🍅Food Nutrition ML🍇

Done by

@Freddy Bicandy

Research Code

https://colab.research.google.com/drive/1CEkQF3WFZZ2quyrrnvk7bFEilhrSk2qJ?usp=sharing

Datasets

FOOD-DATA-GROUP1.csv

FOOD-DATA-GROUP2.csv

FOOD-DATA-GROUP3.csv

FOOD-DATA-GROUP4.csv

FOOD-DATA-GROUP5.csv

Project Description

This is a Food nutritionist Machine learning Model

In the world of fast food and delivery we wanted to help one who is looking for a healthy food based on their need.

therefore our solution is here to help by leveraging your diet plan by specifying the columns specifications your food with their values to make a prediction with a real food name to match your diet result

Project Main parts:

Research

the research here was about Food specifications and the need of each one we have collected a 37 food specifications found 5 large datasets mentioned below which determines the Main characteristics for the need of a human body

Exploratory data analysis (EDA)

and here we were required to study all 5 datasets concatenated by piloting and correlating selected columns in order to get a clear idea about Food types and their characterises (find below the EDA Made with All the visual output)

Expert Guidance

we used various dietitians support ways to check the most important features to be considered to support in our ML Model for their needs based on the collected data we were able to categorise our target audience

Model 1 Gym Enthusiasts

- **Objective**: Maximise **protein** for muscle building while balancing fat and caloric intake.
- **Features**
  - Protein (y as target)
  - Caloric Value
  - Fat
  - Carbohydrates

Model 2 Diet/Weight Loss

- **Objective**: Minimise calories while keeping the food nutritionally rich.
- **Features**
    - Caloric value (y as target)
    - protein
    - dietary Fibre
    - Fat

Heart-Healthy Diet

- **Objective**: Minimise saturated fats and cholesterol.
- **Features**:
    - Saturated Fats (y as target)
    - Cholesterol
    - Monounsaturated Fats
    - Polyunsaturated Fats
- **Constraints**: High fibre, low sodium, and balanced omega fats.

Balanced Nutrition

- **Objective**: Optimise for high nutrient density.
- **Features**:
    - Nutrition Density (y as target)
    - Protein
    - Vitamins (A, C, E, B-family)
    - Minerals (Calcium, Iron, Magnesium, etc.)

Development and Testing

in this phase we have started to develop our model based on various expectations one of which we tried to one hot encoding all cols to train our model in a Classification Technique using the Random-Forest And the Decision-Tree which it perform a low accuracy because it need much more data than we what we have so we decide to shift our training model into Regression in which beside classifying food by names we will be using y as the protein (PREDICATION) and the X will be any 2 or more values given by the user example

X=[”Col1”,”Col2”,etc..]

so we have a function which takes columns names example “Fat”,”Sugar” values=[12,32]

y=will produce the most highly protein contained food with such specifications because after the research we categorised 4 main nutritional goals

as mentioned above we will be have 4 different Models each will be specified for a target audience

Performance and Accuracy

in this phase we used the famous:

- mean_squared_error which measures the average squared difference between predictions and actual values
- r2_score metrics This score represents how well your model explains the variance in the data:
    - $R^2 = 1$ means this is prefect 1*100=100%
    - $R^2 = 0$ No better than predicting the mean.

📉Column Descriptions

Column name	Description
Food	The name or type of the food item.
Caloric Value	Total energy provided by the food, typically measured in kilocalories (kcal) per 100 grams
Fat (in g)	Total amount of fats in grams per 100 grams, including the breakdowns that follow.

Column name	Description
Saturated Fats (in g)	Amount of saturated fats (fats that typically raise the level of cholesterol in the blood) in grams per 100 grams.
Monounsaturated Fats (in g)	Amount of monounsaturated fats (considered heart-healthy fats) in grams per 100 grams.
Polyunsaturated Fats (in g)	Amount of polyunsaturated fats (include essential fats your body needs but can't produce itself) in grams per 100 grams.
Carbohydrates (in g)	Total carbohydrates in grams per 100 grams, including sugars.
Sugars (in g)	Total sugars in grams per 100 grams, a subset of carbohydrates
Protein (in g)	Total proteins in grams per 100 grams, essential for body repair and growth.
Dietary Fiber (in g)	Fiber content in grams per 100 grams, important for digestive health.
Cholesterol (in mg)	Cholesterol content in milligrams per 100 grams, pertinent for cardiovascular health.
Sodium (in mg)	Sodium content in milligrams per 100 grams, crucial for fluid balance and nerve function.
Water (in g)	Water content in grams per 100 grams, which affects the food's energy density.
Vitamin A (in mg)	Amount of Vitamin A in micrograms per 100 grams, important for vision and immune functioning
Vitamin B1 (Thiamine) (in mg)	Essential for glucose metabolism
Vitamin B11 (Folic Acid) (in mg)	Crucial for cell function and tissue growth, particularly important in pregnancy
Vitamin B12 (in mg)	Important for brain function and blood formation.
Vitamin B2 (Riboflavin) (in mg)	Necessary for energy production, cell function, and fat metabolism.
Vitamin B3 (Niacin) (in mg)	Supports digestive system, skin, and nerves health.

Column name	Description
Vitamin B5 (Pantothenic Acid) (in mg)	Necessary for making blood cells, and helps convert food into energy.
Vitamin B6 (in mg)	Important for normal brain development and keeping the nervous and immune systems healthy.
Vitamin C (in mg)	Important for the repair of all body tissues
Vitamin D (in mg)	Crucial for the absorption of calcium, promoting bone growth and health.
Vitamin E (in mg)	Acts as an antioxidant, helping to protect cells from the damage caused by free radicals.
Vitamin K (in mg)	Necessary for blood clotting and bone health.
Calcium (in mg)	Vital for building and maintaining strong bones and teeth.
Copper (in mg)	Helps with the formation of collagen, increases the absorption of iron and plays a role in energy production.
Iron (in mg)	Essential for the creation of red blood cells.
Magnesium (in mg)	Important for many processes in the body including regulation of muscle and nerve function, blood sugar levels, and blood pressure and making protein, bone, and DNA.
Manganese (in mg)	Involved in the formation of bones, blood clotting factors, and enzymes that play a role in fat and carbohydrate metabolism, calcium absorption, and blood sugar regulation.
Phosphorus (in mg)	Helps with the formation of bones and teeth and is necessary for the body to make protein for the growth, maintenance, and repair of cells and tissues.
Potassium (in mg)	Helps regulate fluid balance, muscle contractions, and nerve signals.
Selenium (in mg)	Important for reproduction, thyroid gland function, DNA production, and protecting the body from damage caused by free radicals and from infection.
Zinc (in mg)	Necessary for the immune system to properly function and plays a role in cell division, cell growth, wound healing, and the breakdown of carbohydrates.
Nutrition Density	A metric indicating the nutrient richness of the food per calorie

▼ Import necessary libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

Import Data

```
df1=pd.read_csv("FOOD-DATA-GROUP1.csv")
df2=pd.read_csv("FOOD-DATA-GROUP2.csv")
df3=pd.read_csv("FOOD-DATA-GROUP3.csv")
df4=pd.read_csv("FOOD-DATA-GROUP4.csv")
df5=pd.read_csv("FOOD-DATA-GROUP5.csv")

df=pd.concat([df1,df2,df3,df4,df5],ignore_index=True)
```

Pre-EDA(Pre-Exploratory Data Analysis)

```
print(f"Rows :[{df.shape[0]}],Columns :[{df.shape[1]}]")
df.columns
df.drop(columns=["Unnamed: 0","Unnamed: 0.1"],axis=1,inplace=T
df.head()
```

