

MACHINE LEARNING

SUBMITTED BY,

- Name:Jagannath v v
- Reg.No:-21122024
- class:1st MSc DataScience
- Time: Around 4.30 hr

LAB OVERVIEW

Objectives

Solve the below 10-step problem:

1. Download the 80-cereals dataset from <https://www.kaggle.com/crawford/80-cereals>
2. Load the dataset using pd.read_csv method
3. a. In the column "mfr", replace the column "K" as "Kellogg's", "G" as "Nestle" and all other values as "Other Brands" b. In the column "type", replace "C" with "Type 1" and "H" with "Type 2"
4. Visualise the count of above two features "mfr" and "type" with a bar-plot
5. Describe the five-number summary and boxplots of the features - protien, sugars, fat, carbo
6. Plot Histograms for the features - fat, carbo, sodium, fiber
7. Split the datasets into following ratios: 60:40, 70:30, 80:20. Write down what happens when you give "random_state" parameter with a constant value and what happens if you do not mention the parameter at all.
8. Apply MinMaxScaler() and StandardScaler() to the following features: calories, protien, fat, sodium, fiber, carbo, sugars.
9. Does the standard or min-max scaling make a difference in value distribution? Support your answers with some visualisations on the above dataset.
10. As an extension of 7th step, Generate a new Pandas DataFrame with the following columns based on the Training Dataset: Split Ratio | Random State | Total Number of Entries | Count of Kellogg's | Count of Nestle | Count of Other Brands

PROBLEM_DEFINITONS

Here we try to read new dataset called cereal. And try to explore more in this datasets like visualising through various visualising libraries. And we try to create new datasets with existing details of the current dataset

Approach

Here i refered internet to explore about the datasets and visualise more in it.

Load the dataset using pd.read_csv method

```
In [6]: import pandas as pd #Importing panda
```

```
In [7]: cer=pd.read_csv(r"C:\Users\jagan\Desktop\MSC DATASCIENCE\2nd sem\machine learning\ce
cer
```

Out[7]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	sh
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	
3	All-Bran with Extra Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	
...	
72	Triples	G	C	110	2	1	250	0.0	21.0	3	60	25	
73	Trix	G	C	110	1	1	140	0.0	13.0	12	25	25	
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	25	
75	Wheaties	G	C	100	3	1	200	3.0	17.0	3	110	25	
76	Wheaties Honey Gold	G	C	110	2	1	200	1.0	16.0	8	60	25	

77 rows × 16 columns

3. a. In the column "mfr", replace the column "K" as "Kellogg's", "G" as "Nestle" and all other values as "Other Brands"

b. In the column "type", replace "C" with "Type 1" and "H" with "Type 2"

```
In [8]: cer['mfr'].replace(['K'],['KELLOGGS'],inplace=True) #replacing the value k as kellog
cer
```

Out[8]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	2
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135	

		name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir
2	All-Bran	KELLOGGS		C	70	4	1	260	9.0	7.0	5	320	2
3	All-Bran with Extra Fiber	KELLOGGS		C	50	4	0	140	14.0	8.0	0	330	2
4	Almond Delight		R	C	110	2	2	200	1.0	14.0	8	-1	2
...
72	Triples		G	C	110	2	1	250	0.0	21.0	3	60	2
73	Trix		G	C	110	1	1	140	0.0	13.0	12	25	2
74	Wheat Chex		R	C	100	3	1	230	3.0	17.0	3	115	2
75	Wheaties		G	C	100	3	1	200	3.0	17.0	3	110	2
76	Wheaties Honey Gold		G	C	110	2	1	200	1.0	16.0	8	60	2

77 rows × 16 columns



In [9]:

```
cer['mfr'].replace(['G'], ['NESTLE'], inplace=True)
cer
```

Out[9]:

		name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir
0	100% Bran		N	C	70	4	1	130	10.0	5.0	6	280	2
1	100% Natural Bran		Q	C	120	3	5	15	2.0	8.0	8	135	
2	All-Bran	KELLOGGS		C	70	4	1	260	9.0	7.0	5	320	2
3	All-Bran with Extra Fiber	KELLOGGS		C	50	4	0	140	14.0	8.0	0	330	2
4	Almond Delight		R	C	110	2	2	200	1.0	14.0	8	-1	2
...
72	Triples	NESTLE	C	C	110	2	1	250	0.0	21.0	3	60	2
73	Trix	NESTLE	C	C	110	1	1	140	0.0	13.0	12	25	2
74	Wheat Chex		R	C	100	3	1	230	3.0	17.0	3	115	2
75	Wheaties	NESTLE	C	C	100	3	1	200	3.0	17.0	3	110	2
76	Wheaties Honey Gold	NESTLE	C	C	110	2	1	200	1.0	16.0	8	60	2

77 rows × 16 columns

|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

In [10]:

```
cer['mfr'].replace(['K', 'G'], ['KELLOGS', 'NESTLE'], inplace=True)
cer
```

Out[10]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir			
0	100% Bran	N	C	70	4	1	130	10.0	5.0	6	280	2			
1	100% Natural Bran	Q	C	120	3	5	15	2.0	8.0	8	135				
2	All-Bran	KELLOGGS	C	70	4	1	260	9.0	7.0	5	320	2			
3	All-Bran with Extra Fiber	KELLOGGS	C	50	4	0	140	14.0	8.0	0	330	2			
4	Almond Delight	R	C	110	2	2	200	1.0	14.0	8	-1	2			
...			
72	Triples	NESTLE	C	110	2	1	250	0.0	21.0	3	60	2			
73	Trix	NESTLE	C	110	1	1	140	0.0	13.0	12	25	2			
74	Wheat Chex	R	C	100	3	1	230	3.0	17.0	3	115	2			
75	Wheaties	NESTLE	C	100	3	1	200	3.0	17.0	3	110	2			
76	Wheaties Honey Gold	NESTLE	C	110	2	1	200	1.0	16.0	8	60	2			

77 rows × 16 columns

|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|

In [11]:

```
cer['mfr'].replace(['N', 'Q', 'R', 'P', 'A'], 'OTHERS', inplace=True)
cer
```

Out[11]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir			
0	100% Bran	OTHERS	C	70	4	1	130	10.0	5.0	6	280	2			
1	100% Natural Bran	OTHERS	C	120	3	5	15	2.0	8.0	8	135				
2	All-Bran	KELLOGGS	C	70	4	1	260	9.0	7.0	5	320	2			
3	All-Bran with Extra Fiber	KELLOGGS	C	50	4	0	140	14.0	8.0	0	330	2			

		name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamir
4	Almond Delight		OTHERS	C	110	2	2	200	1.0	14.0	8	-1	2
...
72	Triples		NESTLE	C	110	2	1	250	0.0	21.0	3	60	2
73	Trix		NESTLE	C	110	1	1	140	0.0	13.0	12	25	2
74	Wheat Chex		OTHERS	C	100	3	1	230	3.0	17.0	3	115	2
75	Wheaties		NESTLE	C	100	3	1	200	3.0	17.0	3	110	2
76	Wheaties Honey Gold		NESTLE	C	110	2	1	200	1.0	16.0	8	60	2

77 rows × 16 columns



4. Visualise the count of above two features "mfr" and "type" with a bar-plot

In [123...]

```
import matplotlib as mlp
import seaborn as sns
import matplotlib.pyplot as plt #importing some visualising Libraries
```

In [77]:

```
#show first five rows of record
cer.head(5)
```

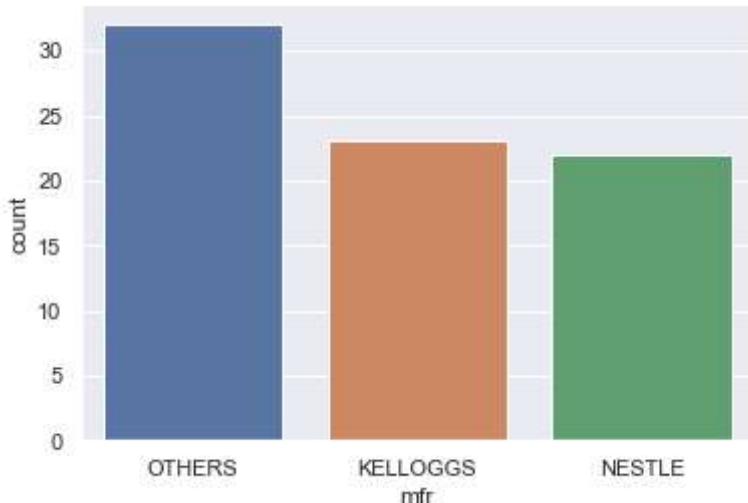
Out[77]:

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins
0	100% Bran	OTHERS	C	70	4	1	130	10.0	5.0	6	280	25
1	100% Natural Bran	OTHERS	C	120	3	5	15	2.0	8.0	8	135	0
2	All-Bran	KELLOGGS	C	70	4	1	260	9.0	7.0	5	320	25
3	All-Bran with Extra Fiber	KELLOGGS	C	50	4	0	140	14.0	8.0	0	330	25
4	Almond Delight	OTHERS	C	110	2	2	200	1.0	14.0	8	-1	25



