Big Data Analytics Mini - Project

Nutrient Analysis

Dataset: food.csv

1. Introduction

This project aims to analyze the nutritional composition of various food items using PySpark, focusing on understanding energy content, macronutrients, and correlations among vitamins and minerals. The dataset contains multiple food components and nutrient values that can reveal trends in food composition and health impacts.

2. Dataset Overview

The dataset used for this project is named 'food.csv'. It contains information such as **food category, description,** and **nutritional values like energy, protein, fat, carbohydrates, and various vitamins.** The data is processed using PySpark to handle large-scale analysis efficiently.

3. Data Preprocessing

Data preprocessing involved loading the CSV file into a PySpark DataFrame, identifying nutrient columns, and handling missing or malformed values using safe type casting. Rows with all null nutritional values were dropped.

MapReduce techniques were used conceptually by applying PySpark transformations (map, filter, reduceByKey) to extract and process nutrient-level data efficiently.

4. Analysis Performed

The following analyses were performed on the dataset:

- Descriptive statistics for key nutrients (Energy, Protein, Fat, Carbohydrates).
- Category-wise nutrient comparison.
- Correlation between nutrients (e.g., Energy vs. Protein).
- Identification of foods with the highest and lowest nutrient values.

5. Visualizations

This project provides the visual representation of the top particularly selected composition of nutrients in the food by the various visual analysis models

1: Energy Distribution across Food Categories

2: Top 10 High Protein Food Items

3: Correlation Heatmap of Nutrients

4: Carbohydrates vs. Energy Relationship

6. Workforce Characteristics

The project was developed through collaborative efforts involving multiple roles:

1. **Data Engineer:** Managed Spark setup, data loading, and cleaning using PySpark (try_cast, null handling, schema correction).

```
import findspark
findspark.init()

import pyspark
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("JupyterPySparkTest").getOrCreate()

print("✅ PySpark is working in Jupyter!")
print("PySpark version:", pyspark.__version__)
print("Spark version:", spark.version)
```

```
✅ PySpark is working in Jupyter!
PySpark version: 4.0.1
Spark version: 4.0.1
```

[2]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import expr, col
from pyspark.sql import functions as F
from pyspark.ml.feature import VectorAssembler, PCA
from pyspark.ml.clustering import KMeans
from pyspark.ml.stat import Correlation
from pyspark.sql.functions import monotonically_increasing_id
import matplotlib.pyplot as plt
```

[3]:
```
# 1. Setup Spark session and load data
spark = SparkSession.builder.appName("FoodAnalysis").getOrCreate()

df = spark.read.csv("food.csv", header=True, inferSchema=True)
df.printSchema()
df

# Clean column names for safe querying
def clean_column(name):
    return name.strip().replace(" ", "_").replace(".", "_").replace("-", "_")

df = df.toDF(*[clean_column(c) for c in df.columns])
```

```
root
 |-- Category: string (nullable = true)
 |-- Description: string (nullable = true)
 |-- Nutrient Data Bank Number: string (nullable = true)
 |-- Data.Alpha Carotene: string (nullable = true)
 |-- Data.Beta Carotene: string (nullable = true)
 |-- Data.Beta Cryptoxanthin: integer (nullable = true)
 |-- Data.Carbohydrate: double (nullable = true)
 |-- Data.Cholesterol: double (nullable = true)
 |-- Data.Choline: double (nullable = true)
 |-- Data.Fiber: double (nullable = true)
 |-- Data.Lutein and Zeaxanthin: double (nullable = true)
 |-- Data.Lycopene: double (nullable = true)
 |-- Data.Niacin: double (nullable = true)
 |-- Data.Protein: double (nullable = true)
 |-- Data.Retinol: double (nullable = true)
 |-- Data.Riboflavin: double (nullable = true)
 |-- Data.Selenium: double (nullable = true)
 |-- Data.Sugar Total: double (nullable = true)
 |-- Data.Thiamin: double (nullable = true)
 |-- Data.Water: double (nullable = true)
 |-- Data.Fat.Monosaturated Fat: double (nullable = true)
 |-- Data.Fat.Polysaturated Fat: double (nullable = true)
 |-- Data.Fat.Saturated Fat: double (nullable = true)
 |-- Data.Fat.Total Lipid: double (nullable = true)
 |-- Data.Major Minerals.Calcium: double (nullable = true)
 |-- Data.Major Minerals.Copper: double (nullable = true)
 |-- Data.Major Minerals.Iron: double (nullable = true)
 |-- Data.Major Minerals.Magnesium: double (nullable = true)
 |-- Data.Major Minerals.Phosphorus: double (nullable = true)
 |-- Data.Major Minerals.Potassium: double (nullable = true)
 |-- Data.Major Minerals.Sodium: integer (nullable = true)
 |-- Data.Major Minerals.Zinc: double (nullable = true)
 |-- Data.Vitamins.Vitamin A - RAE: double (nullable = true)
 |-- Data.Vitamins.Vitamin B12: double (nullable = true)
 |-- Data.Vitamins.Vitamin B6: double (nullable = true)
 |-- Data.Vitamins.Vitamin C: double (nullable = true)
```

[5]:
```python
# 2. Data cleaning

# Identify nutrient columns by prefix "Data_"
nutrient_cols = [c for c in df.columns if c.startswith("Data_")]

# Safely cast nutrient columns to float
for c in nutrient_cols:
    df = df.withColumn(c, expr(f"try_cast({c} as float)"))

# Drop rows where all nutrient columns are null
df = df.dropna(how="all", subset=nutrient_cols)

# Show dataset after cleaning
df.show()

# Count rows and columns
row_count = df.count()
col_count = len(df.columns)
print(f"Row count: {row_count}")
print(f"Column count: {col_count}")

# Drop rows with any nulls (strict clean data)
df_clean = df.dropna()
row_count_clean = df_clean.count()
print(f"Row count after removing all nulls: {row_count_clean}")

# Drop rows with nulls in vitamin columns only
vitamin_cols = [c for c in df.columns if "Vitamin" in c]
df_vitamin_clean = df.dropna(subset=vitamin_cols)
row_count_vitamin_clean = df_vitamin_clean.count()
print(f"Row count after removing nulls in vitamin columns: {row_count_vitamin_clean}")
```

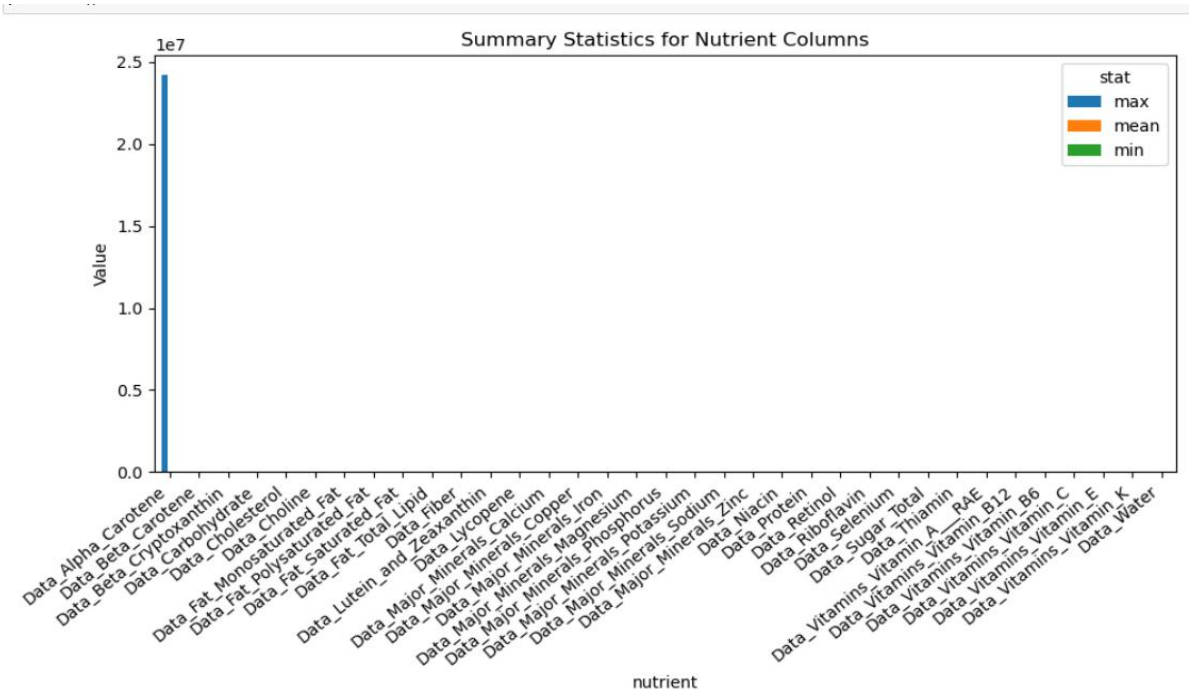
```
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
------+----------------------+----------------------+
|  Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Cryptoxant
a_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Pro
a_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_
rated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Maj
```

```
|Buttermilk|   Buttermilk, whole|                    1111530
4.88|              11.0|          14.6|         0.0|
0.172|            3.7|              4.88|         0.047|      87
1.899|                    3.31|                          115.0|
10.0|                            85.0|
47.0|                         0.46|                          0.036|
0.3|
+----------+--------------------+-------------------------
--------------+----------------+------------+----------+-
---------+----------------+-------------+-----------------+
---------+------------------------+--------------------+---
--------+-------------------------------+-----------------
-+-------------------------+-------------------------------+
------+-------------------------+-------------------------+
only showing top 20 rows

Row count: 7083
Column count: 38
Row count after removing all nulls: 7078
Row count after removing nulls in vitamin columns: 7083
```

2. **Data Analyst:** Performed statistical analysis, trend identification, and summary generation.

