...

$$E[Outcome_{Treatment=1} - Outcome_{Treatment=0}]$$

Step-1: Create a Causal Model

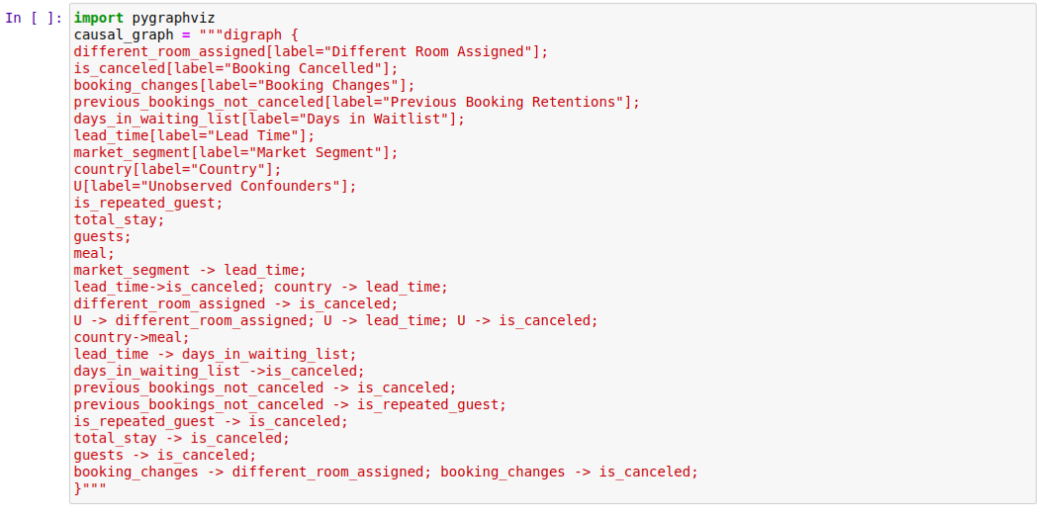
Represent your prior knowledge about the predictive modelling problem as a CI graph using assumptions. Don’t worry, you need not specify the full graph at this stage. Even a partial graph would be enough and the rest can be figured out by *DoWhy* ;-) The causal model, I have thought has been plotted in Fig-2.

Here are a list of assumptions that I have then translated into a Causal Diagram:-

- *Market Segment* has 2 levels, “TA” refers to the “Travel Agents” and “TO” means “Tour Operators” so it should affect the *Lead Time* (which is simply the number of days between booking and arrival).
- *Country* would also play a role in deciding whether a person books early or not (hence more *Lead Time*) and what type of *Meal* a person would prefer.
- *Lead Time* would definitely affected the number of *Days in Waitlist* (There are lesser chances of finding a reservation if you’re booking late). Additionally, higher *Lead Times* can also lead to Cancellations.
- The number of *Days in Waitlist*, the *Total Stay* in nights and the number of *Guests* might affect whether the booking is cancelled or retained.
- *Previous Booking Retentions* would affect whether a customer is a *Repeated Guest* or not. Additionally, both of these variables would affect whether the booking get cancelled or not (Ex- A customer who has retained his past 5 bookings in the past has a higher chance of retaining

this one also. Similarly a person who has been cancelling this booking has a higher chance of repeating the same).