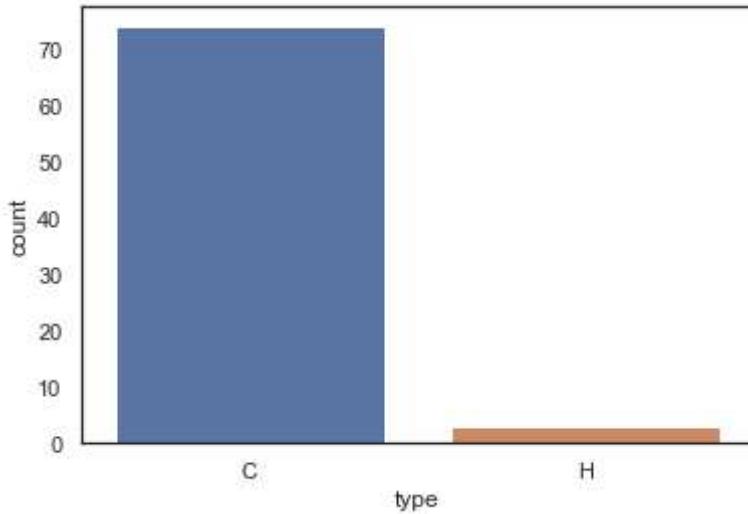
In [78]:

```
#create count plot
sns.set(style="darkgrid")
count = sns.countplot(x='mfr', data=cer)
plt.show()
```



In [80]:

```
sns.set(style="white")
count = sns.countplot(x='type', data=cer)
plt.show()
```



5. Describe the five-number summary and boxplots of the features - protein, sugars, fat, carbo

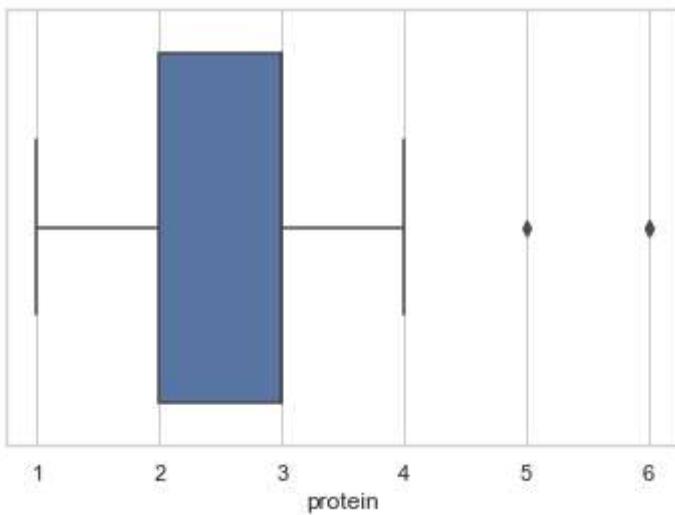
In [118...]

```
cer[['protein', 'fat', 'sugars', 'carbo']].describe()
```

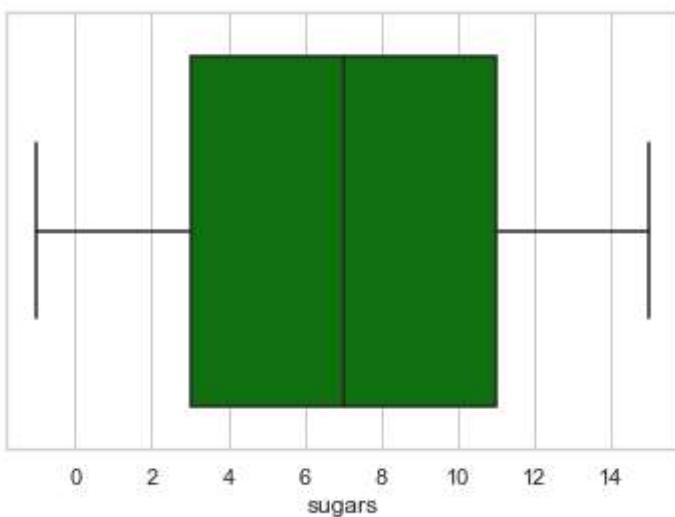
Out[118...]

	protein	fat	sugars	carbo
count	77.000000	77.000000	77.000000	77.000000
mean	2.545455	1.012987	6.922078	14.597403
std	1.094790	1.006473	4.444885	4.278956
min	1.000000	0.000000	-1.000000	-1.000000
25%	2.000000	0.000000	3.000000	12.000000
50%	3.000000	1.000000	7.000000	14.000000
75%	3.000000	2.000000	11.000000	17.000000
max	6.000000	5.000000	15.000000	23.000000

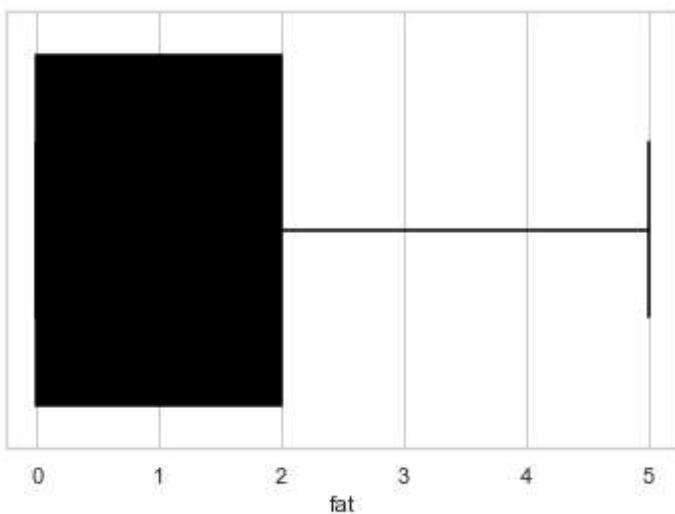
```
In [109... box1 = sns.boxplot(x="protein",data=cer)
```



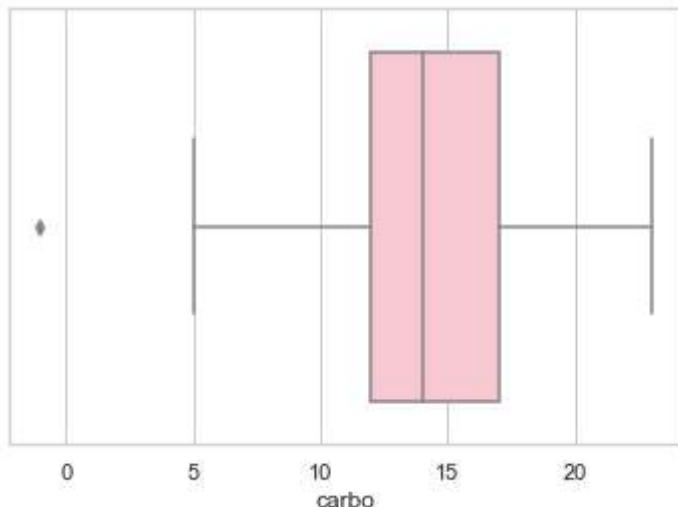
```
In [113... box2=sns.boxplot(x="sugars",data=cer,color="green")
```



```
In [112... box3=sns.boxplot(x="fat",data=cer,color="black")
```



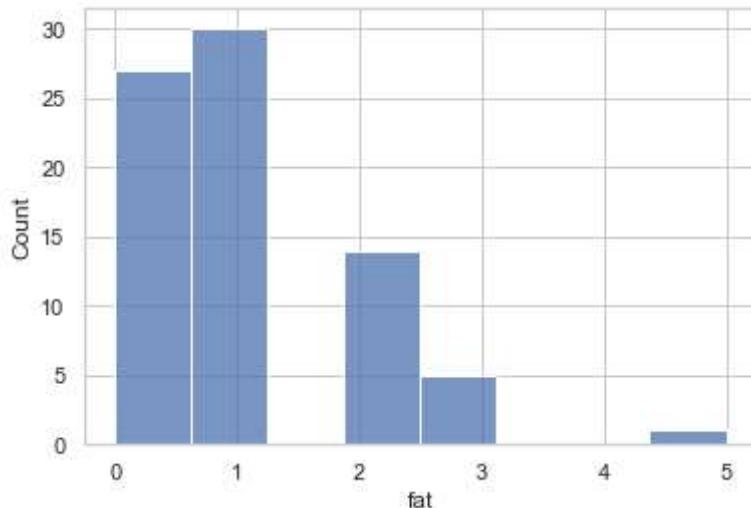
```
In [115... box4=sns.boxplot(x="carbo",data=cer,color="pink")
```



6. Plot Histograms for the features - fat, carbo, sodium, fiber

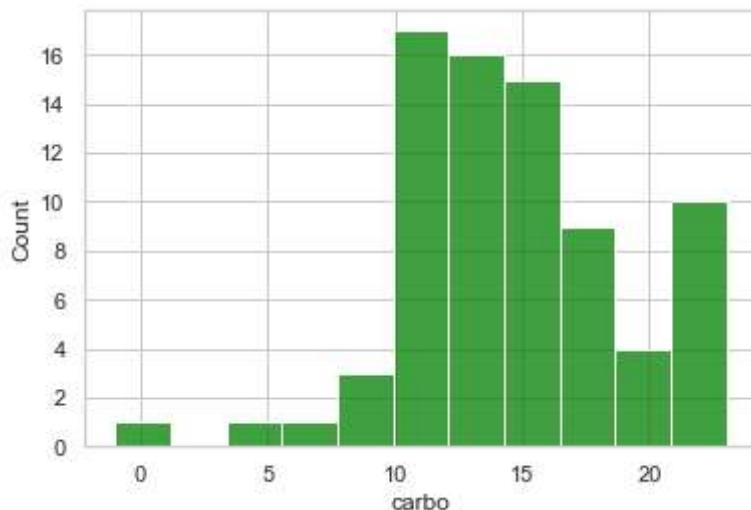
```
In [126... sns.histplot(data=cer, x="fat")]
```

```
Out[126... <AxesSubplot:xlabel='fat', ylabel='Count'>
```