Rows :[2395],Columns :[37]

	food	Caloric Value	Fat	Saturated Fats	Monounsaturated Fats	Polyunsaturated Fats	Carbohydrates	Sugars	Protein	Dietary Fiber	...	Calcium	Copper	Iron	Magnesium	Manganese	Phosphorus	Potassium	Selenium
0	cream cheese	51	5.0	2.9	1.3	0.200	0.8	0.500	0.9	0.0	...	0.008	14.100	0.082	0.027	1.300	0.091	15.5	19.100
1	neufchatel cheese	215	19.4	10.9	4.9	0.800	3.1	2.700	7.8	0.0	...	99.500	0.034	0.100	8.500	0.088	117.300	129.2	0.054
2	requeijao cremoso light catupiry	49	3.6	2.3	0.9	0.000	0.9	3.400	0.8	0.1	...	0.000	0.000	0.000	0.000	0.000	0.000	0.0	0.000
3	ricotta cheese	30	2.0	1.3	0.5	0.002	1.5	0.091	1.5	0.0	...	0.097	41.200	0.097	0.096	4.000	0.024	30.8	43.800
4	cream cheese low fat	30	2.3	1.4	0.6	0.042	1.2	0.900	1.2	0.0	...	22.200	0.072	0.008	1.200	0.098	22.800	37.1	0.034

```
# Clean non unique / duplicates if any
df.shape[0] - df["food"].nunique()
df.duplicated().sum()
df[df.duplicated()]
df.drop_duplicates(inplace=True)
print(df.info())
df.describe().T
```

out:

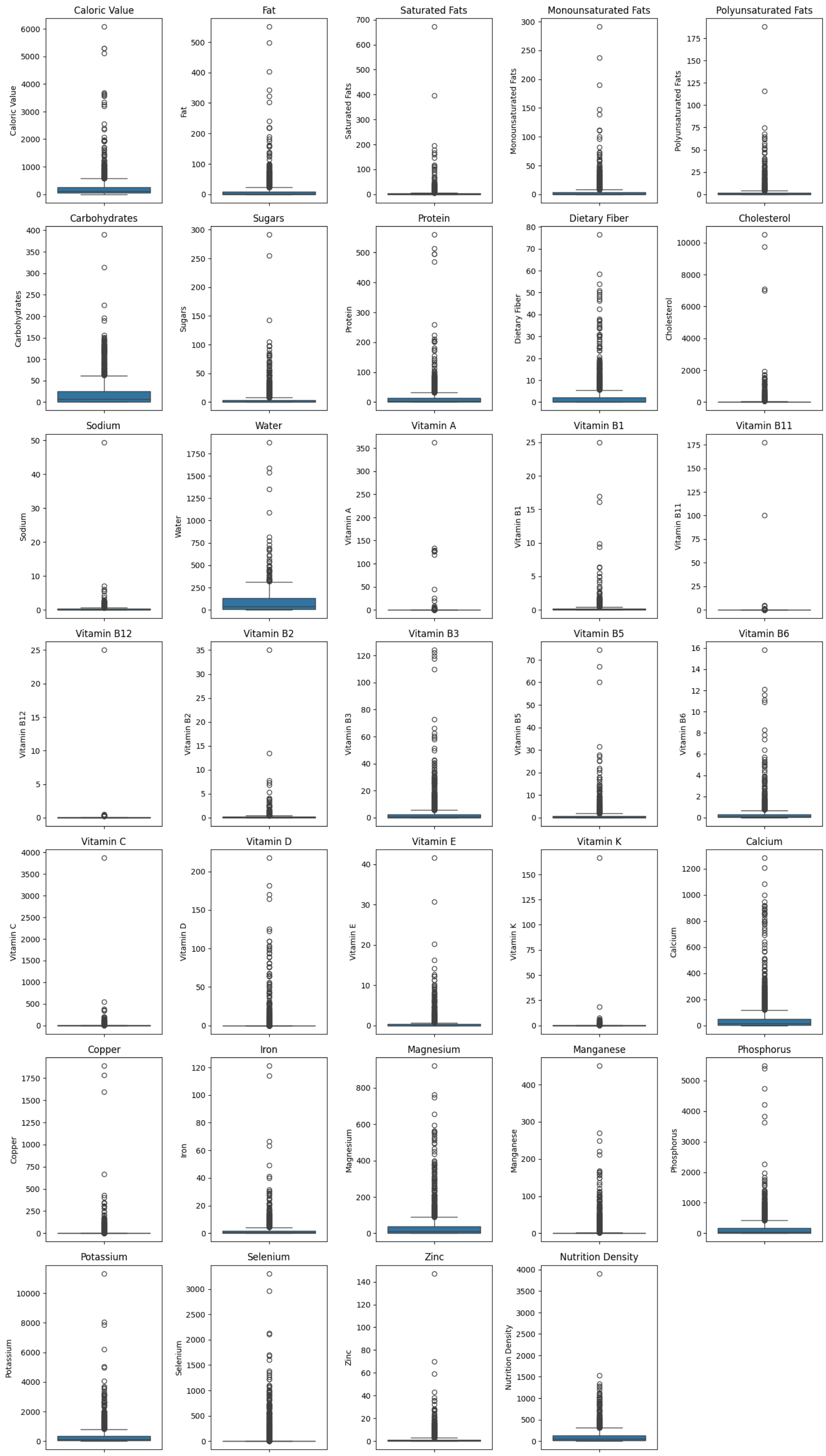
```
RangeIndex: 2395 entries, 0 to 2394
Data columns (total 35 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   food                  2395 non-null   object
 1   Caloric Value         2395 non-null   int64
 2   Fat                   2395 non-null   float64
 3   Saturated Fats        2395 non-null   float64
 4   Monounsaturated Fats  2395 non-null   float64
 5   Polyunsaturated Fats  2395 non-null   float64
 6   Carbohydrates         2395 non-null   float64
 7   Sugars                2395 non-null   float64
 8   Protein               2395 non-null   float64
 9   Dietary Fiber         2395 non-null   float64
 10  Cholesterol           2395 non-null   float64
 11  Sodium                2395 non-null   float64
 12  Water                 2395 non-null   float64
 13  Vitamin A             2395 non-null   float64
 14  Vitamin B1            2395 non-null   float64
 15  Vitamin B11           2395 non-null   float64
 16  Vitamin B12           2395 non-null   float64
 17  Vitamin B2            2395 non-null   float64
 18  Vitamin B3            2395 non-null   float64
 19  Vitamin B5            2395 non-null   float64
 20  Vitamin B6            2395 non-null   float64
 21  Vitamin C             2395 non-null   float64
 22  Vitamin D             2395 non-null   float64
 23  Vitamin E             2395 non-null   float64
 24  Vitamin K             2395 non-null   float64
 25  Calcium               2395 non-null   float64
 26  Copper                2395 non-null   float64
 27  Iron                  2395 non-null   float64
 28  Magnesium             2395 non-null   float64
 29  Manganese             2395 non-null   float64
 30  Phosphorus            2395 non-null   float64
 31  Potassium             2395 non-null   float64
 32  Selenium              2395 non-null   float64
 33  Zinc                  2395 non-null   float64
 34  Nutrition Density     2395 non-null   float64
```

Insight

According to my observation, all columns have outliers due to the significant difference between the mean, 50th percentile, 75th percentile, and maximum values, likely because some foods have very high values.

