

**BIRLA INSTITUTE OF TECHNOLOGY**

**MESRA**



**MASTER OF COMPUTER APPLICATIONS (SEM - IV)**

**2020-2021**

**Machine Learning Project**

**(Documentation of the Assign Project)**

**SUBMITTED TO –**

**Prof.RASHMI RATHI UPADHYAY**

**(Assistant Professor)**

**SUBMITTED BY –**

**KESHAV TYAGI**

**(ROLL – MCA/10054/19)**

# Hotel-Booking-Cancellation-Prediction Project

## **Abstract**

Booking cancellations negatively contribute to the production of accurate forecasts, which comprise a critical tool in the hospitality industry. Research has shown that with today's computational power and advanced machine learning algorithms it is possible to build models to predict bookings cancellation likelihood. However, the effectiveness of these models has never been evaluated in a real environment. To fill this gap and investigate how these models can be implemented in a decision support system and its impact on demand-management decisions, a prototype was built and deployed in two hotels. The prototype, based on an automated machine learning system designed to learn continuously, lead to two important research contributions.

In reservation-based industries, an accurate booking cancellation forecast is of foremost importance to estimate demand. By combining data science tools and capabilities with human judgement and interpretation, this aims to demonstrate how the semiautomatic analysis of the contribute to synthesizing research findings and identify research topics about booking cancellation forecasting.

Furthermore, The data used was obtained through a keyword search in Scopus and Web of Science databases. The methodology presented not only diminishes human bias, but also enhances the fact that data visualisation and text mining techniques facilitate abstraction, expedite analysis, and contribute to the improvement of reviews. Results show that despite the importance of bookings' cancellation forecast in terms of understanding net demand, improving cancellation, and overbooking policies, further research on the subject is still needed.

## **Project introduction:**

The cancellation rate for booking hotels online is high that creates discomfort for many hotels and create a desire to take precautions. Therefore, predicting reservations that can be cancelled will create a surplus value for hotels and hotels

can take action to prevent these cancellations. In my final project, I will try to explore the dataset and explain how to predict future cancelled reservations in advance by machine learning methods.

## What we are going to do in this project:

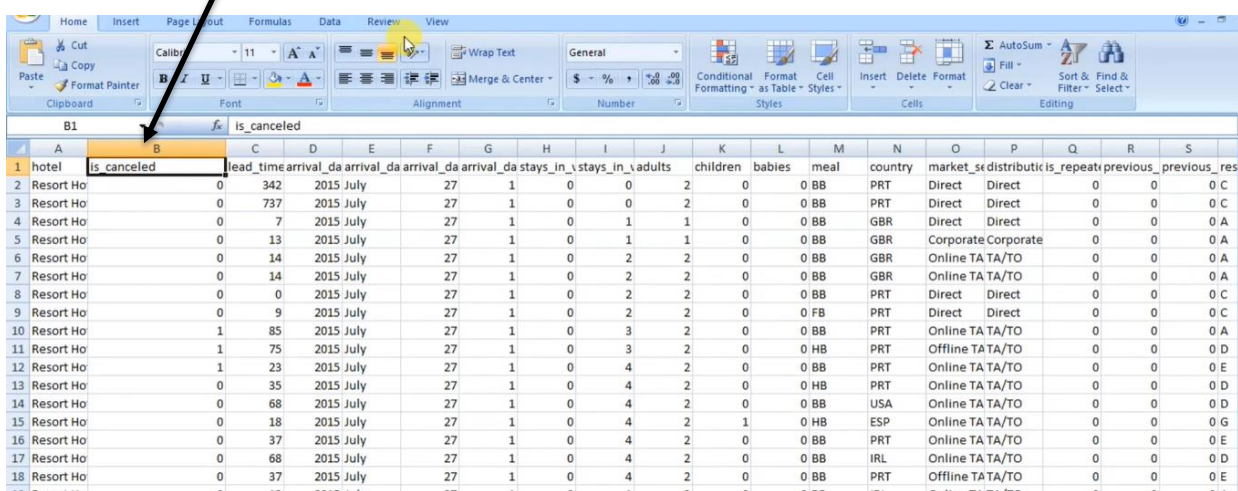
We are using (**Google Colab + Jupyter Notebook**) during the development of this Project.

Advantages of using **Google Colab** :

1. Like Jupyter Notebook + Bonus feature.
2. Hosted By Google.
3. Required Google account.
4. No Installation required.
5. Network speed is very fast.
6. Write Python code directly inside browser.
7. Notebook File stored in Google Drive.
8. CPU,GPU and TPU Support.

Again, I am clarifying the problem statement that we are making such a machine learning model that can predict whether a particular booking which has been done by a user is going to cancel or not.

This is exactly our that **feature** that we have to predict considering all data set.



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
hotel	is_canceled	lead_time	arrival_da	arrival_da	arrival_da	arrival_da	stays_in_v	stays_in_v	adults	children	babies	meal	country	market_sc	distribut	is_repeat	previous	previous_res
Resort Ho	0	342	2015 July		27	1	0	0	2	0	0	BB	PRT	Direct	Direct	0	0	0 C
Resort Ho	0	737	2015 July		27	1	0	0	2	0	0	BB	PRT	Direct	Direct	0	0	0 C
Resort Ho	0	7	2015 July		27	1	0	1	1	0	0	BB	GBR	Direct	Direct	0	0	0 A
Resort Ho	0	13	2015 July		27	1	0	1	1	0	0	BB	GBR	Corporate	Corporate	0	0	0 A
Resort Ho	0	14	2015 July		27	1	0	2	2	0	0	BB	GBR	Online TA	TA/TO	0	0	0 A
Resort Ho	0	14	2015 July		27	1	0	2	2	0	0	BB	GBR	Online TA	TA/TO	0	0	0 A
Resort Ho	0	0	2015 July		27	1	0	2	2	0	0	BB	PRT	Direct	Direct	0	0	0 C
Resort Ho	0	9	2015 July		27	1	0	2	2	0	0	FB	PRT	Direct	Direct	0	0	0 C
Resort Ho	1	85	2015 July		27	1	0	3	2	0	0	BB	PRT	Online TA	TA/TO	0	0	0 A
Resort Ho	1	75	2015 July		27	1	0	3	2	0	0	HB	PRT	Offline TA	TA/TO	0	0	0 D
Resort Ho	1	23	2015 July		27	1	0	4	2	0	0	BB	PRT	Online TA	TA/TO	0	0	0 E
Resort Ho	0	35	2015 July		27	1	0	4	2	0	0	HB	PRT	Online TA	TA/TO	0	0	0 D
Resort Ho	0	68	2015 July		27	1	0	4	2	0	0	BB	USA	Online TA	TA/TO	0	0	0 D
Resort Ho	0	18	2015 July		27	1	0	4	2	1	0	HB	ESP	Online TA	TA/TO	0	0	0 G
Resort Ho	0	37	2015 July		27	1	0	4	2	0	0	BB	PRT	Online TA	TA/TO	0	0	0 E
Resort Ho	0	68	2015 July		27	1	0	4	2	0	0	BB	IRL	Online TA	TA/TO	0	0	0 D
Resort Ho	0	37	2015 July		27	1	0	4	2	0	0	BB	PRT	Offline TA	TA/TO	0	0	0 E