```python
# 3. Summary statistics & visualization

# Compute summary stats (mean, min, max) for all nutrient columns on cleaned df
summary_df = df_clean.agg(
    *[F.mean(c).alias(f"{c}_mean") for c in nutrient_cols],
    *[F.min(c).alias(f"{c}_min") for c in nutrient_cols],
    *[F.max(c).alias(f"{c}_max") for c in nutrient_cols]
)

# Convert summary to Pandas for plotting
summary_pd = summary_df.toPandas().T
summary_pd.columns = ['value']
summary_pd = summary_pd.reset_index()
summary_pd.rename(columns={'index': 'metric'}, inplace=True)
summary_pd['stat'] = summary_pd['metric'].str.split('_').str[-1]
summary_pd['nutrient'] = summary_pd['metric'].apply(lambda x: "_".join(x.split("_")[:-1]))
```

```python
# Pivot dataframe for plotting
pivot = summary_pd.pivot(index='nutrient', columns='stat', values='value')

# Plot bar chart for nutrient summary statistics
pivot.plot(kind='bar', figsize=(10, 6))
plt.ylabel('Value')
plt.title('Summary Statistics for Nutrient Columns')
plt.xticks(rotation=40, ha='right')
plt.tight_layout()
plt.show()
```

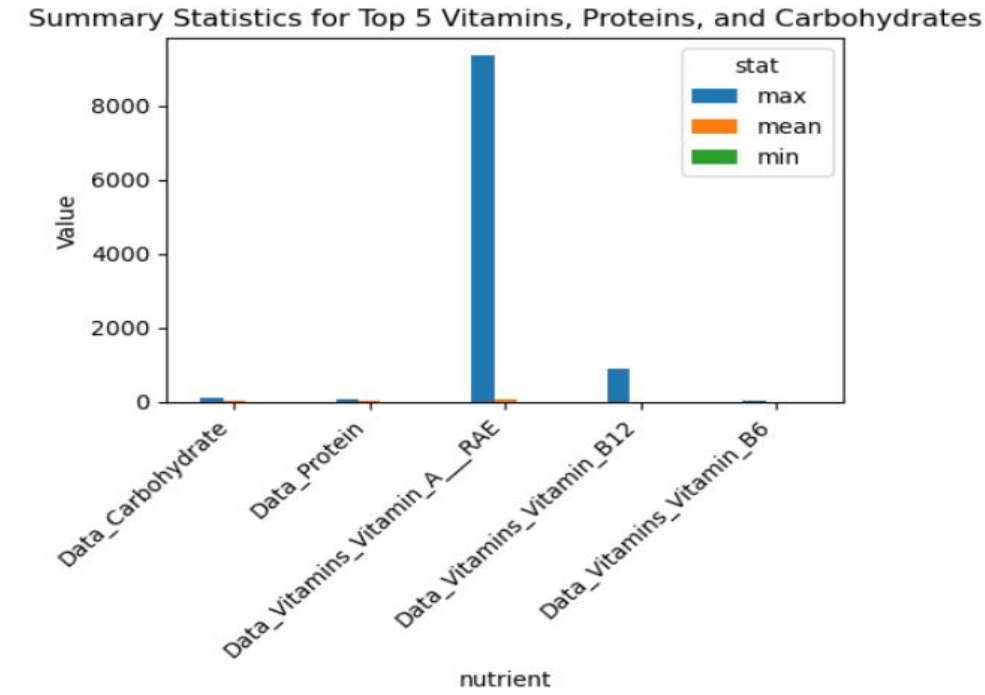



```
# Select top 5 relevant nutrients (Vitamin, Protein, Carbohydrate)
selected_cols = [
    c for c in df.columns
    if any(x in c for x in ["Vitamin", "Protein", "Carbohydrate"])
][:5]

# Summary stats for selected columns
summary_selected = df.agg(
    *[F.mean(c).alias(f"{c}_mean") for c in selected_cols],
    *[F.min(c).alias(f"{c}_min") for c in selected_cols],
    *[F.max(c).alias(f"{c}_max") for c in selected_cols]
)
summary_selected.show()
```

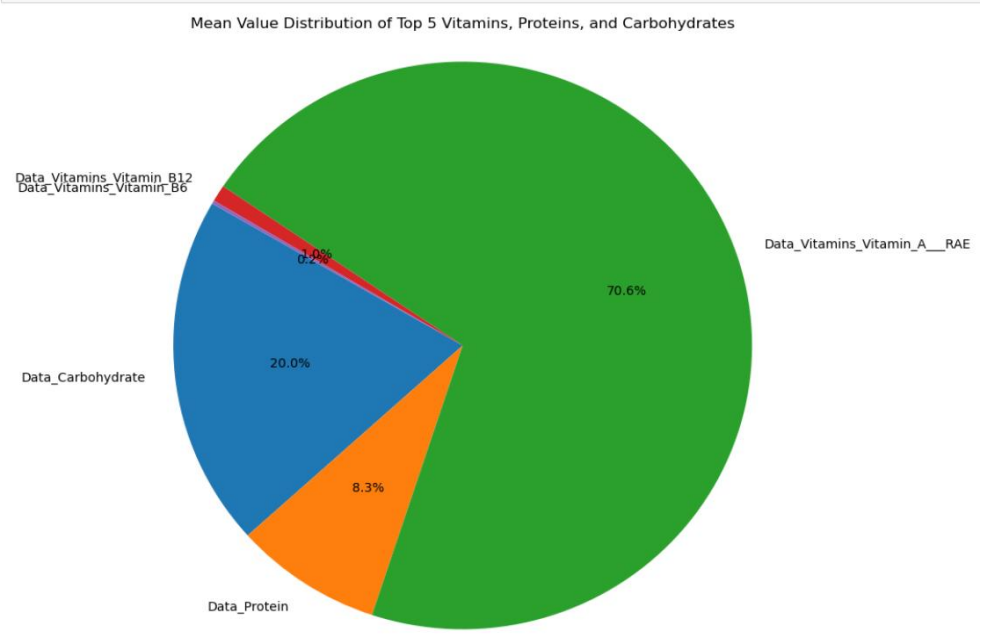
```
+----------------------+-----------------+--------------------------------+-----------------------------+-------
----------+--------------------+---------------+--------------------------------+-----------------------------+-
--------------+--------------------+---------------+--------------------------------+-----------------------------
--------------------+
|Data_Carbohydrate_mean|Data_Protein_mean|Data_Vitamins_Vitamin_A___RAE_mean|Data_Vitamins_Vitamin_B12_mean|Data_Vi
in_B6_mean|Data_Carbohydrate_min|Data_Protein_min|Data_Vitamins_Vitamin_A___RAE_min|Data_Vitamins_Vitamin_B12_min|D
_Vitamin_B6_min|Data_Carbohydrate_max|Data_Protein_max|Data_Vitamins_Vitamin_A___RAE_max|Data_Vitamins_Vitamin_B12_
amins_Vitamin_B6_max|
+----------------------+-----------------+--------------------------------+-----------------------------+-------
----------+--------------------+---------------+--------------------------------+-----------------------------+-
--------------+--------------------+---------------+--------------------------------+-----------------------------
--------------------+
|    20.812247625693203|8.587563600963907|               73.44203162501293|          0.9886954669677196|
4495272865|                 0.0|            0.0|                             0.0|                          0.0|
0.0|               100.0|          78.13|                          9363.0|                        879.0|
14.0|
+----------------------+-----------------+--------------------------------+-----------------------------+-------
----------+--------------------+---------------+--------------------------------+-----------------------------+-
--------------+--------------------+---------------+--------------------------------+-----------------------------
--------------------+
```

[9]:
```
# Convert to Pandas for plotting
summary_pd = summary_selected.toPandas().T
summary_pd.columns = ['value']
summary_pd = summary_pd.reset_index()
summary_pd.rename(columns={'index': 'metric'}, inplace=True)
summary_pd['stat'] = summary_pd['metric'].str.split('_').str[-1]
summary_pd['nutrient'] = summary_pd['metric'].apply(lambda x: "_".join(x.split("_")[:-1]))
pivot = summary_pd.pivot(index='nutrient', columns='stat', values='value')
```

10]:
```
# Plot bar chart
pivot.plot(kind='bar', figsize=(5, 5))
plt.ylabel('Value')
plt.title('Summary Statistics for Top 5 Vitamins, Proteins, and Carbohydrates')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