Exploratory Data Analysis (EDA)

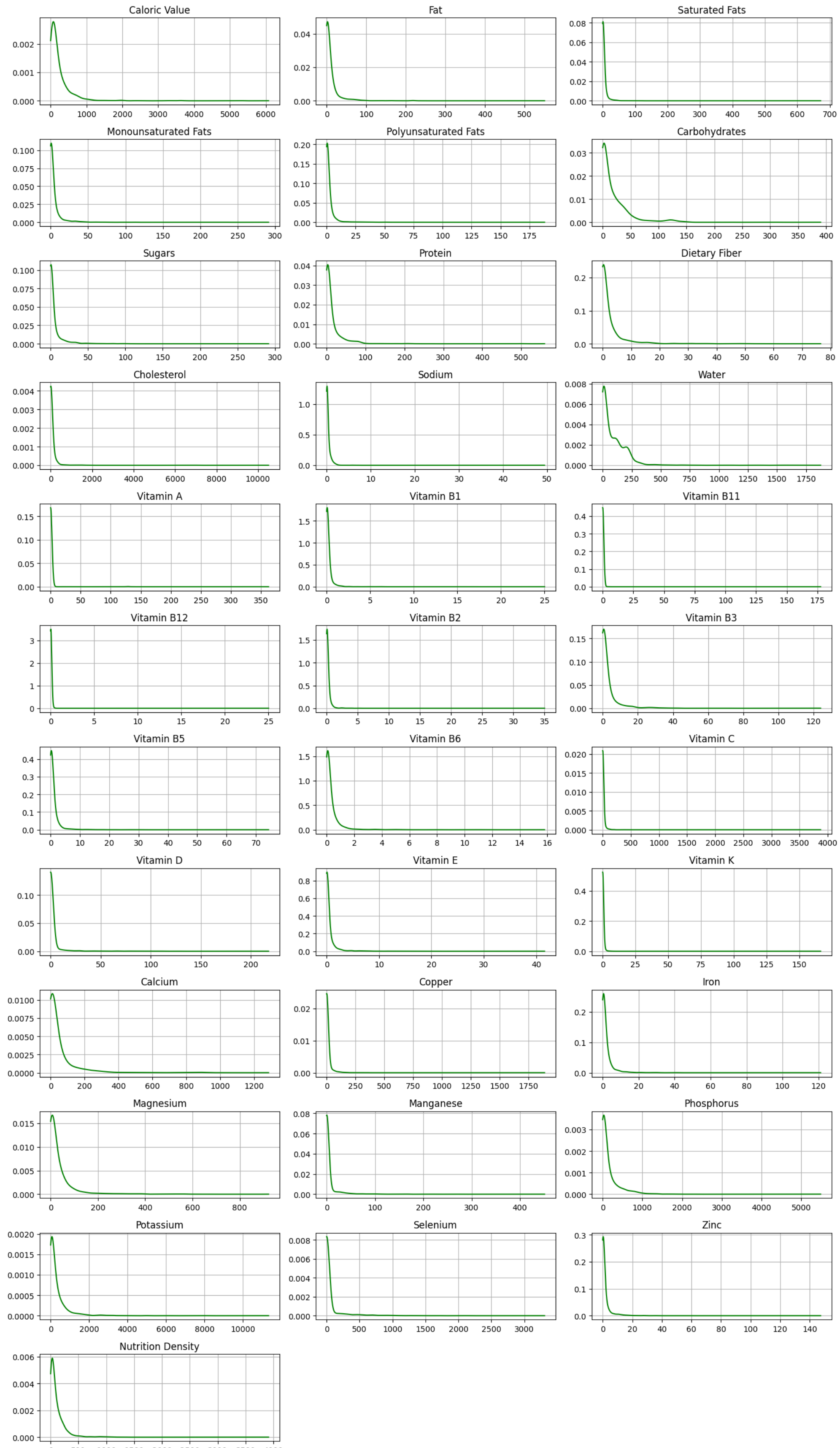
```
# extarct the COL names and remove food
columns_to_plot=df.columns.to_list()
columns_to_plot.remove("food")
# set plot figure size
plt.figure(figsize=(15,30))
for i,column in enumerate (columns_to_plot,1):
  # plot 8 rows with 5 figures each
  plt.subplot(8, 5 , i)
  plt.boxplot(df[column])
  plt.xlabel(column)
plt.show()
```


Caloric Value
Fat
Saturated Fats
Monounsaturated Fats
Polyunsaturated Fats
Carbohydrates
Sugars
Protein
Dietary Fiber
Cholesterol
Sodium
Water
Vitamin A
Vitamin B1
Vitamin B11
Vitamin B12
Vitamin B2
Vitamin B3
Vitamin B5
Vitamin B6
Vitamin C
Vitamin D
Vitamin E
Vitamin K
Calcium
Copper
Iron
Magnesium
Manganese
Phosphorus
Potassium
Selenium
Zinc
Nutrition Density

Skewed Distributions

```
plt.figure(figsize=(15, 30))
for i, column in enumerate(columns_to_plot, 1):
    plt.subplot(14, 3, i)
    data = df[column].dropna()
    density = gaussian_kde(data)
    x = np.linspace(data.min(), data.max(), 1000)
    y = density(x)
    plt.plot(x, y, color="g")
    plt.title(column)
    plt.grid(True)

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


Caloric Value
Fat
Saturated Fats
Monounsaturated Fats
Polyunsaturated Fats
Carbohydrates
Sugars
Protein
Dietary Fiber
Cholesterol
Sodium
Water
Vitamin A
Vitamin B1
Vitamin B11
Vitamin B12
Vitamin B2
Vitamin B3
Vitamin B5
Vitamin B6
Vitamin C
Vitamin D
Vitamin E
Vitamin K
Calcium
Copper
Iron
Magnesium
Manganese
Phosphorus
Potassium
Selenium
Zinc
Nutrition Density

Insights:

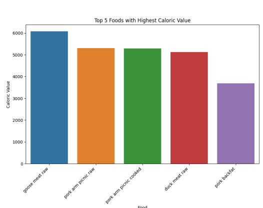
1. **Caloric Value**: The majority of food items have a low caloric value, with a long tail extending towards higher values, indicating that a few foods are very high in calories.
2. **Fat Content (Total, Saturated, Monounsaturated, Polyunsaturated)**: Most foods have low fat content, but there are outliers with significantly higher fat values. Saturated fats, monounsaturated fats, and polyunsaturated fats also show a similar distribution.
3. **Carbohydrates and Sugars**: These plots indicate that most foods have low to moderate carbohydrate content, with a few high carbohydrate foods. Sugars follow a similar distribution pattern.
4. **Protein**: The majority of foods have low to moderate protein content, with a few foods being very high in protein.
5. **Dietary Fiber**: Most foods have low fiber content, but there are some high-fiber foods.
6. **Cholesterol**: The distribution shows that most foods have very low cholesterol, but there are outliers with extremely high cholesterol content.
7. **Sodium**: Similar to cholesterol, most foods are low in sodium, with a few high sodium outliers.
8. **Water Content**: The majority of foods have a low to moderate water content, with a few foods containing very high water content.
9. **Vitamins (A, B1, B11, B12, B2, B3, B5, B6, C, D, E, K)**: Most vitamins are present in low quantities in the majority of foods, with few foods containing high amounts of specific vitamins. This is evident from the sharp peaks and long tails in the distributions.
10. **Minerals (Calcium, Copper, Iron, Magnesium, Manganese, Phosphorus, Potassium, Selenium, Zinc)**: Similar to vitamins, most minerals are present in low quantities in most foods, with a few high-value outliers.
11. **Nutrition Density**: This plot shows that the majority of foods have low to moderate nutrition density, with a few foods having very high nutrition density.

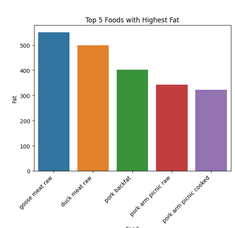
Key Insights:

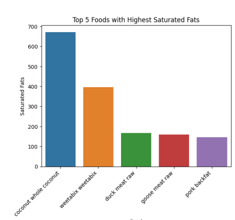
- **Presence of Outliers**: The KDE plots confirm the presence of outliers across all nutritional columns. These outliers represent foods with exceptionally high nutrient values.
- **Skewed Distributions**: Most distributions are right-skewed (long tail to the right), indicating that while most foods have low to moderate nutrient values, there are a few foods with extremely high values.
- **Diet Plan Implications**: The presence of outliers is important for diet planning, as they highlight foods that can significantly contribute to nutrient intake. Removing these outliers would result in losing valuable information about these nutrient-dense foods.

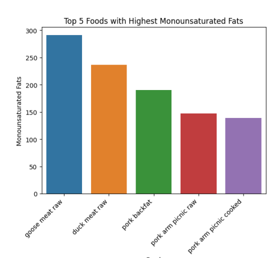
▼ Top 5 Foods with Highest Nutrient Content

