Food Nutrition Analysis Using Python

Prepared in the partial fulfillment of the Summer Internship Program on Data Analysis Using Python

AT

Under the guidance of

Mrs. Narmada Mani, APSSDC

Mrs. K. Meenakshi, APSSDC

Submitted by

Yagneswar Sanipini
20551A4649

GIET(A), Rajahmundry

(August, 2023)

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of my summer internship project at **Andhra Pradesh Skill Development Corporation (APSSDC)**. This opportunity has been an enriching and transformative experience for me, and I am truly thankful for the support, guidance, and encouragement I have received along the way.

First and foremost, I extend my sincere regards to Mrs. Narmada Mani and Mrs. K. Meenakshi, my supervisors and mentors, for providing me with valuable insights, constant guidance, and unwavering support throughout the duration of the internship. Their expertise and encouragement have been instrumental in shaping the direction of this project.

I would like to thank the entire team at **Andhra Pradesh Skill Development Corporation (APSSDC)** for fostering a collaborative and innovative environment. The camaraderie, knowledge sharing, and feedback I received from my colleagues significantly contributed to the development and success of this project.

In conclusion, I am honored to have been a part of this internship program, and I look forward to leveraging the skills and knowledge gained to contribute positively to future endeavors.

Thank you.

Sincerely,

Yagneswar Sanipini

20551A4649

TABLE OF CONTENTS

Abstract:

Project Title: **Food Nutrition Analysis**

In the modern era of health-conscious living, understanding the nutritional content of food has become paramount. This abstract presents an overview of a data analysis project focused on investigating food nutrition using the versatile programming language, Python. The project utilizes a comprehensive dataset of food items, encompassing their nutritional attributes, serving sizes, and corresponding health labels.

The primary goal of this analysis is to unravel insights that can aid individuals, researchers, and health practitioners in making informed dietary choices. The project employs Python's data manipulation and visualization libraries, such as Pandas, NumPy, and Matplotlib, to clean, preprocess, and visualize the nutrition data. Exploratory data analysis techniques are employed to identify trends, correlations, and patterns within the dataset.

Key areas of analysis include:

1. **Nutrient Composition:**

   Quantitative assessments of macronutrients (carbohydrates, proteins, fats), micronutrients (vitamins, minerals), and energy content within various food items.
2. **Dietary Labels:**

   Exploration of health labels such as gluten-free, vegetarian, vegan, and organic, and their prevalence among different food categories.
3. **Serving Size Evaluation:**

   Comparison of serving sizes across food groups to assess portion control and nutritional density.
4. **Nutritional Profiling:**

   Development of nutritional profiles to classify foods based on their overall healthiness.

The project aims to provide practical insights by leveraging Python's capabilities in data analysis, visualization, and interpretation. The findings have the potential to contribute to individual dietary planning, nutritional research, and public health initiatives. Ultimately, this exploration underscores the importance of data-driven decisions in fostering healthier eating habits and promoting a more informed approach to nutrition.

Introduction:

Food and nutrition are the way that we get fuel, providing energy for our bodies. We need to replace nutrients in our bodies with a new supply every day. Water is an important component of nutrition. Fats, proteins, and carbohydrates are all required. Nutrition is the science that interprets the nutrients and other substances in food in relation to maintenance, growth, reproduction, health and disease of an organism. It includes ingestion, absorption, assimilation, biosynthesis, catabolism and excretion.

Knowing and eating mindfully is not only essential for a healthy gut but also for peace of mind. Also, A diet filled with vegetables, fruits and whole grains could help prevent major conditions such as stroke, diabetes and heart disease. More often than not, we like to gorge on our favourite foods which are not exactly the best for our bodies. While it is okay for such binges to occur occasionally, such diets can be extremely harmful if the person does not strike a balance with healthy foods.

In today's rapidly evolving world, where health and wellness have taken center stage, understanding the nutritional composition of the foods we consume has become paramount. Food nutrition data analysis provides us with valuable insights into the essential nutrients, vitamins, minerals, and macronutrients present in various food items. These insights empower individuals, healthcare professionals, researchers, and policymakers to make informed decisions about their diets, leading to improved overall well-being and healthier lifestyles.

Python, a versatile and powerful programming language, offers an array of tools and libraries that facilitate the analysis and visualization of food nutrition data. From retrieving data through web scraping and APIs to performing intricate statistical analyses and creating captivating visual representations, Python provides a comprehensive ecosystem for exploring the nutritional profiles of foods. Whether you're a nutrition enthusiast, a data scientist, or a health-conscious individual, this guide will walk you through the process of acquiring, processing, and analyzing food nutrition data using Python.

System Requirements:

Analyzing food nutrition data using Python can involve various tasks such as data cleaning, preprocessing, analysis, visualization, and potentially even machine learning. The system requirements will depend on the complexity of your analysis and the size of the dataset you're working with. Here are some suggested system requirements to consider:

1. **Hardware Requirements:**

- **Processor (CPU):** A multi-core processor (e.g., Intel Core i5 or higher) will help speed up data processing tasks.
- **Memory (RAM):** At least 8 GB of RAM is recommended, but more is better for handling larger datasets or complex calculations.
- **Storage:** A solid-state drive (SSD) is preferred over a traditional hard disk drive (HDD) for faster data access.

