OBESITY RISK PREDICTION

EXPLORATORY DATA ANALYSIS

Question:

- **How much impact do eating habits and physical conditions have on obesity?**
    - To answer this question, I had to look at the dataset that I chose to give me some quantitative data on each factor and their relation to obesity and potentially predict the level each factor adds to obesity.

Data selection

- **Why did I choose this dataset and what is the purpose of this analysis?**
    - I chose to use the Obesity Risk Prediction dataset to figure out what factors influence obesity and at what level which I can use to predict its occurrence and potentially its levels. Its purpose is to Estimate Obesity Levels Based On Eating Habits and Physical Condition

Dataset

- **About the dataset**
    - The dataset includes data for the estimation of obesity levels in individuals from the countries of Mexico, Peru, and Colombia, based on their eating habits and physical condition.
- **Extracting the data**
    - For our dataset, I had to extract it to be able to use it in my notebook for any analysis that I was to do. To do so, I had to install some packages that helped me retrieve the data and then retrieve the data:

```
[2] ## import the necessary packages
    import pandas as pd
    import numpy as np
    import matplotlib.pylab as plt
    import seaborn as sns
```

```
[3] ## read the CSV file
    df = pd.read_csv('https://raw.githubusercontent.com/kizzanaome/Obesity-Risk-Prediction/main/ObesityDataSet_raw_and
```

- **Previewing the data**
    - To show the contents of the data, I needed to call the `df` function so that the data is displayed graphically. Here is what I run to achieve that:

```
[4]  ## show the extracted data
     df
```

- with that, I was able to show a tabular format of the dataset and show its rows and columns with the values
- From the previewing of the data, I established that the dataset contains 17 attributes and 2111 records, the records are labeled with the class variable NObeyesdad (Obesity Level), which allows classification of the data using the values of Insufficient Weight, Normal Weight, Overweight Level I, Overweight Level II, Obesity Type I, Obesity Type II and Obesity Type III.

- **Understanding the data**
  - To understand the data in depth, I showed a description of each column in the dataset as follows:

    - **Gender** - Biological sex of the individual (Male or Female)
    - **Age** - How old the individual is
    - **Height** - How tall the individual is
    - **Weight** - How light or heavy the individual is
    - **family_history_with_overweight** - Has a family member suffered or suffers from overweight?
    - **FAVC** - Do you eat high caloric food frequently?
    - **FCVC** - Do you usually eat vegetables in your meals?
    - **NCP** - How many main meals do you have daily?
    - **CAEC** - Do you eat any food between meals?
    - **SMOKE** - Do you smoke?
    - **CH20** - How much water do you drink daily?
    - **SCC** - Do you monitor the calories you eat daily?
    - **FAF** - How often do you have physical activity?
    - **TUE** - How much time do you use technological devices such as cell phone, videogames, television, computer and others?
    - **CALC** - How often do you drink alcohol?
    - **MTRANS** - Which transportation do you usually use?
    - **NObeyesdad** - Obesity level

  - I continued to perform a set of queries on the data to get a more in-depth understanding of its structure. These queries I performed were:

```
## show the number of rows and columns
df.shape()
## return the first five records
df.head()
## return the last five records
df.tail()
## show the datatype of each column
df.dtypes()
## show the standard calculations for any numerical values
df.describe()
## show the non-null count and datatype for each column
df.info()
```

  - I used `df.shape` to show the total rows and total columns
  - I used `df.head` to return the first five records
  - I used `df.tail` to view the last five records
  - I used `df.dtypes` to figure out the datatypes of each column

- I used `df.describe` to show the standard calculations for any numerical values
- I used `df.info` to show the non-null count and datatype for each column
- From the above queries, I deduced that:
  - There are 2111 entries, i.e. 2111 rows
  - Each row has a row label (which is the index) with values ranging from 0 to 2110
  - The table has 17 columns, all having a value for each of the rows (all 2111 values are non-null)
  - The columns `Gender`, `family_history_with_overweight`, `FAVC`, `FCVC`, `CAEC`, `SMOKE`, `SCC`, `CALC`, `MTRANS` and `NObeyesdad` consist of textual data (strings which are also known as objects) and the other columns are numerical data with real numbers (also known as float)
  - There are 8 float data types and 9 object data types

Data wrangling

- To perform any kind of analysis and also figure out what factors from my dataset greatly affect obseity, I had to first make sure that my data is clean enough to be used
- I checked if there are any missing values by running the command:

```
## identify missing values and how often they occur in our dataset
df.isnull().sum()
```

- From my findings, I had no missing data as there were no null values, 0 values, or missing values
- Next I checked if there are any white spaces in the column titles with this command

```
[7] ## show the column names
    df.columns
```

```
Index(['Gender', 'Age', 'Height', 'Weight', 'family_history_with_overweight',
       'FAVC', 'FCVC', 'NCP', 'CAEC', 'SMOKE', 'CH2O', 'SCC', 'FAF', 'TUE',
       'CALC', 'MTRANS', 'NObeyesdad'],
      dtype='object')
```

- From the above I established that the columns need no cleaning or remove of white spaces
- Next, I checked for any duplicate values from the dataset and I did that by running the command:

```
# find duplicated rows
df.loc[df.duplicated()]
```

- From this, I derived that the data has 24 duplicated columns that might need cleaning