```
plt.figure(figsize=(10, 6))
for i, column in enumerate(columns_to_plot, 1):
    temp_df = df.sort_values(by=column, ascending=False).hea
    display(temp_df)
    sns.barplot(data=temp_df, x="food", y=column)
    plt.xticks(rotation=45,ha='right',fontsize=10)
    plt.xlabel("Food", fontsize=10)
    plt.ylabel(column, fontsize=10)
    plt.title(f"Top 5 Foods with Highest {column}", fontsize
    plt.show()
```







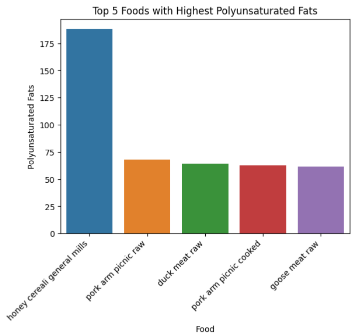


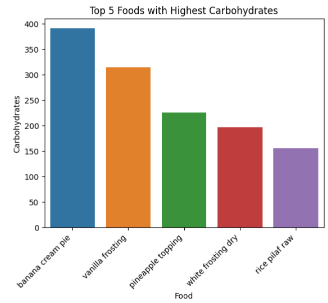
	food	Caloric Value
1437	goose meat raw	6077
1374	pork arm picnic raw	5298
1376	pork arm picnic cooked	5292
1430	duck meat raw	5123
1372	pork backfat	3683

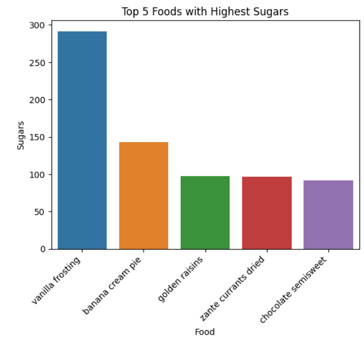
	food	Fat
1437	goose meat raw	550.7
1430	duck meat raw	498.8
1372	pork backfat	402.3
1374	pork arm picnic raw	343.4
1376	pork arm picnic cooked	322.7

	food	Saturated Fats
999	coconut whole coconut	672.0
781	weetabix weetabix	396.1
1430	duck meat raw	167.6
1437	goose meat raw	160.2
1372	pork backfat	146.1

	food	Monounsaturated Fats
1437	goose meat raw	291.1
1430	duck meat raw	237.0
1372	pork backfat	190.3
1374	pork arm picnic raw	147.3
1376	pork arm picnic cooked	139.0



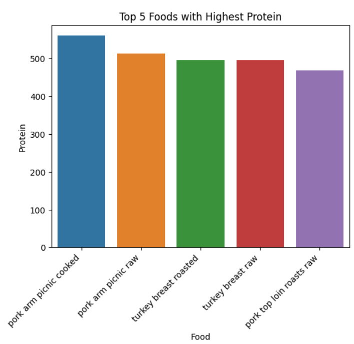


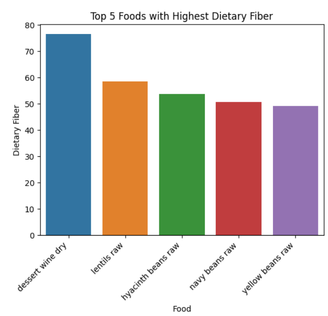


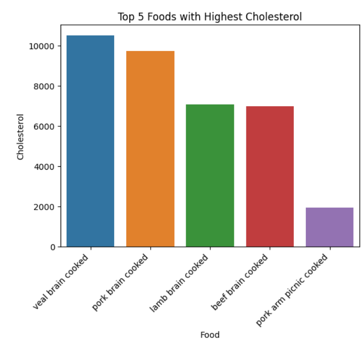
	food	Polyunsaturated Fats
815	honey cereali general mills	188.0
1374	pork arm picnic raw	67.9
1430	duck meat raw	64.4
1376	pork arm picnic cooked	62.7
1437	goose meat raw	61.6

	food	Carbohydrates
707	banana cream pie	390.2
622	vanilla frosting	313.7
625	pineapple topping	225.8
616	white frosting dry	196.4
848	rice pilaf raw	155.7

	food	Sugars
622	vanilla frosting	291.5
707	banana cream pie	143.0
980	golden raisins	97.7
982	zante currants dried	96.9
746	chocolate semisweet	91.6



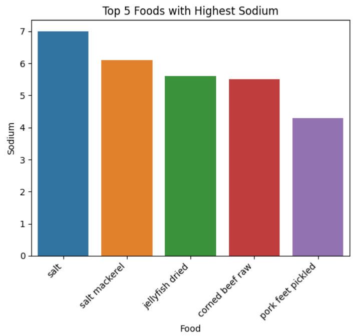


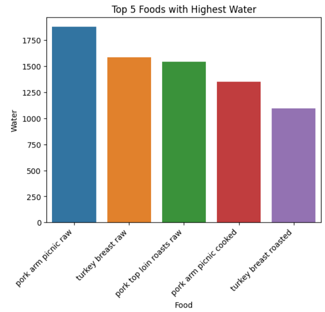


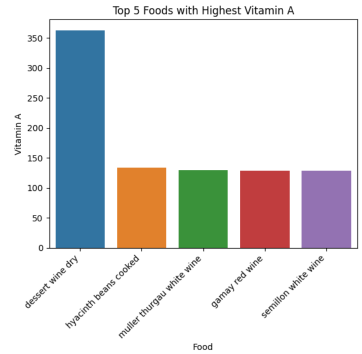
	food	Protein
1376	pork arm picnic cooked	560.3
1374	pork arm picnic raw	513.6
1438	turkey breast roasted	496.1
1406	turkey breast raw	495.6
1339	pork top loin roasts raw	468.4

	food	Dietary Fiber
594	dessert wine dry	76.5
1109	lentils raw	58.6
1099	hyacinth beans raw	53.8
1086	navy beans raw	50.8
1107	yellow beans raw	49.2

	food	Cholesterol
1294	veal brain cooked	10509.0
1320	pork brain cooked	9748.6
1323	lamb brain cooked	7089.2
1299	beef brain cooked	7002.5
1376	pork arm picnic cooked	1936.7



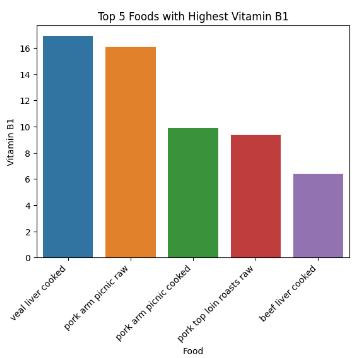


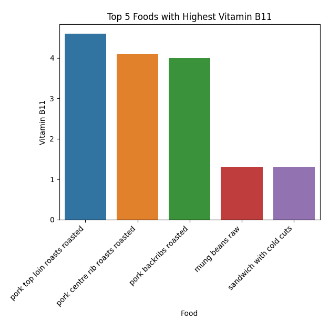


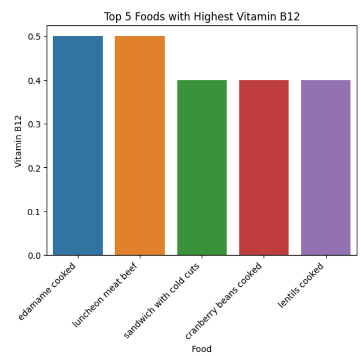
	food	Sodium
1047	salt	7.0
389	salt mackerel	6.1
370	jellyfish dried	5.6
1219	corned beef raw	5.5
1393	pork feet pickled	4.3

	food	Water
1374	pork arm picnic raw	1875.9
1406	turkey breast raw	1585.9
1339	pork top loin roasts raw	1539.6
1376	pork arm picnic cooked	1351.9
1438	turkey breast roasted	1092.4

	food	Vitamin A
594	dessert wine dry	362.7
1091	hyacinth beans cooked	134.1
575	muller thurgau white wine	129.2
570	gamay red wine	128.6
593	semillon white wine	128.1



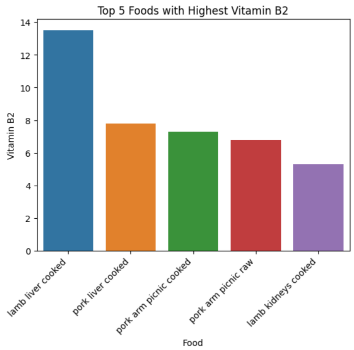




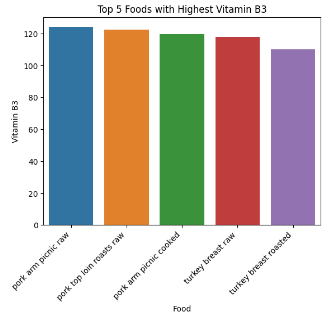
	food	Vitamin B1
1332	veal liver cooked	16.9
1374	pork arm picnic raw	16.1
1376	pork arm picnic cooked	9.9
1339	pork top loin roasts raw	9.4
1318	beef liver cooked	6.4

	food	Vitamin B11
1395	pork top loin roasts roasted	4.6
1385	pork centre rib roasts roasted	4.1
1389	pork backribs roasted	4.0
1103	mung beans raw	1.3
319	sandwich with cold cuts	1.3

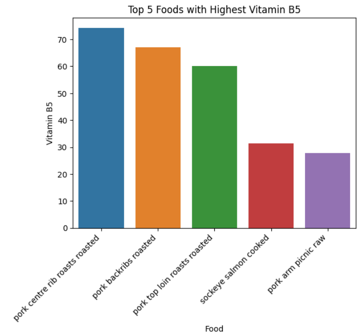
	food	Vitamin B12
1089	edamame cooked	0.5
1259	luncheon meat beef	0.5