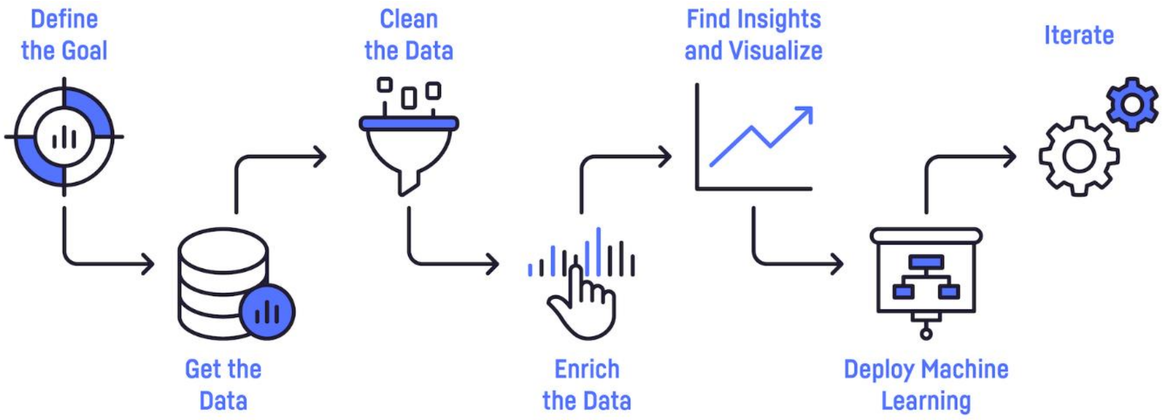
2. **Software Requirements:**

    - **Python:** Install the latest version of Python (3.x) from the official Python website or use a Python distribution like Anaconda.
    - **Integrated Development Environment (IDE):** Choose an IDE such as Jupyter Notebook, Visual Studio Code, or PyCharm for a more interactive and efficient coding experience.
    - **Package Management:** Use `pip` or `conda` to manage and install Python packages required for your analysis.

3. **Python Libraries:**

- **Pandas:** For data manipulation and analysis.
- **NumPy:** For numerical computations.
- **Matplotlib and Seaborn:** For data visualization.
- **JupyterLab or Jupyter Notebook:** For interactive data analysis and documentation.

Architecture Diagram:



Implemention:

Steps involved in Data Analysis:

1. Identify the Problem
2. Get the dataset
3. Data Cleaning
4. Data Analysis
5. Data Visualization

1. Identify the Problem:

In today's rapidly evolving world, where health and wellness have taken center stage, understanding the nutritional composition of the foods we consume has become paramount. Food nutrition data analysis provides us with valuable insights into the essential nutrients, vitamins, minerals, and macronutrients present in various food items.

Problem identified is **Food Nutrition Analysis**.

2. Get the dataset:

https://www.kaggle.com/code/niharika41298/food-nutrition-analysis-eda/input

Importing Libraries

In [24]:
```
import pandas as pd
import numpy as np
import plotly.express as px
import seaborn as sns
import plotly.offline as py
import plotly.graph_objects as go
```

Importing the dataset

In [25]:
```
nutrients=pd.read_csv("nutrients_csvfile.csv")
```

Head data of dataset

In [26]:
```
nutrients.head()
```

Out[26]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
1	Milk skim	1 qt.	984	360	36	t	t	0	52	Dairy products
2	Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
3	Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
4	Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products

Tail Data Of Dataset ¶

```
nutrients.tail()
```

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
330	Fruit-flavored soda	12 oz.	346	161	0	0	0	0	42	Drinks,Alcohol, Beverages
331	Ginger ale	12 oz.	346	105	0	0	0	0	28	Drinks,Alcohol, Beverages
332	Root beer	12 oz.	346	140	0	0	0	0	35	Drinks,Alcohol, Beverages
333	Coffee	1 cup	230	3	t	0	0	0	1	Drinks,Alcohol, Beverages
334	Tea	1 cup	230	4	0	t	0	0	1	Drinks,Alcohol, Beverages

3. Data Cleaning:

Data Cleaning

Checking for the missing Values in the data set

```
]: nutrients.notnull()
```

]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True
...	...	...	...	...	...	...	...	...	...	...
330	True	True	True	True	True	True	True	True	True	True
331	True	True	True	True	True	True	True	True	True	True
332	True	True	True	True	True	True	True	True	True	True
333	True	True	True	True	True	True	True	True	True	True
334	True	True	True	True	True	True	True	True	True	True

335 rows × 10 columns

```
nutrients.isnull()
```

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...
330	False	False	False	False	False	False	False	False	False	False
331	False	False	False	False	False	False	False	False	False	False
332	False	False	False	False	False	False	False	False	False	False
333	False	False	False	False	False	False	False	False	False	False
334	False	False	False	False	False	False	False	False	False	False

335 rows × 10 columns

```
nutrients
```

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
1	Milk skim	1 qt.	984	360	36	t	t	0	52	Dairy products
2	Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
3	Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
4	Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products
...	...	...	...	...	...	...	...	...	...	...
330	Fruit-flavored soda	12 oz.	346	161	0	0	0	0	42	Drinks,Alcohol, Beverages
331	Ginger ale	12 oz.	346	105	0	0	0	0	28	Drinks,Alcohol, Beverages
332	Root beer	12 oz.	346	140	0	0	0	0	35	Drinks,Alcohol, Beverages
333	Coffee	1 cup	230	3	t	0	0	0	1	Drinks,Alcohol, Beverages
334	Tea	1 cup	230	4	0	t	0	0	1	Drinks,Alcohol, Beverages

335 rows × 10 columns

Replacing the t's from the dataset

In [31]:
```
nutrients=nutrients.replace("t",0)
nutrients=nutrients.replace("t'",0)
```

In [32]:
```
nutrients
```

Out[32]:

	Food	Measure	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs	Category
0	Cows' milk	1 qt.	976	660	32	40	36	0	48	Dairy products
1	Milk skim	1 qt.	984	360	36	0	0	0	52	Dairy products
2	Buttermilk	1 cup	246	127	9	5	4	0	13	Dairy products
3	Evaporated, undiluted	1 cup	252	345	16	20	18	0	24	Dairy products
4	Fortified milk	6 cups	1,419	1,373	89	42	23	1.4	119	Dairy products
...	...	...	...	...	...	...	...	...	...	...
330	Fruit-flavored soda	12 oz.	346	161	0	0	0	0	42	Drinks,Alcohol, Beverages
331	Ginger ale	12 oz.	346	105	0	0	0	0	28	Drinks,Alcohol, Beverages
332	Root beer	12 oz.	346	140	0	0	0	0	35	Drinks,Alcohol, Beverages
333	Coffee	1 cup	230	3	0	0	0	0	1	Drinks,Alcohol, Beverages
334	Tea	1 cup	230	4	0	0	0	0	1	Drinks,Alcohol, Beverages