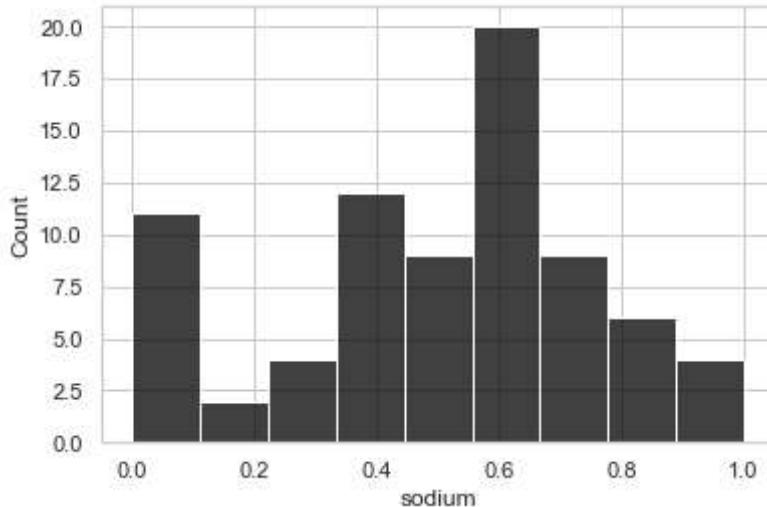
```
In [127... sns.histplot(data=cer, x="carbo", color="green")]
```

```
Out[127... <AxesSubplot:xlabel='carbo', ylabel='Count'>
```



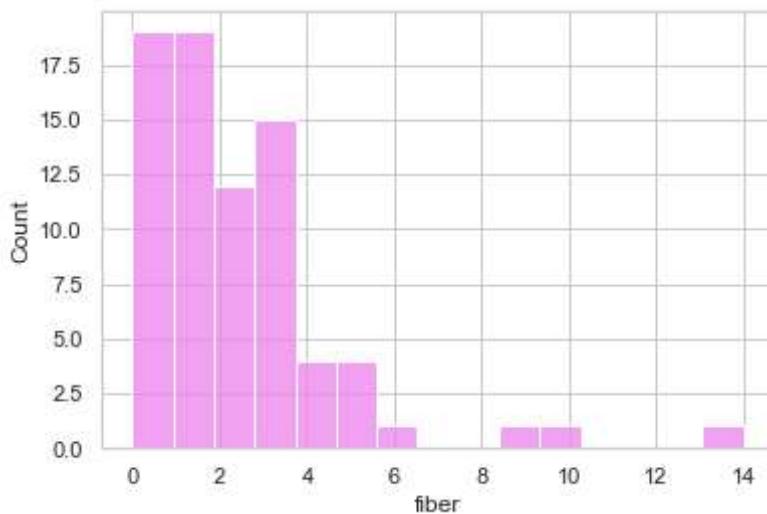
In [239]: `sns.histplot(data=cer,x="sodium",color="black")`

Out[239]: <AxesSubplot:xlabel='sodium', ylabel='Count'>



In [129]: `sns.histplot(data=cer,x="fiber",color="violet")`

Out[129]: <AxesSubplot:xlabel='fiber', ylabel='Count'>



7. Split the datasets into following ratios: 60:40, 70:30, 80:20. Write down what happens when you give "random_state" parameter with a constant value and what happens if you do not mention the parameter at all.

In [12]: `from sklearn.model_selection import train_test_split`

60:40

In [13]: `train_data,test_data=train_test_split(cer,test_size=0.4,random_state=20)
print(train_data)`

			name	mfr	type	calories	protein	fat	\
24			Froot Loops	KELLOGGS	C	110	2	1	
14			Cocoa Puffs	NESTLE	C	110	1	1	
50			Nutri-grain	Wheat	KELLOGGS	90	3	0	
29			Fruity Pebbles	OTHERS	C	110	1	1	

45	Muesli Raisins; Peaches; & Pecans	OTHERS	C	150	4	3
64	Shredded Wheat 'n'Bran	OTHERS	C	90	3	0
60	Raisin Squares	KELLOGGS	C	90	2	0
76	Wheaties Honey Gold	NESTLE	C	110	2	1
44	Muesli Raisins; Dates; & Almonds	OTHERS	C	150	4	3
65	Shredded Wheat spoon size	OTHERS	C	90	3	0
17	Corn Pops	KELLOGGS	C	110	1	0
53	Product 19	KELLOGGS	C	100	3	0
56	Quaker Oat Squares	OTHERS	C	100	4	1
18	Count Chocula	NESTLE	C	110	1	1
54	Puffed Rice	OTHERS	C	50	1	0
61	Rice Chex	OTHERS	C	110	1	0
43	Maypo	OTHERS	H	100	4	1
55	Puffed Wheat	OTHERS	C	50	2	0
32	Grape Nuts Flakes	OTHERS	C	100	3	1
46	Mueslix Crispy Blend	KELLOGGS	C	160	3	2
30	Golden Crisp	OTHERS	C	100	2	0
19	Cracklin' Oat Bran	KELLOGGS	C	110	3	3
59	Raisin Nut Bran	NESTLE	C	100	3	2
58	Raisin Bran	KELLOGGS	C	120	3	1
10	Cap'n'Crunch	OTHERS	C	120	1	2
3	All-Bran with Extra Fiber	KELLOGGS	C	50	4	0
25	Frosted Flakes	KELLOGGS	C	110	1	0
11	Cheerios	NESTLE	C	110	6	2
13	Clusters	NESTLE	C	110	3	2
68	Strawberry Fruit Wheats	OTHERS	C	90	2	0
6	Apple Jacks	KELLOGGS	C	110	2	0
70	Total Raisin Bran	NESTLE	C	140	3	1
37	Honey-comb	OTHERS	C	110	1	0
7	Basic 4	NESTLE	C	130	3	2
67	Special K	KELLOGGS	C	110	6	0
62	Rice Krispies	KELLOGGS	C	110	2	0
16	Corn Flakes	KELLOGGS	C	100	2	0
26	Frosted Mini-Wheats	KELLOGGS	C	100	3	0
40	Kix	NESTLE	C	110	2	1
34	Great Grains Pecan	OTHERS	C	120	3	3
71	Total Whole Grain	NESTLE	C	100	3	1
22	Crispy Wheat & Raisins	NESTLE	C	100	2	1
20	Cream of Wheat (Quick)	OTHERS	H	100	3	0
9	Bran Flakes	OTHERS	C	90	3	0
28	Fruitful Bran	KELLOGGS	C	120	3	0
15	Corn Chex	OTHERS	C	110	2	0

	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
24	125	1.0	11.0	13	30	25	2	1.00	1.00	
14	180	0.0	12.0	13	55	25	2	1.00	1.00	
50	170	3.0	18.0	2	90	25	3	1.00	1.00	
29	135	0.0	13.0	12	25	25	2	1.00	0.75	
45	150	3.0	16.0	11	170	25	3	1.00	1.00	
64	0	4.0	19.0	0	140	0	1	1.00	0.67	
60	0	2.0	15.0	6	110	25	3	1.00	0.50	
76	200	1.0	16.0	8	60	25	1	1.00	0.75	
44	95	3.0	16.0	11	170	25	3	1.00	1.00	
65	0	3.0	20.0	0	120	0	1	1.00	0.67	
17	90	1.0	13.0	12	20	25	2	1.00	1.00	
53	320	1.0	20.0	3	45	100	3	1.00	1.00	
56	135	2.0	14.0	6	110	25	3	1.00	0.50	
18	180	0.0	12.0	13	65	25	2	1.00	1.00	
54	0	0.0	13.0	0	15	0	3	0.50	1.00	
61	240	0.0	23.0	2	30	25	1	1.00	1.13	
43	0	0.0	16.0	3	95	25	2	1.00	1.00	
55	0	1.0	10.0	0	50	0	3	0.50	1.00	
32	140	3.0	15.0	5	85	25	3	1.00	0.88	
46	150	3.0	17.0	13	160	25	3	1.50	0.67	
30	45	0.0	11.0	15	40	25	1	1.00	0.88	
19	140	4.0	10.0	7	160	25	3	1.00	0.50	
59	140	2.5	10.5	8	140	25	3	1.00	0.50	
58	210	5.0	14.0	12	240	25	2	1.33	0.75	
10	220	0.0	12.0	12	35	25	2	1.00	0.75	

3	140	14.0	8.0	0	330	25	3	1.00	0.50
25	200	1.0	14.0	11	25	25	1	1.00	0.75
11	290	2.0	17.0	1	105	25	1	1.00	1.25
13	140	2.0	13.0	7	105	25	3	1.00	0.50
68	15	3.0	15.0	5	90	25	2	1.00	1.00
6	125	1.0	11.0	14	30	25	2	1.00	1.00
70	190	4.0	15.0	14	230	100	3	1.50	1.00
37	180	0.0	14.0	11	35	25	1	1.00	1.33
7	210	2.0	18.0	8	100	25	3	1.33	0.75
67	230	1.0	16.0	3	55	25	1	1.00	1.00
62	290	0.0	22.0	3	35	25	1	1.00	1.00
16	290	1.0	21.0	2	35	25	1	1.00	1.00
26	0	3.0	14.0	7	100	25	2	1.00	0.80
40	260	0.0	21.0	3	40	25	2	1.00	1.50
34	75	3.0	13.0	4	100	25	3	1.00	0.33
71	200	3.0	16.0	3	110	100	3	1.00	1.00
22	140	2.0	11.0	10	120	25	3	1.00	0.75
20	80	1.0	21.0	0	-1	0	2	1.00	1.00
9	210	5.0	13.0	5	190	25	3	1.00	0.67
28	240	5.0	14.0	12	190	25	3	1.33	0.67
15	280	0.0	22.0	3	25	25	1	1.00	1.00

rating

24	32.207582
14	22.736446
50	59.642837
29	28.025765
45	34.139765
64	74.472949
60	55.333142
76	36.187559
44	37.136863
65	72.801787
17	35.782791
53	41.503540
56	49.511874
18	22.396513
54	60.756112
61	41.998933
43	54.850917
55	63.005645
32	52.076897
46	30.313351
30	35.252444
19	40.448772
59	39.703400
58	39.259197
10	18.042851
3	93.704912
25	31.435973
11	50.764999
13	40.400208
68	59.363993
6	33.174094
70	28.592785
37	28.742414
7	37.038562
67	53.131324
62	40.560159
16	45.863324
26	58.345141
40	39.241114
34	45.811716
71	46.658844
22	36.176196
20	64.533816
9	53.313813
28	41.015492
15	41.445019

