

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
C:\ProgramData\Anaconda3\lib\importlib\_bootstrap.py:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incompatibility. Expected 192 from C header, got 216 from PyObject
    return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\importlib\_bootstrap.py:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incompatibility. Expected 192 from C header, got 216 from PyObject
    return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\importlib\_bootstrap.py:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incompatibility. Expected 192 from C header, got 216 from PyObject
    return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\importlib\_bootstrap.py:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incompatibility. Expected 216, got 192
    return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\importlib\_bootstrap.py:219: RuntimeWarning: numpy.ufunc size changed, may indicate binary incompatibility. Expected 192 from C header, got 216 from PyObject
    return f(*args, **kwargs)
```

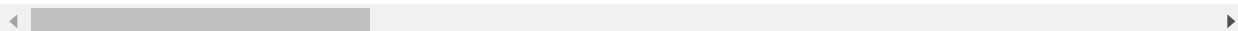
```
In [2]: df=pd.read_csv("C:/Users/DELL/Desktop/pizza.csv")
```

```
In [3]: df.head()
```

Out[3]:

	id	dateAdded	dateUpdated	address	categories	primary
0	AVz3Y-7h3D1zeR_xDAqm	2017-06-30T05:05:40Z	2019-05-01T15:43:09Z	4203 E Kiehl Ave	Pizza,Restaurant,American restaurants,Pizza Pl...	Acco F
1	AVweGPFF_7pvs4fzAAzQ	2016-04-02T04:02:49Z	2019-05-01T15:27:50Z	25 E Camelback Rd	Pizza,Pizza Place,Restaurants	Acco F
2	AVwdRGa9_7pvs4fz4E3K	2016-03-03T18:39:49Z	2019-05-01T12:52:25Z	3703 Paxton Ave	Restaurant,Pizza Place,Restaurants	Acco F
3	AVwdX4psIN2L1WUfvJB1	2016-03-29T05:08:59Z	2019-05-01T12:52:20Z	30495 John R Rd	Pizza,Carry-out food,Pizza Place,Restaurants	Acco F
4	AVwdaeTtkufWRAb55pSH	2016-03-31T02:34:04Z	2019-05-01T12:50:45Z	3600 Eastern Ave	Pizza,American restaurants,Pizza Place,Pizza e...	Acco F

5 rows × 7 columns



In [4]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 24 columns):
id                10000 non-null object
dateAdded         10000 non-null object
dateUpdated       10000 non-null object
address           10000 non-null object
categories        10000 non-null object
primaryCategories 10000 non-null object
city              10000 non-null object
country           10000 non-null object
keys              10000 non-null object
latitude          10000 non-null float64
longitude         10000 non-null float64
menuPageURL       1679 non-null object
menus.amountMax   10000 non-null float64
menus.amountMin   10000 non-null float64
menus.currency    10000 non-null object
menus.dateSeen    10000 non-null object
menus             3718 non-null object
menus.name        10000 non-null object
name              10000 non-null object
postalCode        9996 non-null object
priceRangeCurrency 10000 non-null object
priceRangeMin     10000 non-null int64
priceRangeMax     10000 non-null int64
province          10000 non-null object
dtypes: float64(4), int64(2), object(18)
memory usage: 1.8+ MB
```

In [7]: `df.shape`

Out[7]: (10000, 24)

In [8]: `df.describe()`

Out[8]:

	latitude	longitude	menus.amountMax	menus.amountMin	priceRangeMin	priceRan
<b>count</b>	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
<b>mean</b>	38.371895	-90.063800	14.032670	13.416902	4.655500	27.718750
<b>std</b>	4.573537	17.340379	17.607233	10.742678	9.828412	6.000000
<b>min</b>	21.421326	-157.802443	0.000000	0.000000	0.000000	7.000000
<b>25%</b>	34.422946	-104.801126	8.490000	8.000000	0.000000	25.000000
<b>50%</b>	40.118880	-82.909949	13.390000	12.990000	0.000000	25.000000
<b>75%</b>	40.905048	-75.194741	17.950000	16.990000	0.000000	25.000000
<b>max</b>	64.850500	-71.946170	1395.000000	243.000000	50.000000	55.000000

```
In [9]: df_cat=df.select_dtypes(include=[object])
df_num=df.select_dtypes(include=[np.number])
```

```
In [10]: df_cat.head()
```