You can see how much huge chunk of data we have over here and considering is all these features, we have to build such a machine learning that can predict wheather a particular booking is going to cancel or not.

**But before building such a model**, you have to understand your data or what your data is all about.

So the best way to understand new data is that performing lots of analysis or new data by fact and some amazing insight from this huge chunk of data.

And once we understand about data, what my data is all about, what exactly did i think once we understand our data to a greater extent than we are going to build such a model that can predict what exactly is....

It is back to this future by doing lots of data processing, doing lots of feature encoding techniques, changing techniques ,dealing with missing value and lots of machine learning algorithms, we are going to apply on our data.

### **1.Perform Data Cleaning & Prepare your Data for Modelling Purpose.**

```
# Replace missing values:  
# agent: If no agency is given, booking was most likely  
#         made without one.  
# company: If none given, it was most likely private.  
# rest should be self-explanatory.
```

The screenshot shows a Google Colab notebook titled 'KeshavMIPProject.ipynb'. The left sidebar displays a file explorer with a directory structure: 'drive' > 'MyDrive' > 'My MI Project Files'. The main code area contains the following cells:

```
[8] from google.colab import drive
    drive.mount('/content/drive/')

Mounted at /content/drive/

[10] !pip install -q keras

[11] import keras

[ ]

[12] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns

[13] df=pd.read_csv('/content/drive/MyDrive/My MI Project Files /hotel_bookings.csv')

[14] df.head()
```

The output of the last cell shows the first five rows of the 'hotel\_bookings.csv' file:

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

The status bar at the bottom indicates '0s completed at 11:40 PM'.

The screenshot shows the same Google Colab notebook. The code area now includes additional cells to check the data's shape and for missing values:

```
[15] df.shape

(119390, 32)

[16] df.isna().sum()

hotel
is_canceled
lead_time
```

The output for [16] shows the count of missing values for the specified columns:

Column	Count
hotel	0
is_canceled	0
lead_time	0

The status bar at the bottom indicates '0s completed at 11:40 PM'.

KeshavMIPProject.ipynb - Collaborative | My MIP Project Files - Google Drive | New Tab

colab.research.google.com/drive/14rA24rpeEd\_dnbFUXhIY5nhYQ8xTgh-W#scrollTo=azxwu8ag7SP9

KeshavMIPProject.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- drive
  - MyDrive
    - Classroom
    - Colab Notebooks
    - My MIP Project Files
      - ML\_hotel\_booking\_Pre...
      - data\_dictionary.txt
      - hotel\_bookings.csv
      - 1 st.wav
      - 1005419.pdf
      - 1619210137\_MayankTya...
      - 20200511\_222204.wav
      - APP FORM.pdf
      - Assignment.docx
      - BIT-Mesra-KESHAV TYAGI...
      - Burden R.L., Fairres J.D. Nu...
      - Certificate for KESHAV TY...
      - DS PRESENTATION.gslid...

Disk 69.21 GB available

+ Code + Text

```
df.isna().sum()
```

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

0s completed at 11:40 PM

Type here to search

12:08 AM 24-04-2021

KeshavMIPProject.ipynb - Collaborative | My MIP Project Files - Google Drive | New Tab

colab.research.google.com/drive/14rA24rpeEd\_dnbFUXhIY5nhYQ8xTgh-W#scrollTo=azxwu8ag7SP9

KeshavMIPProject.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Files

- drive
  - MyDrive
    - Classroom
    - Colab Notebooks
    - My MIP Project Files
      - ML\_hotel\_booking\_Pre...
      - data\_dictionary.txt
      - hotel\_bookings.csv
      - 1 st.wav
      - 1005419.pdf
      - 1619210137\_MayankTya...
      - 20200511\_222204.wav
      - APP FORM.pdf
      - Assignment.docx
      - BIT-Mesra-KESHAV TYAGI...
      - Burden R.L., Fairres J.D. Nu...
      - Certificate for KESHAV TY...
      - DS PRESENTATION.gslid...

Disk 69.21 GB available

+ Code + Text

```
[21] def data_clean(df):
      df.fillna(0,inplace=True)
      print(df.isnull().sum())

[22] data_clean(df)
```