In [134...]

print(test_data)

52		Post Nat.	Raisin Bran										
36		Honey Nut	Cheerios										
42		Lucky Charms											
69		Total Corn Flakes											
72		Triples											
41		Life											
0		100% Bran											
31		Golden Grahams											
21		Crispix											
5		Apple Cinnamon	Cheerios										
75		Wheaties											
12		Cinnamon Toast	Crunch										
38		Just Right	Crunchy	Nuggets									
57		Quaker	Oatmeal										
2		All-Bran											
63		Shredded Wheat											
23		Double Chex											
8		Bran Chex											
35		Honey Graham	Ohs										
4		Almond Delight											
27	Fruit & Fibre	Dates; Walnuts;	and Oats										
73		Trix											
74		Wheat Chex											
33		Grape-Nuts											
51		Oatmeal Raisin	Crisp										
47		Multi-Grain	Cheerios										
48		Nut&Honey	Crunch										
49		Nutri-Grain	Almond-Raisin										
1		100% Natural	Bran										
66		Smacks											
39		Just Right	Fruit & Nut										

52	fat	200	6.0	11.0	14	260	25	3	1.33	0.67			
36	1	250	1.5	11.5	10	90	25	1	1.00	0.75			
42	1	180	0.0	12.0	12	55	25	2	1.00	1.00			
69	1	200	0.0	21.0	3	35	100	3	1.00	1.00			
72	1	250	0.0	21.0	3	60	25	3	1.00	0.75			
41	2	150	2.0	12.0	6	95	25	2	1.00	0.67			
0	1	130	10.0	5.0	6	280	25	3	1.00	0.33			
31	1	280	0.0	15.0	9	45	25	2	1.00	0.75			
21	0	220	1.0	21.0	3	30	25	3	1.00	1.00			
5	2	180	1.5	10.5	10	70	25	1	1.00	0.75			
75	1	200	3.0	17.0	3	110	25	1	1.00	1.00			
12	3	210	0.0	13.0	9	45	25	2	1.00	0.75			
38	1	170	1.0	17.0	6	60	100	3	1.00	1.00			
57	2	0	2.7	-1.0	-1	110	0	1	1.00	0.67			
2	1	260	9.0	7.0	5	320	25	3	1.00	0.33			
63	0	0	3.0	16.0	0	95	0	1	0.83	1.00			
23	0	190	1.0	18.0	5	80	25	3	1.00	0.75			
8	1	200	4.0	15.0	6	125	25	1	1.00	0.67			
35	2	220	1.0	12.0	11	45	25	2	1.00	1.00			
4	2	200	1.0	14.0	8	-1	25	3	1.00	0.75			
27	2	160	5.0	12.0	10	200	25	3	1.25	0.67			
73	1	140	0.0	13.0	12	25	25	2	1.00	1.00			
74	1	230	3.0	17.0	3	115	25	1	1.00	0.67			
33	0	170	3.0	17.0	3	90	25	3	1.00	0.25			
51	2	170	1.5	13.5	10	120	25	3	1.25	0.50			
47	1	220	2.0	15.0	6	90	25	1	1.00	1.00			
48	1	190	0.0	15.0	9	40	25	2	1.00	0.67			
49	2	220	3.0	21.0	7	130	25	3	1.33	0.67			
1	5	15	2.0	8.0	8	135	0	3	1.00	1.00			
66	1	70	1.0	9.0	15	40	25	2	1.00	0.75			
39	1	170	2.0	20.0	9	95	100	3	1.30	0.75			

rating

```

52 37.840594
36 31.072217
42 26.734515
69 38.839746
72 39.106174
41 45.328074
0 68.402973
31 23.804043
21 46.895644
5 29.509541
75 51.592193
12 19.823573
38 36.523683
57 50.828392
2 59.425505
63 68.235885
23 44.330856
8 49.120253
35 21.871292
4 34.384843
27 40.917047
73 27.753301
74 49.787445
33 53.371007
51 30.450843
47 40.105965
48 29.924285
49 40.692320
1 33.983679
66 31.230054
39 36.471512

```

length of the train and test data

In [153...]

```

print(len(train_data))
print(len(test_data))

```

```

46
31

```

Without random_parameter

In [154...]

```

train_data,test_data=train_test_split(cer,test_size=0.4)
print(len(train_data))
print(len(test_data))

```

```

46
31

```

70:30

In [28]:

```

train_data1,test_data1=train_test_split(cer,test_size=0.3,random_state=100)
print(train_data1)
print(test_data1)

```

		name	mfr	type	calories	protein	\
37		Honey-comb	OTHERS	C	110	1	
73		Trix	NESTLE	C	110	1	
21		Crispix	KELLOGGS	C	110	2	
70		Total Raisin Bran	NESTLE	C	140	3	
32		Grape Nuts Flakes	OTHERS	C	100	3	
51		Oatmeal Raisin Crisp	NESTLE	C	130	3	
76		Wheaties Honey Gold	NESTLE	C	110	2	
17		Corn Pops	KELLOGGS	C	110	1	

72		Triples	NESTLE	C	110	2
0		100% Bran	OTHERS	C	70	4
18		Count Chocula	NESTLE	C	110	1
39		Just Right Fruit & Nut	KELLOGGS	C	140	3
47		Multi-Grain Cheerios	NESTLE	C	100	2
74		Wheat Chex	OTHERS	C	100	3
50		Nutri-grain Wheat	KELLOGGS	C	90	3
30		Golden Crisp	OTHERS	C	100	2
57		Quaker Oatmeal	OTHERS	H	100	5
62		Rice Krispies	KELLOGGS	C	110	2
45	Muesli Raisins; Peaches; & Pecans		OTHERS	C	150	4
68	Strawberry Fruit Wheats		OTHERS	C	90	2
71	Total Whole Grain		NESTLE	C	100	3
7	Basic 4		NESTLE	C	130	3
3	All-Bran with Extra Fiber		KELLOGGS	C	50	4
56	Quaker Oat Squares		OTHERS	C	100	4
54	Puffed Rice		OTHERS	C	50	1
36	Honey Nut Cheerios		NESTLE	C	110	3
19	Cracklin' Oat Bran		KELLOGGS	C	110	3
13	Clusters		NESTLE	C	110	3
1	100% Natural Bran		OTHERS	C	120	3
31	Golden Grahams		NESTLE	C	110	1
4	Almond Delight		OTHERS	C	110	2
44	Muesli Raisins; Dates; & Almonds		OTHERS	C	150	4
27	Fruit & Fibre Dates; Walnuts; and Oats		OTHERS	C	120	3
2	All-Bran		KELLOGGS	C	70	4
22	Crispy Wheat & Raisins		NESTLE	C	100	2
29	Fruity Pebbles		OTHERS	C	110	1
9	Bran Flakes		OTHERS	C	90	3
59	Raisin Nut Bran		NESTLE	C	100	3
16	Corn Flakes		KELLOGGS	C	100	2
58	Raisin Bran		KELLOGGS	C	120	3
60	Raisin Squares		KELLOGGS	C	90	2
15	Corn Chex		OTHERS	C	110	2
75	Wheaties		NESTLE	C	100	3
34	Great Grains Pecan		OTHERS	C	120	3
14	Cocoa Puffs		NESTLE	C	110	1
66	Smacks		KELLOGGS	C	110	2
53	Product 19		KELLOGGS	C	100	3
52	Post Nat. Raisin Bran		OTHERS	C	120	3
10	Cap'n'Crunch		OTHERS	C	120	1
48	Nut&Honey Crunch		KELLOGGS	C	120	2
67	Special K		KELLOGGS	C	110	6
24	Froot Loops		KELLOGGS	C	110	2
8	Bran Chex		OTHERS	C	90	2

	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
37	0	180	0.0	14.0	11	35	25	1	1.00	1.33	
73	1	140	0.0	13.0	12	25	25	2	1.00	1.00	
21	0	220	1.0	21.0	3	30	25	3	1.00	1.00	
70	1	190	4.0	15.0	14	230	100	3	1.50	1.00	
32	1	140	3.0	15.0	5	85	25	3	1.00	0.88	
51	2	170	1.5	13.5	10	120	25	3	1.25	0.50	
76	1	200	1.0	16.0	8	60	25	1	1.00	0.75	
17	0	90	1.0	13.0	12	20	25	2	1.00	1.00	
72	1	250	0.0	21.0	3	60	25	3	1.00	0.75	
0	1	130	10.0	5.0	6	280	25	3	1.00	0.33	
18	1	180	0.0	12.0	13	65	25	2	1.00	1.00	
39	1	170	2.0	20.0	9	95	100	3	1.30	0.75	
47	1	220	2.0	15.0	6	90	25	1	1.00	1.00	
74	1	230	3.0	17.0	3	115	25	1	1.00	0.67	
50	0	170	3.0	18.0	2	90	25	3	1.00	1.00	
30	0	45	0.0	11.0	15	40	25	1	1.00	0.88	
57	2	0	2.7	-1.0	-1	110	0	1	1.00	0.67	
62	0	290	0.0	22.0	3	35	25	1	1.00	1.00	
45	3	150	3.0	16.0	11	170	25	3	1.00	1.00	
68	0	15	3.0	15.0	5	90	25	2	1.00	1.00	
71	1	200	3.0	16.0	3	110	100	3	1.00	1.00	
7	2	210	2.0	18.0	8	100	25	3	1.33	0.75	