319	sandwich with cold cuts	0.4
1110	cranberry beans cooked	0.4
1112	lentils cooked	0.4



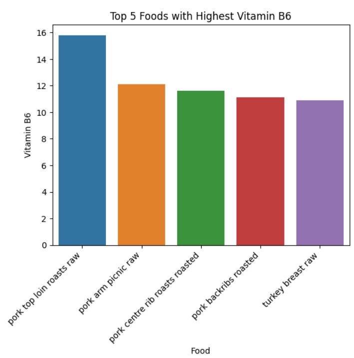
	food	Vitamin B2
1314	lamb liver cooked	13.5
1328	pork liver cooked	7.8
1376	pork arm picnic cooked	7.3
1374	pork arm picnic raw	6.8
1325	lamb kidneys cooked	5.3



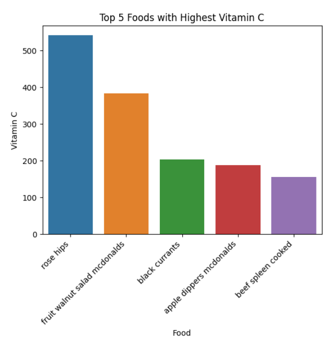
	food	Vitamin B3
1374	pork arm picnic raw	124.0
1339	pork top loin roasts raw	122.2
1376	pork arm picnic cooked	119.5
1406	turkey breast raw	117.7
1438	turkey breast roasted	110.0



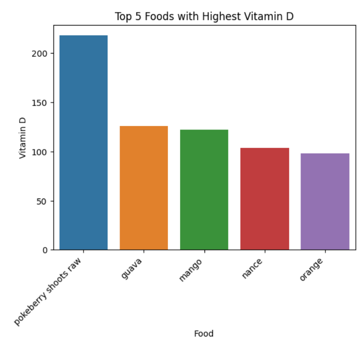
	food	Vitamin B5
1385	pork centre rib roasts roasted	74.3
1389	pork backribs roasted	67.0
1395	pork top loin roasts roasted	60.2
413	sockeye salmon cooked	31.4
1374	pork arm picnic raw	27.9



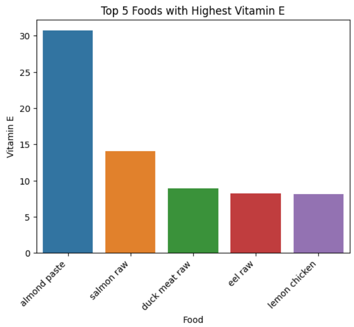
	food	Vitamin B6
1339	pork top loin roasts raw	15.8
1374	pork arm picnic raw	12.1
1385	pork centre rib roasts roasted	11.6
1389	pork backribs roasted	11.1
1406	turkey breast raw	10.9

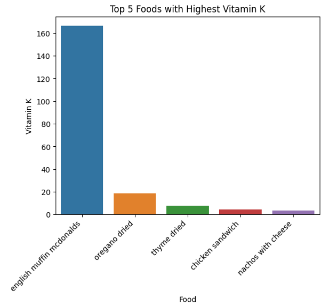


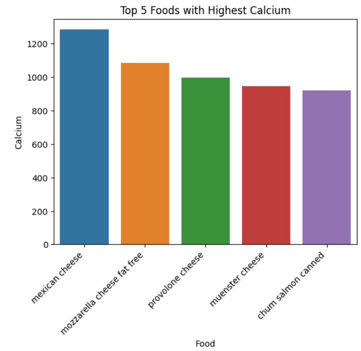
	food	Vitamin C
902	rose hips	541.0
910	fruit walnut salad mcdonalds	383.6
916	black currants	202.7
940	apple dippers mcdonalds	188.4
1306	beef spleen cooked	154.9



	food	Vitamin D
945	pokeberry shoots raw	217.6
923	guava	125.6
951	mango	122.3
998	nance	103.6
888	orange	97.9



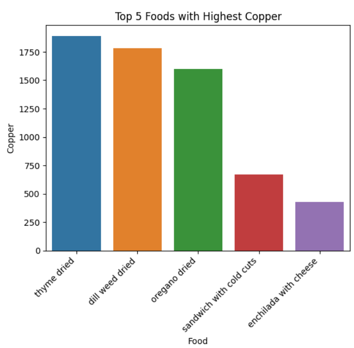


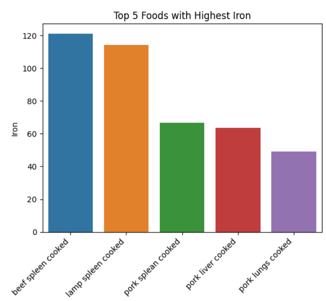


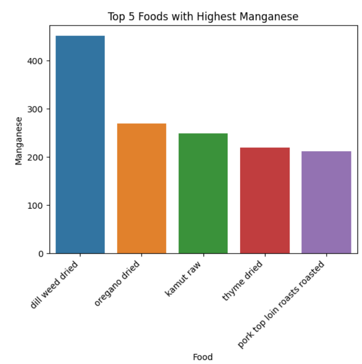
	food	Vitamin E
632	almond paste	30.7
417	salmon raw	14.1
1430	duck meat raw	8.9
451	eel raw	8.2
83	lemon chicken	8.1

	food	Vitamin K
262	english muffin mcdonalds	166.4
1056	oregano dried	18.3
1068	thyme dried	7.5
124	chicken sandwich	4.5
140	nachos with cheese	3.3

	food	Calcium
35	mexican cheese	1283.5
37	mozzarella cheese fat free	1085.9
38	provolone cheese	997.9
31	muenster cheese	946.4
502	chum salmon canned	918.8



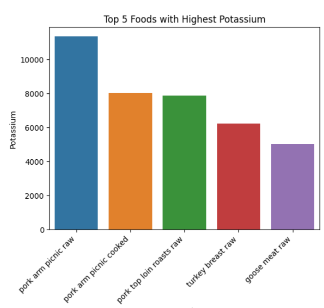


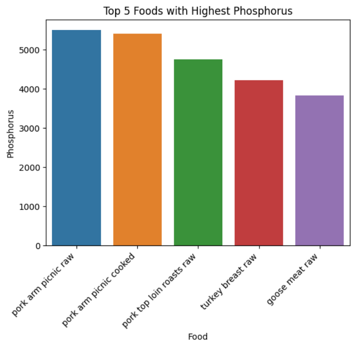


	food	Copper
1068	thyme dried	1890.0
1023	dill weed dried	1784.0
1056	oregano dried	1597.0
319	sandwich with cold cuts	668.6
71	enchilada with cheese	430.2

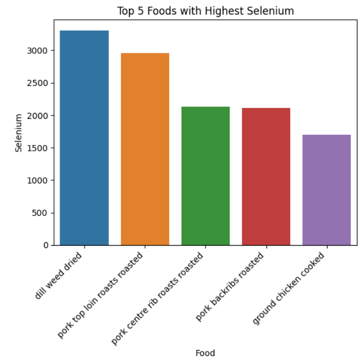
	food	Iron
1306	beef spleen cooked	121.2
1316	lamp spleen cooked	114.1
1358	pork splean cooked	66.5
1328	pork liver cooked	63.4
1333	pork lungs cooked	49.2

	food	Magnesium
775	rice bran	921.6
1126	mothbeans raw	746.8
774	cottonseed kernels roasted	655.6
1099	hyacinth beans raw	594.3
1138	yardlong beans raw	564.5

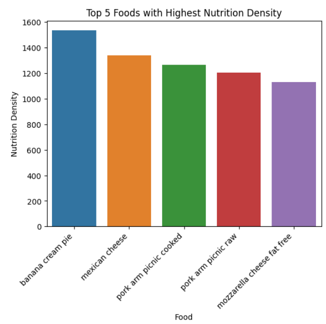




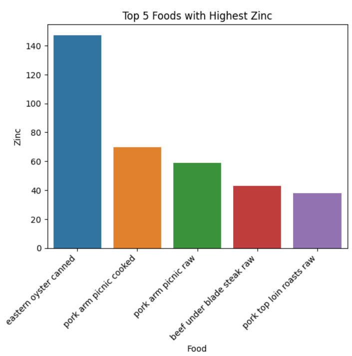
	food	Potassium
1374	pork arm picnic raw	11336.9
1376	pork arm picnic cooked	8039.6
1339	pork top loin roasts raw	7858.1
1406	turkey breast raw	6226.0
1437	goose meat raw	5045.0



	food	Phosphorus
1374	pork arm picnic raw	5490.0
1376	pork arm picnic cooked	5404.8
1339	pork top loin roasts raw	4741.2
1406	turkey breast raw	4211.0
1437	goose meat raw	3832.9



	food	Selenium
1023	dill weed dried	3308.0
1395	pork top loin roasts roasted	2959.5
1385	pork centre rib roasts roasted	2129.8
1389	pork backribs roasted	2107.2
1435	ground chicken cooked	1699.3



	food	Nutrition Density
707	banana cream pie	1533.500
35	mexican cheese	1337.000
1376	pork arm picnic cooked	1264.074
1374	pork arm picnic raw	1202.100
37	mozzarella cheese fat free	1128.200

	food	Zinc
501	eastern oyster canned	147.3
1376	pork arm picnic cooked	69.8
1374	pork arm picnic raw	59.0
1214	beef under blade steak raw	42.9
1339	pork top loin roasts raw	38.0

▼ Top 5 Foods with High Carbs and Low Sugar

