STANLEY

COLLEGE OF ENGINEERING & TECHNOLOGY FOR WOMEN (AUTONOMOUS)

Estd. 2008

STANLEY

COLLEGE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Private Un-aided Non-minority UGC Autonomous Institution)

Nutritional Analysis of Indian Cuisine:

A Regional and Dietary Perspective

Name : N.Laharika

Roll No : 160623747311

Branch : AI&DS

Section : B

📌 Problem Statement

Title: Predicting Calorie Content of Indian Foods Using Nutritional Attributes

Goal:

Indian cuisine includes a wide variety of dishes with complex nutritional compositions. Estimating calorie content manually is often difficult. This project aims to:

- Predict the **calories** of a food item based on its nutrients.
- Support applications like health monitoring apps and diet recommendation engines.

📊 Dataset Overview

DatasetName:Indian_Food_Nutrition_Processed.csv
Rows:1014
Columns: 12

Column	Description
Dish Name	Name of the Indian food dish
Calories (kcal)	Energy in kilocalories
Carbohydrates (g)	Total carbohydrates

Column	Description
Protein (g)	Protein content
Fats (g)	Fat content
Free Sugar (g)	Free sugar content
Fibre (g)	Dietary fiber
Sodium (mg)	Sodium content
Calcium (mg)	Calcium content
Iron (mg)	Iron content
Vitamin C (mg)	Vitamin C (missing in some rows)
Folate (µg)	Folate content (missing in some rows)

🎯 Loading packages and the dataset

#importing packages

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

#importing dataset

```
df = pd.read_csv('/content/Indian_Food_Nutrition_Processed.csv')
df.head()
```

index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
0	Hot tea (Garam Chai)	16.14	2.58	0.39	0.53	2.58	0.0	3.12	14.2	0.02	0.5	1.8
1	Instant coffee	23.16	3.65	0.64	0.75	3.62	0.0	4.92	20.87	0.06	1.51	5.6
2	Espreso coffee	51.54	6.62	1.75	2.14	6.53	0.0	13.98	58.1	0.15	1.51	5.53
3	Iced tea	10.34	2.7	0.03	0.01	2.7	0.0	0.23	1.18	0.02	5.95	1.28
4	Raw mango drink (Aam panna)	35.92	9.05	0.16	0.03	7.49	0.61	79.82	7.08	0.14	45.3	14.05

df.tail()

index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
1009	Sweet and sour tomato pickle (Khatta meetha tamatar ka achaar)	60.88	6.55	1.26	3.24	4.31	2.2	1281.94	15.18	0.54	NaN	NaN
1010	Jhatpat achar with carrot (Jhatpat achaar gajar ke saath)	91.21	6.32	1.98	6.55	3.04	5.08	2067.33	54.31	2.32	NaN	NaN
1011	Tomato chutney (Tamatar ki chutney)	176.07	31.85	0.97	6.01	30.02	1.49	823.65	25.34	0.96	NaN	NaN
1012	Tomato ketchup	33.07	6.48	0.91	0.3	4.68	1.9	177.97	15.33	0.36	NaN	NaN
1013	Bengal 5 Spice Blend (Panch Phoran)	289.79	20.0	18.26	22.16	1.4	18.4	51.43	523.0	13.37	NaN	NaN

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1014 entries, 0 to 1013
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Dish Name          1014 non-null   object 
 1   Calories (kcal)    1014 non-null   float64
 2   Carbohydrates (g)  1014 non-null   float64
 3   Protein (g)        1014 non-null   float64
 4   Fats (g)           1014 non-null   float64
 5   Free Sugar (g)     1014 non-null   float64
 6   Fibre (g)          1014 non-null   float64
 7   Sodium (mg)        1014 non-null   float64
 8   Calcium (mg)       1014 non-null   float64
 9   Iron (mg)          1014 non-null   float64
 10  Vitamin C (mg)     932 non-null    float64
 11  Folate (µg)        932 non-null    float64
dtypes: float64(11), object(1)
memory usage: 95.2+ KB
```

df.columns

```
Index(['Dish Name', 'Calories (kcal)', 'Carbohydrates (g)', 'Protein (g)',
       'Fats (g)', 'Free Sugar (g)', 'Fibre (g)', 'Sodium (mg)',
       'Calcium (mg)', 'Iron (mg)', 'Vitamin C (mg)', 'Folate (µg)'],
      dtype='object')
```

🔧 Preprocessing Steps

Working on Missing Values:

```
df.isnull().sum()
```

	0
Dish Name	0
Calories (kcal)	0
Carbohydrates (g)	0
Protein (g)	0
Fats (g)	0
Free Sugar (g)	0
Fibre (g)	0
Sodium (mg)	0
Calcium (mg)	0
Iron (mg)	0
Vitamin C (mg)	82
Folate (µg)	82

dtype: int64

```
bool_series = pd.isnull(df["Vitamin C (mg)"])

bool_series1 = pd.isnull(df["Folate (µg)"])

missing_data = df[bool_series]

missing_data.head()
```

index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
932	White sauce (thin)	115.17	6.58	3.31	8.41	4.58	0.07	23.91	110.2	0.19	NaN	NaN
933	White sauce (medium)	151.27	7.98	3.36	11.76	4.32	0.14	22.59	103.53	0.22	NaN	NaN
934	White sauce (thick)	209.69	10.24	3.43	17.19	3.9	0.24	20.45	92.74	0.27	NaN	NaN
935	Cheese sauce	194.76	7.08	7.68	15.12	4.53	0.12	317.38	255.14	0.31	NaN	NaN
936	Egg sauce	146.85	5.88	5.97	11.08	3.19	0.12	51.54	89.49	0.65	NaN	NaN

```
missing_data1 = df[bool_series1]

missing_data1.head()
```

index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
932	White sauce (thin)	115.17	6.58	3.31	8.41	4.58	0.07	23.91	110.2	0.19	NaN	NaN
933	White sauce (medium)	151.27	7.98	3.36	11.76	4.32	0.14	22.59	103.53	0.22	NaN	NaN
934	White sauce (thick)	209.69	10.24	3.43	17.19	3.9	0.24	20.45	92.74	0.27	NaN	NaN
935	Cheese sauce	194.76	7.08	7.68	15.12	4.53	0.12	317.38	255.14	0.31	NaN	NaN
936	Egg sauce	146.85	5.88	5.97	11.08	3.19	0.12	51.54	89.49	0.65	NaN	NaN

```
non_missing_values = df.notnull()

non_missing_values.head()
```

	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
0	True	True	True	True	True	True	True	True	True	True	True	True
1	True	True	True	True	True	True	True	True	True	True	True	True
2	True	True	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True	True	True

🧹 Data Cleaning

```
df_clean=df.dropna()

df_clean.head()
```

index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Folate (µg)
0	Hot tea (Garam Chai)	16.14	2.58	0.39	0.53	2.58	0.0	3.12	14.2	0.02	0.5	1.8
1	Instant coffee	23.16	3.65	0.64	0.75	3.62	0.0	4.92	20.87	0.06	1.51	5.6
2	Espreso coffee	51.54	6.62	1.75	2.14	6.53	0.0	13.98	58.1	0.15	1.51	5.53
3	Iced tea	10.34	2.7	0.03	0.01	2.7	0.0	0.23	1.18	0.02	5.95	1.28
4	Raw mango drink (Aam panna)	35.92	9.05	0.16	0.03	7.49	0.61	79.82	7.08	0.14	45.3	14.05

```
df_clean = df_clean.rename(columns ={'Folate (µg)' : 'Vitamin B (µg)'})
```

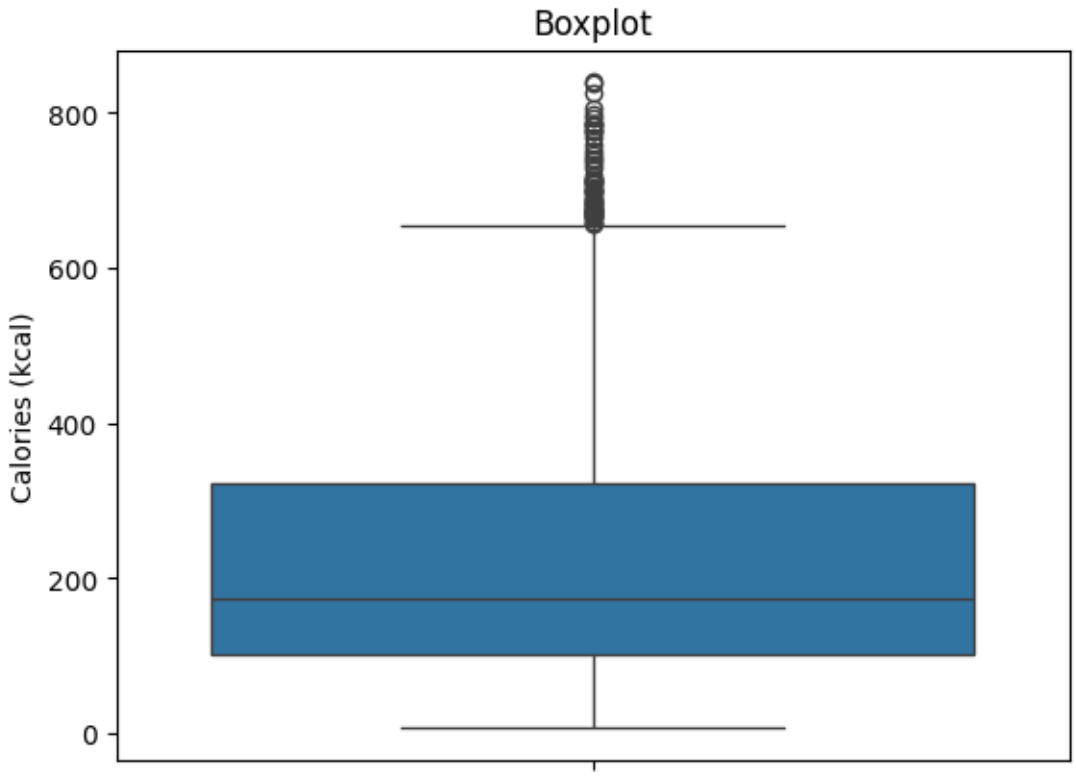
index	Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Vitamin B (µg)
0	Hot tea (Garam Chai)	16.14	2.58	0.39	0.53	2.58	0.0	3.12	14.2	0.02	0.5	1.8
1	Instant coffee	23.16	3.65	0.64	0.75	3.62	0.0	4.92	20.87	0.06	1.51	5.6
2	Espreso coffee	51.54	6.62	1.75	2.14	6.53	0.0	13.98	58.1	0.15	1.51	5.53
3	Iced tea	10.34	2.7	0.03	0.01	2.7	0.0	0.23	1.18	0.02	5.95	1.28
4	Raw mango drink (Aam panna)	35.92	9.05	0.16	0.03	7.49	0.61	79.82	7.08	0.14	45.3	14.05

```
duplicate = df_clean[df_clean.duplicated()]
```

```
duplicate
```

Dish Name	Calories (kcal)	Carbohydrates (g)	Protein (g)	Fats (g)	Free Sugar (g)	Fibre (g)	Sodium (mg)	Calcium (mg)	Iron (mg)	Vitamin C (mg)	Vitamin B (µg)

```
sns.boxplot(df_clean["Calories (kcal)"])
plt.title('Boxplot')
```




```
Q1 = df_clean["Calories (kcal)"].quantile(0.25)
Q3 = df_clean["Calories (kcal)"].quantile(0.75)
```

```
IQR = Q3 - Q1
IQR
```

```
np.float64(222.0775)
```

```
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
lower_bound
```

```
np.float64(-232.56874999999997)
```

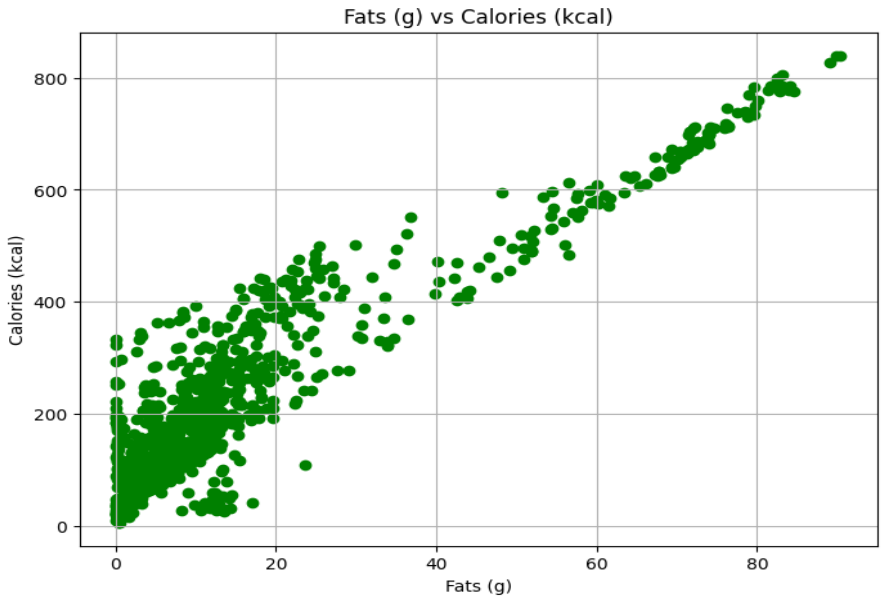
```
upper_bound
```

```
np.float64(655.74125)
```

📈 Exploratory Data Analysis

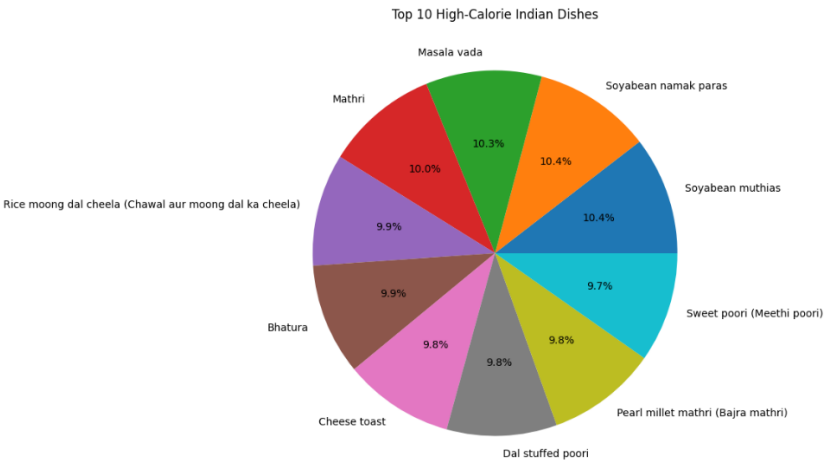
- **Scatter plot:** A Strong Positive Correlation Between Fats and Calorie Content in Indian

```
plt.figure(figsize=(8, 6))
plt.scatter(df['Fats (g)'], df['Calories (kcal)'], color='green')
plt.title('Fats (g) vs Calories (kcal)')
plt.xlabel('Fats (g)')
plt.ylabel('Calories (kcal)')
plt.grid(True)
plt.show()
```



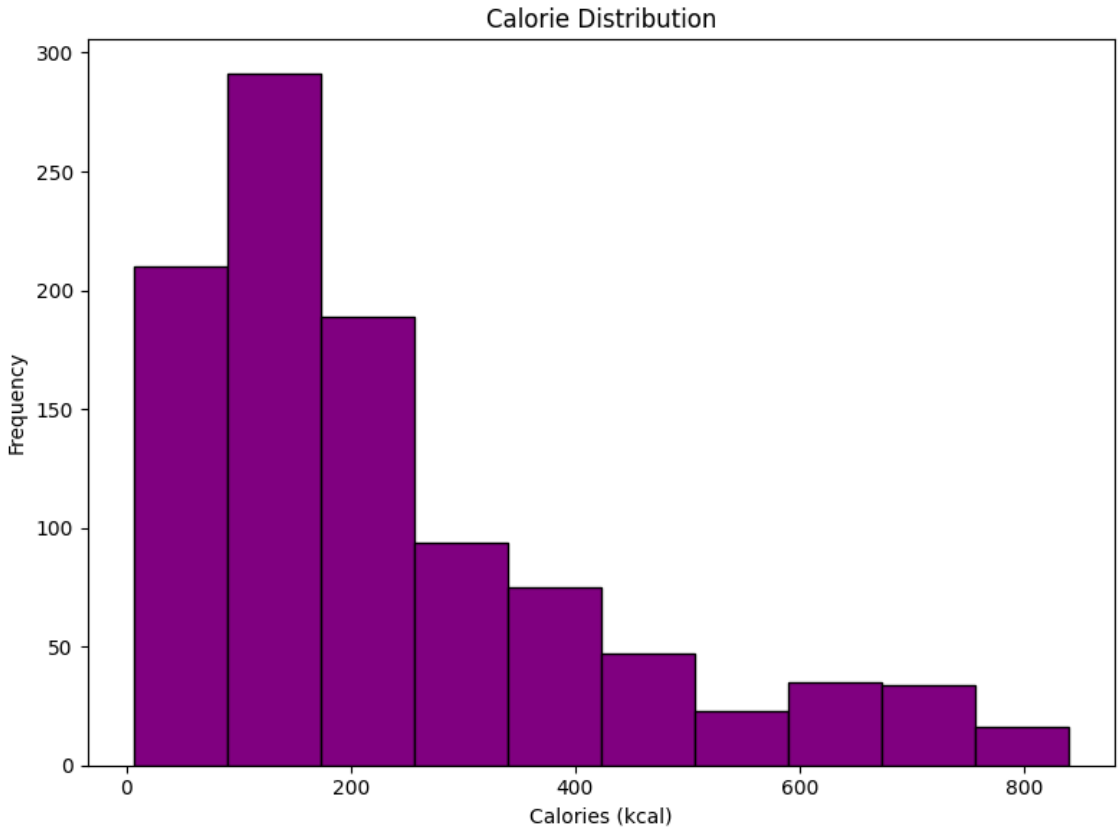
- **Pie Chart:** Top 10 calorie-dense dishes visualized.

```
top_calorie_dishes = df.nlargest(10, 'Calories (kcal)')[['Dish Name', 'Calories (kcal)']]

plt.figure(figsize=(8, 8))

plt.pie(top_calorie_dishes['Calories(kcal)'], labels=top_calorie_dishes['Dish Name'], autopct='%1.1f%%')

plt.title("Top 10 High-Calorie Indian Dishes")

plt.show()
```



- **Histograms:** Showed the distribution of **Calories (kcal)** or **Fats (g)** as the variable.

```
plt.figure(figsize=(8, 6))
plt.hist(df['Calories (kcal)'], bins=10, color='purple', edgecolor='black')
plt.title('Calorie Distribution')
plt.xlabel('Calories (kcal)')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```



✅ Conclusion About Calories for This Project

The analysis clearly shows that calories in Indian food items are primarily influenced by the fat and carbohydrate content. Dishes high in fats—such as paneer-based gravies, deep-fried snacks, and rich sweets—tend to have significantly higher calorie values.

Through visualizations like scatter plots and histograms, a strong positive correlation between fats and calories was observed. The top 10 high-calorie dishes also reaffirmed that traditional Indian foods often combine dense ingredients like oil, ghee, butter, and sugar, contributing to elevated calorie levels.

From a predictive standpoint, calories were accurately modeled using nutritional attributes such as fats, proteins, carbohydrates, and vitamins. This proves that nutritional profiling can be a reliable method for estimating calorie content, especially when actual measurements are unavailable.

Overall, the project emphasizes the need for mindful consumption of high-calorie foods and supports the development of tools for personalized dietary recommendations using calorie estimation.