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

0s completed at 11:40 PM

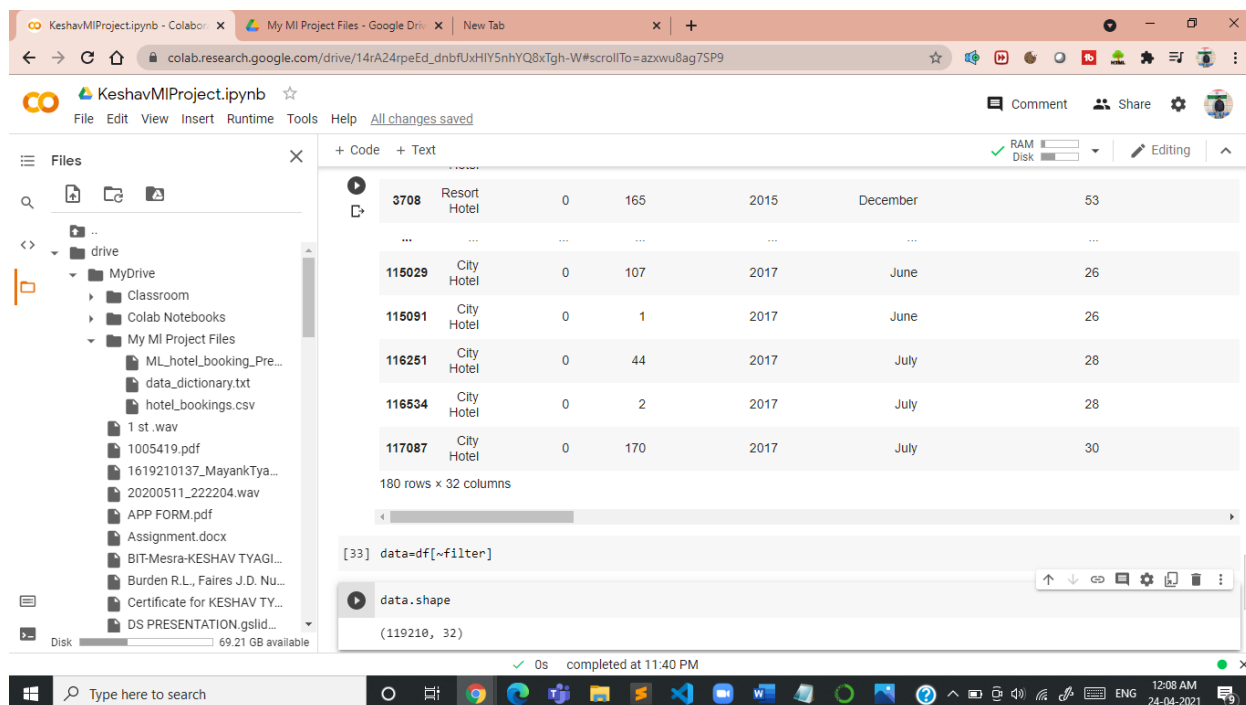
Type here to search

12:08 AM 24-04-2021

### seems to have some dirtiness in data as Adults, babies & children cant be zero at a same time.

### Visualise Entire Dataframe where adult, children & babies are 0

This is exactly that data on which you have to perform all your analysis on which you have to build your machine learning model after doing lots of feature engineering on new data.



The screenshot shows a Google Colab notebook interface. On the left, a file explorer shows the project structure, including a 'My MI Project Files' folder containing 'data\_dictionary.txt' and 'hotel\_bookings.csv'. The main area displays a DataFrame with 180 rows and 32 columns. The visible rows are as follows:

Booking ID	Hotel Name	Adults	Children	Year	Month	Bookings
3708	Resort Hotel	0	165	2015	December	53
...	...	...	...	...	...	...
115029	City Hotel	0	107	2017	June	26
115091	City Hotel	0	1	2017	June	26
116251	City Hotel	0	44	2017	July	28
116534	City Hotel	0	2	2017	July	28
117087	City Hotel	0	170	2017	July	30

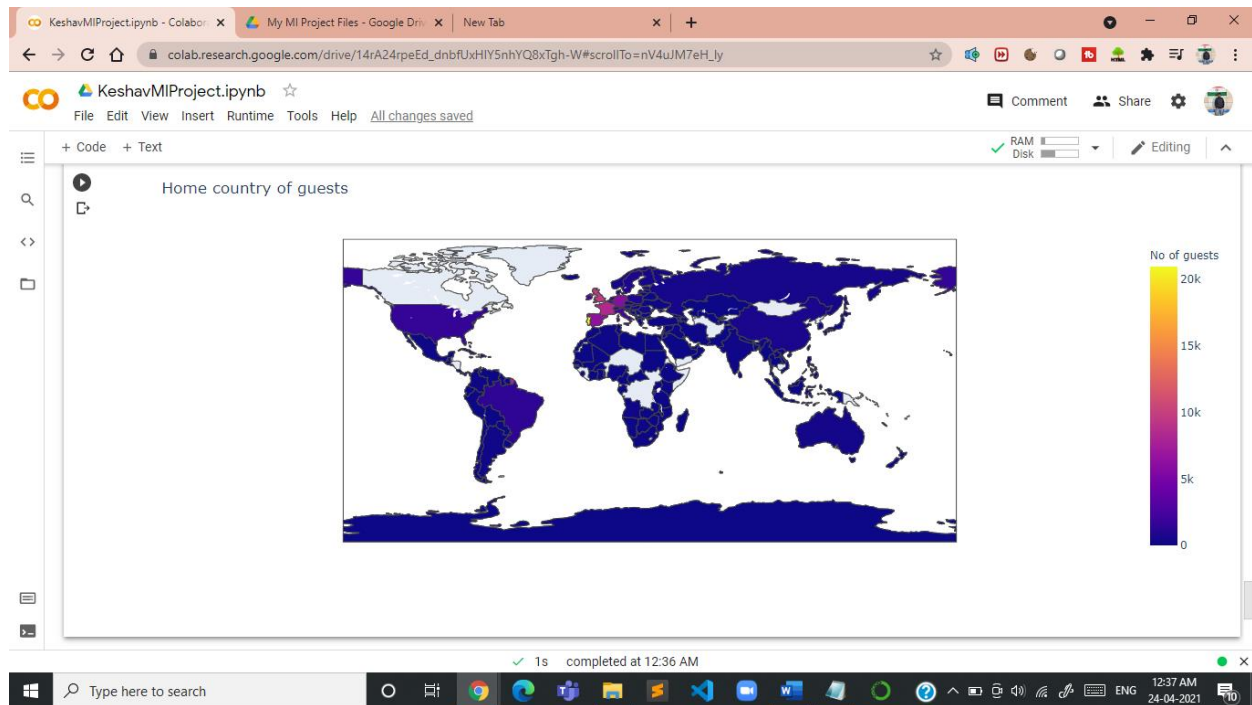
Below the DataFrame, the code cell shows the following execution:

```
[33] data=df[~filter]
data.shape
(119210, 32)
```

The status bar at the bottom indicates the notebook is completed at 11:40 PM.

## Now we find where do the guests come from? Lets perform Spatial Analysis

So **spatial Analysis** is all about whenever you are going to visualize the data on some map so that you get a clear cut yapped from which which location your guests are coming.