```
high_carbs_low_sugars = df.sort_values(by='Carbohydrates', ascending=False).head(5)[["food", 'Carbohydrates', 'Sugars']]
high_carbs_low_sugars = high_carbs_low_sugars.sort_values(by='Sugars', ascending=True).head(5)[["food", 'Carbo
```

```
hydrates', 'Sugars']]
high_carbs_low_sugars
```

Out:

	food	Carbohydrates	Sugars
616	white frosting dry	196.4	0.0
1925	potato chips fat free	190.1	8.4
625	pineapple topping	225.8	71.4
707	banana cream pie	390.2	143.0
622	vanilla frosting	313.7	291.5

▼ Top 5 Foods with Low Carbs and High Sugar

```
high_sugars_low_carbs = df.sort_values(by='Sugars', asc
ending=False).head(5)[["food", 'Sugars', 'Carbohydrate
s']]
high_sugars_low_carbs = high_sugars_low_carbs.sort_valu
es(by='Carbohydrates', ascending=True).head(5)[["food",
'Sugars', 'Carbohydrates']]
high_sugars_low_carbs
```

Out:

	food	Sugars	Carbohydrates
1466	danone low fat alsafi	255.0	0.089
1968	cranberry sauce sweetened	105.0	107.800
2345	sweet potato canned	98.3	133.500
622	vanilla frosting	291.5	313.700
707	banana cream pie	143.0	390.200

Top 5 Foods with High Protein to fiber ratio

```
df['Protein_to_Fiber_Ratio'] = df['Protein'] / df['Dietary Fiber']
df['Protein_to_Fiber_Ratio'].replace([np.inf, -np.inf], np.nan, inplace=True)
df['Protein_to_Fiber_Ratio'].fillna(0, inplace=True)
highest_protein_to_fiber_ratio = df.sort_values(by='Protein_to_Fiber_Ratio', ascending=False).head(5)[['food', 'Protein', 'Dietary Fiber', 'Protein_to_Fiber_Ratio']]
highest_protein_to_fiber_ratio
```

Out:

	food	Protein	Dietary Fiber	Protein_to_Fiber_Ratio
496	sardines in tomato sauce canned	18.6	0.009	2066.666667
34	pimento cheese	31.0	0.100	310.000000
1626	studentenfutter alnatura	3.1	0.031	100.000000
1420	chicken drumstick fried	15.8	0.200	79.000000
1244	turkey pastrami	4.6	0.062	74.193548

Top 5 Foods with Low Cholesterol and High Sodium

```
high_sodium = df.sort_values(by='Sodium', ascending=False).head(5)[["food",'Sodium','Cholesterol']]
high_sodium_low_cholesterol = high_sodium.sort_values(by='Cholesterol', ascending=True)
high_sodium_low_cholesterol
```

Out:

	food	Sodium	Cholesterol
1943	adobo fresco	49.4	0.0
1047	salt	7.0	0.0
370	jellyfish dried	5.6	2.9
389	salt mackerel	6.1	129.2
1219	corned beef raw	5.5	244.9

Top 5 Foods with High Iron and High Calcium

```
high_iron = df.sort_values(by='Iron', ascending=False).head(5)[["food",'Iron','Calcium']]
high_iron_high_calcium= high_iron.sort_values(by='Calcium', ascending=False)
high_iron_high_calcium
```

Out:

	food	Iron	Calcium
1358	pork splean cooked	66.5	38.9
1316	lamp spleen cooked	114.1	38.4
1306	beef spleen cooked	121.2	37.0
1328	pork liver cooked	63.4	35.4
1333	pork lungs cooked	49.2	24.0

▼ Top 5 Foods with High Zinc and High Vitamin C

```
high_zinc = df.sort_values(by='Zinc', ascending=False).head(5)[["food",'Zinc','Vitamin C']]
high_zinc_high_vitamin_C= high_zinc.sort_values(by='Vitamin C', ascending=False)
high_zinc_high_vitamin_C
```

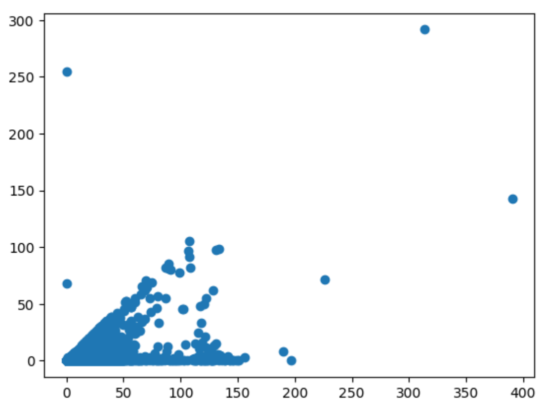
Out:

	food	Zinc	Vitamin C
501	eastern oyster canned	147.3	8.1
1376	pork arm picnic cooked	69.8	0.0
1374	pork arm picnic raw	59.0	0.0
1214	beef under blade steak raw	42.9	0.0
1339	pork top loin roasts raw	38.0	0.0

Correlation

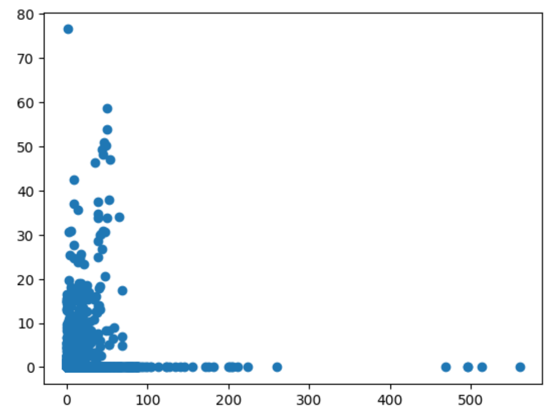
Between Carbohydrates and Sugar