3	0	140	14.0	8.0	0	330	25	3	1.00	0.50
56	1	135	2.0	14.0	6	110	25	3	1.00	0.50
54	0	0	0.0	13.0	0	15	0	3	0.50	1.00
36	1	250	1.5	11.5	10	90	25	1	1.00	0.75
19	3	140	4.0	10.0	7	160	25	3	1.00	0.50
13	2	140	2.0	13.0	7	105	25	3	1.00	0.50
1	5	15	2.0	8.0	8	135	0	3	1.00	1.00
31	1	280	0.0	15.0	9	45	25	2	1.00	0.75
4	2	200	1.0	14.0	8	-1	25	3	1.00	0.75
44	3	95	3.0	16.0	11	170	25	3	1.00	1.00
27	2	160	5.0	12.0	10	200	25	3	1.25	0.67
2	1	260	9.0	7.0	5	320	25	3	1.00	0.33
22	1	140	2.0	11.0	10	120	25	3	1.00	0.75
29	1	135	0.0	13.0	12	25	25	2	1.00	0.75
9	0	210	5.0	13.0	5	190	25	3	1.00	0.67
59	2	140	2.5	10.5	8	140	25	3	1.00	0.50
16	0	290	1.0	21.0	2	35	25	1	1.00	1.00
58	1	210	5.0	14.0	12	240	25	2	1.33	0.75
60	0	0	2.0	15.0	6	110	25	3	1.00	0.50
15	0	280	0.0	22.0	3	25	25	1	1.00	1.00
75	1	200	3.0	17.0	3	110	25	1	1.00	1.00
34	3	75	3.0	13.0	4	100	25	3	1.00	0.33
14	1	180	0.0	12.0	13	55	25	2	1.00	1.00
66	1	70	1.0	9.0	15	40	25	2	1.00	0.75
53	0	320	1.0	20.0	3	45	100	3	1.00	1.00
52	1	200	6.0	11.0	14	260	25	3	1.33	0.67
10	2	220	0.0	12.0	12	35	25	2	1.00	0.75
48	1	190	0.0	15.0	9	40	25	2	1.00	0.67
67	0	230	1.0	16.0	3	55	25	1	1.00	1.00
24	1	125	1.0	11.0	13	30	25	2	1.00	1.00
8	1	200	4.0	15.0	6	125	25	1	1.00	0.67

rating

37	28.742414
73	27.753301
21	46.895644
70	28.592785
32	52.076897
51	30.450843
76	36.187559
17	35.782791
72	39.106174
0	68.402973
18	22.396513
39	36.471512
47	40.105965
74	49.787445
50	59.642837
30	35.252444
57	50.828392
62	40.560159
45	34.139765
68	59.363993
71	46.658844
7	37.038562
3	93.704912
56	49.511874
54	60.756112
36	31.072217
19	40.448772
13	40.400208
1	33.983679
31	23.804043
4	34.384843
44	37.136863
27	40.917047
2	59.425505
22	36.176196
29	28.025765

9 53.313813
 59 39.703400
 16 45.863324
 58 39.259197
 60 55.333142
 15 41.445019
 75 51.592193
 34 45.811716
 14 22.736446
 66 31.230054
 53 41.503540
 52 37.840594
 10 18.042851
 48 29.924285
 67 53.131324
 24 32.207582
 8 49.120253

		name	mfr	type	calories	protein	fat	\
5	Apple Cinnamon	Cheerios	NESTLE	C	110	2	2	
40		Kix	NESTLE	C	110	2	1	
11		Cheerios	NESTLE	C	110	6	2	
64	Shredded Wheat	'n'Bran	OTHERS	C	90	3	0	
23		Double Chex	OTHERS	C	100	2	0	
49	Nutri-Grain	Almond-Raisin	KELLOGGS	C	140	3	2	
63		Shredded Wheat	OTHERS	C	80	2	0	
55		Puffed Wheat	OTHERS	C	50	2	0	
69	Total Corn Flakes		NESTLE	C	110	2	1	
20	Cream of Wheat	(Quick)	OTHERS	H	100	3	0	
42		Lucky Charms	NESTLE	C	110	2	1	
41		Life	OTHERS	C	100	4	2	
25		Frosted Flakes	KELLOGGS	C	110	1	0	
65	Shredded Wheat	spoon size	OTHERS	C	90	3	0	
6		Apple Jacks	KELLOGGS	C	110	2	0	
46	Mueslix	Crispy Blend	KELLOGGS	C	160	3	2	
28		Fruitful Bran	KELLOGGS	C	120	3	0	
12	Cinnamon Toast	Crunch	NESTLE	C	120	1	3	
61		Rice Chex	OTHERS	C	110	1	0	
33		Grape-Nuts	OTHERS	C	110	3	0	
43		Maypo	OTHERS	H	100	4	1	
26	Frosted Mini-Wheats		KELLOGGS	C	100	3	0	
38	Just Right	Crunchy Nuggets	KELLOGGS	C	110	2	1	
35		Honey Graham Ohs	OTHERS	C	120	1	2	

	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
5	180	1.5	10.5	10	70	25	1	1.00	0.75	
40	260	0.0	21.0	3	40	25	2	1.00	1.50	
11	290	2.0	17.0	1	105	25	1	1.00	1.25	
64	0	4.0	19.0	0	140	0	1	1.00	0.67	
23	190	1.0	18.0	5	80	25	3	1.00	0.75	
49	220	3.0	21.0	7	130	25	3	1.33	0.67	
63	0	3.0	16.0	0	95	0	1	0.83	1.00	
55	0	1.0	10.0	0	50	0	3	0.50	1.00	
69	200	0.0	21.0	3	35	100	3	1.00	1.00	
20	80	1.0	21.0	0	-1	0	2	1.00	1.00	
42	180	0.0	12.0	12	55	25	2	1.00	1.00	
41	150	2.0	12.0	6	95	25	2	1.00	0.67	
25	200	1.0	14.0	11	25	25	1	1.00	0.75	
65	0	3.0	20.0	0	120	0	1	1.00	0.67	
6	125	1.0	11.0	14	30	25	2	1.00	1.00	
46	150	3.0	17.0	13	160	25	3	1.50	0.67	
28	240	5.0	14.0	12	190	25	3	1.33	0.67	
12	210	0.0	13.0	9	45	25	2	1.00	0.75	
61	240	0.0	23.0	2	30	25	1	1.00	1.13	
33	170	3.0	17.0	3	90	25	3	1.00	0.25	
43	0	0.0	16.0	3	95	25	2	1.00	1.00	
26	0	3.0	14.0	7	100	25	2	1.00	0.80	
38	170	1.0	17.0	6	60	100	3	1.00	1.00	
35	220	1.0	12.0	11	45	25	2	1.00	1.00	

	rating
5	29.509541
40	39.241114
11	50.764999
64	74.472949
23	44.330856
49	40.692320
63	68.235885
55	63.005645
69	38.839746
20	64.533816
42	26.734515
41	45.328074
25	31.435973
65	72.801787
6	33.174094
46	30.313351
28	41.015492
12	19.823573
61	41.998933
33	53.371007
43	54.850917
26	58.345141
38	36.523683
35	21.871292

In [29]:

```
print(len(train_data1))
print(len(test_data1))
```

```
53
24
```

In [170...]

```
train_data,test_data=train_test_split(cer,test_size=0.3)
print(train_data)
print(test_data)
```

		name	mfr	type	calories	protein	\
50	Nutri-grain	Wheat	KELLOGGS	C	90	3	
7		Basic 4	NESTLE	C	130	3	
65	Shredded Wheat	spoon size	OTHERS	C	90	3	
15		Corn Chex	OTHERS	C	110	2	
17		Corn Pops	KELLOGGS	C	110	1	
41		Life	OTHERS	C	100	4	
16		Corn Flakes	KELLOGGS	C	100	2	
69	Total	Corn Flakes	NESTLE	C	110	2	
43		Maypo	OTHERS	H	100	4	
55		Puffed Wheat	OTHERS	C	50	2	
2	All-Bran	KELLOGGS	C	70	4		
64	Shredded Wheat	'n'Bran	OTHERS	C	90	3	
73		Trix	NESTLE	C	110	1	
22	Crispy Wheat & Raisins		NESTLE	C	100	2	
30		Golden Crisp	OTHERS	C	100	2	
28		Fruitful Bran	KELLOGGS	C	120	3	
60		Raisin Squares	KELLOGGS	C	90	2	
51	Oatmeal Raisin Crisp		NESTLE	C	130	3	
39		Just Right Fruit & Nut	KELLOGGS	C	140	3	
27	Fruit & Fibre Dates; Walnuts; and Oats		OTHERS	C	120	3	
5	Apple Cinnamon Cheerios		NESTLE	C	110	2	
54		Puffed Rice	OTHERS	C	50	1	
11		Cheerios	NESTLE	C	110	6	
10		Cap'n'Crunch	OTHERS	C	120	1	
38	Just Right Crunchy Nuggets		KELLOGGS	C	110	2	
19		Cracklin' Oat Bran	KELLOGGS	C	110	3	
48		Nut&Honey Crunch	KELLOGGS	C	120	2	
1	100% Natural Bran		OTHERS	C	120	3	
56		Quaker Oat Squares	OTHERS	C	100	4	
61		Rice Chex	OTHERS	C	110	1	