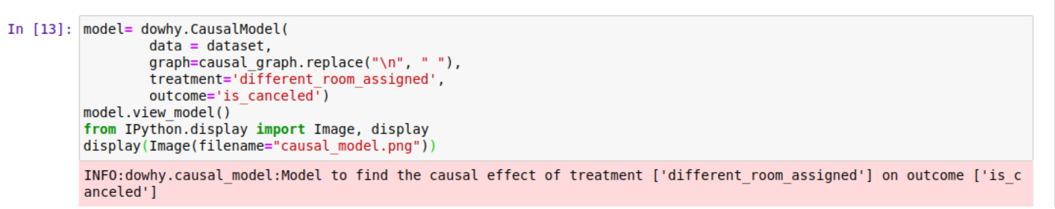
- *Booking Changes* would affect whether the customer is assigned a *different room* or not which might also lead to cancellation.
- Finally, the number of *Booking Changes* being the only confounder affecting *Treatment* and *Outcome* is highly unlikely and its possible that there might be some *Unobsevered Confounders,* regarding which we have no information being captured in our data.



```
import pygraphviz
causal_graph = """digraph {
different_room_assigned[label="Different Room Assigned"];
is_canceled[label="Booking Cancelled"];
booking_changes[label="Booking Changes"];
previous_bookings_not_canceled[label="Previous Booking Retentions"];
days_in_waiting_list[label="Days in Waitlist"];
lead_time[label="Lead Time"];
market_segment[label="Market Segment"];
country[label="Country"];
U[label="Unobserved Confounders"];
is_repeated_guest;
total_stay;
guests;
meal;
market_segment -> lead_time;
lead_time->is_canceled; country -> lead_time;
different_room_assigned -> is_canceled;
U -> different_room_assigned; U -> lead_time; U -> is_canceled;
country->meal;
lead_time -> days_in_waiting_list;
days_in_waiting_list ->is_canceled;
previous_bookings_not_canceled -> is_canceled;
previous_bookings_not_canceled -> is_repeated_guest;
is_repeated_guest -> is_canceled;
total_stay -> is_canceled;
guests -> is_canceled;
booking_changes -> different_room_assigned; booking_changes -> is_canceled;
}"""
```

Here the *Treatment* is assigning the same type of room reserved by the customer during Booking. *Outcome* would be whether the booking was cancelled or not. *Common Causes* represent the variables that according to us have a causal affect on both *Outcome* and *Treatment*. As per our causal assumptions, the 2 variables satisfying this criteria are *Booking Changes* and the *Unobserved Confounders*. So if we are not specifying the graph explicitly

(Not Recommended!), one can also provide these as parameters in the function mentioned below.



```
In [13]: model= dowhy.CausalModel(
                 data = dataset,
                 graph=causal_graph.replace("\n", " "),
                 treatment='different_room_assigned',
                 outcome='is_canceled')
         model.view_model()
         from IPython.display import Image, display
         display(Image(filename="causal_model.png"))

         INFO:dowhy.causal_model:Model to find the causal effect of treatment ['different_room_assigned'] on outcome ['is_c
         anceled']
```

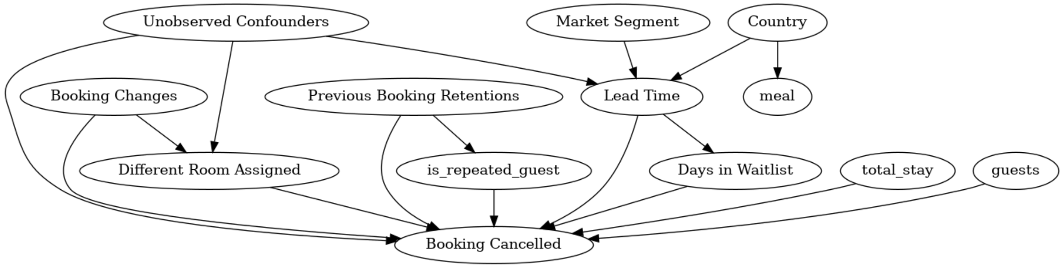


Fig-2: Causal Assumptions specified in terms of a simple graphical model.

Depending on your intuition and assumptions, one might also consider a more detailed Causal diagram as described by Fig-3.