Out[10]:

	id	dateAdded	dateUpdated	address	categories	primary
0	AVz3Y-7h3D1zeR_xDAqm	2017-06-30T05:05:40Z	2019-05-01T15:43:09Z	4203 E Kiehl Ave	Pizza,Restaurant,American restaurants,Pizza Pl...	Acco F
1	AVweGPFF_7pvs4fzAAzQ	2016-04-02T04:02:49Z	2019-05-01T15:27:50Z	25 E Camelback Rd	Pizza,Pizza Place,Restaurants	Acco F
2	AVwdRGa9_7pvs4fz4E3K	2016-03-03T18:39:49Z	2019-05-01T12:52:25Z	3703 Paxton Ave	Restaurant,Pizza Place,Restaurants	Acco F
3	AVwdX4psIN2L1WUfvJB1	2016-03-29T05:08:59Z	2019-05-01T12:52:20Z	30495 John R Rd	Pizza,Carry-out food,Pizza Place,Restaurants	Acco F
4	AVwdaeTtkufWRAb55pSH	2016-03-31T02:34:04Z	2019-05-01T12:50:45Z	3600 Eastern Ave	Pizza,American restaurants,Pizza Place,Pizza e...	Acco F

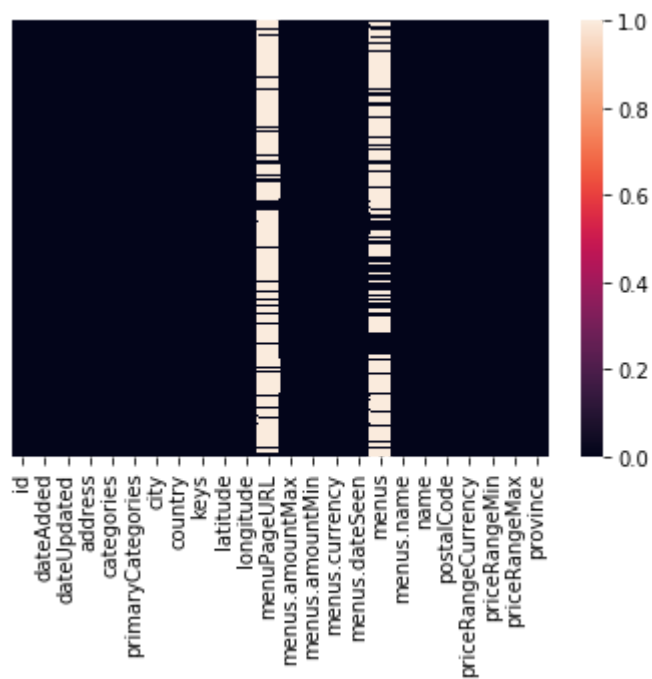
```
In [11]: df_num.head()
```

Out[11]:

	latitude	longitude	menus.amountMax	menus.amountMin	priceRangeMin	priceRangeMax
0	34.832300	-92.183800	7.98	7.98	0	25
1	33.509266	-112.073044	6.00	6.00	0	25
2	39.144883	-84.432685	6.49	6.49	0	25
3	42.516669	-83.106630	5.99	5.99	25	40
4	39.286630	-76.566984	5.49	5.49	0	25

```
In [12]: sns.heatmap(df.isnull(),yticklabels=False)
```

```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x24ab4118320>
```



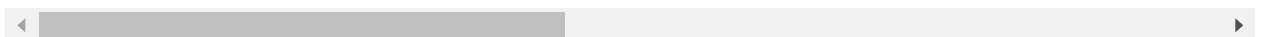
```
In [13]: df_cat.isnull()
```

```
Out[13]:
```

	id	dateAdded	dateUpdated	address	categories	primaryCategories	city	country	key
0	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	Fals
5	False	False	False	False	False	False	False	False	Fals
6	False	False	False	False	False	False	False	False	Fals
7	False	False	False	False	False	False	False	False	Fals
8	False	False	False	False	False	False	False	False	Fals
9	False	False	False	False	False	False	False	False	Fals
10	False	False	False	False	False	False	False	False	Fals
11	False	False	False	False	False	False	False	False	Fals
12	False	False	False	False	False	False	False	False	Fals
13	False	False	False	False	False	False	False	False	Fals
14	False	False	False	False	False	False	False	False	Fals
15	False	False	False	False	False	False	False	False	Fals
16	False	False	False	False	False	False	False	False	Fals
17	False	False	False	False	False	False	False	False	Fals
18	False	False	False	False	False	False	False	False	Fals
19	False	False	False	False	False	False	False	False	Fals
20	False	False	False	False	False	False	False	False	Fals
21	False	False	False	False	False	False	False	False	Fals
22	False	False	False	False	False	False	False	False	Fals
23	False	False	False	False	False	False	False	False	Fals
24	False	False	False	False	False	False	False	False	Fals
25	False	False	False	False	False	False	False	False	Fals
26	False	False	False	False	False	False	False	False	Fals
27	False	False	False	False	False	False	False	False	Fals
28	False	False	False	False	False	False	False	False	Fals
29	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	.
9970	False	False	False	False	False	False	False	False	Fals
9971	False	False	False	False	False	False	False	False	Fals
9972	False	False	False	False	False	False	False	False	Fals

