تکلیف دوم درس مبانی داده کاوی

امید رئیسی (۹۶۲۱۱۶۰۰۱۵)

۱)

الف) متغییرهای **name, mfr, type** اسمی هستند و به ترتیب نام غلات، تولید کننده آن‌ها و نوع سرد و یا گرم بودن را بیان می‌کنند. متغییرهای **shelf, rating** ترتیبی هستند زیرا **shelf** ارتفاع قفسه غلات را بیان می‌کند که با کیفیت و رضایت کاربران از غلات رابطه مستقیم دارد و متغییر **rating** هم امتیاز کاربران به غلات را نشان می‌دهد. سایر متغییرها کمّی و عددی هستند.

ب)

In [10]:
```
import pandas as pd

cereals_df = pd.read_csv(r"./cereals.csv")

# removing non-numeric variables from the data frame
cereals_df.drop(cereals_df.columns[[0, 1, 2, 12, 15]], axis=1, inplace=True)

cereals_df.describe()
```

Out[10]:

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	weight	cups
count	77.000000	77.000000	77.000000	77.000000	77.000000	76.000000	76.000000	75.000000	77.000000	77.000000	77.000000
mean	106.883117	2.545455	1.012987	159.675325	2.151948	14.802632	7.026316	98.666667	28.246753	1.029610	0.821039
std	19.484119	1.094790	1.006473	83.832295	2.383364	3.907326	4.378656	70.410636	22.342523	0.150477	0.232716
min	50.000000	1.000000	0.000000	0.000000	0.000000	5.000000	0.000000	15.000000	0.000000	0.500000	0.250000
25%	100.000000	2.000000	0.000000	130.000000	1.000000	12.000000	3.000000	42.500000	25.000000	1.000000	0.670000
50%	110.000000	3.000000	1.000000	180.000000	2.000000	14.500000	7.000000	90.000000	25.000000	1.000000	0.750000
75%	110.000000	3.000000	2.000000	210.000000	3.000000	17.000000	11.000000	120.000000	25.000000	1.000000	1.000000
max	160.000000	6.000000	5.000000	320.000000	14.000000	23.000000	15.000000	330.000000	100.000000	1.500000	1.500000

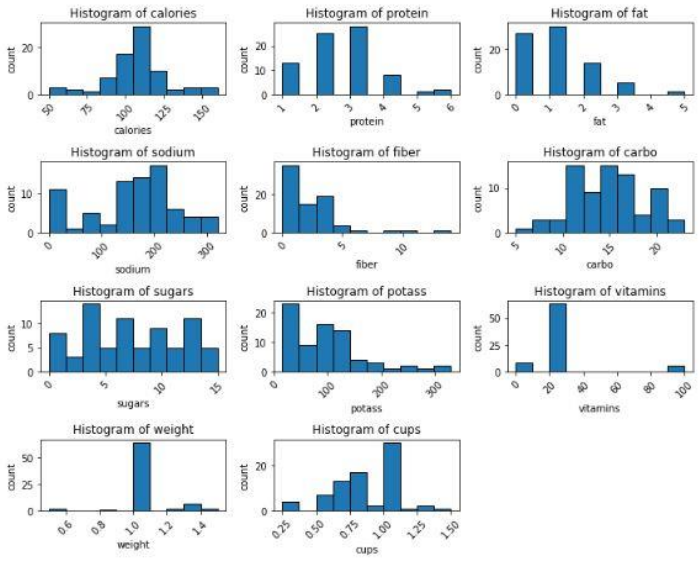
ج)

I. قند، پتاسیم و کربوهیدرات دارای بیشترین تغییر پذیری هستند.

II. متغییرهای پتاسیم، فیبر و چربی دارای چولگی می‌باشند. (در نمودار هیستوگرام این متغییرها نقطه اوج وتراکم مقادیر نمودار از میانگین متغییر فاصله دارد.)

