da-fat

June 11, 2023

[15]:
```
import pandas as pd
import numpy as np
```

[32]:
```
df = pd.read_csv('mcd.csv')
```

1 Preview Analysis

[33]:
```
df.head(3)
```

[33]:
```
   id                    item servesize calories  protien  totalfat  satfat  \
0   0         McVeggie Burger       168      402    10.24     13.83    5.34
1   1     McAloo Tikki Burger       146      339     8.50     11.31    4.27
2   2  McSpicy  Paneer Burger       199      652    20.29     39.45   17.12

   transfat  cholestrol  carbs  sugar  addedsugar   sodium     menu
0      0.16        2.49  56.54   7.90        4.49   706.13  regular
1      0.20        1.47   5.27   7.05        4.07   545.34  regular
2      0.18       21.85  52.33   8.35        5.27  1074.58  regular
```

[34]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 141 entries, 0 to 140
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          141 non-null    int64
 1   item        141 non-null    object
 2   servesize   141 non-null    object
 3   calories    141 non-null    object
 4   protien     139 non-null    float64
 5   totalfat    141 non-null    float64
 6   satfat      138 non-null    float64
 7   transfat    140 non-null    float64
 8   cholestrol  140 non-null    float64
 9   carbs       141 non-null    float64
 10  sugar       140 non-null    float64
 11  addedsugar  138 non-null    float64
 12  sodium      141 non-null    float64
 13  menu        141 non-null    object
dtypes: float64(9), int64(1), object(4)
memory usage: 15.5+ KB
```

[35]:
```
df.describe()
```

[35]:
```
               id     protien    totalfat      satfat    transfat  cholestrol  \
count  141.000000  139.000000  141.000000  138.000000  140.000000  140.000000
mean    70.000000    7.601151   10.060355    5.108797    1.116786   26.509136
std     40.847277    8.347848   10.435455    4.894624    7.345491   50.479092
min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
25%     35.000000    0.720000    0.460000    0.342500    0.077500    1.500000
50%     70.000000    4.980000    7.770000    4.360000    0.150000    8.390000
75%    105.000000   10.945000   14.160000    7.287500    0.250000   31.125000
max    140.000000   39.470000   45.180000   20.460000   75.260000  302.610000

            carbs       sugar  addedsugar       sodium
count  141.000000  140.000000  138.000000   141.000000
mean    30.770851   15.519571   10.561667   362.918809
std     20.664969   15.675504   14.355904   477.792553
min      0.000000    0.000000    0.000000     0.000000
25%     15.630000    2.317500    0.000000    41.990000
50%     29.880000    9.225000    3.740000   150.900000
75%     45.390000   26.965000   19.267500   530.540000
max     93.840000   64.220000   64.220000  2399.490000
```

[36]:
```
df.values
```

[36]:
```
array([[0, 'McVeggie Burger', '168\xa0', …, 4.49, 706.13, 'regular'],
       [1, 'McAloo Tikki Burger', '146\xa0', …, 4.07, 545.34,
        'regular'],
       [2, 'McSpicy\x99 Paneer Burger', '199\xa0', …, 5.27, 1074.58,
        'regular'],
       …,
       [138, 'Cheese Slice', '14', …, 0.0, 15.0, 'condiments'],
       [139, 'Sweet Corn', '40', …, 0.0, 178.95, 'condiments'],
       [140, 'Mixed Fruit Beverage', '180', …, 0.0, 0.04, 'condiments']],
      dtype=object)
```

[37]:
```
df.shape
```

[37]:
```
(141, 14)
```

[38]:
```
df.dtypes
```

```
[38]: id                int64
      item             object
      servesize        object
      calories         object
      protien         float64
      totalfat        float64
      satfat          float64
      transfat        float64
      cholestrol      float64
      carbs           float64
      sugar           float64
      addedsugar      float64
      sodium          float64
      menu             object
      dtype: object
```

```
[39]: df.isnull().sum()
```

```
[39]: id            0
      item          0
      servesize     0
      calories      0
      protien       2
      totalfat      0
      satfat        3
      transfat      1
      cholestrol    1
      carbs         0
      sugar         1
      addedsugar    3
      sodium        0
      menu          0
      dtype: int64
```

```
[40]: df.count()
```

```
[40]: id            141
      item          141
      servesize     141
      calories      141
      protien       139
      totalfat      141
      satfat        138
      transfat      140
      cholestrol    140
      carbs         141
      sugar         140
```

```
addedsugar    138
sodium        141
menu          141
dtype: int64
```

[41]:

```
df1 = df.dropna()
```

[43]:

```
df1.shape
```

[43]:

```
(131, 14)
```

[82]:

```
from statistics import mean
df2 = df.fillna(df.mean())
df2.shape
```

```
<ipython-input-82-93ad42c7e881>:2: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  df2 = df.fillna(df.mean())
```

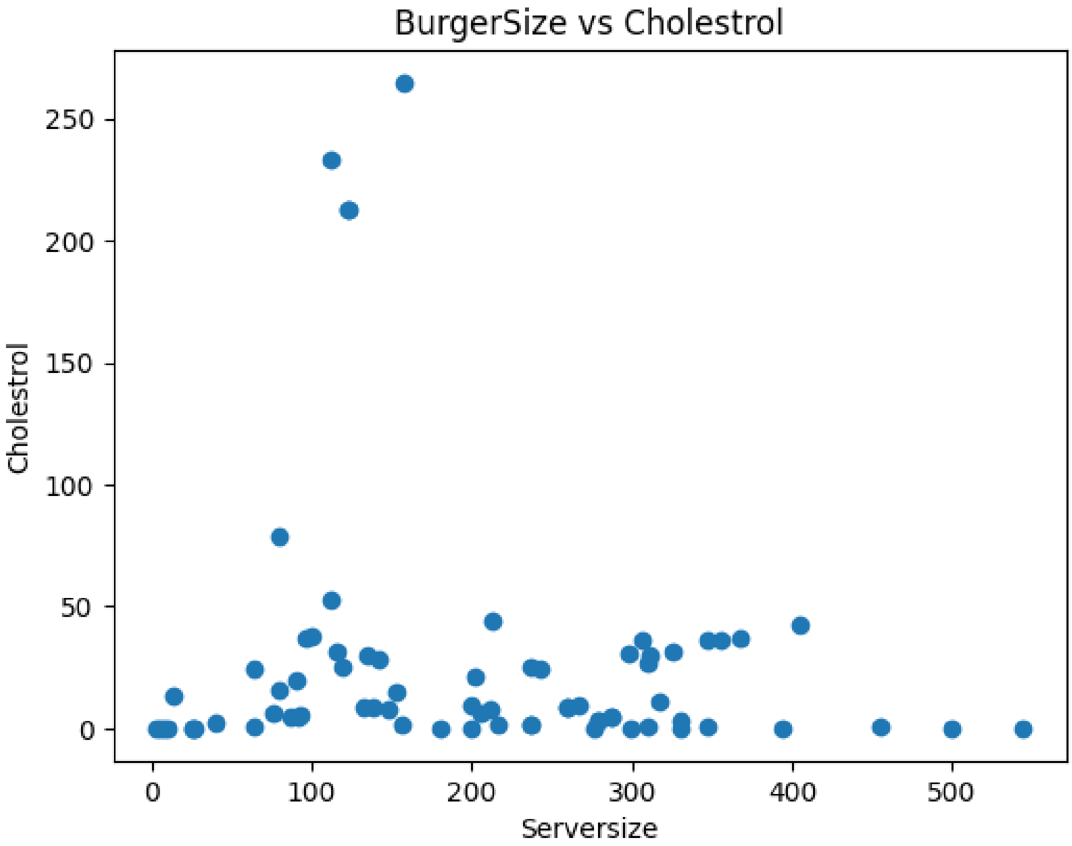
[82]:

```
(141, 14)
```

2 Basic Visualization

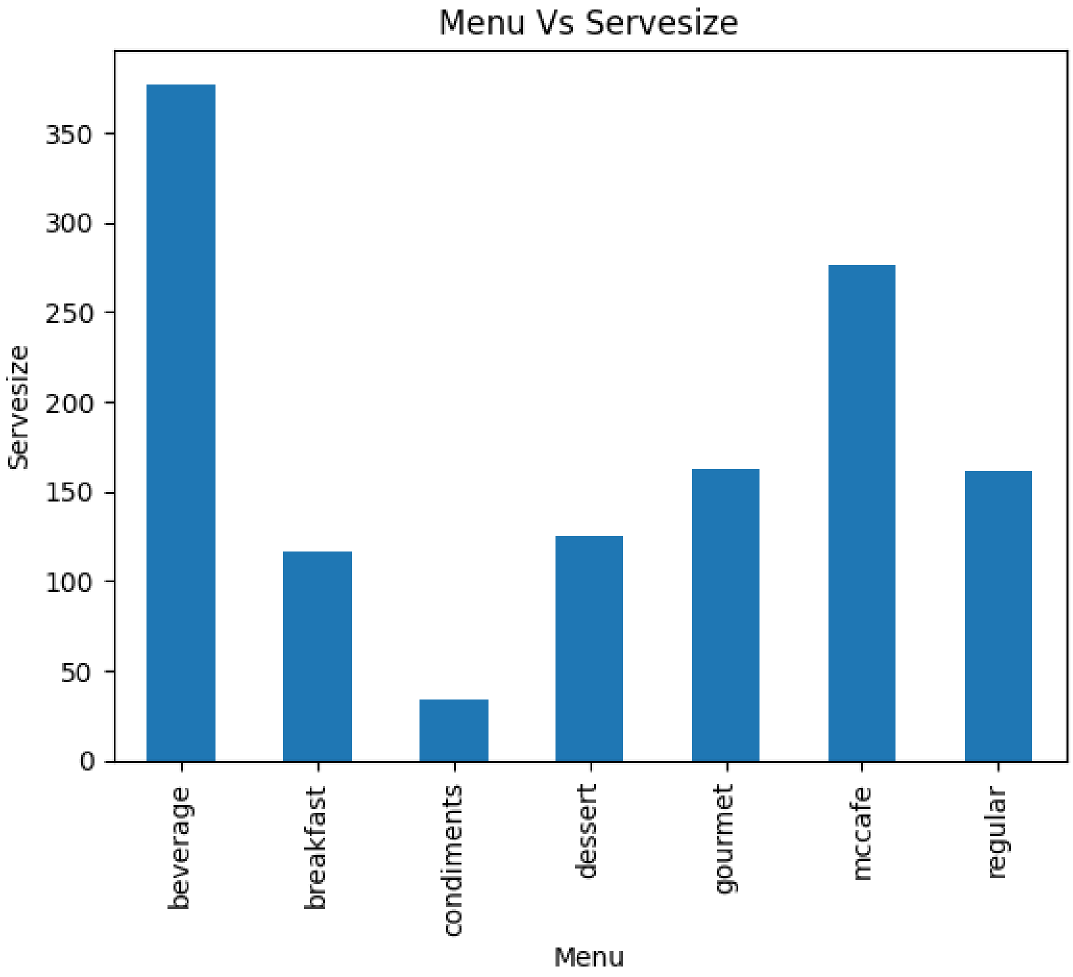
[71]:

```
from matplotlib import pyplot as plt
X = df2['servesize']
Y = df2['cholestrol']
plt.scatter(X,Y)
plt.xlabel('Serversize')
plt.ylabel('Cholestrol')
plt.title('BurgerSize vs Cholestrol')
plt.show()
```



[72]:

```
df2['servesize'] = pd.to_numeric(df2['servesize'])
gd = df2.groupby('menu')['servesize'].mean()
gd.plot(kind = 'bar')
plt.xlabel('Menu')
plt.ylabel('Servesize')
plt.title('Menu Vs Servesize')
plt.show()
```



```
[65]: cm = df2.corr()
      cm
```

```
<ipython-input-65-877562cab6a1>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  cm = df2.corr()
```

```
[65]:                    id  servesize   protien  totalfat    satfat  transfat  \
       id          1.000000  -0.035240 -0.393861 -0.376072 -0.321344 -0.094833
       servesize  -0.035240   1.000000  0.001379 -0.084444  0.029670  0.204322
       protien    -0.393861   0.001379  1.000000  0.864201  0.679893  0.077187
       totalfat   -0.376072  -0.084444  0.864201  1.000000  0.832129  0.061941
       satfat     -0.321344   0.029670  0.679893  0.832129  1.000000  0.211049
       transfat   -0.094833   0.204322  0.077187  0.061941  0.211049  1.000000
```

```
cholestrol -0.228872  -0.122945  0.584133  0.418155  0.355302  0.033002
carbs      -0.076254   0.196151  0.425970  0.535279  0.539640 -0.109547
sugar       0.427337   0.359170 -0.273269 -0.234302 -0.035166 -0.055213
addedsugar  0.462661   0.386028 -0.276281 -0.272879 -0.161092 -0.067510
sodium     -0.343108  -0.059661  0.906338  0.859849  0.608355  0.028684

            cholestrol     carbs     sugar  addedsugar    sodium
id           -0.228872 -0.076254  0.427337    0.462661 -0.343108
servesize    -0.122945  0.196151  0.359170    0.386028 -0.059661
protien       0.584133  0.425970 -0.273269   -0.276281  0.906338
totalfat      0.418155  0.535279 -0.234302   -0.272879  0.859849
satfat        0.355302  0.539640 -0.035166   -0.161092  0.608355
transfat      0.033002 -0.109547 -0.055213   -0.067510  0.028684
cholestrol    1.000000  0.153209 -0.208044   -0.226203  0.473496
carbs         0.153209  1.000000  0.498154    0.447597  0.493693
sugar        -0.208044  0.498154  1.000000    0.899101 -0.306171
addedsugar   -0.226203  0.447597  0.899101    1.000000 -0.251116
sodium        0.473496  0.493693 -0.306171   -0.251116  1.000000
```

3 Feature selection

[88]:

```
from sklearn.feature_selection import SelectKBest,chi2, mutual_info_classif
encoded_df = pd.get_dummies(df2, columns=['item'], drop_first=True)

X = encoded_df.drop('menu', axis=1)
y = encoded_df['menu']
```

[86]:

```
k = 5
selector = SelectKBest(score_func=chi2, k=k)
X_selected = selector.fit_transform(X, y)

feature_indices = selector.get_support(indices=True)
selected_features = X.columns[feature_indices]

print("Selected feature indices:", feature_indices)
print("Selected features:", selected_features)
```

```
Selected feature indices: [ 0  1  2  7 11]
Selected features: Index(['id', 'servesize', 'calories', 'cholestrol',
'sodium'], dtype='object')
```