	id	dateAdded	dateUpdated	address	categories	primaryCategories	city	country	key
9973	False	False	False	False	False	False	False	False	Fals
9974	False	False	False	False	False	False	False	False	Fals
9975	False	False	False	False	False	False	False	False	Fals
9976	False	False	False	False	False	False	False	False	Fals
9977	False	False	False	False	False	False	False	False	Fals
9978	False	False	False	False	False	False	False	False	Fals
9979	False	False	False	False	False	False	False	False	Fals
9980	False	False	False	False	False	False	False	False	Fals
9981	False	False	False	False	False	False	False	False	Fals
9982	False	False	False	False	False	False	False	False	Fals
9983	False	False	False	False	False	False	False	False	Fals
9984	False	False	False	False	False	False	False	False	Fals
9985	False	False	False	False	False	False	False	False	Fals
9986	False	False	False	False	False	False	False	False	Fals
9987	False	False	False	False	False	False	False	False	Fals
9988	False	False	False	False	False	False	False	False	Fals
9989	False	False	False	False	False	False	False	False	Fals
9990	False	False	False	False	False	False	False	False	Fals
9991	False	False	False	False	False	False	False	False	Fals
9992	False	False	False	False	False	False	False	False	Fals
9993	False	False	False	False	False	False	False	False	Fals
9994	False	False	False	False	False	False	False	False	Fals
9995	False	False	False	False	False	False	False	False	Fals
9996	False	False	False	False	False	False	False	False	Fals
9997	False	False	False	False	False	False	False	False	Fals
9998	False	False	False	False	False	False	False	False	Fals
9999	False	False	False	False	False	False	False	False	Fals

10000 rows × 18 columns



```
In [14]: df_cat.isnull().sum()
```

```
Out[14]: id                0
dateAdded                0
dateUpdated              0
address                  0
categories                0
primaryCategories        0
city                     0
country                  0
keys                     0
menuPageURL              8321
menus.currency            0
menus.dateSeen            0
menus                    6282
menus.name                0
name                      0
postalCode                4
priceRangeCurrency        0
province                  0
dtype: int64
```

```
In [29]: df_cat.menuPageURL.fillna(df_cat.menuPageURL.value_counts().idxmax(),inplace=True)
df_cat.menus.fillna(df_cat.menus.value_counts().idxmax(),inplace=True)
df_cat.postalCode.fillna(df_cat.postalCode.value_counts().idxmax(),inplace=True)
```

C:\ProgramData\Anaconda3\lib\site-packages\pandas\core\generic.py:6130: Setting WithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy> (<http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>)

```
self._update_inplace(new_data)
```

In [30]: `df_cat.head()`

Out[30]:

	id	dateAdded	dateUpdated	address	categories	primary
0	AVz3Y-7h3D1zeR_xDAqm	2017-06-30T05:05:40Z	2019-05-01T15:43:09Z	4203 E Kiehl Ave	Pizza,Restaurant,American restaurants,Pizza Pl...	Acco F
1	AVweGPFF_7pvs4fzAAzQ	2016-04-02T04:02:49Z	2019-05-01T15:27:50Z	25 E Camelback Rd	Pizza,Pizza Place,Restaurants	Acco F
2	AVwdRGa9_7pvs4fz4E3K	2016-03-03T18:39:49Z	2019-05-01T12:52:25Z	3703 Paxton Ave	Restaurant,Pizza Place,Restaurants	Acco F
3	AVwdX4psIN2L1WUfvJB1	2016-03-29T05:08:59Z	2019-05-01T12:52:20Z	30495 John R Rd	Pizza,Carry-out food,Pizza Place,Restaurants	Acco F
4	AVwdaeTtkufWRAb55pSH	2016-03-31T02:34:04Z	2019-05-01T12:50:45Z	3600 Eastern Ave	Pizza,American restaurants,Pizza Place,Pizza e...	Acco F

In [31]: `df_num.isnull().sum()`

Out[31]:

latitude	0
longitude	0
menus.amountMax	0
menus.amountMin	0
priceRangeMin	0
priceRangeMax	0
dtype:	int64