```
df[['Protein', 'Sugar']].co
plt.scatter(x=df['Protein'],
```



Between Protein and Dietary Fiber

```
df[['Protein', 'Dietary Fib
plt.scatter(x=df['Protein'],
```



Insight

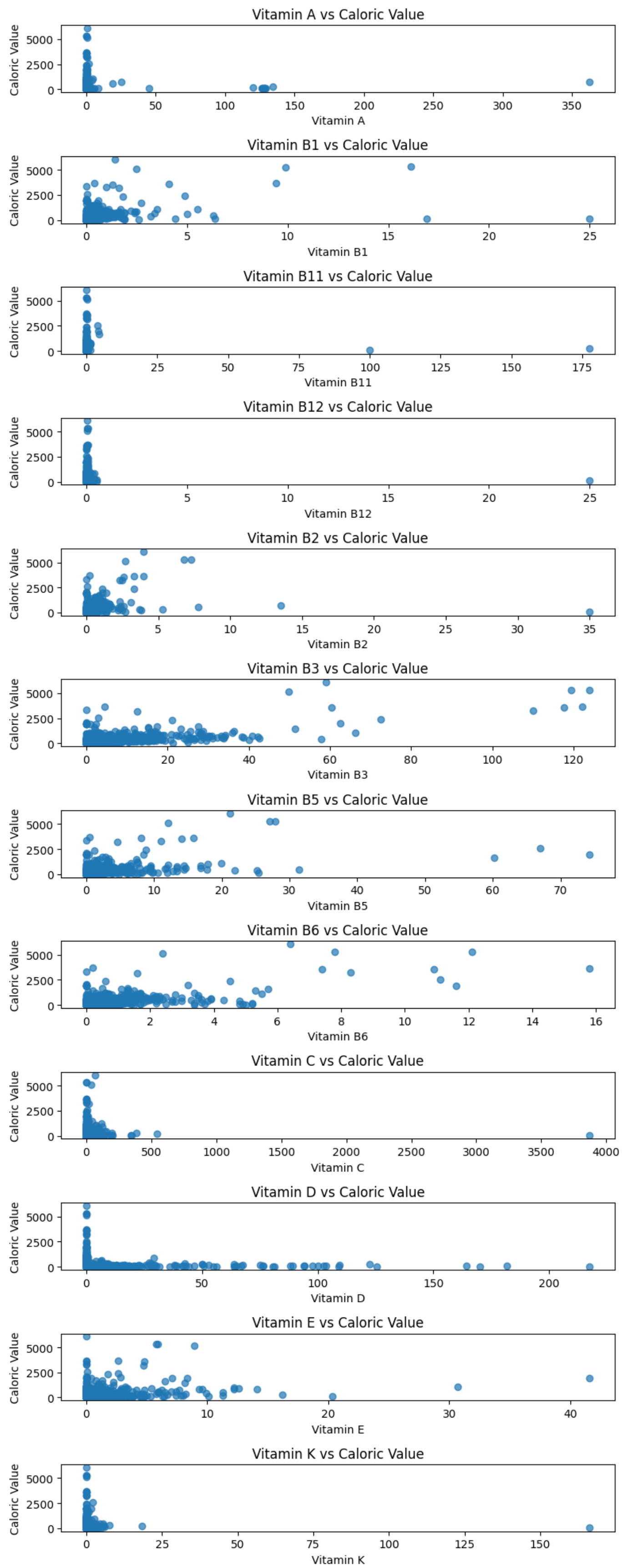
Protein and Dietary Fiber Relationship: The correlation coefficient between protein and dietary fiber is -0.012679, indicating a very weak negative linear relationship. Essentially, the amount of protein in foods is almost completely independent of their dietary fiber content.

Correlation between Protein and other Specifications

```
col=['Vitamin A', 'Vitamin B1', 'Vitamin B11', 'Vitamin B
       'Vitamin B3', 'Vitamin B5', 'Vitamin B6', 'Vitamin
       'Vitamin E', 'Vitamin K','Caloric Value']
correlation_matrix = df[col].corr()
fig, axes = plt.subplots(len(col) - 1, 1, figsize=(8, 20)
for i, feature in enumerate(col[:-1]):
    axes[i].scatter(df[feature], df['Caloric Value'], alp
    axes[i].set_title(f"{feature} vs Caloric Value")
```

```
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel('Caloric Value')

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


Vitamin A vs Caloric Value
Caloric Value
Vitamin A
Vitamin B1 vs Caloric Value
Caloric Value
Vitamin B1
Vitamin B11 vs Caloric Value
Caloric Value
Vitamin B11
Vitamin B12 vs Caloric Value
Caloric Value
Vitamin B12
Vitamin B2 vs Caloric Value
Caloric Value
Vitamin B2
Vitamin B3 vs Caloric Value
Caloric Value
Vitamin B3
Vitamin B5 vs Caloric Value
Caloric Value
Vitamin B5
Vitamin B6 vs Caloric Value
Caloric Value
Vitamin B6
Vitamin C vs Caloric Value
Caloric Value
Vitamin C
Vitamin D vs Caloric Value
Caloric Value
Vitamin D
Vitamin E vs Caloric Value
Caloric Value
Vitamin E
Vitamin K vs Caloric Value
Caloric Value
Vitamin K

Insight

1. **Strong Positive Correlations**:
    - **Vitamin B3 (Niacin)**: The correlation coefficient is 0.713, indicating a strong positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B3.
    - **Vitamin B6 (Pyridoxine)**: The correlation coefficient is 0.661, indicating a strong positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B6.
    - **Vitamin B2 (Riboflavin)**: The correlation coefficient is 0.541, indicating a moderate to strong positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B2.
    - **Vitamin B1 (Thiamine)**: The correlation coefficient is 0.498, indicating a moderate positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B1.
2. **Moderate Positive Correlations**:
    - **Vitamin B5 (Pantothenic Acid)**: The correlation coefficient is 0.486, indicating a moderate positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B5.
    - **Vitamin E**: The correlation coefficient is 0.309, indicating a moderate positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin E.
    - **Vitamin B11**: The correlation coefficient is 0.227, indicating a weak to moderate positive correlation with caloric value. This suggests that foods high in caloric value tend to also be high in Vitamin B11.
3. **Weak Positive Correlations**:
    - **Vitamin A**: The correlation coefficient is 0.003, indicating a very weak positive correlation with caloric value. This suggests that the caloric value is almost independent of Vitamin A content.

- **Vitamin B12 (Cobalamin)**: The correlation coefficient is 0.020, indicating a very weak positive correlation with caloric value. This suggests that the caloric value is almost independent of Vitamin B12 content.
- **Vitamin C**: The correlation coefficient is 0.050, indicating a very weak positive correlation with caloric value. This suggests that the caloric value is almost independent of Vitamin C content.

4. **Weak Negative Correlations**:
    - **Vitamin D**: The correlation coefficient is -0.053, indicating a very weak negative correlation with caloric value. This suggests that the caloric value is almost independent of Vitamin D content.
    - **Vitamin K**: The correlation coefficient is -0.009, indicating a very weak negative correlation with caloric value. This suggests that the caloric value is almost independent of Vitamin K content.

Summary:

- **Vitamins B3, B6, B2, and B1** show the strongest positive correlations with caloric value, indicating that foods rich in calories are likely to contain higher amounts of these vitamins.
- **Vitamins A, B12, C, D, and K** show very weak correlations (positive or negative) with caloric value, suggesting little to no linear relationship.
- **Vitamin E** and **Vitamin B5** also show moderate positive correlations, implying that caloric-rich foods might also have moderate amounts of these vitamins.