0		100% Bran	OTHERS	C	70	4
36		Honey Nut Cheerios	NESTLE	C	110	3
53		Product 19	KELLOGGS	C	100	3
23		Double Chex	OTHERS	C	100	2
46		Mueslix Crispy Blend	KELLOGGS	C	160	3
26		Frosted Mini-Wheats	KELLOGGS	C	100	3
58		Raisin Bran	KELLOGGS	C	120	3
67		Special K	KELLOGGS	C	110	6
66		Smacks	KELLOGGS	C	110	2
74		Wheat Chex	OTHERS	C	100	3
24		Froot Loops	KELLOGGS	C	110	2
76		Wheaties Honey Gold	NESTLE	C	110	2
44		Muesli Raisins; Dates; & Almonds	OTHERS	C	150	4
71		Total Whole Grain	NESTLE	C	100	3
29		Fruity Pebbles	OTHERS	C	110	1
21		Crispix	KELLOGGS	C	110	2
35		Honey Graham Ohs	OTHERS	C	120	1
34		Great Grains Pecan	OTHERS	C	120	3
6		Apple Jacks	KELLOGGS	C	110	2
45		Muesli Raisins; Peaches; & Pecans	OTHERS	C	150	4
32		Grape Nuts Flakes	OTHERS	C	100	3
62		Rice Krispies	KELLOGGS	C	110	2
40		Kix	NESTLE	C	110	2

	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
50	0	170	3.0	18.0	2	90	25	3	1.00	1.00	
7	2	210	2.0	18.0	8	100	25	3	1.33	0.75	
65	0	0	3.0	20.0	0	120	0	1	1.00	0.67	
15	0	280	0.0	22.0	3	25	25	1	1.00	1.00	
17	0	90	1.0	13.0	12	20	25	2	1.00	1.00	
41	2	150	2.0	12.0	6	95	25	2	1.00	0.67	
16	0	290	1.0	21.0	2	35	25	1	1.00	1.00	
69	1	200	0.0	21.0	3	35	100	3	1.00	1.00	
43	1	0	0.0	16.0	3	95	25	2	1.00	1.00	
55	0	0	1.0	10.0	0	50	0	3	0.50	1.00	
2	1	260	9.0	7.0	5	320	25	3	1.00	0.33	
64	0	0	4.0	19.0	0	140	0	1	1.00	0.67	
73	1	140	0.0	13.0	12	25	25	2	1.00	1.00	
22	1	140	2.0	11.0	10	120	25	3	1.00	0.75	
30	0	45	0.0	11.0	15	40	25	1	1.00	0.88	
28	0	240	5.0	14.0	12	190	25	3	1.33	0.67	
60	0	0	2.0	15.0	6	110	25	3	1.00	0.50	
51	2	170	1.5	13.5	10	120	25	3	1.25	0.50	
39	1	170	2.0	20.0	9	95	100	3	1.30	0.75	
27	2	160	5.0	12.0	10	200	25	3	1.25	0.67	
5	2	180	1.5	10.5	10	70	25	1	1.00	0.75	
54	0	0	0.0	13.0	0	15	0	3	0.50	1.00	
11	2	290	2.0	17.0	1	105	25	1	1.00	1.25	
10	2	220	0.0	12.0	12	35	25	2	1.00	0.75	
38	1	170	1.0	17.0	6	60	100	3	1.00	1.00	
19	3	140	4.0	10.0	7	160	25	3	1.00	0.50	
48	1	190	0.0	15.0	9	40	25	2	1.00	0.67	
1	5	15	2.0	8.0	8	135	0	3	1.00	1.00	
56	1	135	2.0	14.0	6	110	25	3	1.00	0.50	
61	0	240	0.0	23.0	2	30	25	1	1.00	1.13	
0	1	130	10.0	5.0	6	280	25	3	1.00	0.33	
36	1	250	1.5	11.5	10	90	25	1	1.00	0.75	
53	0	320	1.0	20.0	3	45	100	3	1.00	1.00	
23	0	190	1.0	18.0	5	80	25	3	1.00	0.75	
46	2	150	3.0	17.0	13	160	25	3	1.50	0.67	
26	0	0	3.0	14.0	7	100	25	2	1.00	0.80	
58	1	210	5.0	14.0	12	240	25	2	1.33	0.75	
67	0	230	1.0	16.0	3	55	25	1	1.00	1.00	
66	1	70	1.0	9.0	15	40	25	2	1.00	0.75	
74	1	230	3.0	17.0	3	115	25	1	1.00	0.67	
24	1	125	1.0	11.0	13	30	25	2	1.00	1.00	
76	1	200	1.0	16.0	8	60	25	1	1.00	0.75	
44	3	95	3.0	16.0	11	170	25	3	1.00	1.00	
71	1	200	3.0	16.0	3	110	100	3	1.00	1.00	

29	1	135	0.0	13.0	12	25	25	2	1.00	0.75
21	0	220	1.0	21.0	3	30	25	3	1.00	1.00
35	2	220	1.0	12.0	11	45	25	2	1.00	1.00
34	3	75	3.0	13.0	4	100	25	3	1.00	0.33
6	0	125	1.0	11.0	14	30	25	2	1.00	1.00
45	3	150	3.0	16.0	11	170	25	3	1.00	1.00
32	1	140	3.0	15.0	5	85	25	3	1.00	0.88
62	0	290	0.0	22.0	3	35	25	1	1.00	1.00
40	1	260	0.0	21.0	3	40	25	2	1.00	1.50

rating

50	59.642837
7	37.038562
65	72.801787
15	41.445019
17	35.782791
41	45.328074
16	45.863324
69	38.839746
43	54.850917
55	63.005645
2	59.425505
64	74.472949
73	27.753301
22	36.176196
30	35.252444
28	41.015492
60	55.333142
51	30.450843
39	36.471512
27	40.917047
5	29.509541
54	60.756112
11	50.764999
10	18.042851
38	36.523683
19	40.448772
48	29.924285
1	33.983679
56	49.511874
61	41.998933
0	68.402973
36	31.072217
53	41.503540
23	44.330856
46	30.313351
26	58.345141
58	39.259197
67	53.131324
66	31.230054
74	49.787445
24	32.207582
76	36.187559
44	37.136863
71	46.658844
29	28.025765
21	46.895644
35	21.871292
34	45.811716
6	33.174094
45	34.139765
32	52.076897
62	40.560159
40	39.241114

		name	mfr	type	calories	protein	fat	sodium	\
25		Frosted Flakes	KELLOGGS	C	110	1	0	200	
68		Strawberry Fruit Wheats	OTHERS	C	90	2	0	15	
49		Nutri-Grain Almond-Raisin	KELLOGGS	C	140	3	2	220	
12		Cinnamon Toast Crunch	NESTLE	C	120	1	3	210	

33	Grape-Nuts	OTHERS	C	110	3	0	170
75	Wheaties	NESTLE	C	100	3	1	200
3	All-Bran with Extra Fiber	KELLOGGS	C	50	4	0	140
4	Almond Delight	OTHERS	C	110	2	2	200
8	Bran Chex	OTHERS	C	90	2	1	200
57	Quaker Oatmeal	OTHERS	H	100	5	2	0
47	Multi-Grain Cheerios	NESTLE	C	100	2	1	220
72	Triples	NESTLE	C	110	2	1	250
9	Bran Flakes	OTHERS	C	90	3	0	210
70	Total Raisin Bran	NESTLE	C	140	3	1	190
59	Raisin Nut Bran	NESTLE	C	100	3	2	140
63	Shredded Wheat	OTHERS	C	80	2	0	0
14	Cocoa Puffs	NESTLE	C	110	1	1	180
52	Post Nat. Raisin Bran	OTHERS	C	120	3	1	200
18	Count Chocula	NESTLE	C	110	1	1	180
42	Lucky Charms	NESTLE	C	110	2	1	180
31	Golden Grahams	NESTLE	C	110	1	1	280
20	Cream of Wheat (Quick)	OTHERS	H	100	3	0	80
37	Honey-comb	OTHERS	C	110	1	0	180
13	Clusters	NESTLE	C	110	3	2	140

	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
25	1.0	14.0	11	25	25	1	1.00	0.75	31.435973
68	3.0	15.0	5	90	25	2	1.00	1.00	59.363993
49	3.0	21.0	7	130	25	3	1.33	0.67	40.692320
12	0.0	13.0	9	45	25	2	1.00	0.75	19.823573
33	3.0	17.0	3	90	25	3	1.00	0.25	53.371007
75	3.0	17.0	3	110	25	1	1.00	1.00	51.592193
3	14.0	8.0	0	330	25	3	1.00	0.50	93.704912
4	1.0	14.0	8	-1	25	3	1.00	0.75	34.384843
8	4.0	15.0	6	125	25	1	1.00	0.67	49.120253
57	2.7	-1.0	-1	110	0	1	1.00	0.67	50.828392
47	2.0	15.0	6	90	25	1	1.00	1.00	40.105965
72	0.0	21.0	3	60	25	3	1.00	0.75	39.106174
9	5.0	13.0	5	190	25	3	1.00	0.67	53.313813
70	4.0	15.0	14	230	100	3	1.50	1.00	28.592785
59	2.5	10.5	8	140	25	3	1.00	0.50	39.703400
63	3.0	16.0	0	95	0	1	0.83	1.00	68.235885
14	0.0	12.0	13	55	25	2	1.00	1.00	22.736446
52	6.0	11.0	14	260	25	3	1.33	0.67	37.840594
18	0.0	12.0	13	65	25	2	1.00	1.00	22.396513
42	0.0	12.0	12	55	25	2	1.00	1.00	26.734515
31	0.0	15.0	9	45	25	2	1.00	0.75	23.804043
20	1.0	21.0	0	-1	0	2	1.00	1.00	64.533816
37	0.0	14.0	11	35	25	1	1.00	1.33	28.742414
13	2.0	13.0	7	105	25	3	1.00	0.50	40.400208

```
In [34]: print(len(train_data1))
print(len(test_data1))
```

53
24

80:20

```
In [31]: train_data2,test_data2=train_test_split(cer,test_size=0.2,random_state=80)
print(train_data2)
print(test_data2)
```

	name	mfr	type	calories	protein	fat	\
36	Honey Nut Cheerios	NESTLE	C	110	3	1	
76	Wheaties Honey Gold	NESTLE	C	110	2	1	
67	Special K	KELLOGGS	C	110	6	0	
75	Wheaties	NESTLE	C	100	3	1	
22	Crispy Wheat & Raisins	NESTLE	C	100	2	1	
..