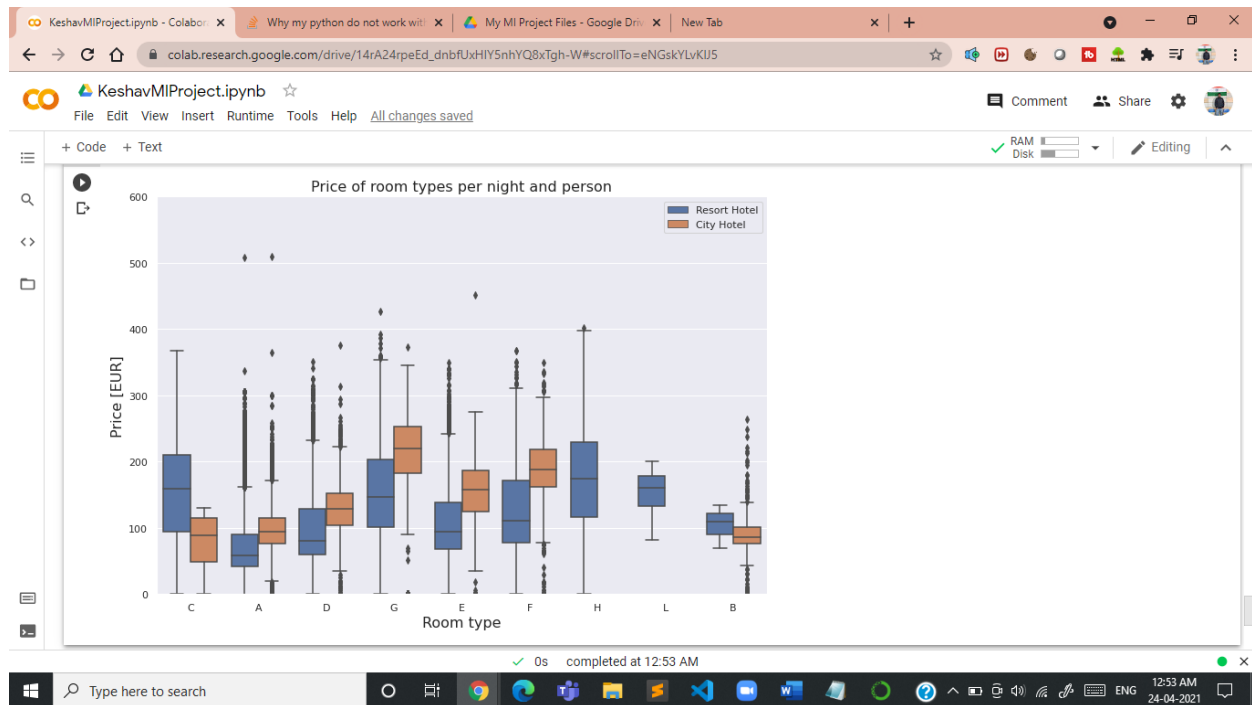
So, that's type of entrance ,how you can fetch stronger data.

***People from all over the world are staying in these two hotels. Most guests are from Portugal and other countries in Europe***

**Now we find how much the guests pay for a night ?Lets use distribution function or of a plot,or you can consider some fancy stuff like box plot.**

Both hotels have different room types and different meal arrangements. Seasonal factors are also important. So the prices vary a lot. Since no currency information is given, but Portugal is part of the European Monetary Union, I assume that all prices are in EUR.





After all you get this visual with respect to each of your room type over here. You will see this blue line exactly a resort hotel and this orange one with middle line exactly your median.

***This figure shows the average price per room, depending on its type and the standard deviation. Note that due to data anonymization rooms with the same type letter may not necessarily be the same across hotels.***

**Now find how does the price per night vary over the year?**

KeshavMIPProject.ipynb

```
final=resort_hotel.merge(city_hotel,on='arrival_date_month')
final.columns=['month','price_for_resort','price_for_city_hotel']
final
```

	month	price_for_resort	price_for_city_hotel
0	April	75.867816	111.962267
1	August	181.205892	118.674598
2	December	68.410104	88.401855
3	February	54.147478	86.520062
4	January	48.761125	82.330983
5	July	150.122528	115.818019
6	June	107.974850	117.874360
7	March	57.056838	90.658533
8	May	76.657558	120.669827
9	November	48.706289	86.946592
10	October	61.775449	102.004672
11	September	96.416860	112.776582

completed at 1:02 AM

***Now we will observe over here is month column is not in order, & if we will visualise we will get improper conclusion***

***so very first we have to provide right hierarchy to the month column***

```
# !pip install sort-dataframeby-monthorweek
## Dependency package needs to be installed
## pip install sorted-months-weekdays
```

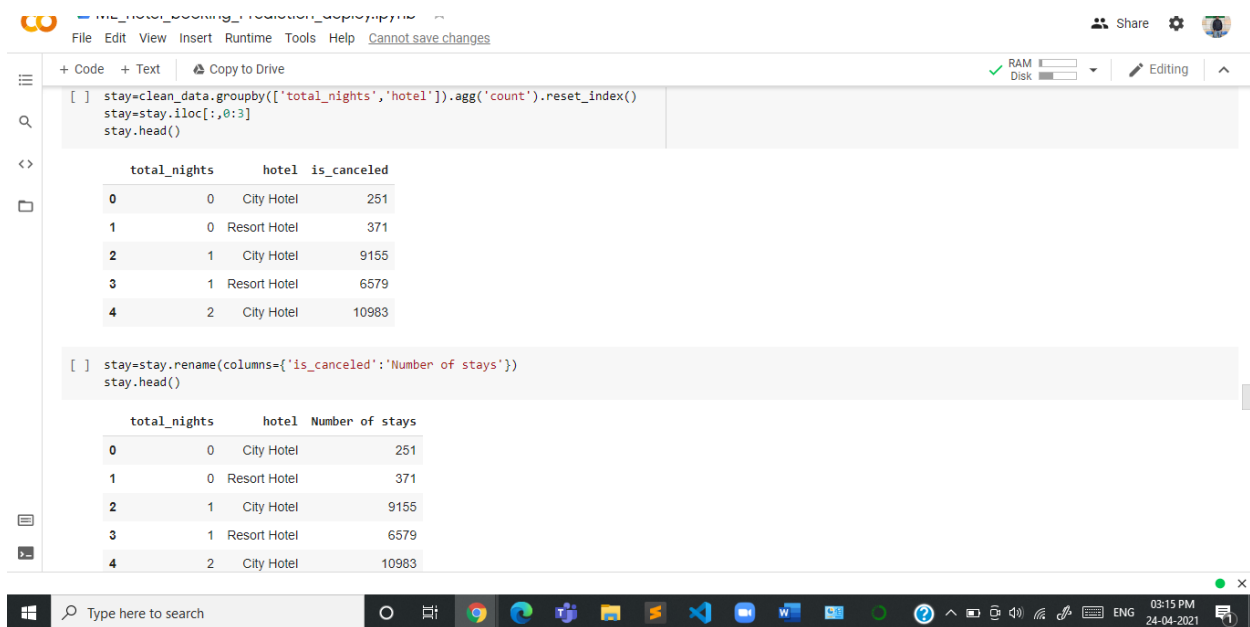
```
final=sort_data(final,'month')
final
```