Conclusion

1. **Correlation Analysis**:
    - **Carbohydrates and Sugars**: Moderate positive correlation (0.5). Foods with high carbohydrates often have higher sugars.
    - **Protein and Dietary Fiber**: Very weak negative correlation (-0.0127). Protein and dietary fiber content are almost independent of each other.
    - **Vitamins and Caloric Value**:

    - Strong positive correlations with Vitamins B3, B6, B2, and B1.
    - Moderate positive correlations with Vitamins B5 and E.
    - Very weak correlations with Vitamins A, B12, C, D, and K.

2. **Top Foods Based on Nutrient Combinations**:
    - **High Carbs, Low Sugar**: White frosting dry, rice pilaf raw, pineapple topping, banana cream pie, vanilla frosting.
    - **Low Carbs, High Sugar**: Zante currants dried, chocolate semisweet, golden raisins, vanilla frosting, banana cream pie.
    - **High Protein to Fiber Ratio**: Sardines in tomato sauce canned, pimento cheese, chicken drumstick fried, turkey pastrami, English muffin with egg cheese sausage.
    - **High Cholesterol, Low Sodium**: Pork brain cooked, veal brain cooked, lamb brain cooked, beef brain cooked, pork arm picnic cooked.
    - **Low Cholesterol, High Sodium**: Salt, jellyfish dried, salt mackerel, corned beef raw, pork feet pickled.
    - **High Iron, High Calcium**: Pork spleen cooked, lamb spleen cooked, beef spleen cooked, pork liver cooked, pork lungs cooked.
    - **High Zinc, High Vitamin C**: Eastern oyster canned, pork arm picnic cooked, pork arm picnic raw, beef under blade steak raw, pork top loin roasts raw.

Food Recommendation ML

```
columns = [
    'food',
    'Carbohydrates',
    'Protein',
    'Fat',
    'Caloric Value',
    'Nutrition Density',
    'Vitamin A',
    'Vitamin B1',
    'Vitamin B6',
```

```
    'Vitamin B12',
    'Vitamin C',
    'Vitamin D',
    'Calcium',
    'Iron',
    'Magnesium',
    'Zinc'
]
```

Model 1 Target Audience [Gym Enthusiasts]

```
# Prepare X,y Cols
X = df.drop(columns=["food", "Protein"])
y = df["Protein"]  # best for muscle grow and repair

# train the model
X_train, Gym_Enthusiasts_test, y_train, y_test = train_test_
split(X, y, test_size=0.2, random_state=42)

Gym_Enthusiasts_model= LinearRegression()
Gym_Enthusiasts_model.fit(X_train, y_train)

y_pred = Gym_Enthusiasts_model.predict(Gym_Enthusiasts_test)

# train the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print(f"R^2 Score:{r2*100}%")
```

Model 2 Target Audience [Diet/Weight Loss]

```
# Prepare X,y Cols
X = df.drop(columns=["food", "Caloric Value"])
```

```
y = df["Caloric Value"]  # Minimise calories while keeping t
he food nutritionally rich.

# train the model
X_train, Diet_Weight_Loss_test, y_train, y_test = train_test
_split(X, y, test_size=0.2, random_state=42)

Diet_Weight_Loss_model= LinearRegression()
Diet_Weight_Loss_model.fit(X_train, y_train)

y_pred = Diet_Weight_Loss_model.predict(Diet_Weight_Loss_tes
t)

# train the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print(f"R^2 Score:{r2*100}%")
```

Model 3 Target Audience [Heart-Healthy Diet]

```
# Prepare X,y Cols
X = df.drop(columns=["food", "Saturated Fats"])
y = df["Saturated Fats"]  # Minimise saturated fats and chol
esterol

# train the model
X_train, Heart_Healthy_Diet_test, y_train, y_test = train_te
st_split(X, y, test_size=0.2, random_state=42)

Heart_Healthy_Diet_model= LinearRegression()
Heart_Healthy_Diet_model.fit(X_train, y_train)

y_pred = Heart_Healthy_Diet_model.predict(Heart_Healthy_Diet
_test)
```

```
# train the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print(f"R^2 Score:{r2*100}%")
```

Model 4 Target Audience [Balanced Nutrition]

```
# Prepare X,y Cols
X = df.drop(columns=["food", "Nutrition Density"])
y = df["Nutrition Density"]  #Optimise for high nutrient den
sity

# train the model
X_train, Balanced_Nutrition_test, y_train, y_test = train_te
st_split(X, y, test_size=0.2, random_state=42)

Balanced_Nutrition_model = LinearRegression()
Balanced_Nutrition_model.fit(X_train, y_train)

y_pred = Balanced_Nutrition_model.predict(Balanced_Nutrition
_test)

# train the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print("Mean Squared Error:", mse)
print(f"R^2 Score:{r2*100}%")
```

▼ Application Model

```
def predict_Model(df, cols_values, Target, model):
    """
    Predicts the output using the trained model and checks f
```

```
or invalid values.
    Returns the top 5 foods with the predicted target value
s.
    """
    # check the cols values if less then 0
    for i, value in enumerate(cols_values, 1):
        if value < 0:
            return value, i, -1, False
    if isinstance(cols_values, pd.Series):
        cols_values = cols_values.values.reshape(1, -1)
    elif isinstance(cols_values, np.ndarray) and cols_value
s.ndim == 1:
        cols_values = cols_values.reshape(1, -1)

    # Make the prediction
    y_pred = model.predict(cols_values)

    # Find top 5 foods with closest target values
    df['Target_diff'] = abs(df[Target] - y_pred[0])
    top_5_foods = df.sort_values(by='Target_diff').head(5)

    return None, None, None, top_5_foods[['food', Target]]
```

▼ Testing

```
# Example Usage
user_input1 = Gym_Enthusiasts_test.iloc[0]
value, value_index, errono, test1_predict = predict_Model
(df, user_input1, "Protein", Gym_Enthusiasts_model)
if test1_predict is False and errono == -1:
    print(f"Error in column index {value_index} where col
umn value {value} cannot be negative.")

user_test2  = Diet_Weight_Loss_test.iloc[0]
value, value_index, errono, test2_predict = predict_Model
(df, user_test2, "Caloric Value", Diet_Weight_Loss_model)
```

```
if test1_predict is False and errono == -1:
    print(f"Error in column index {value_index} where col
umn value {value} cannot be negative.")

user_test3  = Heart_Healthy_Diet_test.iloc[0]
value, value_index, errono, test3_predict = predict_Model
(df, user_test3, "Saturated Fats", Heart_Healthy_Diet_mod
el)
if test1_predict is False and errono == -1:
    print(f"Error in column index {value_index} where col
umn value {value} cannot be negative.")

user_test4  = Balanced_Nutrition_test.iloc[0]
value, value_index, errono, test4_predict = predict_Model
(df, user_test3, "Nutrition Density", Balanced_Nutrition_
model)
if test1_predict is False and errono == -1:
    print(f"Error in column index {value_index} where col
umn value {value} cannot be negative.")
```

▼ Results

```
test1_predict
```

	food	Protein
352	carp cooked	38.9
443	carp raw	38.9
1294	veal brain cooked	38.9
1599	pumpkin squash seeds dried	39.0
538	clams canned	38.8

```
test2_predict
```

	food	Caloric Value
2250	pupusas con frijoles	289
1771	donut with jelly filling	289
1906	russet potato baked	290
398	herring cooked	290
1364	pork top loin chops raw	287

```
test3_predict
```

	food	Saturated Fats
1978	poppyseed dressing	1.0
1293	ostrich top loin raw	1.0
1302	chicken gizzard cooked	1.0
1984	mushroom gravy	1.0
1359	chopped ham	1.0

```
test4_predict
```

	food	Nutrition Density
301	premium crispy chicken ranch blt sandwich mcdo...	219.100
20	brick cheese	219.002
812	rye	219.000
2185	beet greens cooked	218.900
1598	peanuts cooked	218.800