In [32]: `df_num.head()`

Out[32]:

	latitude	longitude	menus.amountMax	menus.amountMin	priceRangeMin	priceRangeMax
0	34.832300	-92.183800	7.98	7.98	0	25
1	33.509266	-112.073044	6.00	6.00	0	25
2	39.144883	-84.432685	6.49	6.49	0	25
3	42.516669	-83.106630	5.99	5.99	25	40
4	39.286630	-76.566984	5.49	5.49	0	25

In [33]: `from sklearn.preprocessing import LabelEncoder`

In [34]: `le=LabelEncoder()`



```
In [35]: df_cat=df_cat.apply(le.fit_transform)

In [37]: df=pd.concat([df_cat,df_num],axis=1)

In [38]: df.head()
```

Out[38]:

menuPageURL	...	name	postalCode	priceRangeCurrency	province	latitude	longitude	menus.an
14	...	1540	1222	0	2	34.832300	-92.183800	
10	...	1519	1409	0	3	33.509266	-112.073044	
119	...	1011	873	0	31	39.144883	-84.432685	
10	...	684	949	0	19	42.516669	-83.106630	
10	...	1581	418	0	18	39.286630	-76.566984	

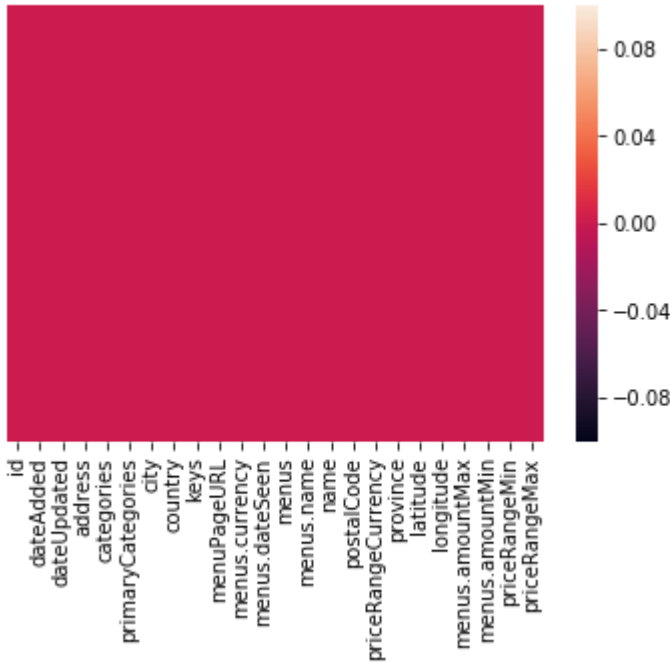


```
In [ ]:

In [ ]:

In [39]: sns.heatmap(df.isnull(),yticklabels=False)
```

Out[39]: <matplotlib.axes.\_subplots.AxesSubplot at 0x24ab8cf5710>



In [79]: `df.head()`

Out[79]:

menus.dateSeen	...	menus.name	name	postalCode	priceRangeCurrency	province	latitude	long
5492	...	1106	1540	1222	0	2	34.832300	-92.18
4602	...	3158	1519	1409	0	3	33.509266	-112.07
5173	...	3110	1011	873	0	31	39.144883	-84.43
634	...	3919	684	949	0	19	42.516669	-83.10
278	...	3382	1581	418	0	18	39.286630	-76.56

In [81]: `df.drop(["priceRangeCurrency"],axis=1,inplace=True)`

In [82]: `df.head()`

Out[82]:

menuPageURL	menus.currency	menus.dateSeen	menus	menus.name	name	postalCode	province
14	0	5492	2707	1106	1540	1222	2
10	0	4602	2707	3158	1519	1409	3
119	0	5173	2792	3110	1011	873	31
10	0	634	2707	3919	684	949	19
10	0	278	2707	3382	1581	418	18

In [87]: `df.drop(["id", "dateAdded", "dateUpdated", "latitude"],axis=1,inplace=True)`

In [88]: `df.head()`

Out[88]:

geURL	menus.currency	menus.dateSeen	menus	menus.name	name	postalCode	province	long
14	0	5492	2707	1106	1540	1222	2	-92.18
10	0	4602	2707	3158	1519	1409	3	-112.07
119	0	5173	2792	3110	1011	873	31	-84.43
10	0	634	2707	3919	684	949	19	-83.10
10	0	278	2707	3382	1581	418	18	-76.56

