Trabalho Final - Python

38BDT / Big Data Mining & Inteligência Artificial / Luiz Carlos Barboza Junior

---

Projeto de conclusão da matéria.

Seguem as orientações:

- *Relatório com 2 (duas) análises de negócio aplicando dois dos algoritmos que aprendemos d seguinte base de dados* (ou alguma outra da sua preferência): o https://www.kaggle.com/da
- *Obrigatória a descrição da análise de negócio realizada, não deve-se ser feita apenas a aplica*
- *Nenhum aluno pode utilizar a mesma BD do outro.*

Alunos

- 333519 - LUIZ FELIPE LOURENÇO MARTINS
- 333185 - MARCOS ANTONIO MACHADO DE BARROS

```
!pwd
```

```
/content
```

```
# Imports

import pandas as pd
import seaborn as sns
```

```
#CSV hotel_bookings gravado no Github

hotel_bookings = pd.read_csv('https://raw.githubusercontent.com/martinslfelipesap/ML-Pytho
hotel_bookings.head(5)
```

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

```
print(hotel_bookings.shape)
hotel_bookings.dtypes
```

```
(119390, 32)
hotel                              object
is_canceled                         int64
lead_time                           int64
arrival_date_year                   int64
arrival_date_month                 object
arrival_date_week_number            int64
arrival_date_day_of_month           int64
stays_in_weekend_nights             int64
stays_in_week_nights                int64
adults                              int64
children                          float64
babies                              int64
meal                               object
country                            object
market_segment                     object
distribution_channel               object
is_repeated_guest                   int64
previous_cancellations              int64
previous_bookings_not_canceled      int64
reserved_room_type                 object
assigned_room_type                 object
booking_changes                     int64
deposit_type                       object
agent                             float64
company                           float64
days_in_waiting_list                int64
customer_type                      object
adr                               float64
required_car_parking_spaces         int64
total_of_special_requests           int64
reservation_status                 object
reservation_status_date            object
dtype: object
```

```
# Recover Company por ter muitos nulos
hotel_bookings=hotel_bookings.drop(['company'],axis=1)
```

```
#Remover o restante de nulos
hotel_bookings=hotel_bookings.dropna(axis=0)
hotel_bookings.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                          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
agent                             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
dtype: int64
```

```
hotel_bookings.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 102894 entries, 3 to 119389
Data columns (total 31 columns):
hotel                           102894 non-null object
is_canceled                     102894 non-null int64
lead_time                       102894 non-null int64
arrival_date_year               102894 non-null int64
arrival_date_month              102894 non-null object
arrival_date_week_number        102894 non-null int64
arrival_date_day_of_month       102894 non-null int64
stays_in_weekend_nights         102894 non-null int64
stays_in_week_nights            102894 non-null int64
adults                          102894 non-null int64
children                        102894 non-null float64
babies                          102894 non-null int64
meal                            102894 non-null object
country                         102894 non-null object
market_segment                  102894 non-null object
distribution_channel            102894 non-null object
is_repeated_guest               102894 non-null int64
previous_cancellations          102894 non-null int64
previous_bookings_not_canceled  102894 non-null int64
reserved_room_type              102894 non-null object
assigned_room_type              102894 non-null object
booking_changes                 102894 non-null int64
deposit_type                    102894 non-null object
agent                           102894 non-null float64
days_in_waiting_list            102894 non-null int64
customer_type                   102894 non-null object
adr                             102894 non-null float64
required_car_parking_spaces     102894 non-null int64
total_of_special_requests       102894 non-null int64
reservation_status              102894 non-null object
reservation_status_date         102894 non-null object
dtypes: float64(3), int64(16), object(12)
memory usage: 25.1+ MB
```

```
#Garantir que não existe Nulo
hotel_bookings.isna().sum()
```

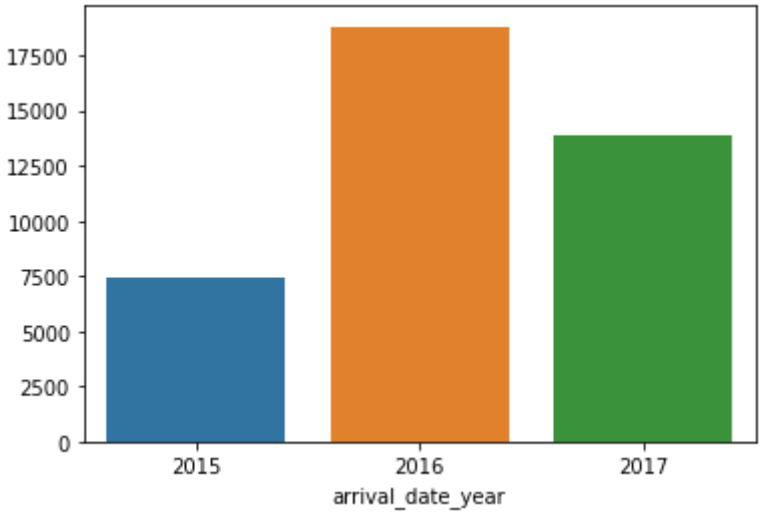
```
hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
children                          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
agent                             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
dtype: int64
```

```
hotel_bookings.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arr
count	102894.000000	102894.000000	102894.000000	102894.000000	
mean	0.390314	111.740092	2016.156977	27.339155	
std	0.487823	107.681013	0.706117	13.279990	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	26.000000	2016.000000	17.000000	
50%	0.000000	79.000000	2016.000000	28.000000	
75%	1.000000	169.000000	2017.000000	38.000000	
max	1.000000	629.000000	2017.000000	53.000000	

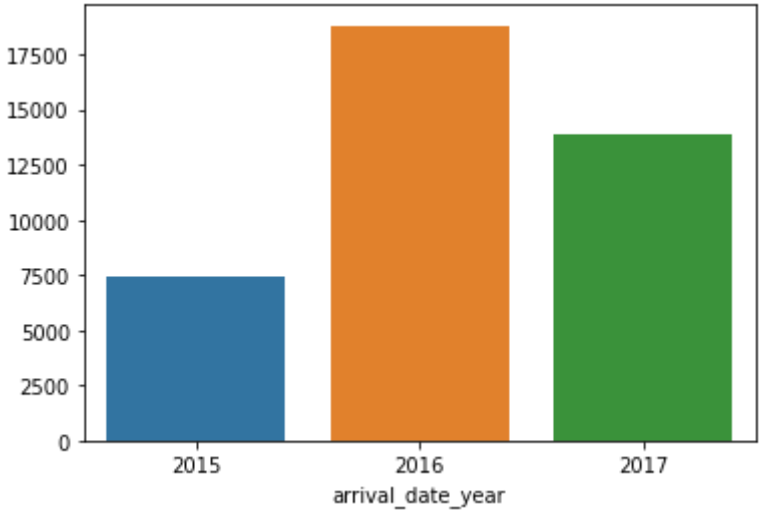
```
cancelaados_ano = hotel_bookings.groupby('arrival_date_year').sum()['is_canceled']
sns.barplot(cancelaados_ano.index, cancelaados_ano.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08f100b128>
```



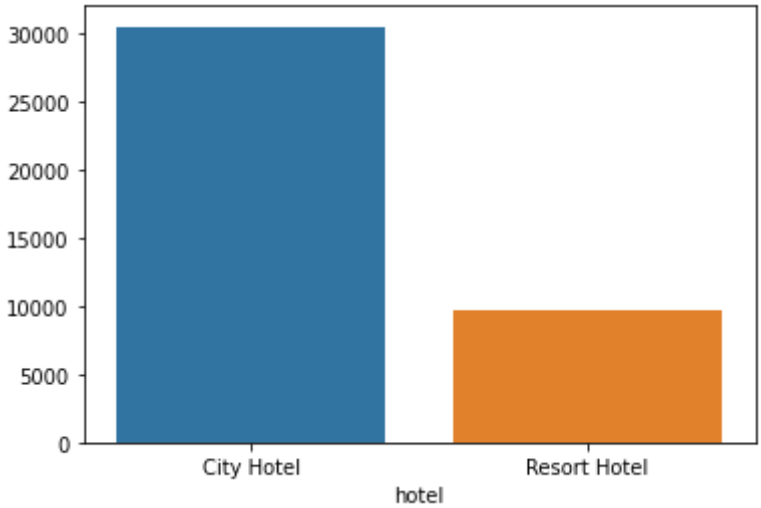

```
cancelaados_ano = hotel_bookings.groupby('arrival_date_year').sum()['is_canceled']
sns.barplot(cancelaados_ano.index, cancelaados_ano.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08f112e4e0>
```



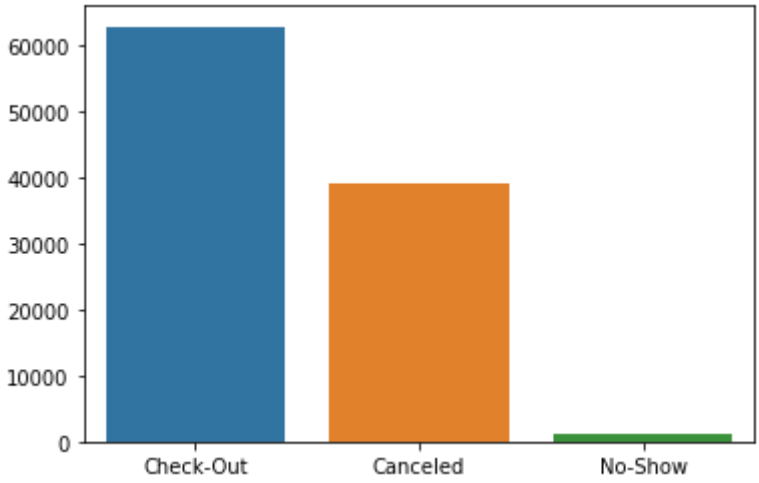

```
cancelaados_hotel = hotel_bookings.groupby('hotel').sum()['is_canceled']
sns.barplot(cancelaados_hotel.index, cancelaados_hotel.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08f0f3ed68>
```



```
reservation_status_qtd = hotel_bookings['reservation_status'].value_counts()
sns.barplot (reservation_status_qtd.index , reservation_status_qtd.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08f0f7a390>
```




```
#Copiando dados de hotel_bookings para hotel_bookings_novo para manter os dados originais
hotel_bookings_novo = hotel_bookings[:]


#Transformando de categorico para numérico
hotel_bookings_novo['hotel'].unique()

# Inserindo os valores na tabela hotel_bookings
hotel_bookings_novo['hotel']=hotel_bookings_novo['hotel'].map({'Resort Hotel':0,'City Hote
hotel_bookings_novo['hotel'].unique()

hotel_bookings_novo.head(10)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_da
3	0	0	13	2015	July	
4	0	0	14	2015	July	
5	0	0	14	2015	July	
7	0	0	9	2015	July	
8	0	1	85	2015	July	
9	0	1	75	2015	July	
10	0	1	23	2015	July	
11	0	0	35	2015	July	
12	0	0	68	2015	July	
13	0	0	18	2015	July	

```
#listando hotel_bookings com alteração na coluna hotel
hotel_bookings_novo.head(5)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_dat
3	0	0	13	2015	July	
4	0	0	14	2015	July	
5	0	0	14	2015	July	
7	0	0	9	2015	July	
8	0	1	85	2015	July	

```
#listandfo hotel_bookings para verificar se a coluna hotel mantém inalterada
hotel_bookings.head(5)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_dat
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	
5	Resort Hotel	0	14	2015	July	
7	Resort Hotel	0	9	2015	July	
8	Resort Hotel	1	85	2015	July	

```
#Transformando de categorico para numérico em hotel_bookings_novo
hotel_bookings_novo['arrival_date_month'].unique()

#Inserindo os valores em hotel_bookings_novo
hotel_bookings_novo['arrival_date_month']=hotel_bookings_novo['arrival_date_month'].map({'
                                                                ,'November':11,'December':
                                                                'April':4,'May':5,'June':6
hotel_bookings_novo['arrival_date_month'].unique()
```

```
array([ 7,  8,  9, 10, 11, 12,  1,  2,  3,  4,  5,  6])
```

```
# Apenas uma análise de correlação
hotel_bookings_novo.corr()
```

	hotel	is_canceled	lead_time	arrival_date_year
hotel	1.000000	0.116237	0.070860	0.023912
is_canceled	0.116237	1.000000	0.277666	0.004561
lead_time	0.070860	0.277666	1.000000	0.041082
arrival_date_year	0.023912	0.004561	0.041082	1.000000
arrival_date_month	-0.001759	0.006457	0.125228	-0.516963
arrival_date_week_number	-0.001085	0.004172	0.120155	-0.530387
arrival_date_day_of_month	0.008135	-0.005468	-0.003472	0.005206
stays_in_weekend_nights	-0.231985	-0.023589	0.047674	0.020693
stays_in_week_nights	-0.280023	0.003918	0.124195	0.032807
adults	-0.023438	0.037057	0.072477	0.061359
children	-0.058166	0.005409	-0.049796	0.065394
babies	-0.048754	-0.032038	-0.021975	-0.009828
is_repeated_guest	-0.050448	-0.029913	-0.055432	-0.022428
previous_cancellations	-0.007864	0.117983	0.100271	-0.134031
previous_bookings_not_canceled	-0.037883	-0.042589	-0.045082	0.015334
booking_changes	-0.059643	-0.131714	-0.000282	0.036468
agent	-0.790229	-0.081939	-0.068753	0.063684
days_in_waiting_list	0.074329	0.052475	0.170352	-0.061742
adr	0.061281	0.023284	-0.112652	0.222059
required_car_parking_spaces	-0.213978	-0.188298	-0.111226	-0.012864
total_of_special_requests	-0.072150	-0.257934	-0.123047	0.121490

```
# Aplica a codificação dos rótulos nos atributos categóricos pelo nome das colunas usando
```

```
from sklearn.preprocessing import LabelEncoder
labelencoder=LabelEncoder()
hotel_bookings_novo['meal']=labelencoder.fit_transform(hotel_bookings_novo['meal'])
hotel_bookings_novo['meal'].unique()
hotel_bookings_novo['country']=labelencoder.fit_transform(hotel_bookings_novo['country'])
hotel_bookings_novo['market_segment']=labelencoder.fit_transform(hotel_bookings_novo['mark
hotel_bookings_novo['distribution_channel']=labelencoder.fit_transform(hotel_bookings_novo
hotel_bookings_novo['reserved_room_type']=labelencoder.fit_transform(hotel_bookings_novo['
hotel_bookings_novo['assigned_room_type']=labelencoder.fit_transform(hotel_bookings_novo['
hotel_bookings_novo['deposit_type']=labelencoder.fit_transform(hotel_bookings_novo['deposi
hotel_bookings_novo['customer_type']=labelencoder.fit_transform(hotel_bookings_novo['custo
hotel_bookings_novo['reservation_status']=labelencoder.fit_transform(hotel_bookings_novo['
```

```
hotel_bookings_novo['reservation_status_date']=labelencoder.fit_transform(hotel_bookings_r

#Visualizando novos valores
hotel_bookings_novo.head(10)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_da
3	0	0	13	2015	7	
4	0	0	14	2015	7	
5	0	0	14	2015	7	
7	0	0	9	2015	7	
8	0	1	85	2015	7	
9	0	1	75	2015	7	
10	0	1	23	2015	7	
11	0	0	35	2015	7	
12	0	0	68	2015	7	
13	0	0	18	2015	7	

```
# Reunindo qual recurso é mais importante usando a função corr()
corr=hotel_bookings_novo.corr()['is_canceled']
corr.abs().sort_values(ascending=False)
```

```
is_canceled                       1.000000
reservation_status                0.921747
deposit_type                      0.459075
country                           0.296281
lead_time                         0.277666
total_of_special_requests         0.257934
required_car_parking_spaces       0.188298
reservation_status_date           0.180957
assigned_room_type                0.165903
booking_changes                   0.131714
previous_cancellations            0.117983
hotel                             0.116237
distribution_channel              0.097841
agent                             0.081939
reserved_room_type                0.061794
customer_type                     0.060653
days_in_waiting_list              0.052475
previous_bookings_not_canceled    0.042589
market_segment                    0.038758
adults                            0.037057
babies                            0.032038
meal                              0.030191
is_repeated_guest                 0.029913
stays_in_weekend_nights           0.023589
adr                               0.023284
arrival_date_month                0.006457
arrival_date_day_of_month         0.005468
children                          0.005409
arrival_date_year                 0.004561
arrival_date_week_number          0.004172
stays_in_week_nights              0.003918
Name: is_canceled, dtype: float64
```

```
#Foi decidido o corte de 0.30, logo o restante dos dados serão ignorados.

colunas=['is_repeated_guest','stays_in_weekend_nights','adr','arrival_date_month','arrival
     'arrival_date_year','arrival_date_week_number','stays_in_week_nights','reservation_st
hotel_bookings_novo=hotel_bookings_novo.drop(colunas,axis=1)
```

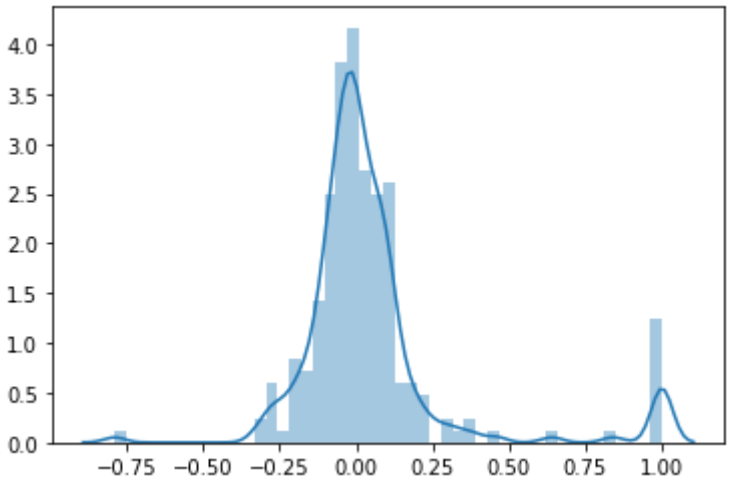
```
hotel_bookings_novo.head(10)
```

	hotel	is_canceled	lead_time	adults	babies	meal	country	market_segment	dis
3	0	0	13	1	0	0	59	2	
4	0	0	14	2	0	0	59	6	
5	0	0	14	2	0	0	59	6	
7	0	0	9	2	0	1	134	3	
8	0	1	85	2	0	0	134	6	
9	0	1	75	2	0	2	134	5	
10	0	1	23	2	0	0	134	6	
11	0	0	35	2	0	2	134	6	
12	0	0	68	2	0	0	166	6	
13	0	0	18	2	0	2	51	6	

```
#Analisando os dados por sb.distplot

corr=hotel_bookings_novo.corr()
sns.distplot(corr)
```

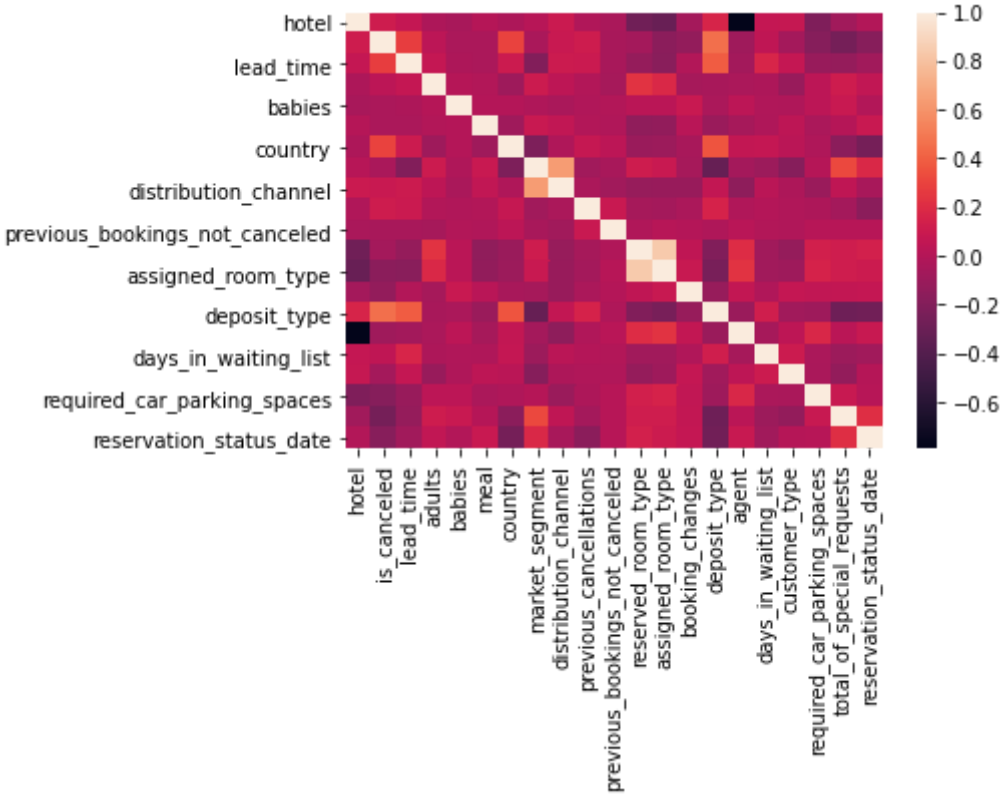
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08e784add8>
```




```
#Analisando os dados por mapa de calor

sns.heatmap(hotel_bookings_novo.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08eb7f5da0>
```



Clusterização usando KMeans

```
from sklearn.cluster import KMeans

k = KMeans(n_clusters=2)
k.fit(hotel_bookings_novo)
```

```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

```
sns.scatterplot(hotel_bookings_novo['deposit_type'], hotel_bookings_novo['is_canceled'], h
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f08a9d36b00>
```

Regressao Linear

```
from sklearn.linear_model import LinearRegression

rl = LinearRegression()

rl.fit(hotel_bookings_novo[['deposit_type']] , hotel_bookings_novo['is_canceled'])
print(rl.coef_,rl.intercept_)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-635d22bd0e44> in <module>()
      3 rl = LinearRegression()
      4
----> 5 rl.fit(hotel_bookings_novo[['deposit_type']] , hotel_bookings_novo['is_cancel
      6 print(rl.coef_,rl.intercept_)

NameError: name 'hotel_bookings_novo' is not defined
```

SEARCH STACK OVERFLOW

```
canc_fut = pd.DataFrame({'deposit_type':[0,1]})
rl.predict(canc_fut)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-1be4fb62bc99> in <module>()
----> 1 canc_fut = pd.DataFrame({'deposit_type':[0,1]})
      2 rl.predict(canc_fut)

NameError: name 'pd' is not defined
```

SEARCH STACK OVERFLOW

```
sns.distplot(canc_fut)
```