In [153]:
```
import pandas as pd
import numpy as np
```

In [154]:
```
df=open(r"C:\Users\feche\.spyder-py3\Curso Python\datasets\cereals\Cereal Data.txt","r")
df
```

Out[154]:
```
<_io.TextIOWrapper name='C:\\Users\\feche\\.spyder-py3\\Curso Python\\datasets\\cereals\\Cereal Data.txt' mode
='r' encoding='cp1252'>
```

In [155]:
```
col_names=pd.read_excel(r"C:\Users\feche\.spyder-py3\Curso Python\datasets\cereals\Cereal data columns.xlsx")
col_names
```

Out[155]:

	Cereal
0	Manufacturer
1	Type
2	Calories
3	Protein
4	Fat
5	Sodium
6	Dietary Fiber
7	Complex Carbohydrates
8	Sugar
9	Display Shelf
10	Potassium
11	Vitamins & Minerals
12	Weight

In [156]:
```
col_names_list=[]
col_names_list.append("Cereal")
col_names_list
```

Out[156]:
```
['Cereal']
```

In [157]:
```
col_names_list2=col_names["Cereal"].tolist()
```

In [158]:
```
for i in range(len(col_names_list2)):
    col_names_list.append(col_names_list2[i])

col_names_list
```

Out[158]:
```
['Cereal',
 'Manufacturer',
 'Type',
 'Calories',
 'Protein',
 'Fat',
 'Sodium',
 'Dietary Fiber',
 'Complex Carbohydrates',
 'Sugar ',
 'Display Shelf',
 'Potassium',
 'Vitamins & Minerals',
 'Weight']
```

In [159]:
```
counter=0
main_dic={}
for col in col_names_list:
    main_dic[col]=[]
main_dic
```

Out[159]:
```
{'Cereal': [],
 'Manufacturer': [],
 'Type': [],
 'Calories': [],
 'Protein': [],
 'Fat': [],
 'Sodium': [],
 'Dietary Fiber': [],
 'Complex Carbohydrates': [],
 'Sugar ': [],
 'Display Shelf': [],
 'Potassium': [],
 'Vitamins & Minerals': [],
 'Weight': []}
```

In [160]:
```
for line in df:
    values=line.strip().split(" ")
    for i in range(len(col_names_list)):
        main_dic[col_names_list[i]].append(values[i])
    counter+=1

counter
```

Out[160]:
```
77
```

In [161]:
```
main_dic
```

```
 'Apple_Cinnamon_Cheerios',
 'Apple_Jacks',
 'Basic_4',
 'Bran_Chex',
 'Bran_Flakes',
 "Cap'n'Crunch",
 'Cheerios',
 'Cinnamon_Toast_Crunch',
 'Clusters',
 'Cocoa_Puffs',
 'Corn_Chex',
 'Corn_Flakes',
 'Corn_Pops',
 'Count_Chocula',
 "Cracklin'_Oat_Bran",
 'Cream_of_Wheat_(Quick)',
 'Crispix',
 'Crispy_Wheat_&_Raisins',
 'Double_Chex',
 'Froot_Loops',
```

In [162]:
```
data=pd.DataFrame(main_dic)
```

In [163]:
```
data.head()
```

Out[163]:

	Cereal	Manufacturer	Type	Calories	Protein	Fat	Sodium	Dietary Fiber	Complex Carbohydrates	Sugar	Display Shelf	Potassium	Vitamins & Minerals
0	100%_Bran	N	C	70	4	1	130	10	5	6	3	280	25
1	100%_Natural_Bran	Q	C	120	3	5	15	2	8	8	3	135	0
2	All-Bran	K	C	70	4	1	260	9	7	5	3	320	25
3	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14	8	0	3	330	25
4	Almond_Delight	R	C	110	2	2	200	1	14	8	3	-1	25

In [185]:
```
data.shape
```

Out[185]:
```
(77, 15)
```

In [164]:
```
data[["Calories", "Protein", "Fat", "Sodium","Dietary Fiber", "Complex Carbohydrates", "Sugar ", "Potassium", "V
data.dtypes
```