In [89]: `df.shape`

Out[89]: (10000, 16)

```
In [84]: from sklearn.model_selection import KFold
```

```
In [104]: kf=KFold(n_splits=10)
```

```
In [123]: x=df.iloc[:,[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14]].values  
y=df.iloc[:,15].values
```

```
In [109]: for a,b in kf.split(x):  
    x_train,x_test=x[a],x[b]  
    y_train,y_test=y[a],y[b]  
    x1_train,x1_test=x[a],x[b]  
    y1_train,y1_test=y[a],y[b]  
    x2_train,x2_test=x[a],x[b]  
    y2_train,y2_test=y[a],y[b]  
    x3_train,x3_test=x[a],x[b]  
    y3_train,y3_test=y[a],y[b]  
    x4_train,x4_test=x[a],x[b]  
    y4_train,y4_test=y[a],y[b]  
    x5_train,x5_test=x[a],x[b]  
    y5_train,y5_test=y[a],y[b]
```

```
In [110]: from sklearn.tree import DecisionTreeClassifier  
from sklearn.neighbors import KNeighborsClassifier  
from sklearn.naive_bayes import GaussianNB  
from sklearn.svm import LinearSVC  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.linear_model import LogisticRegression
```

```
In [111]: dt=DecisionTreeClassifier()  
knn=KNeighborsClassifier()  
nb=GaussianNB()  
svm=LinearSVC()  
rf=RandomForestClassifier()  
lr=LogisticRegression()
```

```
In [112]: dt.fit(x_train,y_train)
          knn.fit(x1_train,y1_train)
          nb.fit(x2_train,y2_train)
          svm.fit(x3_train,y3_train)
          rf.fit(x4_train,y4_train)
          lr.fit(x5_train,y5_train)
```

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:931: Convergence Warning: Liblinear failed to converge, increase the number of iterations.

"the number of iterations.", ConvergenceWarning)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246: FutureWarning: The default value of n\_estimators will change from 10 in version 0.20 to 100 in 0.22.

"10 in version 0.20 to 100 in 0.22.", FutureWarning)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

FutureWarning)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:460: FutureWarning: Default multi\_class will be changed to 'auto' in 0.22. Specify the multi\_class option to silence this warning.

"this warning.", FutureWarning)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\svm\base.py:931: Convergence Warning: Liblinear failed to converge, increase the number of iterations.

"the number of iterations.", ConvergenceWarning)

```
Out[112]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                             intercept_scaling=1, max_iter=100, multi_class='warn',
                             n_jobs=None, penalty='l2', random_state=None, solver='warn',
                             tol=0.0001, verbose=0, warm_start=False)
```

```
In [113]: pred=dt.predict(x_test)
          pred1=knn.predict(x1_test)
          pred2=nb.predict(x2_test)
          pred3=svm.predict(x3_test)
          pred4=rf.predict(x4_test)
          pred5=lr.predict(x5_test)
```

```
In [117]: from sklearn.metrics import confusion_matrix
```

```
In [118]: cm=confusion_matrix(y_test,pred)
          cm1=confusion_matrix(y1_test,pred1)
          cm2=confusion_matrix(y2_test,pred2)
          cm3=confusion_matrix(y3_test,pred3)
          cm4=confusion_matrix(y4_test,pred4)
          cm5=confusion_matrix(y5_test,pred5)
```

```
In [119]: cm,cm1,cm2,cm3,cm4,cm5
```

```
Out[119]: (array([[684, 134,  0],
                  [122,  56,  0],
                  [  3,   1,  0]], dtype=int64), array([[765,  52,  1],
                  [158,  20,  0],
                  [  4,   0,  0]], dtype=int64), array([[660, 144,  14,  0],
                  [  0,   0,  0,  0],
                  [144,  31,  3,  0],
                  [  4,   0,  0,  0]], dtype=int64), array([[818,  0,  0],
                  [178,  0,  0],
                  [  4,   0,  0]], dtype=int64), array([[800,  18,  0],
                  [147,  31,  0],
                  [  2,   2,  0]], dtype=int64), array([[ 0,  0,  0,  0],
                  [  0, 818,  0,  0],
                  [  1, 177,  0,  0],
                  [  0,   4,  0,  0]], dtype=int64))
```

```
In [120]: from sklearn.metrics import accuracy_score
```

```
In [121]: a=accuracy_score(y_test,pred)
          b=accuracy_score(y1_test,pred1)
          c=accuracy_score(y2_test,pred2)
          d=accuracy_score(y3_test,pred3)
          e=accuracy_score(y4_test,pred4)
          f=accuracy_score(y5_test,pred5)
```

```
In [122]: a,b,c,d,e,f
```

```
Out[122]: (0.74, 0.785, 0.663, 0.818, 0.831, 0.818)
```

```
In [ ]:
```