

```
!pip -q install plotly --upgrade
```

Importando bibliotecas

Abrindo um arquivo CSV do drive

```
C:\Users\Felipe\AppData\Local\Temp\ipykernel_452\2162366225.py:1:
FutureWarning: In a future version of pandas all arguments of read_csv
except for the argument 'filepath_or_buffer' will be keyword-only.
    base = pd.read_csv('restaurante.csv', ';', usecols= lambda x:x not
in ['Exemplo'])
```

```
#base2 = pd.read_csv('restaurante.csv', ';', usecols=['Alternativo',
'Bar'])
#base2
```

	Alternativo	Bar	Sex/Sab	Fome	Cliente	Preço	Chuva	Res	Tipo
0 Tempo \									
0-10	Sim	Nao	Nao	Sim	Alguns	RRR	Nao	Sim	Frances
1 30-60	Sim	Nao	Nao	Sim	Cheio	R	Nao	Nao	Tailandes
2 0-10	Nao	Sim	Nao	Nao	Alguns	R	Nao	Nao	Hamburger
3 out/30	Sim	Nao	Sim	Sim	Cheio	R	Sim	Nao	Tailandes
4 >60	Sim	Nao	Sim	Nao	Cheio	RRR	Nao	Sim	Frances
5 0-10	Nao	Sim	Nao	Sim	Alguns	RR	Sim	Sim	Italiano

6 0-10	Nao	Sim	Nao	Nao	Nenhum	R	Sim	Nao	Hamburger
7 0-10	Nao	Nao	Nao	Sim	Alguns	RR	Sim	Sim	Tailandes
8 >60	Nao	Sim	Sim	Nao	Cheio	R	Sim	Nao	Hamburger
9 out/30	Sim	Sim	Sim	Sim	Cheio	RRR	Nao	Sim	Italiano
10 0-10	Nao	Nao	Nao	Nao	Nenhum	R	Nao	Nao	Tailandes
11 30-60	Sim	Sim	Sim	Sim	Cheio	R	Nao	Nao	Hamburger

	Conc
0	Sim
1	Nao
2	Sim
3	Sim
4	Nao
5	Sim
6	Nao
7	Sim
8	Nao
9	Nao
10	Nao
11	Sim

```
base.head(3)
```

Alternativo	Bar	Sex/Sab	Fome	Cliente	Preco	Chuva	Res	Tipo	
0 Tempo \									
0-10	Sim	Nao	Nao	Sim	Alguns	RRR	Nao	Sim	Frances
1 30-60	Sim	Nao	Nao	Sim	Cheio	R	Nao	Nao	Tailandes
2 0-10	Nao	Sim	Nao	Nao	Alguns	R	Nao	Nao	Hamburger

	Conc
0	Sim
1	Nao
2	Sim

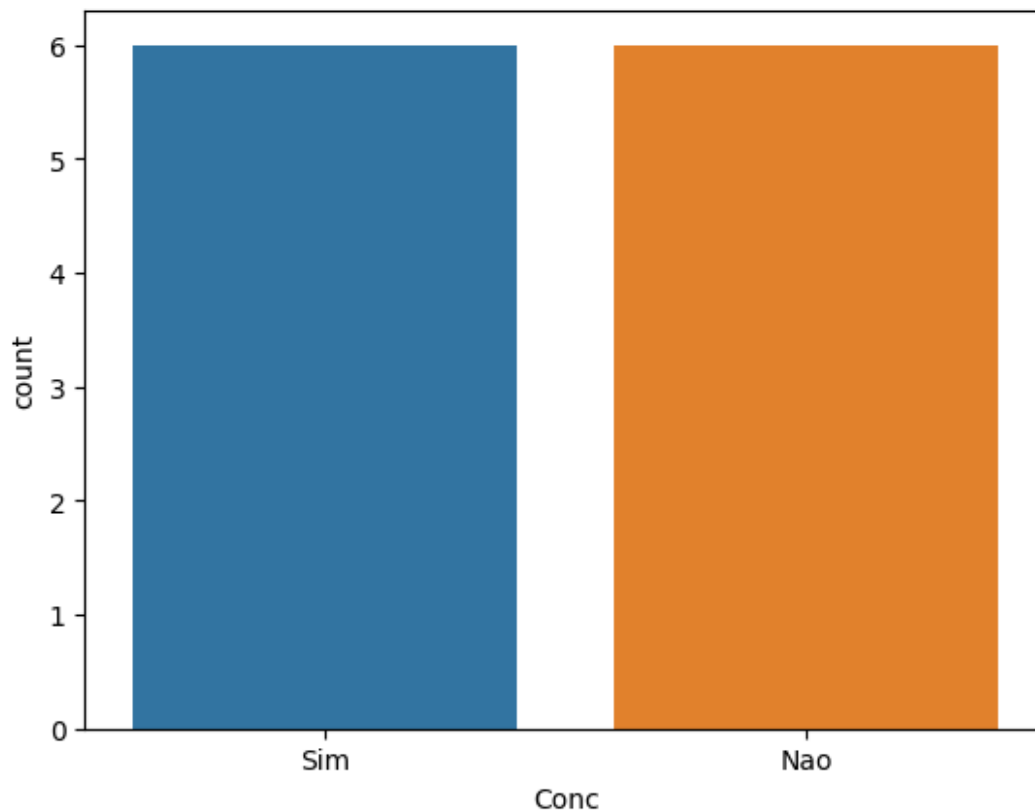
```
base.tail(2)
```

Tempo \ Alternativo	Bar	Sex/Sab	Fome	Cliente	Preço	Chuva	Res	Tipo	
10-10	Nao	Nao	Nao	Nao	Nenhum	R	Nao	Nao	Tailandes
11-30-60	Sim	Sim	Sim	Sim	Cheio	R	Nao	Nao	Hamburger

```
Conc
10  Nao
11  Sim
```

Contando quantidade de instâncias

```
np.unique(base['Conc'], return_counts=True)
(array(['Nao', 'Sim'], dtype=object), array([6, 6], dtype=int64))
sns.countplot(x = base['Conc']);
```



Separando os atributos de entrada e de classe

```
X_prev = base.iloc[:, 0:10].values
X_prev
array([[ 'Sim', 'Nao', 'Nao', 'Sim', 'Alguns', 'RRR', 'Nao', 'Sim',
        'Frances', '0-10'],
       [ 'Sim', 'Nao', 'Nao', 'Sim', 'Cheio', 'R', 'Nao', 'Nao',
        'Tailandes', '30-60'],
       [ 'Nao', 'Sim', 'Nao', 'Nao', 'Alguns', 'R', 'Nao', 'Nao',
        'Hamburger', '0-10'],
       [ 'Sim', 'Nao', 'Sim', 'Sim', 'Cheio', 'R', 'Sim', 'Nao',
```

```

    'Tailandes', 'out/30'],
    ['Sim', 'Nao', 'Sim', 'Nao', 'Cheio', 'RRR', 'Nao', 'Sim',
     'Frances', '>60'],
    ['Nao', 'Sim', 'Nao', 'Sim', 'Alguns', 'RR', 'Sim', 'Sim',
     'Italiano', '0-10'],
    ['Nao', 'Sim', 'Nao', 'Nao', 'Nenhum', 'R', 'Sim', 'Nao',
     'Hamburger', '0-10'],
    ['Nao', 'Nao', 'Nao', 'Sim', 'Alguns', 'RR', 'Sim', 'Sim',
     'Tailandes', '0-10'],
    ['Nao', 'Sim', 'Sim', 'Nao', 'Cheio', 'R', 'Sim', 'Nao',
     'Hamburger', '>60'],
    ['Sim', 'Sim', 'Sim', 'Sim', 'Cheio', 'RRR', 'Nao', 'Sim',
     'Italiano', 'out/30'],
    ['Nao', 'Nao', 'Nao', 'Nao', 'Nenhum', 'R', 'Nao', 'Nao',
     'Tailandes', '0-10'],
    ['Sim', 'Sim', 'Sim', 'Sim', 'Cheio', 'R', 'Nao', 'Nao',
     'Hamburger', '30-60']], dtype=object)

```

```
X_prev_label = base.iloc[:, 0:10]
```

```
X_prev_label
```

	Alternativo	Bar	Sex/Sab	Fome	Cliente	Preco	Chuva	Res	Tipo
Tempo									
0	Sim	Nao	Nao	Sim	Alguns	RRR	Nao	Sim	Frances
0-10									
1	Sim	Nao	Nao	Sim	Cheio	R	Nao	Nao	Tailandes
30-60									
2	Nao	Sim	Nao	Nao	Alguns	R	Nao	Nao	Hamburger
0-10									
3	Sim	Nao	Sim	Sim	Cheio	R	Sim	Nao	Tailandes
out/30									
4	Sim	Nao	Sim	Nao	Cheio	RRR	Nao	Sim	Frances
>60									
5	Nao	Sim	Nao	Sim	Alguns	RR	Sim	Sim	Italiano
0-10									
6	Nao	Sim	Nao	Nao	Nenhum	R	Sim	Nao	Hamburger
0-10									
7	Nao	Nao	Nao	Sim	Alguns	RR	Sim	Sim	Tailandes
0-10									
8	Nao	Sim	Sim	Nao	Cheio	R	Sim	Nao	Hamburger
>60									
9	Sim	Sim	Sim	Sim	Cheio	RRR	Nao	Sim	Italiano
out/30									
10	Nao	Nao	Nao	Nao	Nenhum	R	Nao	Nao	Tailandes
0-10									
11	Sim	Sim	Sim	Sim	Cheio	R	Nao	Nao	Hamburger
30-60									

```
X_prev[:,5]
```

```
array(['RRR', 'R', 'R', 'R', 'RRR', 'RR', 'R', 'RR', 'R', 'RRR', 'R',
      'R'],
      dtype=object)
```

```
y_classe = base.iloc[:, 10].values
```

```
y_classe
```

```
array(['Sim', 'Nao', 'Sim', 'Sim', 'Nao', 'Sim', 'Nao', 'Sim', 'Nao',
      'Nao', 'Nao', 'Sim'], dtype=object)
```

Podemos fazer esta mesma divisão (separar as variáveis de entrada e saída) usando os comandos abaixo

```
#X = base.copy()
#y = X.pop('Conc')
```

Tratamento de dados categóricos

*LabelEncoder - Vamos tratar os dados categóricos colocando 1, 2, 3 e etc**

```
from sklearn.preprocessing import LabelEncoder
```

```
label_encoder = LabelEncoder() #binarizador
```

```
X_prev[:,0]
```

```
array(['Sim', 'Sim', 'Nao', 'Sim', 'Sim', 'Nao', 'Nao', 'Nao', 'Nao',
      'Sim', 'Nao', 'Sim'], dtype=object)
```

```
X_prev
```

```
array([[ 'Sim', 'Nao', 'Nao', 'Sim', 'Alguns', 'RRR', 'Nao', 'Sim',
        'Frances', '0-10'],
       [ 'Sim', 'Nao', 'Nao', 'Sim', 'Cheio', 'R', 'Nao', 'Nao',
        'Tailandes', '30-60'],
       [ 'Nao', 'Sim', 'Nao', 'Nao', 'Alguns', 'R', 'Nao', 'Nao',
        'Hamburger', '0-10'],
       [ 'Sim', 'Nao', 'Sim', 'Sim', 'Cheio', 'R', 'Sim', 'Nao',
        'Tailandes', 'out/30'],
       [ 'Sim', 'Nao', 'Sim', 'Nao', 'Cheio', 'RRR', 'Nao', 'Sim',
        'Frances', '>60'],
       [ 'Nao', 'Sim', 'Nao', 'Sim', 'Alguns', 'RR', 'Sim', 'Sim',
        'Italiano', '0-10'],
       [ 'Nao', 'Sim', 'Nao', 'Nao', 'Nenhum', 'R', 'Sim', 'Nao',
        'Hamburger', '0-10'],
       [ 'Nao', 'Nao', 'Nao', 'Sim', 'Alguns', 'RR', 'Sim', 'Sim',
        'Tailandes', '0-10'],
       [ 'Nao', 'Sim', 'Sim', 'Nao', 'Cheio', 'R', 'Sim', 'Nao',
        'Hamburger', '>60'],
       [ 'Sim', 'Sim', 'Sim', 'Sim', 'Cheio', 'RRR', 'Nao', 'Sim',
        'Italiano', 'out/30'],
       [ 'Nao', 'Nao', 'Nao', 'Nao', 'Nenhum', 'R', 'Nao', 'Nao',
```

```

        'Tailandes', '0-10'],
        ['Sim', 'Sim', 'Sim', 'Sim', 'Cheio', 'R', 'Nao', 'Nao',
        'Hamburger', '30-60']], dtype=object)

label_encoder_Alternativo = LabelEncoder()
label_encoder_Bar = LabelEncoder()
label_encoder_SexSab = LabelEncoder()
label_encoder_fome = LabelEncoder()
label_encoder_chuva = LabelEncoder()
label_encoder_Res = LabelEncoder()
label_encoder_preco = LabelEncoder()
label_encoder_cliente = LabelEncoder()
label_encoder_tempo = LabelEncoder()

X_prev[:,0] = label_encoder_Alternativo.fit_transform(X_prev[:,0])
X_prev[:,1] = label_encoder_Bar.fit_transform(X_prev[:,1])
X_prev[:,2] = label_encoder_SexSab.fit_transform(X_prev[:,2])
X_prev[:,3] = label_encoder_fome.fit_transform(X_prev[:,3])
X_prev[:,4] = label_encoder_cliente.fit_transform(X_prev[:,4])
X_prev[:,5] = label_encoder_preco.fit_transform(X_prev[:,5])
X_prev[:,6] = label_encoder_chuva.fit_transform(X_prev[:,6])
X_prev[:,7] = label_encoder_Res.fit_transform(X_prev[:,7]) # 4, 5, 9
X_prev[:,9] = label_encoder_tempo.fit_transform(X_prev[:,9])

X_prev
array([[1, 0, 0, 1, 0, 2, 0, 1, 'Frances', 0],
       [1, 0, 0, 1, 1, 0, 0, 0, 'Tailandes', 1],
       [0, 1, 0, 0, 0, 0, 0, 0, 'Hamburger', 0],
       [1, 0, 1, 1, 1, 0, 1, 0, 'Tailandes', 3],
       [1, 0, 1, 0, 1, 2, 0, 1, 'Frances', 2],
       [0, 1, 0, 1, 0, 1, 1, 1, 'Italiano', 0],
       [0, 1, 0, 0, 2, 0, 1, 0, 'Hamburger', 0],
       [0, 0, 0, 1, 0, 1, 1, 1, 'Tailandes', 0],
       [0, 1, 1, 0, 1, 0, 1, 0, 'Hamburger', 2],
       [1, 1, 1, 1, 1, 2, 0, 1, 'Italiano', 3],
       [0, 0, 0, 0, 2, 0, 0, 0, 'Tailandes', 0],
       [1, 1, 1, 1, 1, 0, 0, 0, 'Hamburger', 1]], dtype=object)

```

OneHotEncoder - Agora vamos binarizar atributos não ordinais

Contando quantas opções de resposta tem cada atributo

```
len(np.unique(base['Cliente']))
```

3

```

from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer

```

X_prev

```
array([[1, 0, 0, 1, 0, 2, 0, 1, 'Frances', 0],
       [1, 0, 0, 1, 1, 0, 0, 0, 'Tailandes', 1],
       [0, 1, 0, 0, 0, 0, 0, 0, 'Hamburger', 0],
       [1, 0, 1, 1, 1, 0, 1, 0, 'Tailandes', 3],
       [1, 0, 1, 0, 1, 2, 0, 1, 'Frances', 2],
       [0, 1, 0, 1, 0, 1, 1, 1, 'Italiano', 0],
       [0, 1, 0, 0, 2, 0, 1, 0, 'Hamburger', 0],
       [0, 0, 0, 1, 0, 1, 1, 1, 'Tailandes', 0],
       [0, 1, 1, 0, 1, 0, 1, 0, 'Hamburger', 2],
       [1, 1, 1, 1, 1, 2, 0, 1, 'Italiano', 3],
       [0, 0, 0, 0, 2, 0, 0, 0, 'Tailandes', 0],
       [1, 1, 1, 1, 1, 0, 0, 0, 'Hamburger', 1]], dtype=object)
```

```
X_prev[:,0:9]
```

```
array([[1, 0, 0, 1, 0, 2, 0, 1, 'Frances'],
       [1, 0, 0, 1, 1, 0, 0, 0, 'Tailandes'],
       [0, 1, 0, 0, 0, 0, 0, 0, 'Hamburger'],
       [1, 0, 1, 1, 1, 0, 1, 0, 'Tailandes'],
       [1, 0, 1, 0, 1, 2, 0, 1, 'Frances'],
       [0, 1, 0, 1, 0, 1, 1, 1, 'Italiano'],
       [0, 1, 0, 0, 2, 0, 1, 0, 'Hamburger'],
       [0, 0, 0, 1, 0, 1, 1, 1, 'Tailandes'],
       [0, 1, 1, 0, 1, 0, 1, 0, 'Hamburger'],
       [1, 1, 1, 1, 1, 2, 0, 1, 'Italiano'],
       [0, 0, 0, 0, 2, 0, 0, 0, 'Tailandes'],
       [1, 1, 1, 1, 1, 0, 0, 0, 'Hamburger']], dtype=object)
```

```
onehotencoder_restaurante = ColumnTransformer(transformers=[('OneHot',
OneHotEncoder(), [8])], remainder='passthrough')
```

```
X_prev= onehotencoder_restaurante.fit_transform(X_prev)
```

```
X_prev
```

```
array([[1.0, 0.0, 0.0, 0.0, 1, 0, 0, 1, 0, 2, 0, 1, 0],
       [0.0, 0.0, 0.0, 1.0, 1, 0, 0, 1, 1, 0, 0, 0, 1],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
       [0.0, 0.0, 0.0, 1.0, 1, 0, 1, 1, 1, 0, 1, 0, 3],
       [1.0, 0.0, 0.0, 0.0, 1, 0, 1, 0, 1, 2, 0, 1, 2],
       [0.0, 0.0, 1.0, 0.0, 0, 1, 0, 1, 0, 1, 1, 1, 0],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 0, 0, 2, 0, 1, 0, 0],
       [0.0, 0.0, 0.0, 1.0, 0, 0, 0, 1, 0, 1, 1, 1, 0],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 1, 0, 1, 0, 1, 0, 2],
       [0.0, 0.0, 1.0, 0.0, 1, 1, 1, 1, 1, 2, 0, 1, 3],
       [0.0, 0.0, 0.0, 1.0, 0, 0, 0, 0, 2, 0, 0, 0, 0],
       [0.0, 1.0, 0.0, 0.0, 1, 1, 1, 1, 1, 0, 0, 0, 1]], dtype=object)
```

```
X_prev.shape
```

```
(12, 13)
```

Esta mesma transformação OneHotEncoder pode ser feita com o comando abaixo

```
#base = pd.get_dummies(X_prev_label)
#base.head()
```

Método de amostragem Holdout

```
from sklearn.model_selection import train_test_split
```

X_prev

```
array([[1.0, 0.0, 0.0, 0.0, 1, 0, 0, 1, 0, 2, 0, 1, 0],
       [0.0, 0.0, 0.0, 1.0, 1, 0, 0, 1, 1, 0, 0, 0, 1],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 0, 0, 0, 0, 0, 0, 0],
       [0.0, 0.0, 0.0, 1.0, 1, 0, 1, 1, 1, 0, 1, 0, 3],
       [1.0, 0.0, 0.0, 0.0, 1, 0, 1, 0, 1, 2, 0, 1, 2],
       [0.0, 0.0, 1.0, 0.0, 0, 1, 0, 1, 0, 1, 1, 1, 0],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 0, 0, 2, 0, 1, 0, 0],
       [0.0, 0.0, 0.0, 1.0, 0, 0, 0, 1, 0, 1, 1, 1, 0],
       [0.0, 1.0, 0.0, 0.0, 0, 1, 1, 0, 1, 0, 1, 0, 2],
       [0.0, 0.0, 1.0, 0.0, 1, 1, 1, 1, 1, 2, 0, 1, 3],
       [0.0, 0.0, 0.0, 1.0, 0, 0, 0, 0, 2, 0, 0, 0, 0],
       [0.0, 1.0, 0.0, 0.0, 1, 1, 1, 1, 1, 0, 0, 0, 1]], dtype=object)
```

y_classe

```
array(['Sim', 'Nao', 'Sim', 'Sim', 'Nao', 'Sim', 'Nao', 'Sim', 'Nao',
       'Nao', 'Nao', 'Sim'], dtype=object)
```

y_classe.shape

(12,)

```
#X_train_ds, X_test_ds, y_train_ds, y_test_ds = train_test_split(X, y,
test_size=0.3, random_state=123, shuffle=True, stratify=y)
X_treino, X_teste, y_treino, y_teste = train_test_split(X_prev,
y_classe, test_size = 0.20, random_state = 0)
```

X_treino.shape

(9, 13)

X_teste.shape

(3, 13)

X_teste

```
array([[0.0, 1.0, 0.0, 0.0, 0, 1, 0, 0, 2, 0, 1, 0, 0],
       [0.0, 1.0, 0.0, 0.0, 1, 1, 1, 1, 1, 0, 0, 0, 1],
       [1.0, 0.0, 0.0, 0.0, 1, 0, 1, 0, 1, 2, 0, 1, 2]], dtype=object)
```

y_treino


```
array(['Nao', 'Sim', 'Nao', 'Nao', 'Sim', 'Nao', 'Sim', 'Sim', 'Sim'],  
      dtype=object)  
  
y_teste  
  
array(['Nao', 'Sim', 'Nao'], dtype=object)  
  
import pickle  
  
with open('restaurante.pkl', mode = 'wb') as f:  
    pickle.dump([X_treino, X_teste, y_treino, y_teste], f)
```