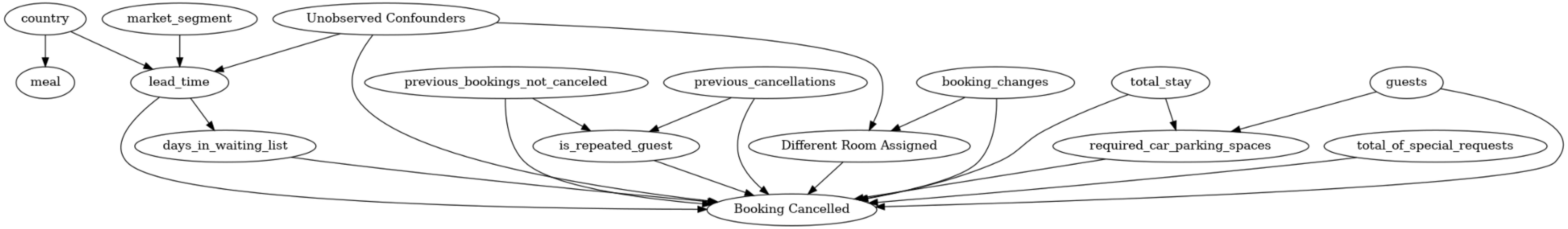


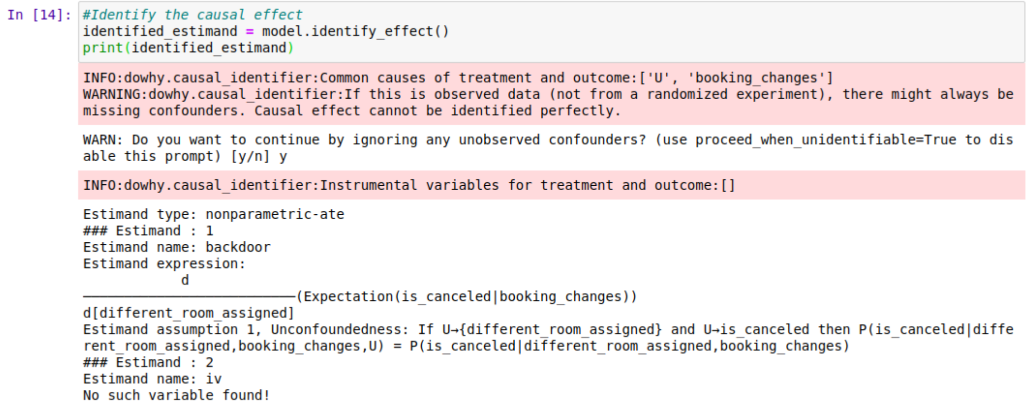
Fig-3: Causal Assumptions specified in terms of a graphical model with greater details.

Step-2: Identify Cause

We say that *Treatment* causes *Outcome* if changing *Treatment* leads to a change in *Outcome* keeping everything else constant.

Keeping everything else constant can be thought by imagining a *Counterfactual* World or a paralell universe where everything was same uptill the point when the treatment was introduced. In the *factual* world the treatment was provided whereas in the *Counterfactual* world it wasn't. Thus any change in *Outcome* that we might be observing would be solely due to the *Treatment*.

Thus in this step, by using properties of the causal graph, we identify the causal effect to be estimated.



```
In [14]: #Identify the causal effect
         identified_estimand = model.identify_effect()
         print(identified_estimand)
```

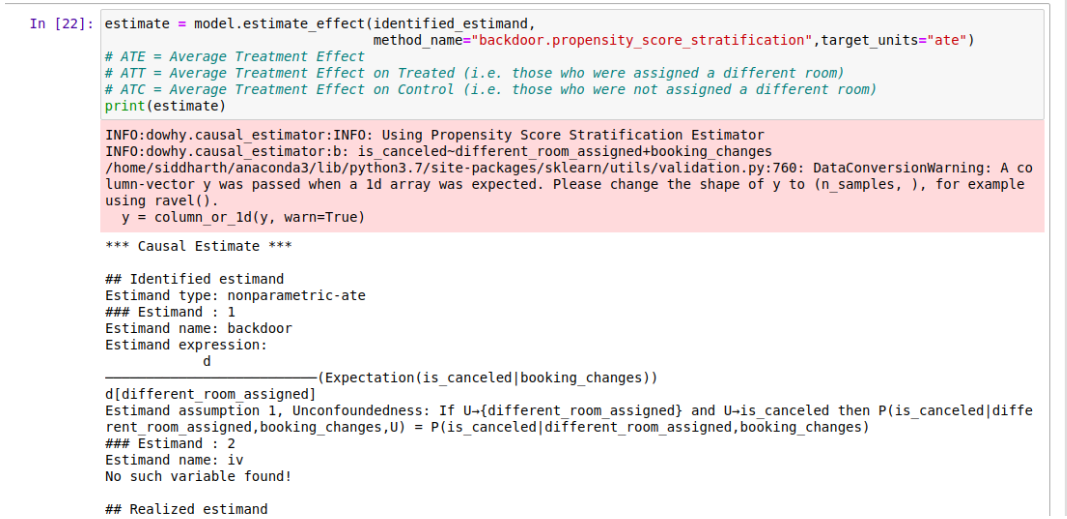
```
INFO:dowhy.causal_identifier:Common causes of treatment and outcome:['U', 'booking_changes']
WARNING:dowhy.causal_identifier:If this is observed data (not from a randomized experiment), there might always be
missing confounders. Causal effect cannot be identified perfectly.

WARN: Do you want to continue by ignoring any unobserved confounders? (use proceed_when_unidentifiable=True to dis
able this prompt) [y/n] y

INFO:dowhy.causal_identifier:Instrumental variables for treatment and outcome:[]

Estimand type: nonparametric-ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
            d
──────────────────────────(Expectation(is_canceled|booking_changes))
d[different_room_assigned]
Estimand assumption 1, Unconfoundedness: If U→{different_room_assigned} and U→is_canceled then P(is_canceled|diffe
rent_room_assigned,booking_changes,U) = P(is_canceled|different_room_assigned,booking_changes)
### Estimand : 2
Estimand name: iv
No such variable found!
```