50	Nutri-grain Wheat	KELLOGGS	C	90	3	0				
10	Cap'n'Crunch	OTHERS	C	120	1	2				
38	Just Right Crunchy Nuggets	KELLOGGS	C	110	2	1				
60	Raisin Squares	KELLOGGS	C	90	2	0				
47	Multi-Grain Cheerios	NESTLE	C	100	2	1				
						\				
36	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
36	250	1.5	11.5	10	90	25	1	1.0	0.75	
76	200	1.0	16.0	8	60	25	1	1.0	0.75	
67	230	1.0	16.0	3	55	25	1	1.0	1.00	
75	200	3.0	17.0	3	110	25	1	1.0	1.00	
22	140	2.0	11.0	10	120	25	3	1.0	0.75	
..
50	170	3.0	18.0	2	90	25	3	1.0	1.00	
10	220	0.0	12.0	12	35	25	2	1.0	0.75	
38	170	1.0	17.0	6	60	100	3	1.0	1.00	
60	0	2.0	15.0	6	110	25	3	1.0	0.50	
47	220	2.0	15.0	6	90	25	1	1.0	1.00	
	rating									
36	31.072217									
76	36.187559									
67	53.131324									
75	51.592193									
22	36.176196									
..	...									
50	59.642837									
10	18.042851									
38	36.523683									
60	55.333142									
47	40.105965									

[61 rows x 16 columns]

	name	mfr	type	calories	protein	fat	sodium	\		
53	Product 19	KELLOGGS	C	100	3	0	320			
2	All-Bran	KELLOGGS	C	70	4	1	260			
29	Fruity Pebbles	OTHERS	C	110	1	1	135			
58	Raisin Bran	KELLOGGS	C	120	3	1	210			
59	Raisin Nut Bran	NESTLE	C	100	3	2	140			
40	Kix	NESTLE	C	110	2	1	260			
13	Clusters	NESTLE	C	110	3	2	140			
62	Rice Krispies	KELLOGGS	C	110	2	0	290			
73	Trix	NESTLE	C	110	1	1	140			
6	Apple Jacks	KELLOGGS	C	110	2	0	125			
11	Cheerios	NESTLE	C	110	6	2	290			
46	Mueslix Crispy Blend	KELLOGGS	C	160	3	2	150			
12	Cinnamon Toast Crunch	NESTLE	C	120	1	3	210			
18	Count Chocula	NESTLE	C	110	1	1	180			
0	100% Bran	OTHERS	C	70	4	1	130			
63	Shredded Wheat	OTHERS	C	80	2	0	0			
	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating	\
53	1.0	20.0	3	45	100	3	1.00	1.00	41.503540	
2	9.0	7.0	5	320	25	3	1.00	0.33	59.425505	
29	0.0	13.0	12	25	25	2	1.00	0.75	28.025765	
58	5.0	14.0	12	240	25	2	1.33	0.75	39.259197	
59	2.5	10.5	8	140	25	3	1.00	0.50	39.703400	
40	0.0	21.0	3	40	25	2	1.00	1.50	39.241114	
13	2.0	13.0	7	105	25	3	1.00	0.50	40.400208	
62	0.0	22.0	3	35	25	1	1.00	1.00	40.560159	
73	0.0	13.0	12	25	25	2	1.00	1.00	27.753301	
6	1.0	11.0	14	30	25	2	1.00	1.00	33.174094	
11	2.0	17.0	1	105	25	1	1.00	1.25	50.764999	
46	3.0	17.0	13	160	25	3	1.50	0.67	30.313351	
12	0.0	13.0	9	45	25	2	1.00	0.75	19.823573	
18	0.0	12.0	13	65	25	2	1.00	1.00	22.396513	
0	10.0	5.0	6	280	25	3	1.00	0.33	68.402973	
63	3.0	16.0	0	95	0	1	0.83	1.00	68.235885	

```
In [32]: print(len(train_data))
print(len(test_data))
```

46
31

without parameter

```
In [33]: train_data,test_data=train_test_split(cer,test_size=0.2)
print(train_data2)
print(test_data2)
```

		name	mfr	type	calories	protein	fat	\			
36		Honey Nut Cheerios	NESTLE	C	110	3	1				
76		Wheaties Honey Gold	NESTLE	C	110	2	1				
67		Special K	KELLOGGS	C	110	6	0				
75		Wheaties	NESTLE	C	100	3	1				
22		Crispy Wheat & Raisins	NESTLE	C	100	2	1				
..				
50		Nutri-grain Wheat	KELLOGGS	C	90	3	0				
10		Cap'n'Crunch	OTHERS	C	120	1	2				
38		Just Right Crunchy Nuggets	KELLOGGS	C	110	2	1				
60		Raisin Squares	KELLOGGS	C	90	2	0				
47		Multi-Grain Cheerios	NESTLE	C	100	2	1				
		sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	\
36		250	1.5	11.5	10	90	25	1	1.0	0.75	
76		200	1.0	16.0	8	60	25	1	1.0	0.75	
67		230	1.0	16.0	3	55	25	1	1.0	1.00	
75		200	3.0	17.0	3	110	25	1	1.0	1.00	
22		140	2.0	11.0	10	120	25	3	1.0	0.75	
..	
50		170	3.0	18.0	2	90	25	3	1.0	1.00	
10		220	0.0	12.0	12	35	25	2	1.0	0.75	
38		170	1.0	17.0	6	60	100	3	1.0	1.00	
60		0	2.0	15.0	6	110	25	3	1.0	0.50	
47		220	2.0	15.0	6	90	25	1	1.0	1.00	
		rating									
36		31.072217									
76		36.187559									
67		53.131324									
75		51.592193									
22		36.176196									
..		...									
50		59.642837									
10		18.042851									
38		36.523683									
60		55.333142									
47		40.105965									

[61 rows x 16 columns]

		name	mfr	type	calories	protein	fat	sodium	\
53		Product 19	KELLOGGS	C	100	3	0	320	
2		All-Bran	KELLOGGS	C	70	4	1	260	
29		Fruity Pebbles	OTHERS	C	110	1	1	135	
58		Raisin Bran	KELLOGGS	C	120	3	1	210	
59		Raisin Nut Bran	NESTLE	C	100	3	2	140	
40		Kix	NESTLE	C	110	2	1	260	
13		Clusters	NESTLE	C	110	3	2	140	
62		Rice Krispies	KELLOGGS	C	110	2	0	290	
73		Trix	NESTLE	C	110	1	1	140	
6		Apple Jacks	KELLOGGS	C	110	2	0	125	
11		Cheerios	NESTLE	C	110	6	2	290	
46		Mueslix Crispy Blend	KELLOGGS	C	160	3	2	150	
12		Cinnamon Toast Crunch	NESTLE	C	120	1	3	210	
18		Count Chocula	NESTLE	C	110	1	1	180	
0		100% Bran	OTHERS	C	70	4	1	130	

63	Shredded Wheat			OTHERS	C	80	2	0	0
	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
53	1.0	20.0	3	45	100	3	1.00	1.00	41.503540
2	9.0	7.0	5	320	25	3	1.00	0.33	59.425505
29	0.0	13.0	12	25	25	2	1.00	0.75	28.025765
58	5.0	14.0	12	240	25	2	1.33	0.75	39.259197
59	2.5	10.5	8	140	25	3	1.00	0.50	39.703400
40	0.0	21.0	3	40	25	2	1.00	1.50	39.241114
13	2.0	13.0	7	105	25	3	1.00	0.50	40.400208
62	0.0	22.0	3	35	25	1	1.00	1.00	40.560159
73	0.0	13.0	12	25	25	2	1.00	1.00	27.753301
6	1.0	11.0	14	30	25	2	1.00	1.00	33.174094
11	2.0	17.0	1	105	25	1	1.00	1.25	50.764999
46	3.0	17.0	13	160	25	3	1.50	0.67	30.313351
12	0.0	13.0	9	45	25	2	1.00	0.75	19.823573
18	0.0	12.0	13	65	25	2	1.00	1.00	22.396513
0	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
63	3.0	16.0	0	95	0	1	0.83	1.00	68.235885

In [176...]

```
print(len(train_data))
print(len(test_data))
```

61
16

From the above program we executed the program with random parameter and without random parameter. basically we should get different types values i guess but here i got same values.

8. Apply MinMaxScaler() and StandardScaler() to the following features: calories, protien, fat, sodium, fiber, carbo, sugars.

MinMaxScalar()

The MinMaxscaler is a type of scaler that scales the minimum and maximum values to be 0 and 1 respectively

In [233...]

```
from sklearn.preprocessing import MinMaxScaler
```

In [242...]

```
scaler=MinMaxScaler()# defining a standard scalar
cer[['protein','calories','fat','sodium','fiber','carbo','sugars']] = scaler.fit_transform(cer)
```

Out[242...]

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass
0	100% Bran	OTHERS	C	0.181818		0.6	0.2	0.406250	0.714286	0.250000	0.4375
1	100% Natural Bran	OTHERS	C	0.636364		0.4	1.0	0.046875	0.142857	0.375000	0.5625
2	All-Bran	KELLOGGS	C	0.181818		0.6	0.2	0.812500	0.642857	0.333333	0.3750
3	All-Bran with Extra Fiber	KELLOGGS	C	0.000000		0.6	0.0	0.437500	1.000000	0.375000	0.0625
4	Almond Delight	OTHERS	C	0.545455		0.2	0.4	0.625000	0.071429	0.625000	0.5625

		name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass
...
72	Triples	NESTLE	C	0.545455	0.2	0.2	0.781250	0.000000	0.916667	0.2500	60	
73	Trix	NESTLE	C	0.545455	0.0	0.2	0.437500	0.000000	0.583333	0.8125	25	
74	Wheat Chex	OTHERS	C	0.454545	0.4	0.2	0.718750	0.214286	0.750000	0.2500	115	
75	Wheaties	NESTLE	C	0.454545	0.4	0.2	0.625000	0.214286	0.750000	0.2500	110	
76	Wheaties Honey Gold	NESTLE	C	0.545455	0.2	0.2	0.625000	0.071429	0.708333	0.5625	60	

77 rows × 16 columns



StandardScaler()

StandardScaler is used to resize the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1.