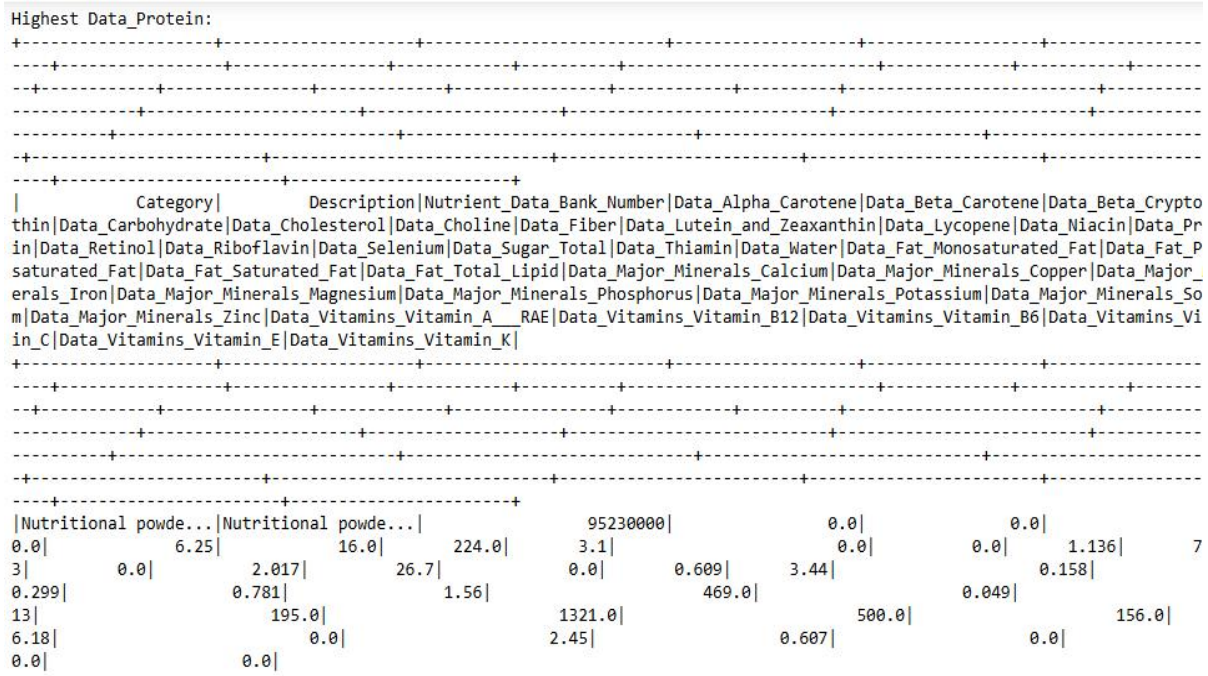
[11]:
```
# Pie chart of mean distribution for selected nutrients
mean_values = pivot['mean']
plt.figure(figsize=(9, 9))
plt.pie(mean_values, labels=mean_values.index, autopct='%1.1f%%', startangle=150)
plt.title('Mean Value Distribution of Top 5 Vitamins, Proteins, and Carbohydrates')
plt.axis('equal')
plt.show()
```




```
: # 4. Highest and lowest values for selected nutrients
for col in selected_cols:
    print(f"\nHighest {col}:")
    df_clean.orderBy(df_clean[col].desc()).limit(1).show()
    print(f"Lowest {col}:")
    df_clean.orderBy(df_clean[col].asc()).limit(1).show()
```

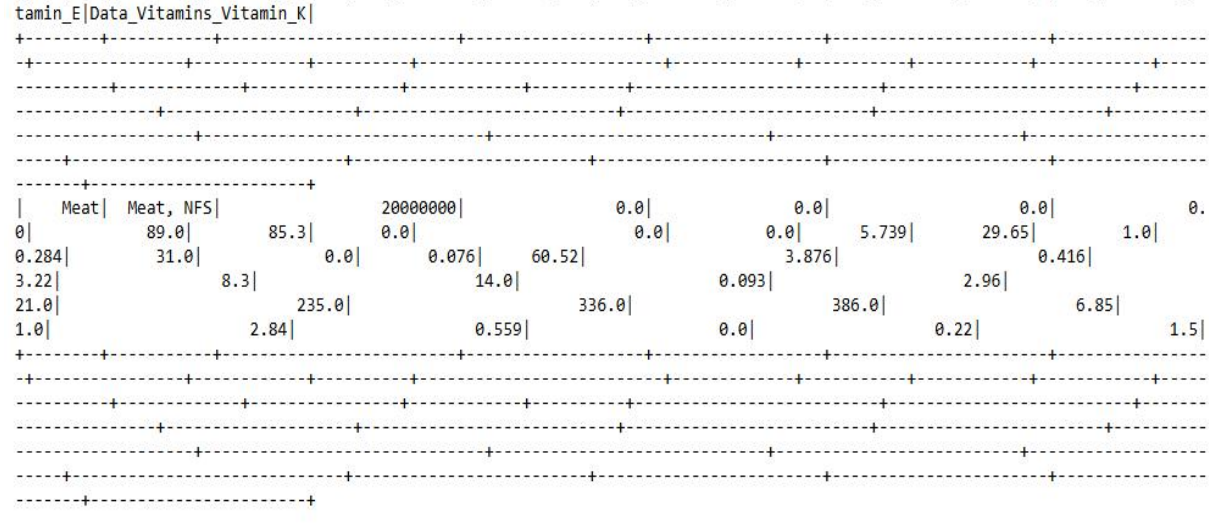
```
Highest Data_Carbohydrate:
+----------------+--------------------+-------------------------+-------------------+---------------
+-----------------+----------------+------------+----------+-------------------------+-------------
-----------+---------------+-------------+----------------+------------+----------+-----------------
---------+----------------------+--------------------+---------------------------+-----------------
------+-------------------------------+-------------------------------+-----------------------------+--
----------------------+-----------------------------+-------------------------+---------------------
+----------------------+----------------------+
|        Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carot
|Data_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene
ata_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosatu
rated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Mineral
s_Iron|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Da
ta_Major_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin
|Data_Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+----------------+--------------------+-------------------------+-------------------+---------------
+-----------------+----------------+------------+----------+-------------------------+-------------
-----------+---------------+-------------+----------------+------------+----------+-----------------
---------+----------------------+--------------------+---------------------------+-----------------
------+-------------------------------+-------------------------------+-----------------------------+--
----------------------+-----------------------------+-------------------------+---------------------
+----------------------+----------------------+
|Sugar substitute|Sugar substitute,...|                91108000|               0.0|
|           100.0|             0.0|        0.0|      0.0|                      0.0|          0.0
0.0|            0.0|          0.0|            0.0|        0.0|      0.0|                       0.
0|                0.0|                0.0|                      0.0|                       0.0
0.0|                       0.0|                      0.0|                     0.0|
0.0|                   0.0|                   0.0|                   0.0|                    0
+----------------+--------------------+-------------------------+-------------------+---------------
+-----------------+----------------+------------+----------+-------------------------+-------------
          .               .            .          .                .          .
```

Highest Data_Protein:



```
|           Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Crypto
thin|Data_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Pr
in|Data_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_P
saturated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_
erals_Iron|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_So
m|Data_Major_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vi
in_C|Data_Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+--------------------+--------------------+-----------------------+-------------------+------------------+----------------
----+----------------+----------------+------------+----------+-------------------------+-------------+----------+-------
--+------------+--------------+-------------+---------------+------------+----------+-------------------------+----------
------------+----------------------+--------------------+-----------------------------+-------------------------+-----------
----------+-----------------------------+-----------------------------+---------------------------+-------------------------+----------
-+-------------------------+---------------------------+-------------------------+-----------------------+----------------
----+-----------------------+-----------------------+
|Nutritional powde...|Nutritional powde...|               95230000|                0.0|               0.0|
0.0|             6.25|           16.0|      224.0|       3.1|                        0.0|          0.0|     1.136|       7
3|         0.0|          2.017|         26.7|             0.0|       0.609|      3.44|                    0.158|
0.299|                0.781|                1.56|                    469.0|                    0.049|
13|                        195.0|                      1321.0|                     500.0|                     156.0|
6.18|                         0.0|                     2.45|                   0.607|                     0.0|
0.0|                   0.0|
```