	month	price_for_resort	price_for_city_hotel
0	January	48.761125	82.330983
1	February	54.147478	86.520062
2	March	57.056838	90.658533
3	April	75.867816	111.962267
4	May	76.657558	120.669827
5	June	107.974850	117.874360
6	July	150.122528	115.818019
7	August	181.205892	118.674598
8	September	96.416860	112.776582
9	October	61.775449	102.004672
10	November	48.706289	86.946592
11	December	68.410104	88.401855

```
[81] nx.line(final, x='month', y=[f'price_for_resort', 'price_for_city_hotel'], title='Room price per night over the Months')
```

**Conclusion-->> This clearly shows that the prices in the Resort hotel are much higher during the summer (no surprise here)., The price of the city hotel varies less and is most expensive during spring and autumn.**

## How long do people stay at the hotels?



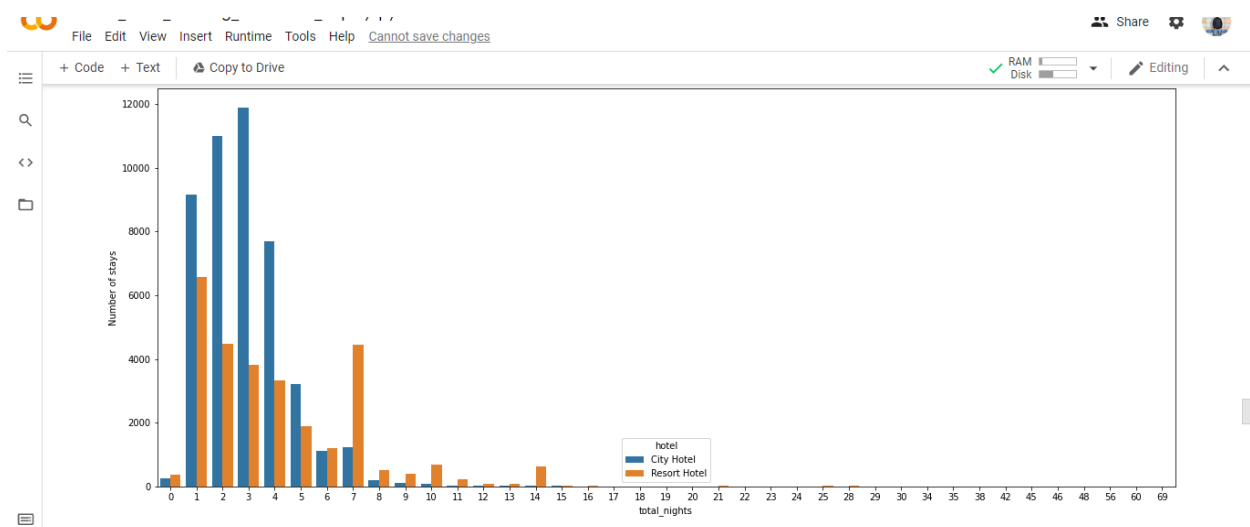
The screenshot shows a Jupyter Notebook interface with the following code and output:

```
[ ] stay=clean_data.groupby(['total_nights','hotel']).agg('count').reset_index()
    stay=stay.iloc[:,0:3]
    stay.head()
```

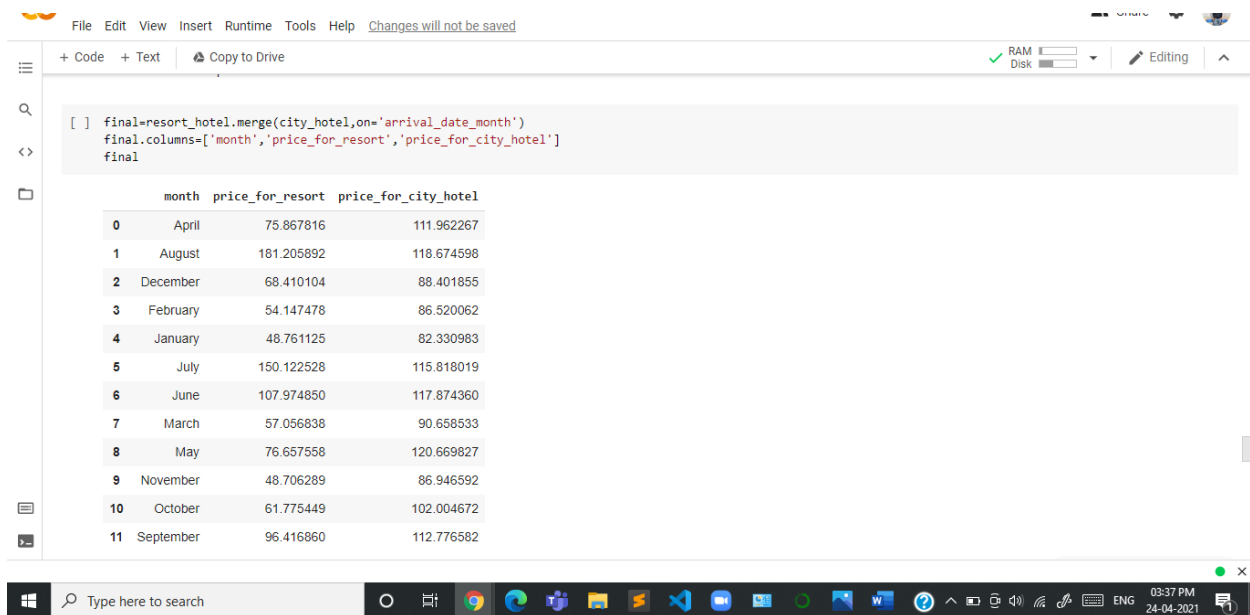
	total_nights	hotel	is_canceled
0	0	City Hotel	251
1	0	Resort Hotel	371
2	1	City Hotel	9155
3	1	Resort Hotel	6579
4	2	City Hotel	10983

```
[ ] stay=stay.rename(columns={'is_canceled':'Number of stays'})
    stay.head()
```

	total_nights	hotel	Number of stays
0	0	City Hotel	251
1	0	Resort Hotel	371
2	1	City Hotel	9155
3	1	Resort Hotel	6579
4	2	City Hotel	10983



## How does the price per night vary over the year?



*now we will observe over here is month column is not in order, & if we will visualise we will get improper conclusion.*

*so very first we have to provide right hierarchy to the month column.*

```
## !pip install sort-dataframeby-monthorweek
## Dependency package needs to be installed
## pip install sorted-months-weekdays
```



**Conclusion-->> This clearly shows that the prices in the Resort hotel are much higher during the summer (no surprise here)., The price of the city hotel varies less and is most expensive during spring and autumn.**

## **Which are the most busy month or in which months Guests are high?**

File Edit View Insert Runtime Tools Help *Changes will not be saved*

+ Code + Text Copy to Drive

RAM Disk Editing

```
final_rush=sort_data(final_rush,'month')
final_rush
```

	month	no of guests in resort	no of guest in city hotel
0	January	1866	2249
1	February	2308	3051
2	March	2571	4049
3	April	2550	4010
4	May	2535	4568
5	June	2037	4358
6	July	3137	4770
7	August	3257	5367
8	September	2102	4283
9	October	2575	4326
10	November	1975	2676
11	December	2014	2377

final\_rush dtypes



## Conclusion

The City hotel has more guests during spring and autumn, when the prices are also highest.

In July and August there are less visitors, although prices are lower.

Guest numbers for the Resort hotel go down slightly from June to September, which is also when the prices are highest.

Both hotels have the fewest guests during the winter.

## How long do people stay at the hotels?

Till now we have analyzed our data and we all understand our data in a very proper way. That what exactly the trend from where my other guests are, what exactly the distribution of the prize and how exactly the prices of a room that I am in, which month my guest is higher. We all have analyzed this trend as well.

So that's a time for your machine learning aspect.

Now, the very first task with respect to your machine learning approach is exactly you have to select some important features using correlation concept for your machine learning model.

```
co_relation=data.corr()  
co_relation
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
is_canceled	1.000000	0.292876	0.016622	0.008315	-0.005948	-0.001323	
lead_time	0.292876	1.000000	0.040334	0.127046	0.002306	0.085985	
arrival_date_year	0.016622	0.040334	1.000000	-0.540373	-0.000121	0.021694	
arrival_date_week_number	0.008315	0.127046	-0.540373	1.000000	0.066572	0.018629	
arrival_date_day_of_month	-0.005948	0.002306	-0.000121	0.066572	1.000000	-0.016225	
stays_in_weekend_nights	-0.001323	0.085985	0.021694	0.018629	-0.016225	1.000000	
stays_in_week_nights	0.025542	0.166892	0.031203	0.016047	-0.028362	0.494175	
adults	0.058182	0.117575	0.030266	0.026567	-0.001754	0.094759	
children	0.004851	-0.037878	0.054710	0.005556	0.014550	0.046135	
babies	-0.032569	-0.021003	-0.013192	0.010417	-0.000235	0.018607	
is_repeated_guest	-0.083745	-0.123209	0.010281	-0.031125	-0.006471	-0.086009	
previous_cancellations	0.110139	0.086025	-0.119905	0.035493	-0.027027	-0.012769	
previous_bookings_not_canceled	-0.057365	-0.073599	0.029234	-0.021009	-0.000306	-0.042859	
booking_changes	-0.144832	0.002230	0.031416	0.006311	0.011266	0.050191	
agent	-0.046770	-0.013114	0.056438	-0.018225	0.000159	0.162411	
company	-0.083594	-0.085854	0.033682	-0.032912	0.003667	-0.080783	

```
co_relation=data.corr()["is_canceled"]
co_relation
```

```
is_canceled      1.000000
lead_time        0.292876
arrival_date_year 0.016622
arrival_date_week_number 0.008315
arrival_date_day_of_month -0.005948
stays_in_weekend_nights -0.001323
stays_in_week_nights 0.025542
adults           0.058182
children         0.004851
babies           -0.032569
is_repeated_guest -0.083745
previous_cancellations 0.110139
previous_bookings_not_canceled -0.057365
```

```

booking_changes          -0.144832
agent                    -0.046770
company                  -0.083594
days_in_waiting_list    0.054301
adr                      0.046492
required_car_parking_spaces -0.195701
total_of_special_requests -0.234877
Name: is_canceled, dtype: float64

```

After all the eng" these are all my categorical features that we have to take care of it.

```

Index(['hotel', 'arrival_date_month', 'meal',
      'market_segment',
      'distribution_channel', 'reserved_room_type',
      'deposit_type',
      'customer_type', 'year', 'month', 'day',
      'cancellation'],
      dtype='object')

```

### **Now our aim is to derived some feature from the data.**