Step-3: Estimate identified cause

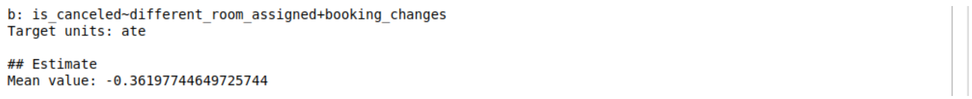
Causal effect is the magnitude by which the *Outcome* changes due to a *unit change* in *Treatment*. Since Estimation is a statistical procedure, it's much simpler compared to the other steps. *DoWhy* provides a number of methods which can be used to calculate the identified causal estimate.

The method being used in our example is the *Propensity Score Stratification Estimator*. One can think of propensity scores as a measure of an observations tendency to be treated. They are to be estimated or modeled from data and not observed. These scores are calculated by training a Machine Learning Classifier such as *Logistic Regression* or *Random Forests* in order to predict the *Treatment* variable given Covariates(all variables except *Outcome, Treatement*). Stratification is a technique for identifying paired subpopulations whose covariate distributions are similar(similarity measured using different types of norms).

For an apt explanation of these methods & techniques the reader is referred to this Tutorial given by the creators of *DoWhy* at KDD'18.



```
In [22]: estimate = model.estimate_effect(identified_estimand,
                                          method_name="backdoor.propensity_score_stratification",target_units="ate")
         # ATE = Average Treatment Effect
         # ATT = Average Treatment Effect on Treated (i.e. those who were assigned a different room)
         # ATC = Average Treatment Effect on Control (i.e. those who were not assigned a different room)
         print(estimate)

INFO:dowhy.causal_estimator:INFO: Using Propensity Score Stratification Estimator
INFO:dowhy.causal_estimator:b: is_canceled~different_room_assigned+booking_changes
/home/siddharth/anaconda3/lib/python3.7/site-packages/sklearn/utils/validation.py:760: DataConversionWarning: A co
lumn-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example
using ravel().
  y = column_or_1d(y, warn=True)

*** Causal Estimate ***

## Identified estimand
Estimand type: nonparametric-ate
### Estimand : 1
Estimand name: backdoor
Estimand expression:
            d
──────────────────────────(Expectation(is_canceled|booking_changes))
d[different_room_assigned]
Estimand assumption 1, Unconfoundedness: If U→{different_room_assigned} and U→is_canceled then P(is_canceled|diffe
rent_room_assigned,booking_changes,U) = P(is_canceled|different_room_assigned,booking_changes)
### Estimand : 2
Estimand name: iv
No such variable found!

## Realized estimand
```



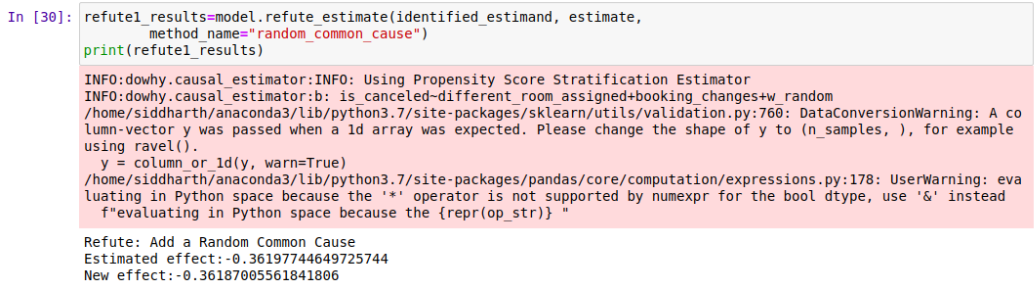
```
b: is_canceled~different_room_assigned+booking_changes
Target units: ate

## Estimate
Mean value: -0.36197744649725744
```

Step-4: Refute obtained results

Remember that the *Causal* part does not come from data, rather it comes from your *assumptions* (Step-1) that were used to *identify*(Step-2) and *estimate*(Step-3) the cause. Data is simply used for statistical estimation.