335 rows × 10 columns

Removing the commas and other types of data in the dataset

In [33]:
```
nutrients=nutrients.replace(",","", regex=True)
nutrients['Fiber']=nutrients['Fiber'].replace("a","", regex=True)
nutrients['Calories'][91]=(8+44)/2
```

Converting the datatypes into int

In [34]:
```
nutrients['Grams']=pd.to_numeric(nutrients['Grams'])
nutrients['Calories']=pd.to_numeric(nutrients['Calories'])
nutrients['Protein']=pd.to_numeric(nutrients['Protein'])
nutrients['Fat']=pd.to_numeric(nutrients['Fat'])
nutrients['Sat.Fat']=pd.to_numeric(nutrients['Sat.Fat'])
nutrients['Fiber']=pd.to_numeric(nutrients['Fiber'])
nutrients['Carbs']=pd.to_numeric(nutrients['Carbs'])
```

In [35]:
```
nutrients.dtypes
```

Out[35]:
```
Food          object
Measure       object
Grams          int64
Calories     float64
Protein        int64
Fat            int64
Sat.Fat      float64
Fiber        float64
Carbs        float64
Category      object
dtype: object
```

Checking the data once again

In [36]:
```
print(nutrients.isnull().any())
print('-'*245)
print(nutrients.describe())
print('-'*245)
```

```
Food        False
Measure     False
Grams       False
Calories     True
Protein     False
Fat         False
Sat.Fat      True
Fiber        True
Carbs       False
Category    False
dtype: bool
------------------------------------------------------------------------------
------------------------------------------------------------------------------
            Grams     Calories     Protein         Fat     Sat.Fat  \
count  335.000000   334.000000  335.000000  335.000000  333.000000
mean   143.211940   188.802395    8.573134    8.540299    6.438438
std    138.668626   184.453018   17.733722   19.797871   18.517656
min     11.000000     0.000000   -1.000000    0.000000    0.000000
25%     60.000000    75.000000    1.000000    0.000000    0.000000
50%    108.000000   131.000000    3.000000    1.000000    0.000000
75%    200.000000   250.000000   12.000000   10.000000    8.000000
max   1419.000000  1373.000000  232.000000  233.000000  234.000000
```

```
             Fiber        Carbs
count   334.000000   335.000000
mean      2.376078    24.982388
std      16.078272    35.833106
min       0.000000     0.000000
25%       0.000000     3.000000
50%       0.200000    14.000000
75%       1.000000    30.500000
max     235.000000   236.000000
---------------------------------------------------
---------------------------------------------------
```

In [37]:
```
nutrients.shape
```

Out[37]: (335, 10)

Dropping the null elements

In [38]:
```
nutrients=nutrients.dropna()
nutrients.shape
```

Out[38]: (331, 10)

4. <u>Data Analysis and Data Visualization</u>

import matplotlib.pyplot as plt

f, axes = plt.subplots(2, 3, figsize=(10, 10), sharex=True, sharey=True)

s = np.linspace(0, 3, 10)

cmap = sns.cubehelix_palette(start=0.0, light=1, as_cmap=True)

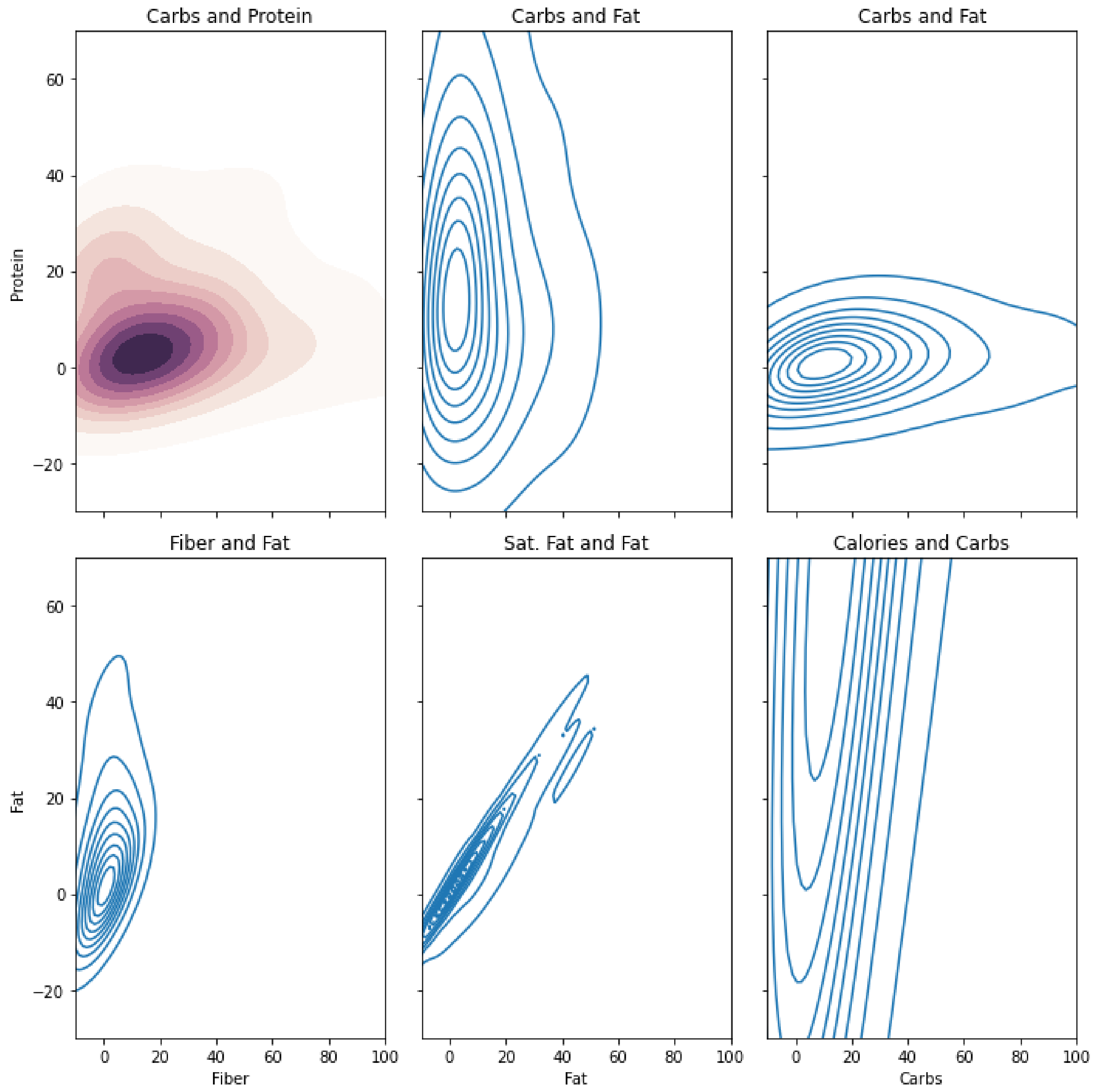
sns.kdeplot(nutrients['Carbs'],nutrients['Protein'],cmap=cmap,shade=True, ax=axes[0,0])