```

data_cat['year']=data_cat['reservation_status_date'].dt
.year
data_cat['month']=data_cat['reservation_status_date'].d
t.month
data_cat['day']=data_cat['reservation_status_date'].dt.
day

```

**It means yoy have to convert these string data into some integer format using some feature encoding.**

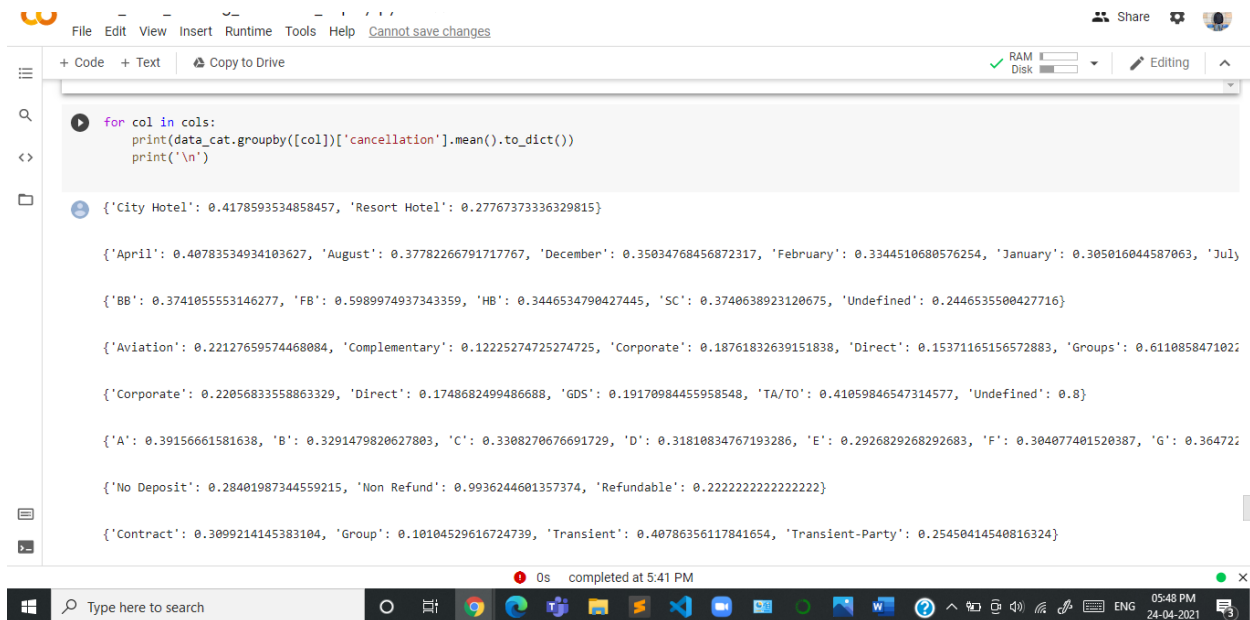


## Now we are going to apply feature encoding technique.

### Perform Mean Encoding Technique

```
cols=data_cat.columns[0:8]
cols
```

```
Index(['hotel', 'arrival_date_month', 'meal',
      'market_segment',
      'distribution_channel', 'reserved_room_type',
      'deposit_type',
      'customer_type'],
      dtype='object')
```



```
for col in cols:
    print(data_cat.groupby([col])['cancellation'].mean().to_dict())
    print('\n')
```

```
{'City Hotel': 0.4178593534858457, 'Resort Hotel': 0.27767373336329815}

{'April': 0.40783534934103627, 'August': 0.37782266791717767, 'December': 0.35034768456872317, 'February': 0.3344510680576254, 'January': 0.305016044587063, 'July': 0.3741055553146277, 'FB': 0.5989974937343359, 'HB': 0.3446534790427445, 'SC': 0.3740638923120675, 'Undefined': 0.2446535500427716}

{'Aviation': 0.22127659574468084, 'Complementary': 0.12225274725274725, 'Corporate': 0.18761832639151838, 'Direct': 0.15371165156572883, 'Groups': 0.6110858471022}

{'Corporate': 0.2205683358863329, 'Direct': 0.1748682499486688, 'GDS': 0.19170984455958548, 'TA/TO': 0.41059846547314577, 'Undefined': 0.8}

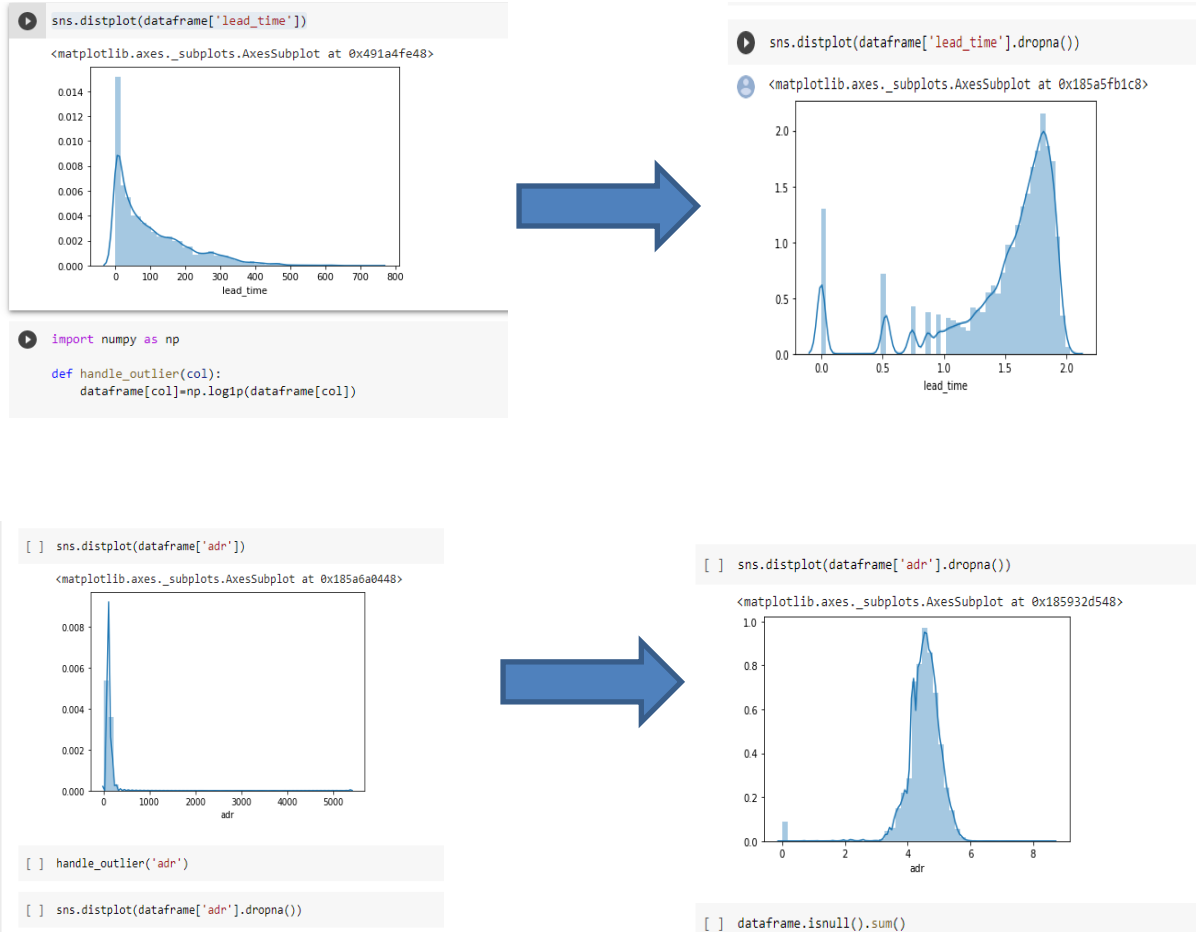
{'A': 0.39156661581638, 'B': 0.3291479820627803, 'C': 0.3308270676691729, 'D': 0.31810834767193286, 'E': 0.2926829268292683, 'F': 0.304077401520387, 'G': 0.364722}