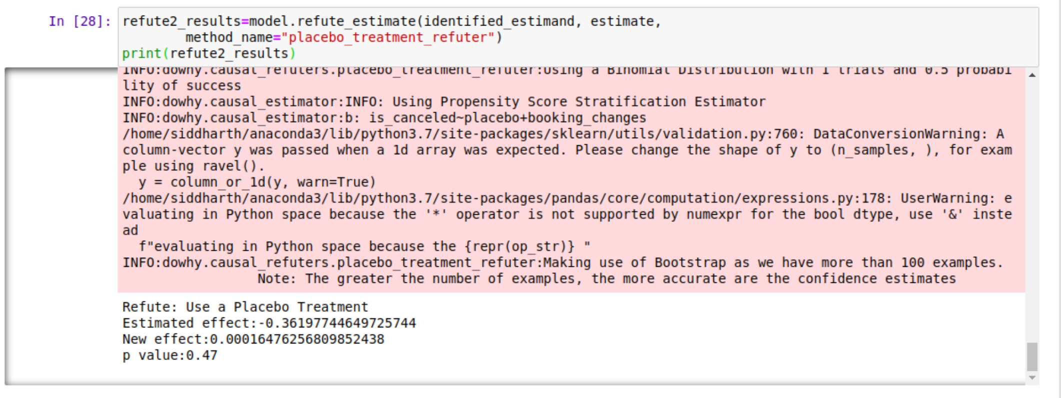
Thus it becomes verify our assumptions and challenge their validity in more than one way if possible. *DoWhy* offers Multiple Robustness Checks that can be used to test the validity of our assumptions:-

- **Random Common Cause**:- *Adds randomly drawn covariates to data and re-runs the analysis to see if the causal estimate changes or not. If our assumption was originally correct then there shouldn't much variation in the causal estimate.*



```
In [30]: refute1_results=model.refute_estimate(identified_estimand, estimate,
                method_name="random_common_cause")
         print(refute1_results)

INFO:dowhy.causal_estimator:INFO: Using Propensity Score Stratification Estimator
INFO:dowhy.causal_estimator:b: is_canceled~different_room_assigned+booking_changes+w_random
/home/siddharth/anaconda3/lib/python3.7/site-packages/sklearn/utils/validation.py:760: DataConversionWarning: A co
lumn-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example
using ravel().
  y = column_or_1d(y, warn=True)
/home/siddharth/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py:178: UserWarning: eva
luating in Python space because the '*' operator is not supported by numexpr for the bool dtype, use '&' instead
  f"evaluating in Python space because the {repr(op_str)} "

Refute: Add a Random Common Cause
Estimated effect:-0.36197744649725744
New effect:-0.36187005561841806
```