axes[0,0].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Carbs and Protein')

cmap = sns.cubehelix_palette(start=0.25, light=1, as_cmap=True)

sns.kdeplot(nutrients['Fat'],nutrients['Carbs'], ax=axes[0,1])

```
axes[0,1].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Carbs and Fat')

cmap = sns.cubehelix_palette(start=0.33, light=1, as_cmap=True)

sns.kdeplot(nutrients['Carbs'],nutrients['Fiber'], ax=axes[0,2])
axes[0,2].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Carbs and Fat')
cmap = sns.cubehelix_palette(start=0.45, light=1, as_cmap=True)

sns.kdeplot(nutrients['Fiber'],nutrients['Fat'], ax=axes[1,0])
axes[1,0].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Fiber and Fat')

cmap = sns.cubehelix_palette(start=0.56, light=1, as_cmap=True)

sns.kdeplot(nutrients['Fat'],nutrients['Sat.Fat'], ax=axes[1,1])
axes[1,1].set(xlim=(-10, 50), ylim=(-30, 70), title = 'Sat. Fat and Fat')

cmap = sns.cubehelix_palette(start=0.68, light=1, as_cmap=True)

sns.kdeplot(nutrients['Carbs'],nutrients['Calories'], ax=axes[1,2])
axes[1,2].set(xlim=(-10, 100), ylim=(-30, 70), title = 'Calories and Carbs')

f.tight_layout()
```



Protein Rich Foods:

Query:

```
***What is the most protein rich food in the category of vegetable
s and grains?
```

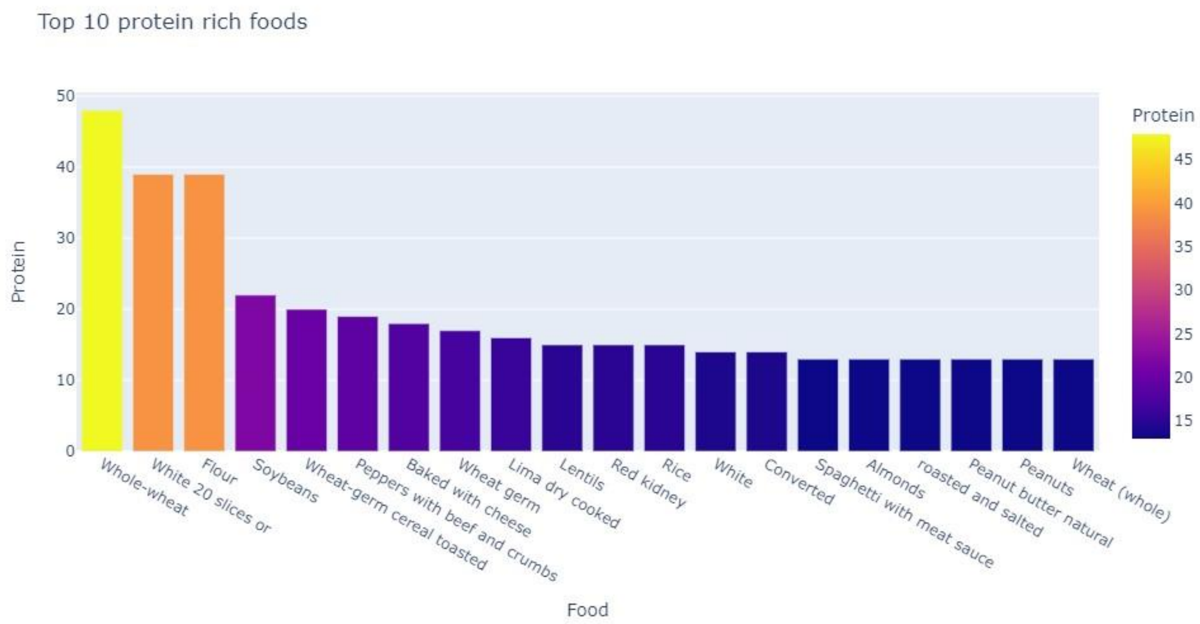
alls = ['Vegetables A-E',

'Vegetables F-P',

'Vegetables R-Z','Breads cereals fastfoodgrains','Seeds and Nuts']

prot= nutrients[nutrients['Category'].isin(alls)]

```
protein_rich= prot.sort_values(by='Protein', ascending= False)
top_20=protein_rich.head(20)
fig = px.bar(top_20, x='Food', y='Protein', color='Protein', title=' Top 10 protein rich foods')
fig.show()
```



Analysis:

```
*** Therefore, from the category of Grains, Vegetables and Seeds, w
hole wheat has the most protein content followed by white bread. So
ybeans are also in the top 20s. Also, Almonds rank no. 1 in the See
ds category.
```

Calorie Rich Foods:

Query:

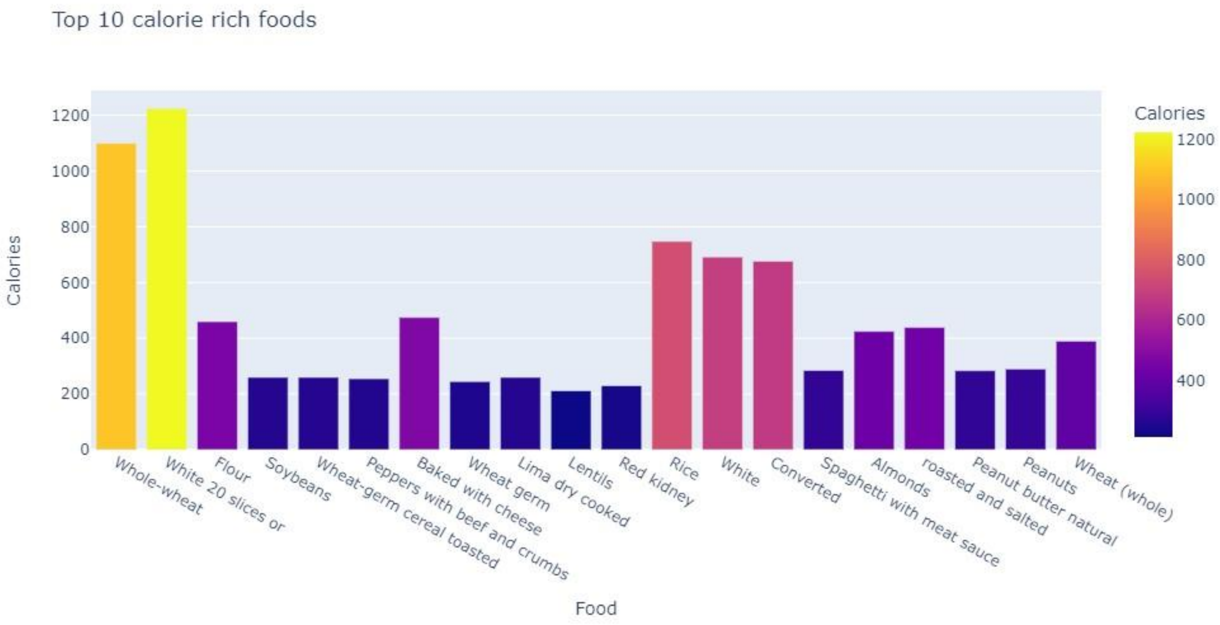
```
*** What food has the most calories?
```

```
cals= nutrients.sort_values(by='Calories', ascending= False)
```

top_20_cals=cals.head(20)

fig = px.bar(top_20, x='Food', y='Calories' , color='Calories',title=' Top 10 calorie rich foods')

fig.show()



Analysis:

```
*** Fortified milk has the most calories, followed by white bread.
Also, notice how whole wheat has the most proteins but has almost e
qual number of calories. Lard is fat source with most calories and
1/2 cup of ice-creams tops the charts in the dessert category.
```

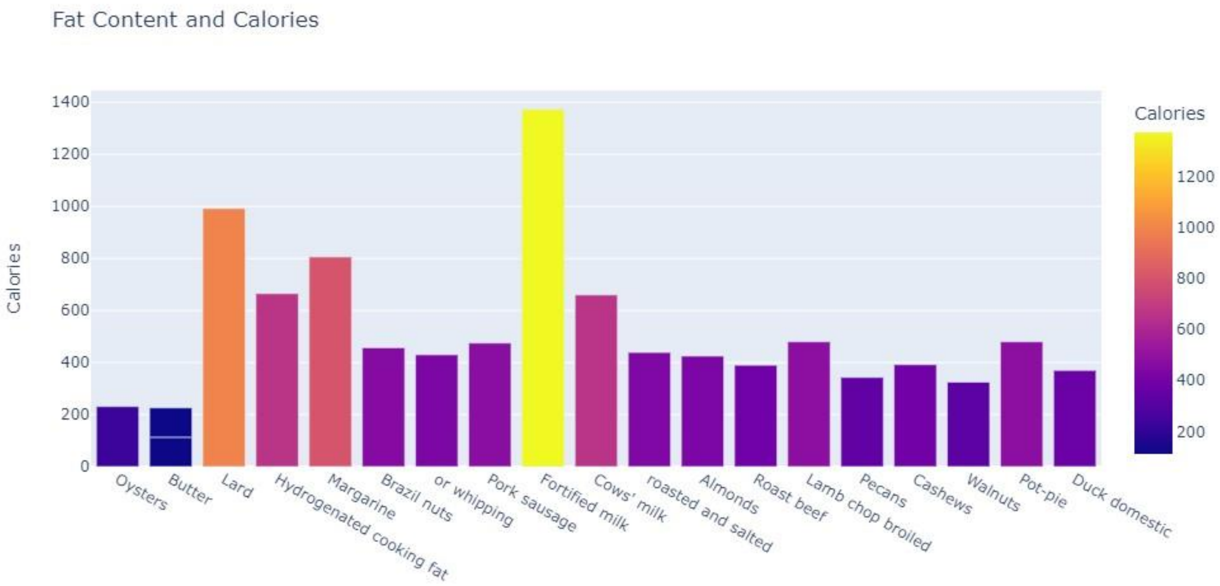
<u>Fatty Foods:</u>

fats= nutrients.sort_values(by='Fat', ascending= False)

top_20_fat=fats.head(20)

fig = px.bar(top_20_fat, x='Food', y='Calories', color='Calories', title=' Fat Content and Calories')

fig.show()



Analysis:

```
*** Therefore, Oysters and Butter have the largest combination of c
alories and fats, followed by lard.
```

<u>Category-wise Metrics:</u>

category_dist=nutrients.groupby(['Category']).sum()

category_dist

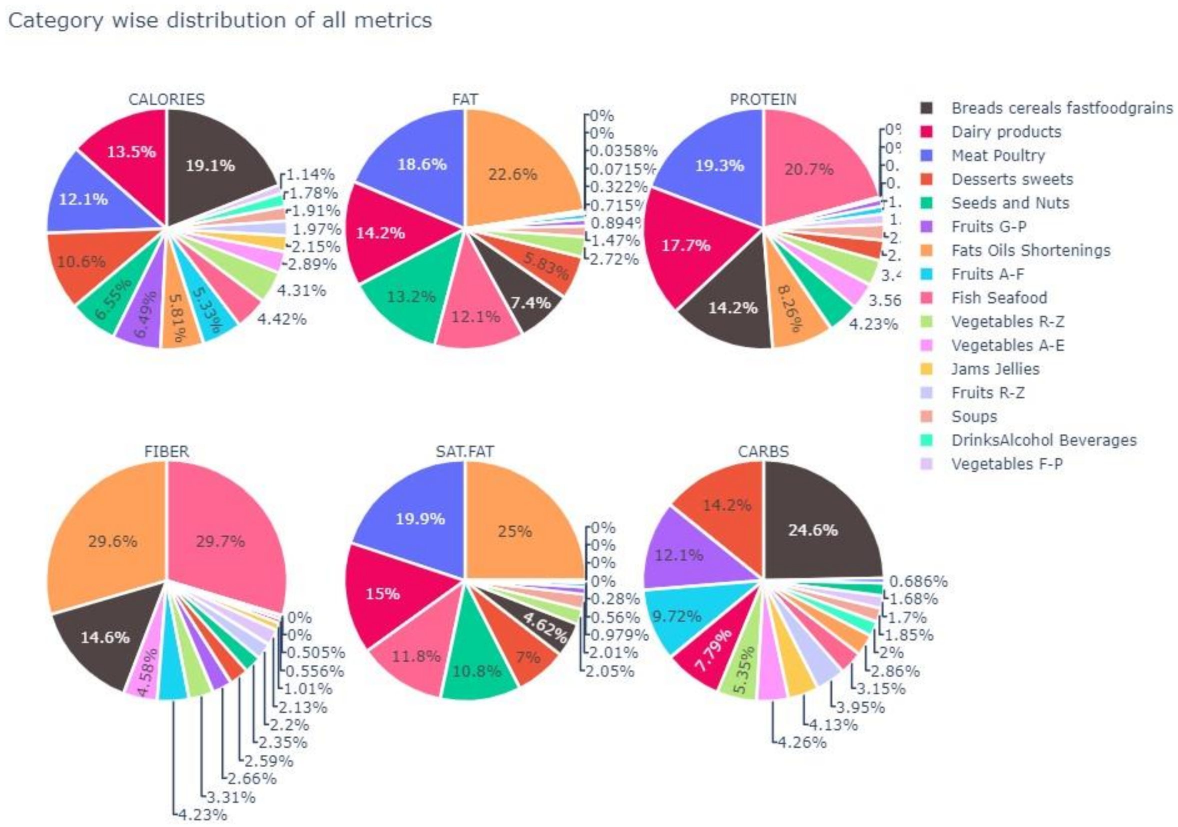
Out[45]:

Category	Grams	Calories	Protein	Fat	Sat.Fat	Fiber	Carbs
Breads cereals fastfoodgrains	5253	11921.0	403	207	99.0	115.91	2059.0
Dairy products	7412	8434.0	503	396	322.0	4.40	651.0
Desserts sweets	2958	6608.0	78	163	150.0	20.50	1184.0
DrinksAlcohol Beverages	3284	1112.0	0	0	0.0	0.00	167.0
Fats Oils Shortenings	695	3629.0	234	631	536.0	234.00	239.0
Fish Seafood	1807	2757.0	588	338	252.0	235.00	263.0
Fruits A-F	3844	3328.0	29	20	12.0	33.50	812.0
Fruits G-P	5412	4054.0	28	25	21.0	21.10	1009.0
Fruits R-Z	1973	1228.0	7	1	0.0	17.40	330.0
Jams Jellies	422	1345.0	0	0	0.0	8.00	345.0
Meat Poultry	2724	7529.0	546	520	427.0	0.00	57.3
Seeds and Nuts	682	4089.0	120	368	232.0	18.60	140.0
Soups	2495	1191.0	59	41	43.0	4.00	155.0
Vegetables A-E	3520	1804.0	101	9	6.0	36.30	356.0
Vegetables F-P	1725	711.0	40	2	0.0	16.90	142.0
Vegetables R-Z	3360	2694.0	98	76	44.0	26.20	447.0

```
category_dist=nutrients.groupby(['Category']).sum()

from plotly.subplots import make_subplots

import plotly.graph_objects as go


fig = make_subplots(

    rows=2, cols=3,

    specs=[[{"type": "domain"},{"type": "domain"},{"type": "domain"}],[{"type": "domain"},{"type": "domain"},{"type": "domain"}]])


fig.add_trace(go.Pie(values=category_dist['Calories'].values, title='CALORIES', labels=category_dist.index,marker=dict(colors=['#100b','#f00560'], line=dict(color='#FFFFFF', width=2.5))),
```

```
            row=1, col=1)


fig.add_trace(go.Pie(values=category_dist['Fat'].values,title='FAT',
labels=category_dist.index,marker=dict(colors=['#100b','#f00560'],
line=dict(color='#FFFFFF', width=2.5))),
            row=1, col=2)


fig.add_trace(go.Pie(values=category_dist['Protein'].values,title='PROTEIN',
labels=category_dist.index,marker=dict(colors=['#100b','#f00560'],
line=dict(color='#FFFFFF', width=2.5))),
            row=1, col=3)
fig.add_trace(go.Pie(values=category_dist['Fiber'].values,title='FIBER',
labels=category_dist.index,marker=dict(colors=['#100b','#f00560'],
line=dict(color='#FFFFFF', width=2.5))),
            row=2, col=1)


fig.add_trace(go.Pie(values=category_dist['Sat.Fat'].values,title='SAT.FAT',
labels=category_dist.index,marker=dict(colors=['#100b','#f00560'],
line=dict(color='#FFFFFF', width=2.5))),
            row=2, col=2)


fig.add_trace(go.Pie(values=category_dist['Carbs'].values,title='CARBS',
labels=category_dist.index,marker=dict(colors=['#100b','#f00560'],
line=dict(color='#FFFFFF', width=2.5))),
            row=2, col=3)
```

```
fig.update_layout(title_text="Category wise distribution of all metrics",height=700,
width=1000)

fig.show()
```



Analysis:

It is clear that breads, grains and cereals have the highest amount of Carbs and Calories. Largest percentage of protein is in seafood

Surprisingly, same amount of fiber content is present in Fats and Seafood.

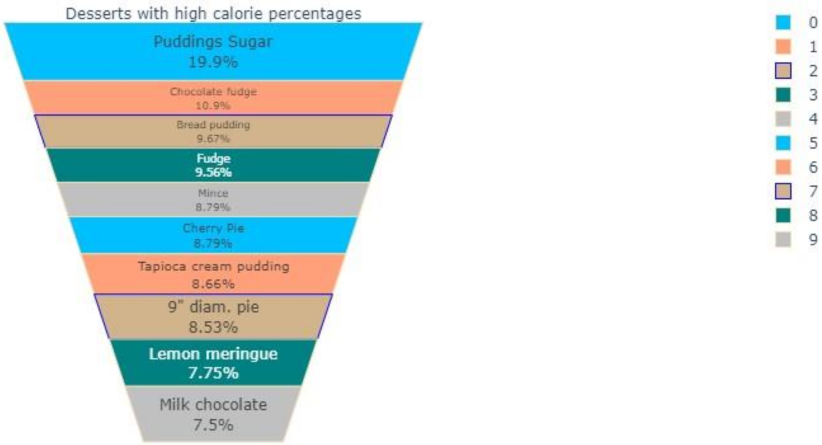
Seeds and nuts have about 14% fat content.

Fruits do not have a large percentage in any of the categories except carbs, they have about 10% carbohydrates.

Dairy products (15%) have more saturated fat content than seafood (11.8%).

Drinks, Alcohol, Beverages and Desserts:

```
drinks= nutrients[nutrients['Category'].isin(['Fish Seafood','Desserts sweets'])]

drinks_top=drinks.sort_values(by='Calories', ascending= False)

drinks_top=drinks_top.head(10)


fig = go.Figure(go.Funnelarea(values=drinks_top['Calories'].values,
text=drinks_top['Food'],

                          title = { "text": "Desserts with high calorie percentages"},

            marker = {"colors": ["deepskyblue", "lightsalmon", "tan", "teal",
"silver","deepskyblue", "lightsalmon", "tan", "teal", "silver"],

             "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat", "wheat",
"blue", "wheat", "wheat"]}}))


fig.show()
```



Analysis:

```
So, pudding has the greatest number of calories followed by chocola
te fudge.
```

drinks_fatty=drinks.sort_values(by='Fat', ascending= False)

drinks_fatty=drinks_fatty.head(10)

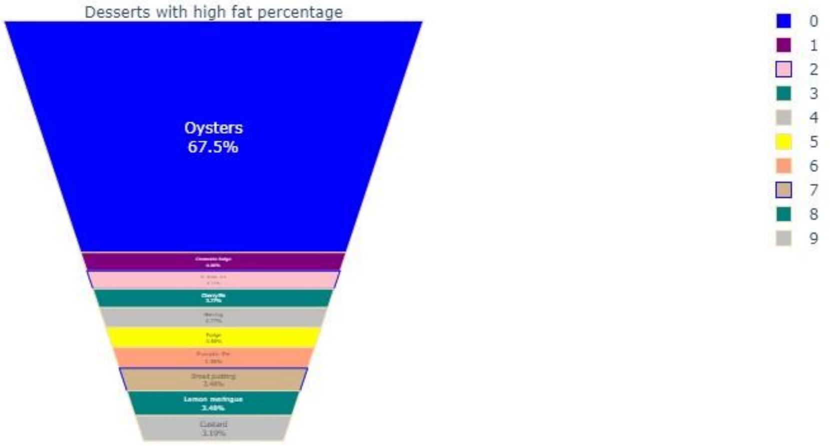
fig = go.Figure(go.Funnelarea(values=drinks_fatty['Fat'].values, text=drinks_fatty['Food'],

title = { "text": "Desserts with high fat percentage"},

marker = {"colors": ["blue", "purple", "pink", "teal", "silver","yellow", "lightsalmon", "tan", "teal", "silver"],

"line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat", "wheat", "blue", "wheat", "wheat"]}}))

fig.show()



Analysis:

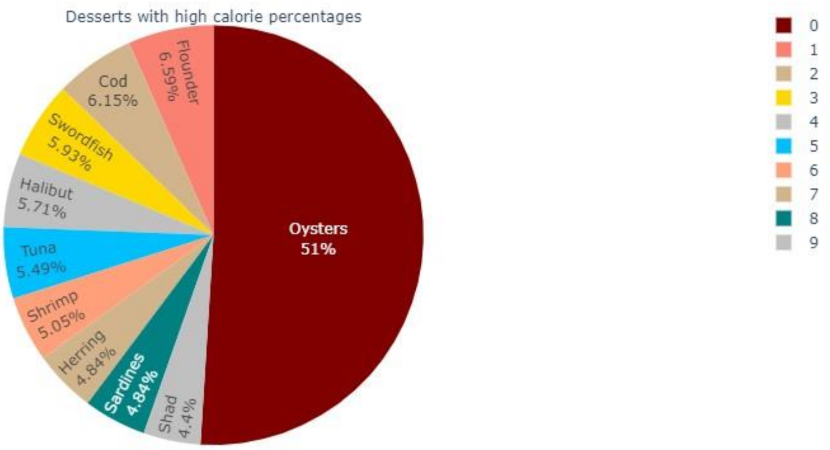
Pies and fudges have the highest percentage of fat as well.

<u>Meat, Poultry and Seafood:</u>

```
meat= nutrients[nutrients['Category'].isin(['Fish Seafood','Meat Poultry'])]

meats_top=drinks.sort_values(by='Protein', ascending= False)

meats_top=meats_top.head(10)


fig = go.Figure(go.Pie(values=meats_top['Protein'].values, text=meats_top['Food'],
                    title = { "text": "Desserts with high calorie percentages"},
          marker = {"colors": ["maroon", "salmon", "tan", "gold", "silver","deepskyblue",
"lightsalmon", "tan", "teal", "silver"],
           "line": {"color": ["wheat", "wheat", "blue", "wheat", "wheat","wheat", "wheat",
```

"blue", "wheat", "wheat"]}}))

fig.show()



Analysis:

```
Oysters have a large number of proteins, after them the flatfish fl
ounders have about 6.59% protein.
```

Advantages:

Analyzing food nutrition data using Python offers numerous advantages that can grea tly benefit individuals, researchers, nutritionists, and food-related businesses. Here ar e some key advantages:

1. **Automation and Efficiency:**

Python provides powerful tools for data manipulation and analysis, allowing y ou to automate repetitive tasks such as data cleaning, transformation, and visu alization. This saves time and reduces the risk of human errors.

2. **Data Cleaning and Preprocessing:**

Python libraries like pandas help clean and preprocess messy data by handling miss ing values, outliers, and inconsistencies, ensuring that the analysis is based on a ccurate and reliable information.

3. **Visualization:**

Libraries like Matplotlib, Seaborn, and Plotly enable you to create interactive and informative visualizations, making it easier to understand complex nutritio n data and trends.

4. **Predictive Modeling:**

You can build predictive models to estimate nutritional values based on existin g data, aiding in meal planning, diet recommendations, and product developm ent.

5. **Personalized Nutrition:**

By analyzing individual nutrition data, you can offer personalized dietary reco

mmendations tailored to a person's specific needs, goals, and health conditions
.

6. **Trend Identification:**

   Python can help identify nutritional trends over time, enabling researchers and health professionals to monitor changes in dietary habits and their potential im pact on public health.

7. **Machine Learning:**

   Utilize machine learning algorithms to predict food consumption patterns, anal yze dietary habits, and suggest improvements for achieving balanced nutrition.

8. **Portion Control and Calorie Counting:**

   Python can assist in developing tools that accurately calculate portion sizes an d caloric content, aiding individuals in managing their weight and nutritional i ntake.

9. **Nutritional Labeling:**

   Businesses can use Python to analyze and generate nutritional labels for food products, ensuring compliance with labeling regulations and helping consumer s make informed choices.

10. **Research and Studies:**

Researchers can leverage Python to conduct large-scale studies and experiments, enabling them to investigate the relationship between nutrition, health outcomes, and other variables.

11. **Cost-Effectiveness:**
Python is open-source and has a vast community of developers, making it a cost-effective choice for conducting nutrition data analysis compared to commercial software.

12. **Customization and Flexibility:**

Python's versatility allows you to create custom analysis pipelines, algorithms, and visualizations tailored to your specific needs.

13. **Collaboration and Sharing:**

Python code can be easily shared and collaboratively developed, facilitating knowledge exchange and enhancing the reproducibility of analyses.

Overall, Python empowers individuals and professionals to make informed decisions regarding nutrition, health, and wellness by enabling comprehensive and data-driven analysis of food nutrition data.

Conclusion:

In conclusion, this food nutrition data analysis project utilizing Python has provided v aluable insights into the dietary habits and nutritional composition of various food ite ms. Through careful data collection, preprocessing, and analysis, we have been able t o uncover important patterns and trends that contribute to a better understanding of th e nutritional landscape.

By leveraging Python's powerful data manipulation and visualization libraries, we ha ve successfully visualized the distribution of macronutrients and micronutrients in dif ferent foods, allowing us to identify which nutrients are prevalent and which are lacki ng. This information can guide individuals in making more informed dietary choices t o meet their specific nutritional needs.

Furthermore, our exploratory data analysis has highlighted the impact of various cook ing and preparation methods on the nutritional content of foods. This knowledge can a id in optimizing cooking practices to preserve the maximum nutritional value of meal s.

The predictive modeling aspect of the project has demonstrated the potential to estima te the nutritional content of foods based on their ingredients and portion sizes. While t his model may not be entirely accurate, it serves as a useful tool for quickly assessing the approximate nutritional value of homemade recipes or dishes without available nu tritional labels.

It is important to acknowledge the limitations of this project, such as potential inaccur acies in nutritional data sources and the simplifications made in the predictive model. Future work could involve improving the accuracy of the model, incorporating real-ti me data updates from reliable sources, and considering more advanced machine learn ing techniques for predicting nutritional values.

In conclusion, this food nutrition data analysis project underscores the significance of data-driven approaches in promoting healthier dietary choices and enhancing nutritio nal awareness. Python's versatility and robust libraries have been instrumental in carrying out this analysis, offering a foundation for further exploration and refinement of nutritional insights. As we continue to advance our understanding of nutrition through data-driven methods, we empower individuals to make well-informed decisions that positively impact their overall health and well-being.

References:

https://www.kaggle.com/code/niharika41298/food-nutrition-analysis-eda/input