{'No Deposit': 0.28401987344559215, 'Non Refund': 0.9936244601357374, 'Refundable': 0.2222222222222222}

{'Contract': 0.3099214145383104, 'Group': 0.10104529616724739, 'Transient': 0.40786356117841654, 'Transient-Party': 0.25450414540816324}
```

## Now we Handle Outliers in our data

```
sns.distplot(dataframe['lead_time'])
```



*And this time you will see it is somehow very close to your normal distribution so now this is exactly that you really want for your machine learning purpose.*

So, from this n number of feature you have to select some subset of features that are going to contribute more to my machine learning model.

```
# select a suitable alpha (equivalent of penalty).
# The bigger the alpha the less features that will be s
elected.
# remember to set the seed, the random state in this fu
nction.
```

```

# let's print the number of total and selected features

# this is how we can make a list of the selected features
selected_feat = cols[(feature_sel_model.get_support())]
print('total features: {}'.format((x.shape[1])))
print('selected features: {}'.format(len(selected_feat)
))
total features: 28
selected features: 16
selected_feat
Index(['deposit_type', 'year', 'month', 'day',
      'lead_time',
      'arrival_date_week_number',
      'stays_in_week_nights', 'adults',
      'children', 'previous_cancellations',
      'previous_bookings_not_canceled',
      'booking_changes', 'company', 'adr',
      'required_car_parking_spaces',
      'total_of_special_requests'],
      dtype='object')

```

Now it's time to apply our machine learning algorithm on data because up to a greater extent, our data is ready to apply your machine learning algorithm on our data.

Now we apply machine learning algorithm.

And then,

### **Cross validate our model.**

So for that, what we have to do we very first split our data in the form of training as well as your testing, because once you have trained data, you can train your model.

And once you have test data and you can check what exactly is the accuracy of your model.

## Splitting dataset & model Building

```

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,train_size=0.75,random_state=0)

from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(x_train,y_train)

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

[ ] y_pred=logreg.predict(x_test)

[ ] from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm

array([[15767, 2872],
       [ 4112, 7052]], dtype=int64)

[ ] from sklearn.metrics import accuracy_score

```

```

logreg.fit(x_train,y_train)

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)

[ ] y_pred=logreg.predict(x_test)

[ ] from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,y_pred)
cm

array([[15767, 2872],
       [ 4112, 7052]], dtype=int64)

[ ] from sklearn.metrics import accuracy_score
score=accuracy_score(y_test,y_pred)
score

0.7656611750494917

[ ]

```

Now you will see my Logistic Regression model has somewhere close to seventy seven percent accuracy. But this is actually not my exact accuracy .

**So how I can achieve my exact accuracy so in such scenarios you have to cross validate my model.**

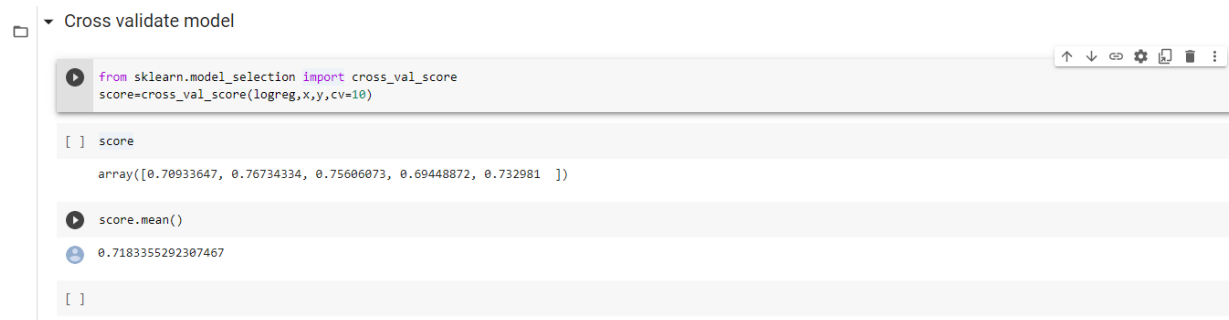
**If I change this random state value to any big number than the accuracy will be changed to any other percentage so than you will be thinking what is going here now in this we have to cross validate our model.**

## **random\_state**

**So whenever you will get your accuracy without cross validate your model, you have to always say I have achieved accuracy within this, this range.**

**It means now to get the exact accuracy you have to cross, validate your model for this.**

## **Cross validate model**



The screenshot shows a Jupyter Notebook interface with a tab titled "Cross validate model". The code cell contains the following Python code:

```
from sklearn.model_selection import cross_val_score
score=cross_val_score(logreg,x,y,cv=10)
```

The output of the code is displayed in two parts. The first part shows the variable `score` as an array of 10 values:

```
[ ] score
array([0.70933647, 0.76734334, 0.75606073, 0.69448872, 0.732981  ])
```

The second part shows the mean of the scores calculated by `score.mean()`:

```
[ ] score.mean()
0.7183355292307467
```

**And now you can see over here it has accuracy of somewhere close to 73 percent. It means you're 73 percent predictions are going to be correct.**

## **How can this model be used?**

1. This model will be helpful in highlighting the bookings which have high propensity of cancellation.
2. The hotels can use this model to impose high cancellation fee on bookings which the model has detected as likely to be cancelled. This will discourage the customer from cancelling or the booking will be done by a user who is less likely to cancel.

**I was able to predict the cancelation with 72.7% accuracy. I hope you find this project as interesting as I did!**

**Code-** [https://da.gd/my\\_code](https://da.gd/my_code)