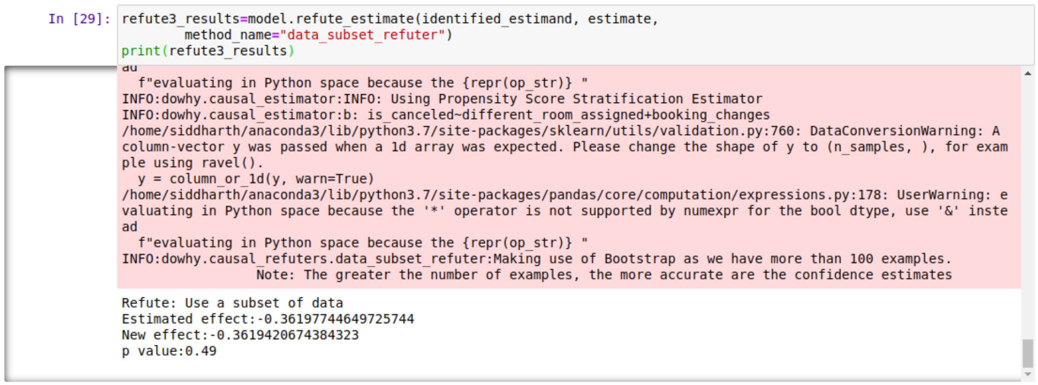
- **Placebo Treatment Refuter**:- *Randomly assigns any covariate as a treatment and re-runs the analysis. If our assumptions were correct then* *this newly found out estimate should go to 0.*



```
In [28]: refute2_results=model.refute_estimate(identified_estimand, estimate,
                 method_name="placebo_treatment_refuter")
         print(refute2_results)
```

```
INFO:dowhy.causal_refuters.placebo_treatment_refuter:Using a Binomial Distribution with 1 trials and 0.5 probabi
lity of success
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Stratification Estimator
INFO:dowhy.causal_estimator:b: is_canceled~placebo+booking_changes
/home/siddharth/anaconda3/lib/python3.7/site-packages/sklearn/utils/validation.py:760: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exam
ple using ravel().
  y = column_or_1d(y, warn=True)
/home/siddharth/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py:178: UserWarning: e
valuating in Python space because the '*' operator is not supported by numexpr for the bool dtype, use '&' inste
ad
  f"evaluating in Python space because the {repr(op_str)} "
INFO:dowhy.causal_refuters.placebo_treatment_refuter:Making use of Bootstrap as we have more than 100 examples.
                 Note: The greater the number of examples, the more accurate are the confidence estimates

Refute: Use a Placebo Treatment
Estimated effect:-0.36197744649725744
New effect:0.00016476256809852438
p value:0.47
```

- **Data Subset Refuter**:- *Creates subsets of the data(similar to cross-validation) and checks whether the causal estimates vary across subsets. If our assumptions were correct there shouldn't be much variation.*



```
In [29]: refute3_results=model.refute_estimate(identified_estimand, estimate,
                 method_name="data_subset_refuter")
         print(refute3_results)
```

```
ad
  f"evaluating in Python space because the {repr(op_str)} "
INFO:dowhy.causal_estimator:INFO: Using Propensity Score Stratification Estimator
INFO:dowhy.causal_estimator:b: is_canceled~different_room_assigned+booking_changes
/home/siddharth/anaconda3/lib/python3.7/site-packages/sklearn/utils/validation.py:760: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exam
ple using ravel().
  y = column_or_1d(y, warn=True)
/home/siddharth/anaconda3/lib/python3.7/site-packages/pandas/core/computation/expressions.py:178: UserWarning: e
valuating in Python space because the '*' operator is not supported by numexpr for the bool dtype, use '&' inste
ad
  f"evaluating in Python space because the {repr(op_str)} "
INFO:dowhy.causal_refuters.data_subset_refuter:Making use of Bootstrap as we have more than 100 examples.
                 Note: The greater the number of examples, the more accurate are the confidence estimates

Refute: Use a subset of data
Estimated effect:-0.36197744649725744
New effect:-0.3619420674384323
p value:0.49
```

By using multiple robustness checks, we have validated that our causal assumptions were indeed correct!

This tells us that on an average the Probability of a hotel booking being cancelled decreases by ~36% when the Person is assigned the same room compared to the case when he is assigned a different room than what he had chosen during booking.

So next time if the hotel staff is in a pickle and has to assign its customer a different room, they know how much of an impact their actions would have on that booking being cancelled.

Bonus Tips:-

Best Practices for doing Causal Inference on observational data:-

1. Be sure to follow the four steps viz. ***Model, Identify, Estimate, Refute***.
2. ***Aim for simplicity,*** If your analysis is too complicated, it's most likely wrong!
3. Try at least ***2 methods with different assumptions***. Higher confidence in estimates if both methods agree.

A big thanks to Dr. Amit Sharma & Dr. Emre Kiciman from Microsoft Research for creating this amazing library!

The complete jupyter notebook along with the dataset is now a part of Microsoft Dowhy repository on github.

microsoft/dowhy

Permalink Dismiss GitHub is home to over 50 million developers

github.com

References:-

- *"The Book of Why"* by Prof. Judea Pearl.
- "The Elements of Causal Inference" by Prof. Bernhard Schölkopf.
- Breezy Talk on Causal Inference given at Northwestern University by Dr. Amit Sharma.
- KDD'18 Tutorial given by Dr. Amit Sharma
- https://github.com/microsoft/dowhy

Causality Machine Learning Predictive Analytics Causal Inference Booking

About Help Legal