Out[164]:
```
Cereal                   object
Manufacturer             object
Type                     object
Calories                  int64
Protein                   int64
Fat                       int64
Sodium                    int64
Dietary Fiber           float64
Complex Carbohydrates   float64
Sugar                     int64
Display Shelf            object
Potassium                 int64
Vitamins & Minerals       int64
Weight                   object
dtype: object
```

In [165]:
```
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import RFE
from sklearn.svm import SVR
```

In [166]:
```
feature_cols=["Protein", "Fat", "Sodium","Dietary Fiber", "Complex Carbohydrates", "Sugar ", "Potassium", "Vitar
```

In [167]:
```
col_names=data.columns.values.tolist()
col_names
```

Out[167]:
```
['Cereal',
 'Manufacturer',
 'Type',
 'Calories',
 'Protein',
 'Fat',
 'Sodium',
 'Dietary Fiber',
 'Complex Carbohydrates',
 'Sugar ',
 'Display Shelf',
 'Potassium',
 'Vitamins & Minerals',
 'Weight']
```

In [168]:
```
X=data[feature_cols]
Y=data["Calories"]
```

In [169]:
```
estimator=SVR(kernel="linear")
selector=RFE(estimator,8,step=1)
selector=selector.fit(X,Y)
```

In [170]:
```
selector.support_
```

Out[170]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True])
```

In [171]:
```
selector.ranking_
```

Out[171]:
```
array([1, 1, 1, 1, 1, 1, 1, 1])
```

In [172]:
```
lm=LinearRegression()
lm.fit(X,Y)
```

Out[172]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [173]:
```
print(lm.intercept_)
print(lm.coef_)
```

```
22.11145682760973
[ 5.74639942e+00  8.43287251e+00  2.45646695e-03 -9.39108086e-01
  2.81413513e+00  3.22898332e+00 -6.64605092e-03  1.55397354e-02]
```

In [174]:
```
lm.score(X,Y)
```

Out[174]:
```
0.8357452532148784
```

In [175]:
```
list(zip(feature_cols,lm.coef_))
```

Out[175]:
```
[('Protein', 5.746399418090031),
 ('Fat', 8.432872512221525),
 ('Sodium', 0.0024564669528756933),
 ('Dietary Fiber', -0.9391080862390954),
 ('Complex Carbohydrates', 2.814135127424553),
 ('Sugar ', 3.228983316982189),
 ('Potassium', -0.006646050916592121),
 ('Vitamins & Minerals', 0.015539735440727198)]
```

In [176]:
```
calories_pred=lm.predict(X)
```

In [191]:
```
data["Calories_Pred"]=calories_pred
data.head(10)
```

Out[191]:

	Cereal	Manufacturer	Type	Calories	Protein	Fat	Sodium	Dietary Fiber	Complex Carbohydrates	Sugar	Display Shelf	Potassium	Vita Min
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6	3	280	
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8	3	135	
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	3	320	
3	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0	8.0	0	3	330	
4	Almond_Delight	R	C	110	2	2	200	1.0	14.0	8	3	-1	
5	Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5	10.5	10	1	70	
6	Apple_Jacks	K	C	110	2	0	125	1.0	11.0	14	2	30	
7	Basic_4	G	C	130	3	2	210	2.0	18.0	8	3	100	
8	Bran_Chex	R	C	90	2	1	200	4.0	15.0	6	1	125	
9	Bran_Flakes	P	C	90	3	0	210	5.0	13.0	5	3	190	

In [178]:
```
calories_mean=np.mean(data["Calories"])
calories_mean
```

Out[178]: 106.88311688311688

In [179]:
```
SSD=sum((data["Calories"]-data["Calories_Pred"])**2)
SSD
```

Out[179]: 4739.069421530211

In [180]:
```
RSE=np.sqrt(SSD/(len(data)-len(feature_cols)-1))
RSE
```

Out[180]: 8.348185274384145

In [184]:
```
error=RSE/calories_mean
error
```

Out[184]: 0.07810574315037414

In [ ]:
```

```