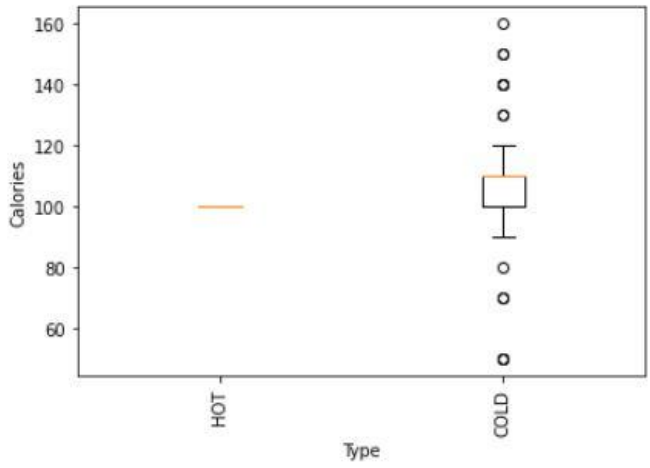
III. متغییر **max** در جدول قسمت **ب** کرانه متغییرها را نشان می‌دهد که برای مثال قند و پتاسیم به ترتیب ۱۵ و ۳۳۰ می‌باشد.

```
In [20]: import pandas as pd
         import matplotlib.pyplot as plt

         cereals_df = pd.read_csv(r"./cereals.csv")

         # removing non-numeric variables from the data frame
         cereals_df.drop(cereals_df.columns[[0, 1, 2, 12, 15]], axis=1, inplace=True)

         fig, axs = plt.subplots(nrows=4, ncols=3, figsize=(10, 8))

         axs = axs.ravel()

         for variable_pair in enumerate(cereals_df.columns):
             axs[variable_pair[0]].hist(cereals_df[variable_pair[1]], edgecolor="black")
             axs[variable_pair[0]].set_xlabel(variable_pair[1])
             axs[variable_pair[0]].set_ylabel("count")
             axs[variable_pair[0]].set_title(f"Histogram of {variable_pair[1]}")
             axs[variable_pair[0]].tick_params(axis="x", rotation=45)

         fig.delaxes(axs[11])

         fig.tight_layout()
         plt.show()
```



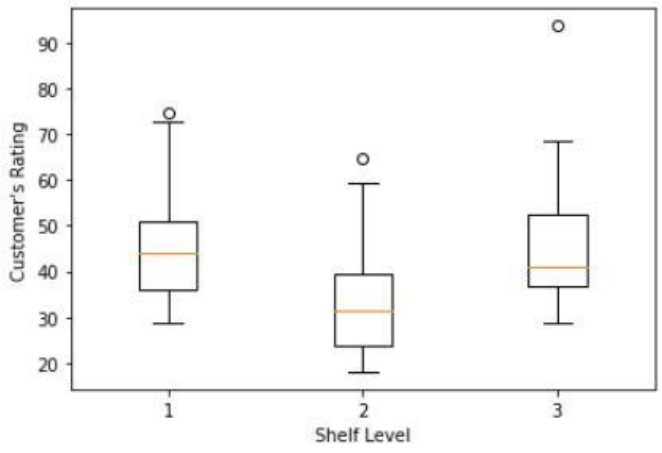
د) نمودار زیر نشان می‌دهد که تعداد غلات از نوع گرم خیلی کم است و با این تعداد کم نمی‌توان مقایسه‌ای بین دو نوع سرد و گرم انجام داد.

In [22]:
```
import pandas as pd
import matplotlib.pyplot as plt


cereals_df = pd.read_csv(r"./cereals.csv")

data_hot = [d.calories for d in cereals_df.iloc if d.type == "H"]
data_cold = [d.calories for d in cereals_df.iloc if d.type == "C"]

data_for_plot = [data_hot, data_cold]

plt.boxplot(data_for_plot, labels=["HOT", "COLD"])
plt.xlabel("Type")
plt.ylabel("Calories")
plt.xticks(rotation=90)
plt.show()
```



ه) با توجه به نمودار زیر سطح ارتفاع قفسه از زمین در امتیاز دهی کاربران مؤثر است و بهتر است برای بررسی بیشتر داده‌های قفسه شماره ۲ صرف نظر شوند.

```
In [24]: import pandas as pd
         import matplotlib.pyplot as plt


         cereals_df = pd.read_csv(r"./cereals.csv")

         data_1 = [d.rating for d in cereals_df.iloc if d.shelf == 1]
         data_2 = [d.rating for d in cereals_df.iloc if d.shelf == 2]
         data_3 = [d.rating for d in cereals_df.iloc if d.shelf == 3]

         data_for_plot = [data_1, data_2, data_3]

         plt.boxplot(data_for_plot, labels=["1", "2", "3"])
         plt.xlabel("Shelf Level")
         plt.ylabel("Customer's Rating")
         plt.show()
```



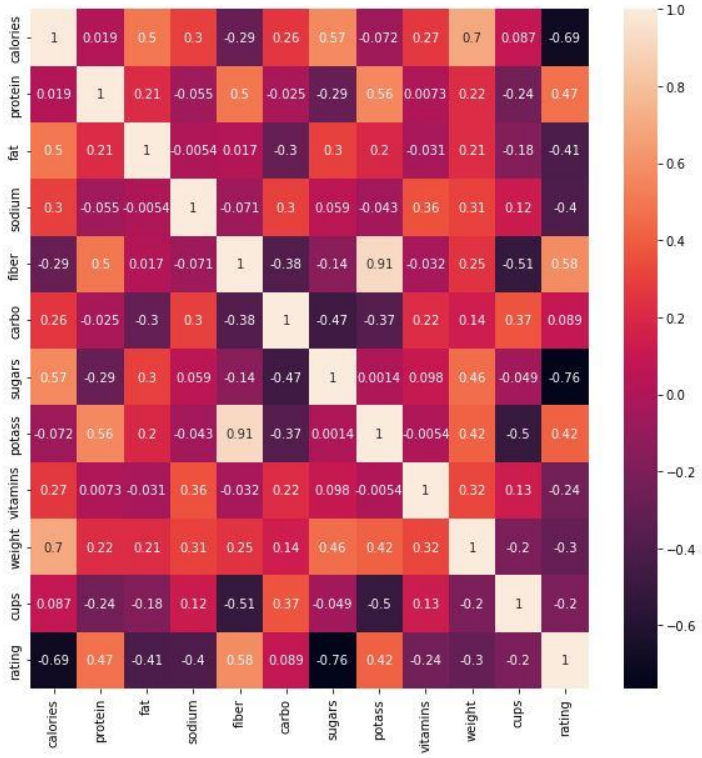
و)

I. زوج متغییر های **[(potas, fiber), (suger, rating), (calories, rating)]** دارای بیشترین همبستگی هستند.

II. متغییرهای که دارای همبستگی بالا می‌باشند باید یکی از آن‌ها حذف شود این کار اصولا با روش‌های مختلف انجام می‌شود که اساس کار آن‌ها سعی و خطا و تاثیر متغییرهای جدید بر مدل می‌باشد. در این حالت که با متغییرهای عددی سروکار داریم روش **PCA** روش مناسبی برای این کار می‌باشد.

III. در صورت نرمال‌سازی داده‌ها متغییرها دارای میانگین، واریانس و در نتیجه انحراف معیار جدید می‌شوند و تغییر این شاخص ها در مقادیر ماتریس همبستگی تأثیر زیادی می‌گذارد و ممکن است همبستگی‌های جدید به وجود آمده و همبستگی‌های قبلی از بین بروند.

```
In [30]: import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

         cereals_df = pd.read_csv(r"./cereals.csv")
         cereals_df.drop(cereals_df.columns[[0, 1, 2, 12]], axis=1, inplace=True)
         fig, ax = plt.subplots(figsize=(10, 10))
         sns.heatmap(cereals_df.corr(), xticklabels=cereals_df.columns, yticklabels=cereals_df.columns, annot=True, ax=ax)
         plt.show()
```



ز) مؤلفه اصلی اول که در روش **PCA** کاربرد دارد در واقع خطی است که بیشترین تغییر پذیری را درنمودار پراکنشی دو متغییر با همبستگی بالا را شامل می‌شود.در این نمودار که به هدف بررسی امکان حذف متغییر با همبستگی بالا رسم می‌شود این خط دارای بیشترین واریانس است و همچنین مجموع فواصل نقاط مختلف نمودار از این خط دارای کمترین مقدار ممکن است.

(2

		رده پیش‌بینی شده	
		کلاهبردار	غیرکلاهبردار
رده واقعی	کلاهبردار	30	32
	غیرکلاهبردار	58	920

نرخ خطای کل از رابطه زیر بدست می‌آید.

$$error = \frac{n_{1,2} + n_{2,1}}{n}$$

که $n_{1,2}$ و $n_{2,1}$ تعداد پیش‌بینی‌های اشتباه را نشان می‌دهند.

$$error = \frac{32 + 58}{1040} = 0.086$$

این رابطه نشان می‌دهد که این مدل در تقریبا ۹٪ داده‌ها مرتکب اشتباه می شود.

3)

الف)

```
In [10]: import pandas as pd
         from dmba import classificationSummary


         prediction_df = pd.read_csv(r"./prediction_table.csv")

         cutoffs = [0.25, 0.50, 0.75]

         for cutoff in cutoffs:
             predicted_values = [1 if p > cutoff else 0 for p in prediction_df.propensity_1]

             print(f"Classification Summary for cutoff = {cutoff}")
             classificationSummary(
                 prediction_df.actual_class, predicted_values, class_names=["class 0", "class 1"]
             )
             print()

Classification Summary for cutoff = 0.25
Confusion Matrix (Accuracy 0.6000)

        Prediction
 Actual class 0 class 1
class 0       9       8
class 1       0       3

Classification Summary for cutoff = 0.5
Confusion Matrix (Accuracy 0.9000)

        Prediction
 Actual class 0 class 1
class 0      15       2
class 1       0       3

Classification Summary for cutoff = 0.75
Confusion Matrix (Accuracy 0.9500)

        Prediction
 Actual class 0 class 1
class 0      17       0
class 1       1       2
```

نرخ خطا از رابطه $\boldsymbol{error = 1 - accuracy}$ بدست می‌آید و همچنین حساسیت (**sensitivity**) و وضوح (**specifity**) از روابط زیر بدست می‌آیند: (در اینجا پیشبینی کلاس **1** از پیشبینی کلاس **0** مهمتر است.)

$$specifity = \frac{n_{1,1}}{n_{1,1} + n_{1,2}}$$

$$sensitivity = \frac{n_{2,2}}{n_{2,1} + n_{2,2}}$$

Cutoff = 0.25	$error = 1 - 0.60 = 0.40$
	$sesitivity = \frac{3}{3+0} = 1.0$
	$specifity = \frac{9}{9+8} = 0.53$
Cutoff = 0.50	$error = 1 - 0.90 = 0.10$
	$sesitivity = \frac{3}{3+0} = 1.0$
	$specifity = \frac{15}{15+2} = 0.88$
Cutoff = 0.75	$error = 1 - 0.95 = 0.05$
	$sesitivity = \frac{2}{2+1} = 0.66$
	$specifity = \frac{17}{17+0} = 1.0$

ب)

In [11]:
```
import pandas as pd
from dmba import liftChart
import matplotlib.pyplot as plt


prediction_df = pd.read_csv(r"./prediction_table.csv")
prediction_df.sort_values(by=["propensity_1"], ascending=False, inplace=True)

fig, ax = plt.subplots()
liftChart(prediction_df.propensity_1, ax=ax)

ax.set_title("Lift Chart for Class Prediction")
ax.set_ylabel("Lift of Class 1 selection")
ax.set_xlabel("Percentage of Data")

plt.tight_layout()
plt.show()
```