Lowest Data_Carbohydrate:

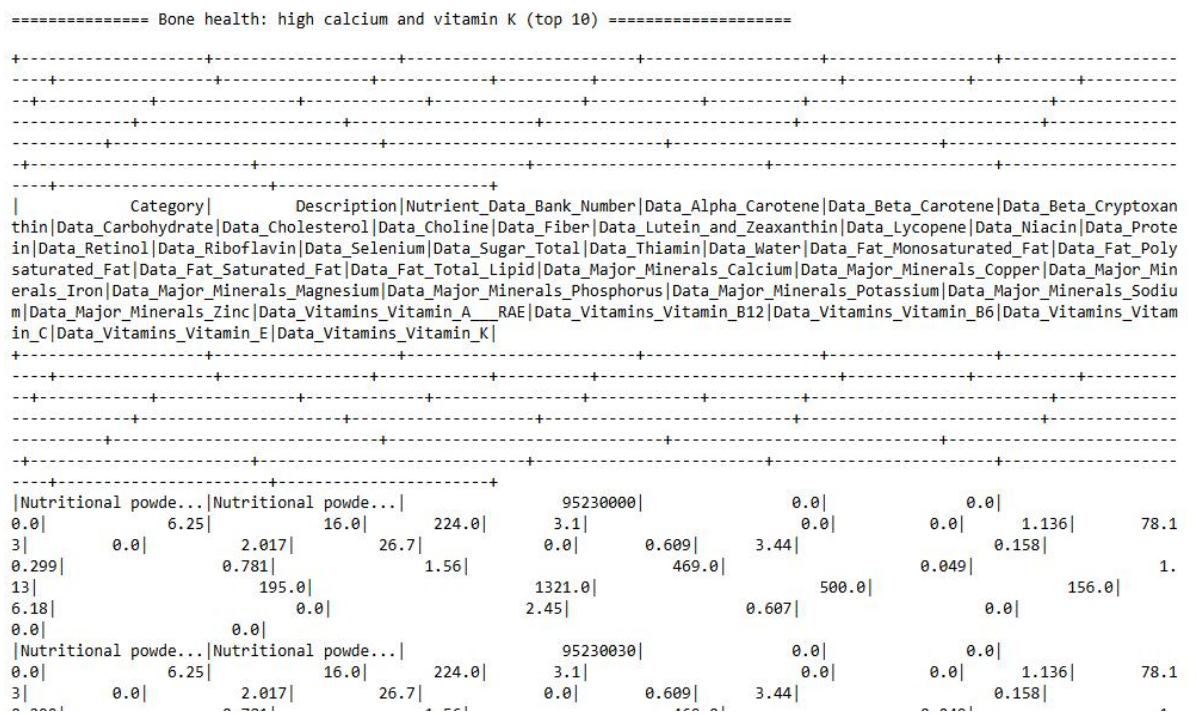


```
|Category|Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Cryptoxanthin|Data_Carbohydrat
e|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Protein|Data_Retinol|Data_
Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_Polysaturated_Fat|Data_Fa
t_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_Minerals_Iron|Data_Major
_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_Sodium|Data_Major_Minerals
_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vitamin_C|Data_Vitamins_Vi
tamin_E|Data_Vitamins_Vitamin_K|
+--------+----------+-------------------------+-------------------+------------------+-----------------------+----------------
-+----------------+------------+----------+-------------------------+-------------+----------+------------+-----------+-----
----------+-------------+---------------+------------+----------+-------------------------+-------------------------+-------
--------------+--------------------+-----------------------+-------------------------+-----------------------+----------
------------------+-----------------------------+-----------------------------+---------------------------+-------------------------+----------
-----+-----------------------------+-------------------------+-----------------------+-----------------------+----------------
------+-----------------------+
|    Meat|  Meat, NFS|                20000000|                0.0|               0.0|                    0.0|              0.
0|           89.0|        85.3|       0.0|                      0.0|          0.0|      5.739|       29.65|         1.0|
0.284|         31.0|             0.0|       0.076|     60.52|                    3.876|                    0.416|
3.22|                 8.3|                         14.0|                    0.093|                    2.96|
21.0|                         235.0|                         336.0|                     386.0|                     6.85|
1.0|                    2.84|                  0.559|                    0.0|                   0.22|                    1.5|
+--------+----------+-------------------------+-------------------+------------------+-----------------------+----------------
-+----------------+------------+----------+-------------------------+-------------+----------+------------+-----------+-----
----------+-------------+---------------+------------+----------+-------------------------+-------------------------+-------
--------------+--------------------+-----------------------+-------------------------+-----------------------+----------
------------------+-----------------------------+-----------------------------+---------------------------+-------------------------+----------
-----+-----------------------------+-------------------------+-----------------------+-----------------------+----------------
------+-----------------------+
```

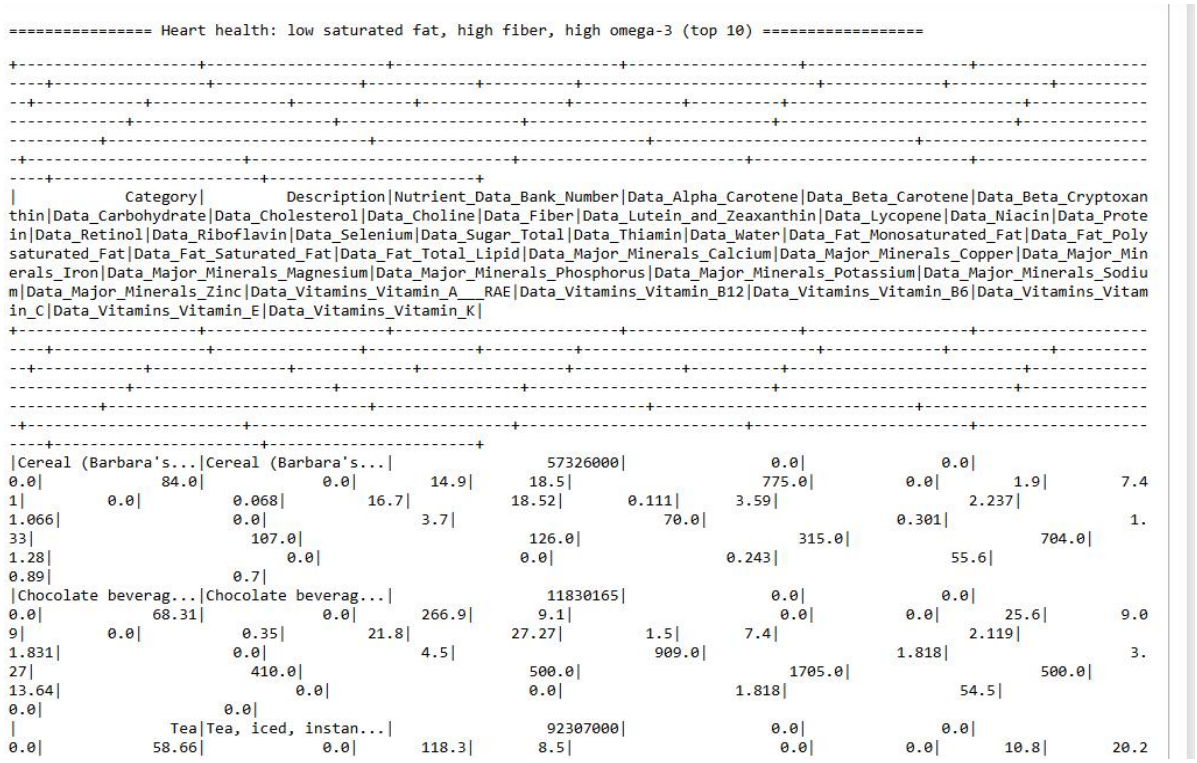
```python
# 5. Specialized food recommendations

# High-protein foods (top 10)
print("=========== High Protein Food (top 10) ==============")
df_clean.orderBy(df_clean['Data_Protein'].desc()).limit(10).show()

# Diabetes-friendly foods: low carbohydrates and sugars (top 10)
print("=========== Diabetes-friendly foods: low carbohydrates and sugars (top 10) ==============")
df_clean.orderBy(
    df_clean['Data_Carbohydrate'],
    df_clean['Data_Sugar_Total']
).limit(10).show()

# Bone health: high calcium and vitamin K (top 10)
print("=============== Bone health: high calcium and vitamin K (top 10) ====================")
df_clean.orderBy(
    df_clean['Data_Protein'].desc(),
    df_clean['Data_Vitamins_Vitamin_K'].desc()
).limit(10).show()

# Heart health: low saturated fat, high fiber, high monounsaturated fat (top 10)
print("================ Heart health: low saturated fat, high fiber, high omega-3 (top 10) ==================")
df_clean.orderBy(
    df_clean['Data_Fat_Saturated_Fat'],
    df_clean['Data_Fiber'].desc(),
    df_clean['Data_Fat_Monosaturated_Fat'].desc()
).limit(10).show()
```

```
=========== High Protein Food (top 10) ==============
+--------------------+--------------------+--------------------------+--------------------+------------------+----------------
----+--------------------+-----------------+-------------+-----------------------------+-----------------+-------------+-------
--+-------------+----------------+-------------+-----------------+-----------+------------+-------------------------+----------
-------------+----------------------+-----------------------+----------------------------+-------------------------+------------
----------+--------------------------+----------------------------+---------------------------------+--------------------------
-+-------------------------+------------------------------+-------------------------------+-----------------------------+-------
----+------------------------+-----------------------+
|            Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Crypto:
thin|Data_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Pro
in|Data_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_Po
saturated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_I
erals_Iron|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_Soc
m|Data_Major_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vi
in_C|Data_Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+--------------------+--------------------+--------------------------+--------------------+------------------+----------------
----+--------------------+-----------------+-------------+-----------------------------+-----------------+-------------+-------
--+-------------+----------------+-------------+-----------------+-----------+------------+-------------------------+----------
-------------+----------------------+-----------------------+----------------------------+-------------------------+------------
----------+--------------------------+----------------------------+---------------------------------+--------------------------
-+-------------------------+------------------------------+-------------------------------+-----------------------------+-------
----+------------------------+-----------------------+
|Nutritional powde...|Nutritional powde...|                  95230000|                0.0|               0.0|
0.0|             6.25|            16.0|       224.0|       3.1|                       0.0|          0.0|      1.136|       7:
3|        0.0|          2.017|         26.7|           0.0|       0.609|    3.44|                          0.158|
0.299|                 0.781|               1.56|                        469.0|                      0.049|
13|                    195.0|                          1321.0|                         500.0|                             156.0|
6.18|                          0.0|                         2.45|                      0.607|                         0.0|
0.0|                   0.0|
|Nutritional powde...|Nutritional powde...|                  95230030|                0.0|               0.0|
0.0|             6.25|            16.0|       224.0|       3.1|                       0.0|          0.0|      1.136|       7:
3|        0.0|          2.017|         26.7|           0.0|       0.609|    3.44|                          0.158|
0.299|                 0.781|               1.56|                        469.0|                      0.049|
13|                    195.0|                          1321.0|                         500.0|                             156.0|
6.18|                          0.0|                         2.45|                      0.607|                         0.0|
0.0|                   0.0|
```

```
=========== Diabetes-friendly foods: low carbohydrates and sugars (top 10) ==============

+----------+--------------------+--------------------------+-------------------+------------------+-----------------------+-----
-----------+----------------+------------+----------+--------------------------+-------------+-----------+------------+-------
-----+---------------+------------+----------------+------------+----------+--------------------------+--------------------------
---+----------------------+-----------------------+---------------------------+--------------------------+--------------------
+-----------------------------+-------------------------------+-----------------------------+--------------------------+--------
----------------+------------------------------+-------------------------+------------------------+-----------------------+-----
-----------------+-----------------------+
|  Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Cryptoxanthin|Data_
Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Protein|Data_Re
tinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_Polysaturated_
Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_Minerals_Iron
|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_Sodium|Data_Maj
or_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vitamin_C|Data_
Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+----------+--------------------+--------------------------+-------------------+------------------+-----------------------+-----
-----------+----------------+------------+----------+--------------------------+-------------+-----------+------------+-------
-----+---------------+------------+----------------+------------+----------+--------------------------+--------------------------
---+----------------------+-----------------------+---------------------------+--------------------------+--------------------
+-----------------------------+-------------------------------+-----------------------------+--------------------------+--------
----------------+------------------------------+-------------------------+------------------------+-----------------------+-----
-----------------+-----------------------+
|     Steak|Steak, NS as to t...|                 21001000|                0.0|               0.0|                    0.0|
0.0|            81.0|        93.0|       0.0|                       0.0|          0.0|      7.908|       29.23|        1.0|
0.237|          33.7|           0.0|       0.082|     62.46|                      2.926|                        0.333|
2.643|                6.75|                   16.0|                      0.083|                      2.38|
23.0|                     238.0|                          371.0|                          387.0|                        5.77|
1.0|                1.98|                      0.676|                    0.0|                      0.3|                      1.5|
|Beef steak|Beef steak, broil...|                 21101120|                0.0|               0.0|                    0.0|
0.0|            87.0|        86.7|       0.0|                       0.0|          0.0|      7.126|       27.06|        3.0|
0.215|          29.8|           0.0|       0.075|     58.14|                      6.005|                        0.595|
5.576|               13.44|                   17.0|                      0.079|                      2.26|
21.0|                     218.0|                          341.0|                          382.0|                        5.21|
```



```
=============== Bone health: high calcium and vitamin K (top 10) ====================

+--------------------+--------------------+-------------------------+------------------+-----------------+------------------
----+----------------+----------------+------------+----------+-------------------------+-------------+----------+----------
--+------------+---------------+-------------+-----------------+------------+----------+-------------------------+-------------
------------+---------------------+-------------------+--------------------------+-------------------------+--------------
---------+-----------------------------+-------------------------+-----------------------+-----------------------+------------------
-+-----------------------+----------------------------+----------------------+----------------------+------------------
----+-----------------------+-----------------------+
|            Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Cryptoxan
thin|Data_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Prote
in|Data_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_Poly
saturated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_Min
erals_Iron|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_Sodiu
m|Data_Major_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vitam
in_C|Data_Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+--------------------+--------------------+-------------------------+------------------+-----------------+------------------
----+----------------+----------------+------------+----------+-------------------------+-------------+----------+----------
--+------------+---------------+-------------+-----------------+------------+----------+-------------------------+-------------
------------+---------------------+-------------------+--------------------------+-------------------------+--------------
---------+-----------------------------+-------------------------+-----------------------+-----------------------+------------------
-+-----------------------+----------------------------+----------------------+----------------------+------------------
----+-----------------------+-----------------------+
|Nutritional powde...|Nutritional powde...|                95230000|               0.0|              0.0|
0.0|             6.25|           16.0|       224.0|       3.1|                      0.0|          0.0|      1.136|       78.1
3|         0.0|          2.017|         26.7|             0.0|       0.609|      3.44|                     0.158|
0.299|                0.781|                1.56|                     469.0|                    0.049|                     1.
13|                     195.0|                     1321.0|                    500.0|                    156.0|
6.18|                      0.0|                    2.45|                   0.607|                      0.0|
0.0|                    0.0|
|Nutritional powde...|Nutritional powde...|                95230030|               0.0|              0.0|
0.0|             6.25|           16.0|       224.0|       3.1|                      0.0|          0.0|      1.136|       78.1
3|         0.0|          2.017|         26.7|             0.0|       0.609|      3.44|                     0.158|
```



```
=============== Heart health: low saturated fat, high fiber, high omega-3 (top 10) ==================

+--------------------+--------------------+-------------------------+------------------+-----------------+------------------
----+----------------+----------------+------------+----------+-------------------------+-------------+----------+----------
--+------------+---------------+-------------+-----------------+------------+----------+-------------------------+-------------
------------+---------------------+-------------------+--------------------------+-------------------------+--------------
---------+-----------------------------+-------------------------+-----------------------+-----------------------+------------------
-+-----------------------+----------------------------+----------------------+----------------------+------------------
----+-----------------------+-----------------------+
|            Category|         Description|Nutrient_Data_Bank_Number|Data_Alpha_Carotene|Data_Beta_Carotene|Data_Beta_Cryptoxan
thin|Data_Carbohydrate|Data_Cholesterol|Data_Choline|Data_Fiber|Data_Lutein_and_Zeaxanthin|Data_Lycopene|Data_Niacin|Data_Prote
in|Data_Retinol|Data_Riboflavin|Data_Selenium|Data_Sugar_Total|Data_Thiamin|Data_Water|Data_Fat_Monosaturated_Fat|Data_Fat_Poly
saturated_Fat|Data_Fat_Saturated_Fat|Data_Fat_Total_Lipid|Data_Major_Minerals_Calcium|Data_Major_Minerals_Copper|Data_Major_Min
erals_Iron|Data_Major_Minerals_Magnesium|Data_Major_Minerals_Phosphorus|Data_Major_Minerals_Potassium|Data_Major_Minerals_Sodiu
m|Data_Major_Minerals_Zinc|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|Data_Vitamins_Vitam
in_C|Data_Vitamins_Vitamin_E|Data_Vitamins_Vitamin_K|
+--------------------+--------------------+-------------------------+------------------+-----------------+------------------
----+----------------+----------------+------------+----------+-------------------------+-------------+----------+----------
--+------------+---------------+-------------+-----------------+------------+----------+-------------------------+-------------
------------+---------------------+-------------------+--------------------------+-------------------------+--------------
---------+-----------------------------+-------------------------+-----------------------+-----------------------+------------------
-+-----------------------+----------------------------+----------------------+----------------------+------------------
----+-----------------------+-----------------------+
|Cereal (Barbara's...|Cereal (Barbara's...|                57326000|               0.0|              0.0|
0.0|            84.0|            0.0|        14.9|      18.5|                    775.0|          0.0|        1.9|        7.4
1|         0.0|          0.068|         16.7|            18.52|       0.111|      3.59|                     2.237|
1.066|                  0.0|                3.7|                      70.0|                    0.301|                     1.
33|                     107.0|                     126.0|                    315.0|                    704.0|
1.28|                      0.0|                    0.0|                   0.243|                     55.6|
0.89|                    0.7|
|Chocolate beverag...|Chocolate beverag...|                11830165|               0.0|              0.0|
0.0|           68.31|            0.0|       266.9|       9.1|                      0.0|          0.0|       25.6|        9.0
9|         0.0|          0.35|         21.8|            27.27|         1.5|       7.4|                     2.119|
1.831|                  0.0|                4.5|                     909.0|                    1.818|                     3.
27|                     410.0|                     500.0|                   1705.0|                    500.0|
13.64|                      0.0|                    0.0|                   1.818|                     54.5|
0.0|                    0.0|
|                Tea|Tea, iced, instan...|                92307000|               0.0|              0.0|
0.0|           58.66|            0.0|       118.3|       8.5|                      0.0|          0.0|       10.8|       20.2
```