In [243...]

```
scaler1=StandardScaler()# defining a standard scalar
cer[['protein','calories','fat','sodium','fiber','carbo','sugars']] = scaler1.fit_transform(cer)
```

Out[243...]

		name	mfr	type	calories	protein	fat	sodium	fiber	carbo	s
0	100% Bran	OTHERS	C	-1.905397	1.337319	-0.012988	-0.356306	3.314439	-2.257639	-0.21	
1	100% Natural Bran	OTHERS	C	0.677623	0.417912	3.987349	-1.737087	-0.064172	-1.551936	0.24	
2	All-Bran	KELLOGGS	C	-1.905397	1.337319	-0.012988	1.204578	2.892113	-1.787170	-0.41	
3	All-Bran with Extra Fiber	KELLOGGS	C	-2.938605	1.337319	-1.013072	-0.236238	5.003745	-1.551936	-1.51	
4	Almond Delight	OTHERS	C	0.161019	-0.501495	0.987096	0.484170	-0.486498	-0.140530	0.24	
...
72	Triples	NESTLE	C	0.161019	-0.501495	-0.012988	1.084510	-0.908824	1.506111	-0.81	
73	Trix	NESTLE	C	0.161019	-1.420902	-0.012988	-0.236238	-0.908824	-0.375764	1.11	
74	Wheat Chex	OTHERS	C	-0.355585	0.417912	-0.012988	0.844374	0.358155	0.565173	-0.81	
75	Wheaties	NESTLE	C	-0.355585	0.417912	-0.012988	0.484170	0.358155	0.565173	-0.81	
76	Wheaties Honey Gold	NESTLE	C	0.161019	-0.501495	-0.012988	0.484170	-0.486498	0.329939	0.24	

77 rows × 16 columns

9. Does the standard or min-max scaling make a difference in value distribution? Support your answers with some visualisations on the above dataset.

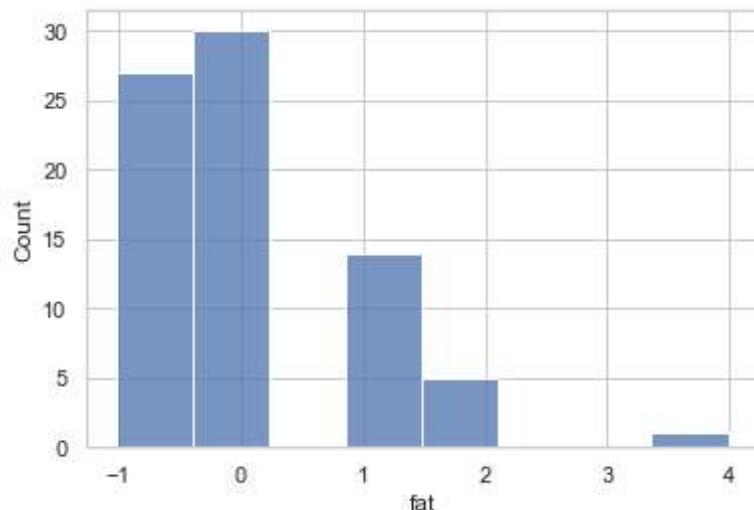
yes,it make difference

- because the MinMaxscaler is a type of scaler that scales the minimum and maximum values to be 0 and 1 respectively.
- and the StandardScaler scales all values between min and max so that they fall within a range from min to max

In [252...]

```
sns.histplot(data=cer,x='fat')
```

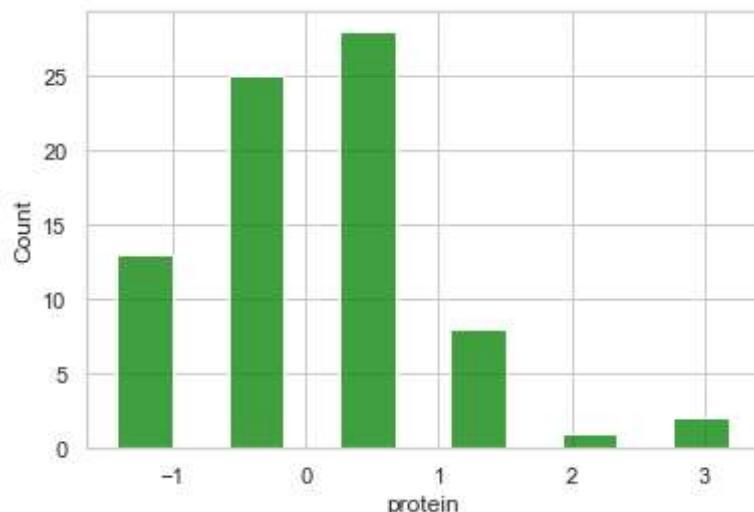
Out[252...]



In [246...]

```
sns.histplot(data=cer,x='protein',color="green")
```

Out[246...]



10. As an extension of 7th step, Generate a new Pandas DataFrame with the following columns based on the Training

Dataset: Split Ratio | Random State | Total Number of Entries | Count of Kellogg's | Count of Nestle | Count of Other Brands

In [100]:

```
import pandas as pd

df = pd.DataFrame(columns = ["Split Ratio", "Random_state", "Total num.of Entries", "Count of kellogs", "count of Nestle", "Count of Other Brands"])
df
```

Out[100]:

	Split Ratio	Random_state	Total num.of Entries	Count of kellogs	count of Nestle	Count of Other Brands
--	-------------	--------------	----------------------	------------------	-----------------	-----------------------

split ratio

60:40

In [88]:

```
train_data['mfr'].count()
```

Out[88]: 61

In [89]:

```
train_data['mfr'].value_counts()
```

```
Out[89]: OTHERS      26
KELLOGGS    18
NESTLE       17
Name: mfr, dtype: int64
```

In [90]:

```
data_1 = {"Split Ratio": "60:40", "Random_state": 20, "Total num.of Entries": 61, "Count of kellogs": 16, "count of Nestle": 11, "Count of Other Brands": 19}
df = df.append(data_1, ignore_index=True)
```

In [91]:

```
data_1
```

```
Out[91]: {'Split Ratio': '60:40',
          'Random_state': 20,
          'Total num.of Entries': 61,
          'Count of kellogs': 16,
          'count of Nestle': 11,
          'Count of Other Brands': 19}
```

split ratio :70:30

In [92]:

```
train_data1['mfr'].count()
```

Out[92]: 53

In [93]:

```
train_data1['mfr'].value_counts()
```

```
Out[93]: OTHERS      21
KELLOGGS    16
NESTLE       16
Name: mfr, dtype: int64
```

In [94]:

```
data_2 = {"Split Ratio": "70:30", "Random_state": 100, "Total num.of Entries": 53, "Count of kellogs": 16, "count of Nestle": 11, "Count of Other Brands": 19}
df = df.append(data_2, ignore_index=True)
data_2
```

```
Out[94]: {'Split Ratio': '70:30',
           'Random_state': 100,
           'Total num.of Entries': 53,
           'Countof kellogs': 16,
           'count of Nestle': 16,
           'Count of Other Brands': 21}
```

split_ratio 80:20

```
In [95]: train_data2['mfr'].count()
```

```
Out[95]: 61
```

```
In [96]: train_data2['mfr'].value_counts()
```

```
Out[96]: OTHERS      29
KELLOGGS    17
NESTLE      15
Name: mfr, dtype: int64
```

```
In [102...]: data_3 = {"Split Ratio": "80:20", "Random_state": 80, "Total num.of Entries": 61, "Countof kellogs": 17, "count of Nestle": 15, "Count of Other Brands": 29}
df = df.append(data_3, ignore_index=True)
data_3
```

```
Out[102...]: {'Split Ratio': '80:20',
               'Random_state': 80,
               'Total num.of Entries': 61,
               'Countof kellogs': 17,
               'count of Nestle': 15,
               'Count of Other Brands': 29}
```

```
In [104...]: df
```

	Split Ratio	Random_state	Total num.of Entries	Countof kellogs	count of Nestle	Count of Other Brands
0	60:40	20	61	16	11	19
1	70:30	100	53	16	16	21
2	80:20	80	61	17	15	29
3	80:20	80	61	17	15	29

CONCLUSION

Here we understand more about visualizing part and about some libraries regarding Visualisation . And also how to load the datasets and after loading datasets we understand how to split and work on the training and testing data and we tried to do some mathematical operations such as MinMaxScaler and standard scaler.In Machine Learning, StandardScaler is used to resize the distribution of values so that the mean of the observed values is 0 and the standard deviation is 1 and the MinMaxscaler is a type of scaler that scales the minimum and maximum values to be 0 and 1 respectively.And we understand how to create a dataframe using existing data.

REFERENCE

- <https://www.geeksforgeeks.org/seaborn-barplot-method-in-python/>
- <https://www.kite.com/python/answers/how-to-replace-column-values-in-a-pandas-dataframe-in-python>
- <https://www.javatpoint.com/pandas-dataframe-describe>,
<https://www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/>, -
- <https://www.geeksforgeeks.org/matplotlib-pyplot-hist-in-python>,
<https://www.geeksforgeeks.org/how-to-make-histograms-with-density-plots-with-seaborn-histplot/>
- <https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/>

In []: