McDonald's Menu Nutritional Analysis – Project

Instruction

McDonald's is a global fast-food chain known for its diverse menu offerings. As a data analyst, your task is to analyze the nutritional content of the menu items available at McDonald's outlets. This analysis will provide valuable insights into the calorie count and nutrition facts of various menu items.

Objectives:

1. Extract meaningful information from the McDonald's menu nutritional dataset.
2. Perform exploratory data analysis to understand the nutritional distribution and trends.
3. Create visualizations to present the calorie count and nutrition facts of different menu items.
4. Identify healthy and less healthy menu options based on nutritional content.

Deliverables:

A report (PDF) containing:
Description of data analysis approach and methodology.
Exploratory data analysis findings and insights.
Visualizations depicting nutritional information.
Source code used for data preprocessing, analysis, and visualization.
Tasks/Activities List:
Data Collection: Download the McDonald's menu nutritional dataset from this link.
Data Preprocessing:
Load and inspect the dataset.
Handle missing values and data cleaning if necessary.
Exploratory Data Analysis (EDA):
Analyze the distribution of calorie counts across menu items.
Explore the nutritional content (e.g., fat, protein, carbohydrates) of different items.
Identify trends and patterns in the dataset.
Data Visualization:
Create bar charts, histograms, and box plots to visualize calorie distribution and nutritional content.
Compare nutritional characteristics of different food categories (e.g., burgers, salads, desserts).
Nutrition-Based Insights:
Identify menu items with the highest and lowest calorie counts.
Determine the average nutritional content of popular menu categories.
Documentation and Reporting:
Summarize the findings and insights from the analysis.
Explain how the nutritional analysis could benefit McDonald's customers and the organization.

Task/ Activities List:

Data Collection:

- The data set was downloaded, observed the trends on nutrition contents.

Data Pre-processing:

1. Load and inspect the dataset.
2. Handle missing values and data cleaning if necessary.

- The dataset is uploaded to the Jupyter notebook, imported pandas to with the dataset, as depicted below:
  Syntax:
  import pandas as pd
  import seaborn as sns
  Nutrition = pd.read_csv('Nutrical Dataset.csv')
  Nutrition.shape
  Nutrition.info()

```
import pandas as pd
import seaborn as sns
```

```
Nutrition = pd.read_csv('Nutrical Dataset.csv')
```

```
Nutrition.head()
```

	Category	Item	Serving Size	Calories	Calories from Fat	Total Fat	Total Fat (% Daily Value)	Saturated Fat	Saturated Fat (% Daily Value)	Trans Fat	...	Carbohydrates	Carbohydrates (% Daily Value)	Dietary Fiber	Dietary Fiber (% Daily Value)	Sugars	Protein	Vi
0	Breakfast	Egg McMuffin	4.8 oz (136 g)	300	120	13.0	20	5.0	25	0.0	...	31	10	4	17	3	17	
1	Breakfast	Egg White Delight	4.8 oz (135 g)	250	70	8.0	12	3.0	15	0.0	...	30	10	4	17	3	18	
2	Breakfast	Sausage McMuffin	3.9 oz (111 g)	370	200	23.0	35	8.0	42	0.0	...	29	10	4	17	2	14	
3	Breakfast	Sausage McMuffin with Egg	5.7 oz (161 g)	450	250	28.0	43	10.0	52	0.0	...	30	10	4	17	2	21	

Exploratory Data Analysis (EDA):

1. **Analyse the distribution of calorie counts across menu items.**

- We've loaded the dataset, analysed the trends. Let's understand a statistical summary for the calories in the dataset.
  Syntax:
  print(Nutrition['Calories'].describe())

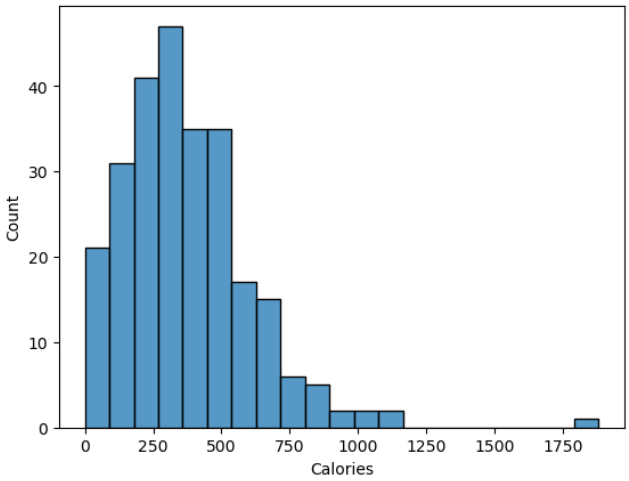
```
print(Nutrition['Calories'].describe())
```

```
count     260.000000
mean      368.269231
std       240.269886
min         0.000000
25%       210.000000
50%       340.000000
75%       500.000000
max      1880.000000
Name: Calories, dtype: float64
```

- Univariate Analysis: Analysing Calorie Distribution:
- To visualize the distribution of calories, plot a histogram.

Syntax:

sns.histplot(Nutrition['Calories'])



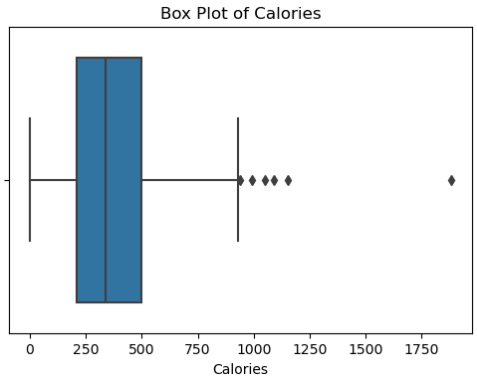
Nutrition.groupby('Item')['Calories'].describe()

Item	count	mean	std	min	25%	50%	75%	max
1% Low Fat Milk Jug	1.0	100.0	NaN	100.0	100.0	100.0	100.0	100.0
Apple Slices	1.0	15.0	NaN	15.0	15.0	15.0	15.0	15.0
Bacon Buffalo Ranch McChicken	1.0	430.0	NaN	430.0	430.0	430.0	430.0	430.0
Bacon Cheddar McChicken	1.0	480.0	NaN	480.0	480.0	480.0	480.0	480.0
Bacon Clubhouse Burger	1.0	720.0	NaN	720.0	720.0	720.0	720.0	720.0
...	...	...	...	...	...	...	...	...
Sweet Tea (Medium)	1.0	180.0	NaN	180.0	180.0	180.0	180.0	180.0
Sweet Tea (Small)	1.0	150.0	NaN	150.0	150.0	150.0	150.0	150.0
Vanilla Shake (Large)	1.0	820.0	NaN	820.0	820.0	820.0	820.0	820.0
Vanilla Shake (Medium)	1.0	660.0	NaN	660.0	660.0	660.0	660.0	660.0
Vanilla Shake (Small)	1.0	530.0	NaN	530.0	530.0	530.0	530.0	530.0

- To understand the spread of calorie values, we go with Box plot:

Syntax:

import matplotlib.pyplot as plt
plt.figure(figsize=(6,4))
sns.boxplot(x=Nutrition['Calories'])
plt.title('Box Plot of Calories')
plt.show()



- To understand the unique items in the “Items” columns, we perform the following syntax:

  Print(‘The unique items are: ’, Nutrition[‘Items’].unique())
- We get almost all the items present, as the dataset consist of unique items vs calories values.
- Items, with the maximum calories:

  Syntax:

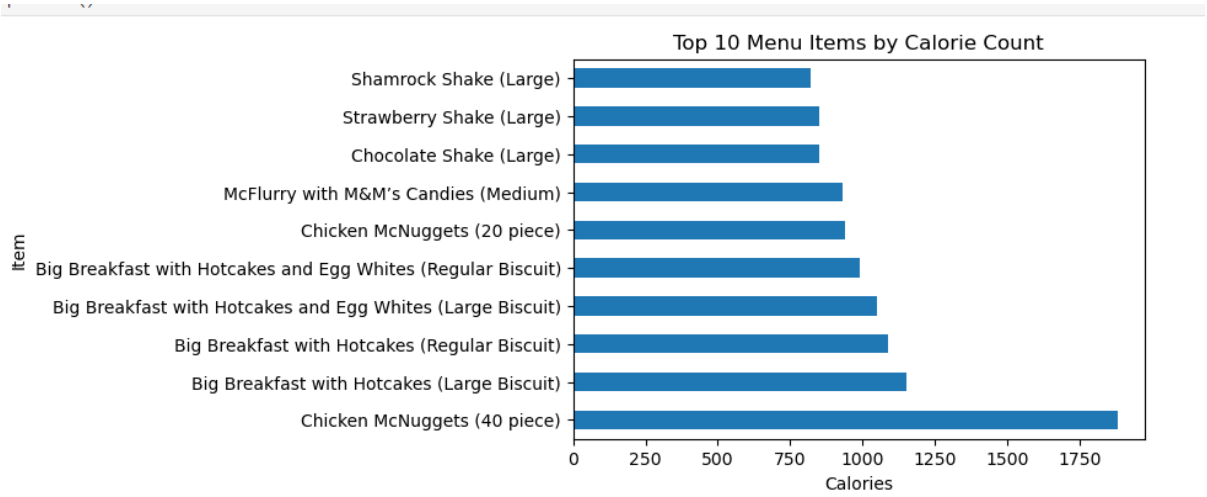
  top_items = Nutrition.nlargest(10, 'Calories')
  print(top_items[['Item', 'Calories']])

	Item	Calories
82	Chicken McNuggets (40 piece)	1880
32	Big Breakfast with Hotcakes (Large Biscuit)	1150
31	Big Breakfast with Hotcakes (Regular Biscuit)	1090
34	Big Breakfast with Hotcakes and Egg Whites (La...	1050
33	Big Breakfast with Hotcakes and Egg Whites (Re...	990
81	Chicken McNuggets (20 piece)	940
253	McFlurry with M&M’s Candies (Medium)	930
246	Strawberry Shake (Large)	850
249	Chocolate Shake (Large)	850
243	Vanilla Shake (Large)	820

- Bivariate analysis: Relationship between “Items” and “Calories”: Creating a bar plot of Items and avg. calorie count.

  Syntax:

  plt.figure(figsize=(8,4))
  top_items = Nutrition.groupby('Item')['Calories'].mean().nlargest(10)
  top_items.plot(kind='barh')
  plt.title('Top 10 Menu Items by Calorie Count')
  plt.xlabel('Calories')
  plt.ylabel('Item')
  plt.show()



- Checking for any extremely high calories rates, so to consider Outliers. To retrieve the Z-scores we perform the following.

  Syntax:

  from scipy import stats
  z_scores = stats.zscore(Nutrition['Calories'])
  outliers = Nutrition[abs(z_scores) > 3] # Items with z-scores greater than 3
  print(outliers[['Item', 'Calories']])

	Item	Calories
31	Big Breakfast with Hotcakes (Regular Biscuit)	1090
32	Big Breakfast with Hotcakes (Large Biscuit)	1150
82	Chicken McNuggets (40 piece)	1880

- **Insights:**
  1. Calorie Distribution: For the given dataset, the skewness of calories is said to be 1.452 which indicates positive skewness (right skewness). Most of the calories are concentrated on the lower end. The distribution is asymmetry.
  2. Skewness greater than 1 suggests a highly skewed distribution. There are few high-calorie items comparatively.

     Syntax:

     skew_calories = Nutrition['Calories'].skew()
     print("Skewness of Calories:", skew_calories)
     O/P : Skewness of Calories: 1.4524981730922621

  3. The top menu items with the highest calories rates are Chicken McNuggests, Big breakfast with Hotcakes – regular and large biscuits.
  4. The menu with the lowest calorie rate is Diet Coke (Small).

     Syntax:

     min_calorie_item = Nutrition.loc[Nutrition['Calories'].idxmin(), 'Item']
     print("Item with the lowest calorie:", min_calorie_item)

Explore the nutritional content of different items.

- Let's understand a statistical summary for the proteins, cholesterol and sugars in the dataset.

  Syntax:

  print(Nutrition[['Cholesterol', 'Protein', 'Sugars']].describe())

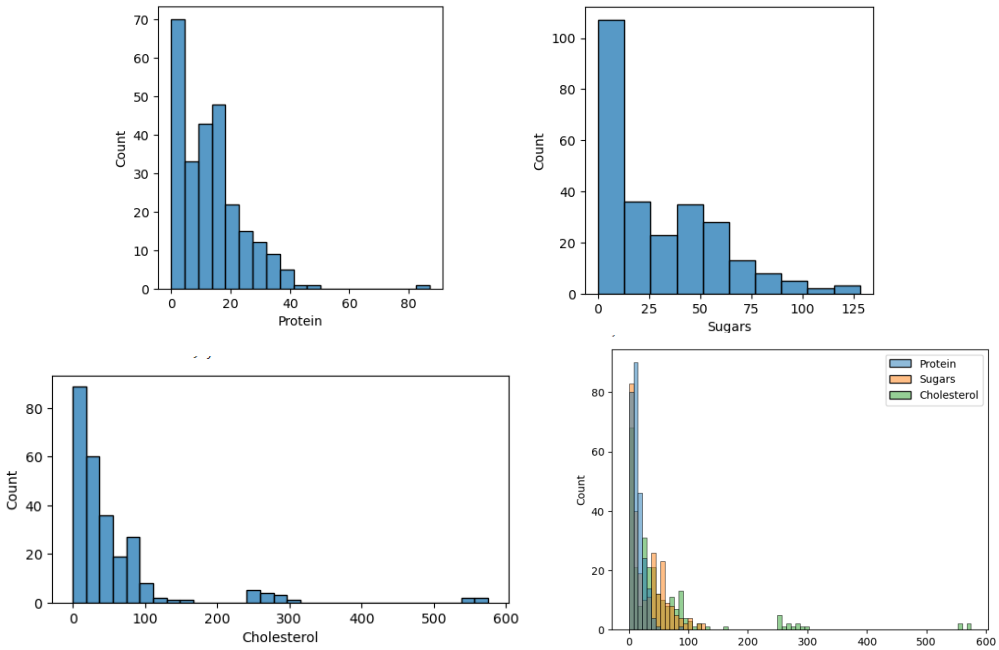
	Cholesterol	Protein	Sugars
count	260.000000	260.000000	260.000000
mean	54.942308	13.338462	29.423077
std	87.269257	11.426146	28.679797
min	0.000000	0.000000	0.000000
25%	5.000000	4.000000	5.750000
50%	35.000000	12.000000	17.500000
75%	65.000000	19.000000	48.000000
max	575.000000	87.000000	128.000000

- Univariate Analysis: Analysing Calorie Distribution:
- To visualize the distribution of calories, plot a histogram.

  Syntax:

  plt.figure(figsize=(4,4))
  sns.histplot(Nutrition['Protein'])
  plt.figure(figsize=(4,4))
  sns.histplot(Nutrition['Sugars'])
  plt.figure(figsize=(6,3))
  sns.histplot(Nutrition['Cholesterol'])

sns.histplot(Nutrition[['Protein', 'Sugars', 'Cholesterol']])



- Let's describe Cholesterol, Proteins and sugar in the menu.

  Syntax

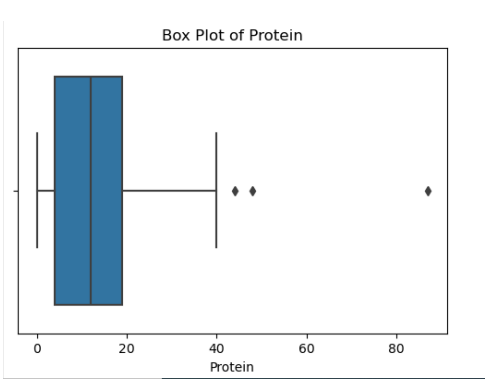
  Nutrition.groupby('Item')[['Sugars', 'Protein', 'Cholesterol']].describe()

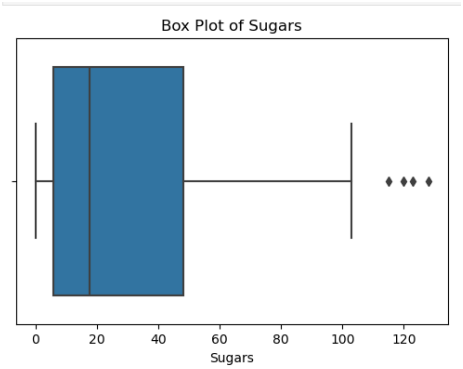
	Sugars								Protein										Cholesterol							
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max	count	mean	std	min	25%	50%	75%	max
Item																										
1% Low Fat Milk Jug	1.0	12.0	NaN	12.0	12.0	12.0	12.0	12.0	1.0	8.0	...	8.0	8.0	1.0	10.0	NaN	10.0	10.0	10.0	10.0	10.0
Apple Slices	1.0	3.0	NaN	3.0	3.0	3.0	3.0	3.0	1.0	0.0	...	0.0	0.0	1.0	0.0	NaN	0.0	0.0	0.0	0.0	0.0
Bacon Buffalo Ranch McChicken	1.0	6.0	NaN	6.0	6.0	6.0	6.0	6.0	1.0	20.0	...	20.0	20.0	1.0	50.0	NaN	50.0	50.0	50.0	50.0	50.0
Bacon Cheddar McChicken	1.0	6.0	NaN	6.0	6.0	6.0	6.0	6.0	1.0	22.0	...	22.0	22.0	1.0	65.0	NaN	65.0	65.0	65.0	65.0	65.0
Bacon Clubhouse Burger	1.0	14.0	NaN	14.0	14.0	14.0	14.0	14.0	1.0	39.0	...	39.0	39.0	1.0	115.0	NaN	115.0	115.0	115.0	115.0	115.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Sweet Tea (Medium)	1.0	45.0	NaN	45.0	45.0	45.0	45.0	45.0	1.0	1.0	...	1.0	1.0	1.0	0.0	NaN	0.0	0.0	0.0	0.0	0.0
Sweet Tea (Small)	1.0	36.0	NaN	36.0	36.0	36.0	36.0	36.0	1.0	1.0	...	1.0	1.0	1.0	0.0	NaN	0.0	0.0	0.0	0.0	0.0
Vanilla Shake (Large)	1.0	101.0	NaN	101.0	101.0	101.0	101.0	101.0	1.0	18.0	...	18.0	18.0	1.0	90.0	NaN	90.0	90.0	90.0	90.0	90.0
Vanilla Shake (Medium)	1.0	81.0	NaN	81.0	81.0	81.0	81.0	81.0	1.0	14.0	...	14.0	14.0	1.0	75.0	NaN	75.0	75.0	75.0	75.0	75.0
Vanilla Shake (Small)	1.0	63.0	NaN	63.0	63.0	63.0	63.0	63.0	1.0	11.0	...	11.0	11.0	1.0	60.0	NaN	60.0	60.0	60.0	60.0	60.0

- To understand the spread of calorie values, we go with Box plot:

  Syntax:

  import matplotlib.pyplot as plt
  plt.figure(figsize=(6,4))
  sns.boxplot(x=Nutrition['x'])
  plt.title('Box Plot of Calories')
  plt.show()





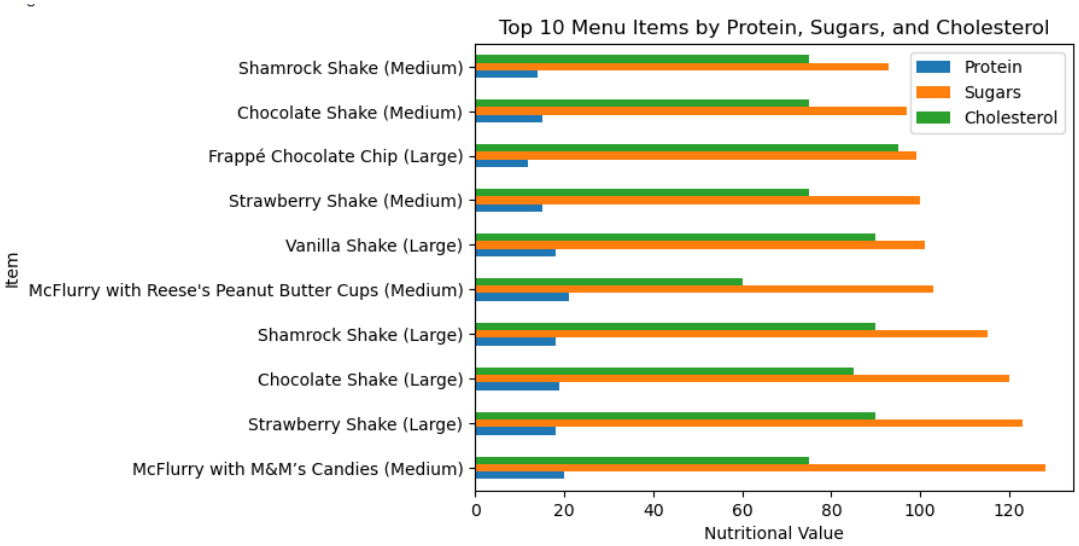
- Let's understand the top items with high Protein, Sugars and cholesterol levels:
  Syntax:

  top_items = Nutrition.nlargest(10, ['Protein', 'Sugars', 'Cholesterol'])
  print(top_items[['Item', 'Protein', 'Sugars', 'Cholesterol']])

	Item	Protein	Cholesterol	Sugars
82	Chicken McNuggets (40 piece)	87	265	1
47	Double Quarter Pounder with Cheese	48	160	10
81	Chicken McNuggets (20 piece)	44	135	0
64	Bacon Clubhouse Grilled Chicken Sandwich	40	110	14
60	Premium Grilled Chicken Club Sandwich	40	105	9
51	Bacon Clubhouse Burger	39	115	14
44	Quarter Pounder with Bacon & Cheese	37	105	12
45	Quarter Pounder with Bacon Habanero Ranch	37	105	10
31	Big Breakfast with Hotcakes (Regular Biscuit)	36	575	17
32	Big Breakfast with Hotcakes (Large Biscuit)	36	575	17

- Bivariate analysis: To understand the relationship between 'Items' and "Cholesterol, Proteins and Sugars"
  Syntax:
  plt.figure(figsize=(8,6))

  top_items = Nutrition.groupby('Item')[['Protein', 'Sugars', 'Cholesterol']].mean().nlargest(10, 'Sugars')
  top_items.plot(kind='barh', stacked=False)
  plt.title('Top 10 Menu Items by Protein, Sugars, and Cholesterol')
  plt.xlabel('Nutritional Value')
  plt.ylabel('Item')
  plt.show()



- Checking for any extremely high Sugar, Protein and cholesterol rates, so to consider Outliers. To retrieve the Z-scores we perform the following.
- *Items having z-score > 3:*
  Syntax:

```
from scipy import stats
columns = ['Protein', 'Sugars', 'Cholesterol']
z_scores = Nutrition[columns].apply(stats.zscore)
outliers = Nutrition[(z_scores.abs() > 3).any(axis=1)]
print(outliers[['Item', 'Protein', 'Sugars', 'Cholesterol']])
```

```
                                          Item  Protein  Sugars  \
27                Big Breakfast (Regular Biscuit)       28       3
28                  Big Breakfast (Large Biscuit)       28       3
31   Big Breakfast with Hotcakes (Regular Biscuit)       36      17
32     Big Breakfast with Hotcakes (Large Biscuit)       36      17
47              Double Quarter Pounder with Cheese       48      10
82                    Chicken McNuggets (40 piece)       87       1
246                     Strawberry Shake (Large)       18     123
249                      Chocolate Shake (Large)       19     120
253           McFlurry with M&M's Candies (Medium)       20     128

     Cholesterol
27           555
28           555
31           575
32           575
47           160
82           265
246           90
249           85
253           75
```

- **<u>Insights:</u>**
  1. Nutrients Distribution: For the given dataset, the skewness of Protein is 1.57, Sugars is 1.03 and for Cholesterol is 3.79.
  2. The skewness follows the pattern: Sugars< Protein < Cholesterol, being highly skewed. The distribution is asymmetry.
  3. There are few items with high cholesterol, making the dataset to be highly- right skewed.

     <u>Syntax:</u>

```
columns = ['Protein', 'Sugars', 'Cholesterol']
skew_Nutrition = Nutrition[columns].skew()
print("The skewness for :", skew_Nutrition)
```

O/P : The skewness for : Protein 1.579924
Sugars 1.031940
Cholesterol 3.798907

4. The menu with the better healthy nutrients having less sugar, cholesterol and proteins are:

Syntax:

columns = ['Cholesterol', 'Protein', 'Sugars']
min_calorie_item = Nutrition.loc[Nutrition[columns].idxmin(), 'Item']
print("Item with the better health ratio:", min_calorie_item)
O/P: Item with the better health ratio:
38 Hash Brown
101 Apple Slices
38 Hash Brown

Nutrition-Based Insights:

1. **Identify menu items with the highest and lowest calorie counts.**
   - The top menu items with the highest calories rates are Chicken McNuggests, Big breakfast with Hotcakes – regular and large biscuits.
   - The menu with the lowest calorie rate is Diet Coke (Small).
2. **Determine the average nutritional content of popular menu categories.**

Syntax:

average_nutrition = Nutrition.groupby('Item')[['Calories', 'Sugars', 'Cholesterol', 'Total Fat', 'Carbohydrates (% Daily Value)', 'Sodium']].mean()
popular_items_avg = average_nutrition.loc[['Hamburger', 'Premium Crispy Chicken Classic Sandwich', 'Baked Apple Pie']]
print(popular_items_avg)

```
                                  Calories  Sugars  Cholesterol  \
Item
Hamburger                            240.0     6.0         30.0
Premium Crispy Chicken Classic Sandwich  510.0    10.0         45.0
Baked Apple Pie                      250.0    13.0          0.0

                                  Total Fat  \
Item
Hamburger                              8.0
Premium Crispy Chicken Classic Sandwich       22.0
Baked Apple Pie                       13.0

                                  Carbohydrates (% Daily Value)  Sodium
Item
Hamburger                                                 11.0   480.0
Premium Crispy Chicken Classic Sandwich                   18.0   990.0
Baked Apple Pie                                           11.0   170.0
```

DOCUMENTATION:

- From the above dataset, we found Chicken Nuggets possess high calorie counts and diet coke holds the least amount. One who watches their diet should also observe the nutrient contents. Thanks to EDA.
- Calories and Cholesterol levels are highly right-skewed, resulting that certain items possess hold higher level of such nutrients – which aren't good for health concerns.
- One to enjoy such a meal, they can hold a better frequency on consuming meals from McDonald's between intervals, possibly monthly once.
- If customers are following this trend, it would be a huge decline on the revenue of the restaurant. Hence, it'd be high time for the restaurant to concentrate on the nutrient values and provide better meal to their customers.
- The restaurant should primarily focus on calorie and cholesterol contents in their dishes, resulting in a healthy meal and increasing better revenues.