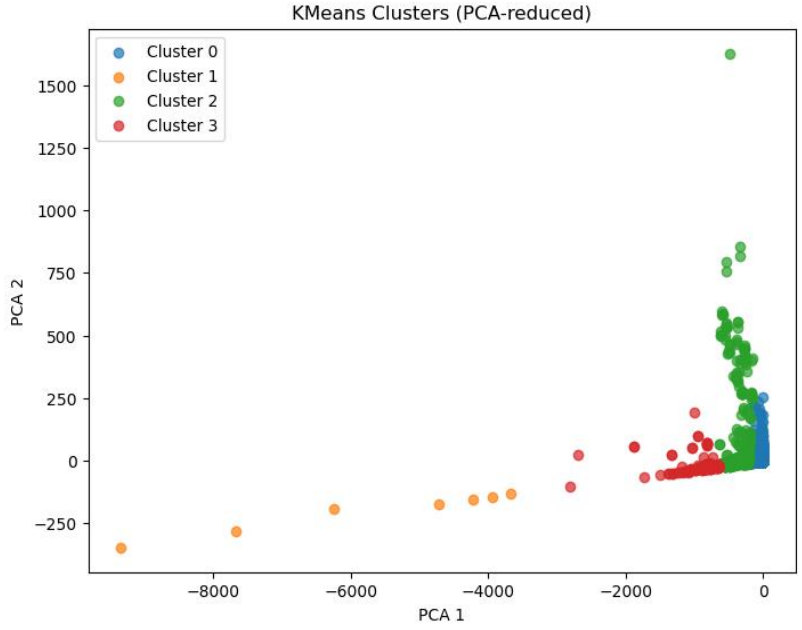
3. ML Engineer: Built clustering (KMeans) and dimensionality reduction (PCA) models for pattern detection.

```python
# 6. Machine Learning Analysis

# Feature selection (Vitamin, Protein, Carbohydrate, Fat, Fiber)
features = [c for c in df.columns if any(x in c for x in ["Vitamin", "Protein", "Carbohydrate", "Fat", "Fiber"])]

assembler = VectorAssembler(inputCols=features, outputCol="features")
df_features = assembler.transform(df).select("features")

# KMeans clustering (k=4)
kmeans = KMeans(k=4, seed=42)
model = kmeans.fit(df_features)
clusters = model.transform(df_features)

print("✅ Clustered Foods")
clusters.show(5, truncate=False)

# Find similar foods by Euclidean distance to the first food item
from pyspark.ml.linalg import Vectors

target = df_features.limit(1).collect()[0]["features"]
similar = (df_features.rdd
           .map(lambda row: (float(Vectors.squared_distance(row["features"], target)), row["features"]))
           .sortByKey()
           .take(6))

print("\n✅ Top 5 similar foods:")
for dist, feats in similar[1:]:
    print(f"Distance={dist:.4f}, Features={feats}")

# Correlation matrix of nutrient features
vector_df = assembler.transform(df).select("features")
corr_matrix = Correlation.corr(vector_df, "features").head()[0]
corr_array = corr_matrix.toArray()
print("\n✅ Correlation matrix shape:", corr_array.shape)
print(corr_array)
```

```
✅ Clustered Foods
+----------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------+----------+
|features
|prediction|
+----------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------+----------+
|[6.889999866485596,0.0,1.0299999713897705,1.6579999923706055,0.4970000088214874,2.009000062942505,4.380000114440918,61.0,0.050
00000074505806,0.010999999940395355,5.0,0.07999999821186066,0.30000001192092896]                          |0         |
|[4.869999885559082,0.0,3.3399999141693115,0.4259999990463257,0.06499999761581421,1.1640000343322754,1.9900000095367432,59.0,0.
5600000023841858,0.05999999865889549,0.10000000149011612,0.029999999329447746,0.20000000298023224]|0         |
|[4.670000076293945,0.0,3.2799999713897705,0.6880000233650208,0.1080000028014183,1.8600000143051147,3.200000047683716,32.0,0.54
00000214576721,0.06100000068545415,0.0,0.05000000074505806,0.30000001192092896]                          |0         |
|[4.460000038146973,0.0,3.0999999046325684,0.9990000128746033,0.12800000607967377,2.1540000438690186,3.4600000381469727,29.0,0.
36000001430511475,0.03400000184774399,0.8999999761581421,0.07999999821186066,0.30000001192092896] |0         |
|[4.670000076293945,0.0,3.2799999713897705,0.6880000233650208,0.1080000028014183,1.8600000143051147,3.200000047683716,32.0,0.54
00000214576721,0.06100000068545415,0.0,0.05000000074505806,0.30000001192092896]                          |0         |
+----------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------------+----------+
only showing top 5 rows

✅ Top 5 similar foods:
Distance=4.0157, Features=[7.400000095367432,0.0,1.7200000286102295,1.3250000476837158,0.718999981880188,1.5119999647140503,3.5
49999952316284,60.0,0.20000000298023224,0.03999999910593033,5.900000095367432,0.5899999737739563,0.0]
Distance=4.0157, Features=[7.400000095367432,0.0,1.7200000286102295,1.3250000476837158,0.718999981880188,1.5119999647140503,3.5
49999952316284,60.0,0.20000000298023224,0.03999999910593033,5.900000095367432,0.5899999737739563,0.0]
Distance=4.0157, Features=[7.400000095367432,0.0,1.7200000286102295,1.3250000476837158,0.718999981880188,1.5119999647140503,3.5
49999952316284,60.0,0.20000000298023224,0.03999999910593033,5.900000095367432,0.5899999737739563,0.0]
Distance=4.0157, Features=[7.400000095367432,0.0,1.7200000286102295,1.3250000476837158,0.718999981880188,1.5119999647140503,3.5
49999952316284,60.0,0.20000000298023224,0.03999999910593033,5.900000095367432,0.5899999737739563,0.0]
Distance=4.0157, Features=[7.400000095367432,0.0,1.7200000286102295,1.3250000476837158,0.718999981880188,1.5119999647140503,3.5
49999952316284,60.0,0.20000000298023224,0.03999999910593033,5.900000095367432,0.5899999737739563,0.0]

✅ Correlation matrix shape: (13, 13)
[[ 1.          0.32864258 -0.16771915  0.04580604  0.1000772   0.07554255
   0.09808108  0.04739915 -0.02049868  0.17034773  0.07390755  0.10042638
  -0.0913621 ]
 [ 0.32864258  1.         -0.02505399  0.20299966  0.27644682  0.03593452
   0.08350884  0.06331013  0.12199638  0.48191885  0.06784769  0.14478561
   0.03618866]
 [-0.16771915 -0.02505399  1.          0.22427035  0.08142519  0.18116615
   0.2265446   0.02939101  0.1013274   0.20934402 -0.15045437  0.07541236
  -0.09602871]
```

```python
# 7. Cluster visualization with PCA

# Add row IDs to align DataFrames
df_with_id = df.withColumn("row_id", monotonically_increasing_id())
df_features_with_id = df_features.withColumn("row_id", monotonically_increasing_id())
clusters_with_id = clusters.withColumn("row_id", monotonically_increasing_id())

# Join cluster assignments to the original DataFrame
df_with_clusters = (
    df_with_id
    .join(clusters_with_id.select("row_id", "prediction"), on="row_id")
    .drop("row_id")
    .withColumnRenamed("prediction", "cluster")
)

display(df_with_clusters)

# 2D visualization using PCA
pca = PCA(k=2, inputCol="features", outputCol="pca_features")
pca_model = pca.fit(df_features)
df_pca = pca_model.transform(df_features_with_id)

# Join cluster assignments to PCA DataFrame
df_pca = (
    df_pca
    .join(clusters_with_id.select("row_id", "prediction"), on="row_id")
    .withColumnRenamed("prediction", "cluster")
)

# Convert to Pandas for matplotlib plotting
pdf = df_pca.select("pca_features", "cluster").toPandas()
pdf["x"] = pdf["pca_features"].apply(lambda v: v[0])
pdf["y"] = pdf["pca_features"].apply(lambda v: v[1])

import matplotlib.pyplot as plt

plt.figure(figsize=(8,6))
for cluster in sorted(pdf["cluster"].unique()):
    subset = pdf[pdf["cluster"] == cluster]
    plt.scatter(subset["x"], subset["y"], label=f"Cluster {cluster}", alpha=0.7)
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.title("KMeans Clusters (PCA-reduced)")
plt.legend()
plt.show()
```

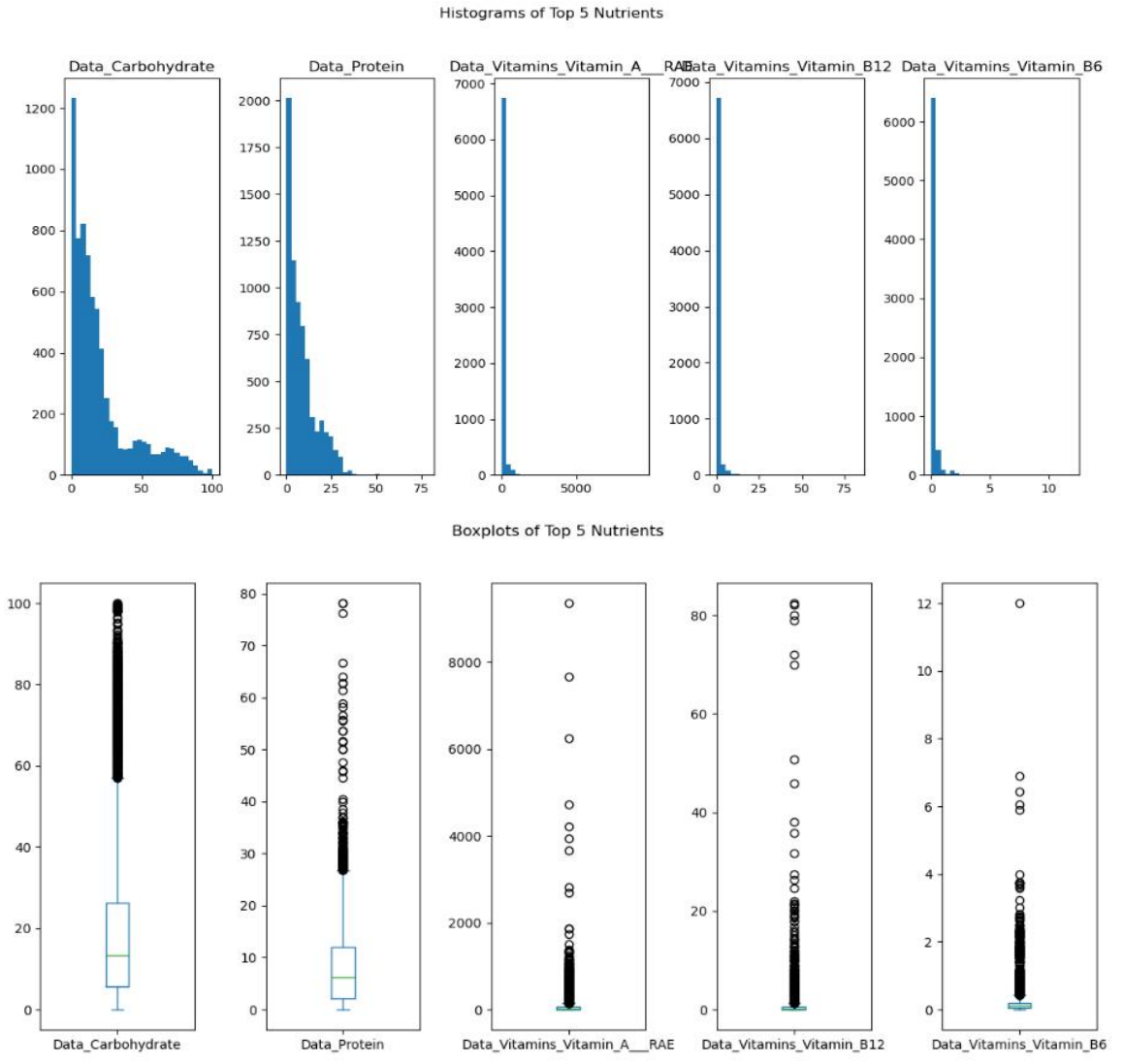
```
DataFrame[Category: string, Description: string, Nutrient_Data_Bank_Number: string, Data_Alpha_Carotene: float, Data_Beta_Carotene: float, Data_Beta_Cryptoxanthin: float, Data_Carbohydrate: float, Data_Cholesterol: float, Data_Choline: float, Data_Fiber: float, Data_Lutein_and_Zeaxanthin: float, Data_Lycopene: float, Data_Niacin: float, Data_Protein: float, Data_Retinol: float, Data_Riboflavin: float, Data_Selenium: float, Data_Sugar_Total: float, Data_Thiamin: float, Data_Water: float, Data_Fat_Monosaturated_Fat: float, Data_Fat_Polysaturated_Fat: float, Data_Fat_Saturated_Fat: float, Data_Fat_Total_Lipid: float, Data_Major_Minerals_Calcium: float, Data_Major_Minerals_Copper: float, Data_Major_Minerals_Iron: float, Data_Major_Minerals_Magnesium: float, Data_Major_Minerals_Phosphorus: float, Data_Major_Minerals_Potassium: float, Data_Major_Minerals_Sodium: float, Data_Major_Minerals_Zinc: float, Data_Vitamins_Vitamin_A___RAE: float, Data_Vitamins_Vitamin_B12: float, Data_Vitamins_Vitamin_B6: float, Data_Vitamins_Vitamin_C: float, Data_Vitamins_Vitamin_E: float, Data_Vitamins_Vitamin_K: float, cluster: int]
```



Analysis on Normalized dataset(only on top-5 nutrients)

4. Visualization Specialist: Designed bar charts, boxplots, and PCA plots for clear interpretation.

```
# Histograms and boxplots for selected nutrients
df_plot = df_clean.select(selected_cols).toPandas()

df_plot.hist(bins=30, figsize=(12, 6), layout=(1, 5), grid=False)
plt.suptitle('Histograms of Top 5 Nutrients')
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()

df_plot.plot(kind='box', subplots=True, layout=(1, 5), figsize=(12, 6), sharey=False)
plt.suptitle('Boxplots of Top 5 Nutrients')
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```



```
# Use only selected nutrient columns, e.g., top 5
selected_cols = [
    c for c in df.columns
    if any(x in c for x in ["Vitamin", "Protein", "Carbohydrate"])
][:5]

# Convert to RDD of Rows
rdd_rows = df.select(*selected_cols).rdd
```

```
# Convert each row to tuple of floats
rdd_tuples = rdd.map(lambda row: tuple(float(x) if x is not None else 0 for x in row))

# Compute column-wise min and max using reduce
min_values = rdd_tuples.reduce(lambda x, y: tuple(min(a,b) for a,b in zip(x,y)))
max_values = rdd_tuples.reduce(lambda x, y: tuple(max(a,b) for a,b in zip(x,y)))

print("Mins:", min_values)
print("Maxs:", max_values)
```

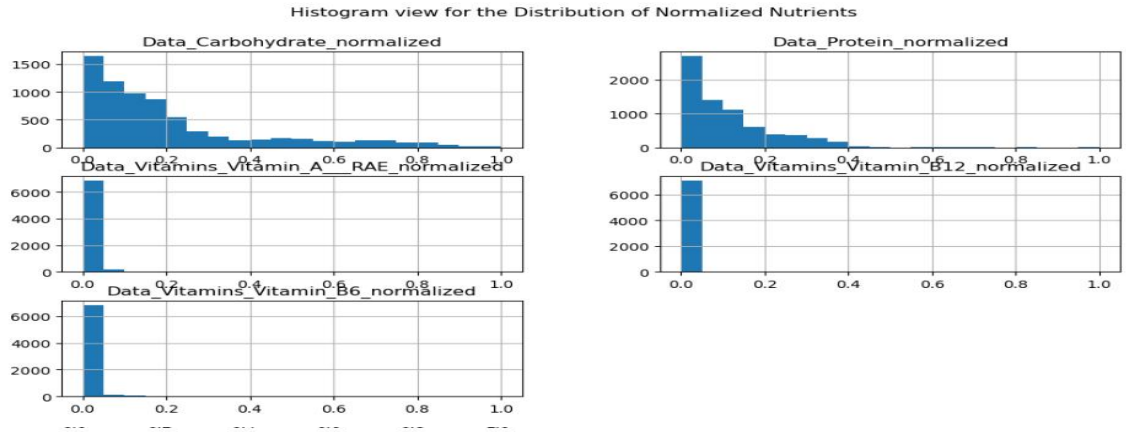
```
Mins: (0.0, 0.0, 0.0, 0.0, 0.0)
Maxs: (100.0, 78.12999725341797, 9363.0, 879.0, 14.0)
```

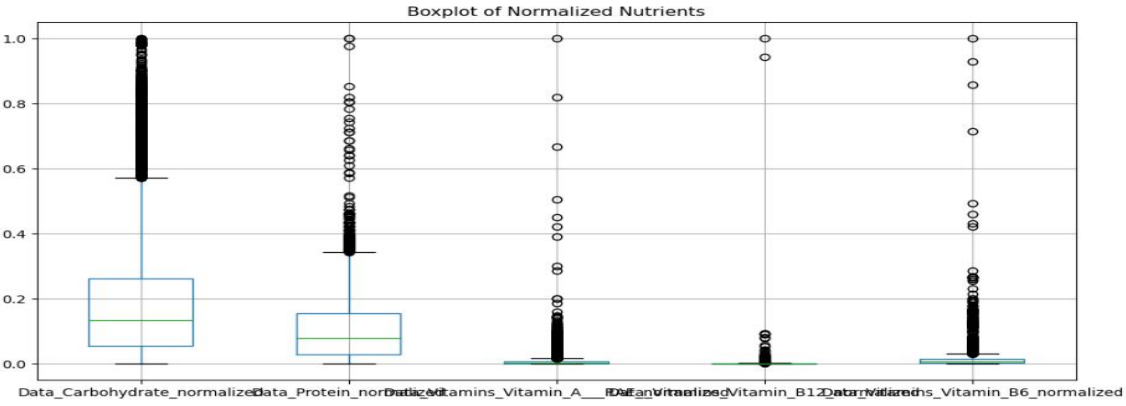
```
def normalize(row):
    normalized = []
    for i, val in enumerate(row):
        if max_values[i] != min_values[i]:
            norm_val = (val - min_values[i]) / (max_values[i] - min_values[i])
        else:
            norm_val = 0.0  # Handle constant columns
        normalized.append(norm_val)
    return tuple(normalized)

rdd_normalized = rdd_tuples.map(normalize)
```

```
from pyspark.sql.types import StructType, StructField, FloatType

# Define schema with FloatType for each selected column
schema = StructType([StructField(col, FloatType(), True) for col in selected_cols])

# Create DataFrame with schema
normalized_df = spark.createDataFrame(rdd_rows, schema)

normalized_df.show(5)
```

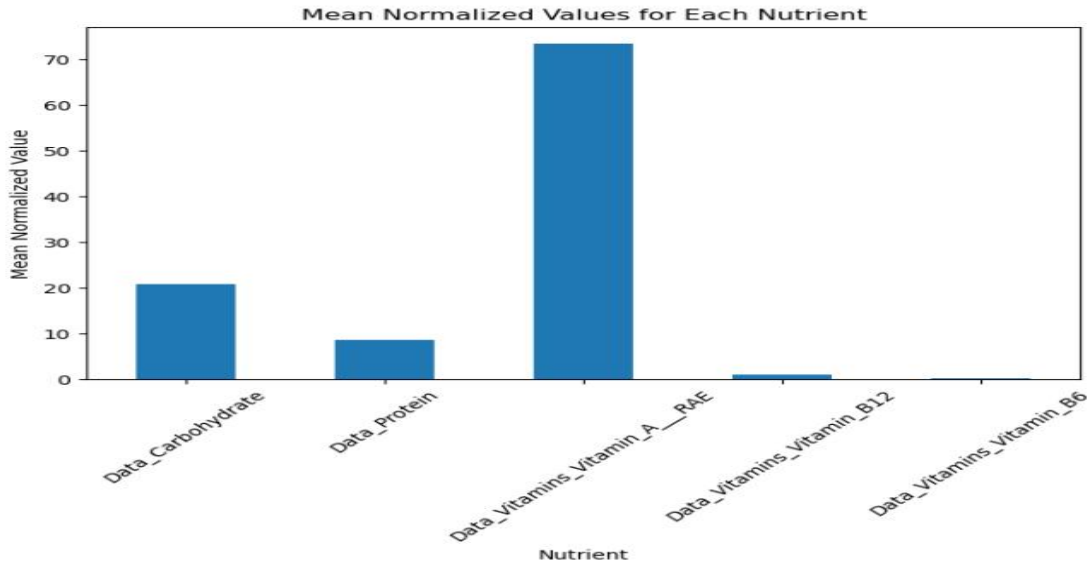
```
+-----------------+------------+---------------------------+------------------------+------------------------+
|Data_Carbohydrate|Data_Protein|Data_Vitamins_Vitamin_A___RAE|Data_Vitamins_Vitamin_B12|Data_Vitamins_Vitamin_B6|
+-----------------+------------+---------------------------+------------------------+------------------------+
|             6.89|        1.03|                       61.0|                    0.05|                   0.011|
|             4.87|        3.34|                       59.0|                    0.56|                    0.06|
|             4.67|        3.28|                       32.0|                    0.54|                   0.061|
|             4.46|         3.1|                       29.0|                    0.36|                   0.034|
|             4.67|        3.28|                       32.0|                    0.54|                   0.061|
+-----------------+------------+---------------------------+------------------------+------------------------+
only showing top 5 rows
```

```
normalized_pd = df_normalized.toPandas()
import matplotlib.pyplot as plt

# Histogram to view the distribution of each nutrient
normalized_pd.hist(column=[c for c in normalized_pd.columns if c != "Category"], bins=20, figsize=(12, 6))

plt.suptitle("Histogram view for the Distribution of Normalized Nutrients")
plt.show()

# Boxplot - visualize spread and outliers per nutrient
normalized_pd.boxplot(column=[c for c in normalized_pd.columns if c != "Category"], figsize=(12,6))
plt.title("Boxplot of Normalized Nutrients")
plt.show()
```






```python
# Bar plot of mean normalized values per nutrient
mean_values = normalized_pd.mean()
mean_values.plot(kind='bar', figsize=(8, 5))
plt.title('Mean Normalized Values for Each Nutrient')
plt.ylabel('Mean Normalized Value')
plt.xlabel('Nutrient')
plt.xticks(rotation=45)
plt.show()
```



7. Insights and Observations

- Foods from categories such as snacks and fast food tend to have higher energy and fat content.
- Fruits and vegetables are rich in vitamins but low in energy content.
- A positive correlation was observed between protein and energy levels.
- Outliers in certain categories may indicate data recording inconsistencies or unique food compositions.

8. Conclusion

This project demonstrates how PySpark can be effectively used for large-scale nutritional data analysis. Through data cleaning, transformation, and visualization, we gain meaningful insights into food compositions that could inform health, diet planning, and food industry research.