Assignment 001: **Machine Learning**

Student Name: **Ndibalekera Diana Rhoda**

RegNo: **2024/HD05/21951U**

Dataset Title

National Health and Nutrition Health Survey 2013-2014 (NHANES) Age Prediction Subset

Donated on 9/21/2023

About

The National Health and Nutrition Examination Survey (NHANES), administered by the Centers for Disease Control and Prevention (CDC), collects extensive health and nutritional information from a diverse U.S. population. Though expansive, the dataset is often too broad for specific analytical purposes. In this sub-dataset, their focus was narrowed to predicting respondents' age by extracting a subset of features from the larger NHANES dataset. The selected features included physiological measurements, lifestyle choices, and biochemical markers, which were hypothesized to have strong correlations with age.

Variables Table

Variable Name	Role	Type	Demographic	Description	Units	Missing Values
SEQN	ID	Continuous		Respondent Sequence Number		no
age_group	Target	Categorical	Age	Respondent's Age Group (senior/non-senior)		no
RIDAGEYR	Other	Continuous	Age	Respondent's Age		no
RIAGENDR	Feature	Continuous	Gender	Respondent's Gender		no
PAQ605	Feature	Continuous		If the respondent engages in moderate or vigorous-intensity sports, fitness, or recreational activities in the typical week		no
BMXBMI	Feature	Continuous		Respondent's Body Mass Index		no
LBXGLU	Feature	Continuous		Respondent's Blood Glucose after fasting		no
DIQ010	Feature	Continuous		If the Respondent is diabetic		no
LBXGLT	Feature	Continuous		Respondent's Oral		no
LBXIN	Feature	Continuous		Respondent's Blood Insulin Levels		no

Colab Findings

Interactions started with library importations to colab.And then proceded to calling the dataset csv file which I uploaded to google colab

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

[ ] df = pd.read_csv('/content/national+health+and+nutrition+health+survey+2013-2014+(nhanes)+age+prediction+subset.zip')
```

Data Wrangling

Firstly, were my interactions with the Colab. Under this, I was looking for duplicates in the data, however I there was none. Also checked for the uniqueness of the data and finally integrated one of the columns with the data type of object and converted it to Boolean. Thus, found out that I had to maintain my entire dataset after finding out that all data columns count.

```
#columns and rows
df.shape
```

```
(2278, 10)
```

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2278 entries, 0 to 2277
Data columns (total 10 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   SEQN       2278 non-null   float64
 1   age_group  2278 non-null   object 
 2   RIDAGEYR   2278 non-null   float64
 3   RIAGENDR   2278 non-null   float64
 4   PAQ605     2278 non-null   float64
 5   BMXBMI     2278 non-null   float64
 6   LBXGLU     2278 non-null   float64
 7   DIQ010     2278 non-null   float64
 8   LBXGLT     2278 non-null   float64
 9   LBXIN      2278 non-null   float64
dtypes: float64(9), object(1)
memory usage: 178.1+ KB
```

```
[ ] #data uniqueness
    df.nunique()
```

	0
SEQN	2278
age_group	2
RIDAGEYR	69
RIAGENDR	2
PAQ605	3
BMXBMI	340
LBXGLU	101
DIQ010	3
LBXGLT	232
LBXIN	1424

dtype: int64

```
[ ] df.describe().T
```

	count	mean	std	min	25%	50%	75%	
SEQN	2278.0	78691.853819	2921.365151	73564.00	76171.75	78749.00	81213.50	83727
RIDAGEYR	2278.0	41.795874	20.156111	12.00	24.00	41.00	58.00	80
RIAGENDR	2278.0	1.511414	0.499979	1.00	1.00	2.00	2.00	2
PAQ605	2278.0	1.822651	0.398918	1.00	2.00	2.00	2.00	7
BMXBMI	2278.0	27.955180	7.248962	14.50	22.80	26.80	31.20	70
LBXGLU	2278.0	99.553117	17.889834	63.00	91.00	97.00	104.00	405
DIQ010	2278.0	2.016242	0.185556	1.00	2.00	2.00	2.00	3
LBXGLT	2278.0	114.978929	47.061239	40.00	87.00	105.00	130.00	604
LBXIN	2278.0	11.834794	9.718812	0.14	5.86	9.04	14.44	102

```
[ ] df.duplicated().sum()
```

```
0
```

```
[ ] #Gabbage
    for i in df.select_dtypes(include='object').columns:
      print(df[i].value_counts())
      print("***"*10)
```

```
age_group
Adult     1914
Senior     364
Name: count, dtype: int64
******************************
```

```
[ ] pd.get_dummies(df,columns=['age_group'], drop_first=True)
```

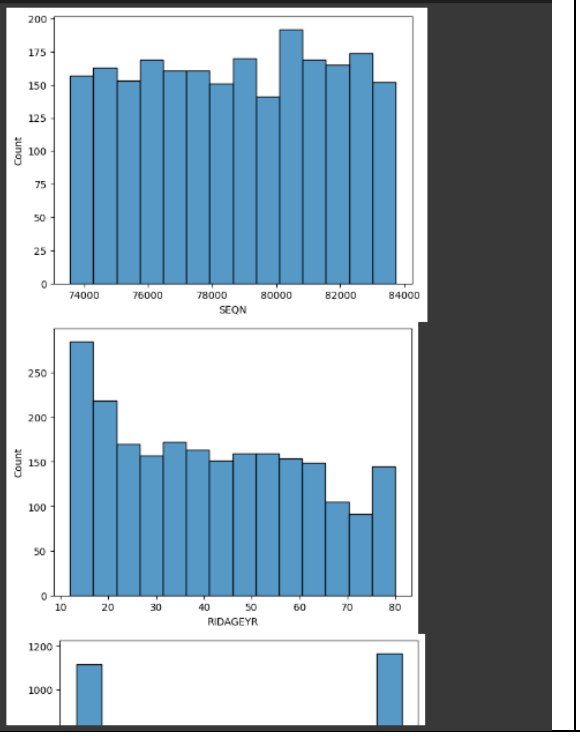
	SEQN	RIDAGEYR	RIAGENDR	PAQ605	BMXBMI	LBXGLU	DIQ010	LBXGLT	LBXIN	age_group_Senior
0	73564.0	61.0	2.0	2.0	35.7	110.0	2.0	150.0	14.91	False
1	73568.0	26.0	2.0	2.0	20.3	89.0	2.0	80.0	3.85	False
2	73576.0	16.0	1.0	2.0	23.2	89.0	2.0	68.0	6.14	False
3	73577.0	32.0	1.0	2.0	28.9	104.0	2.0	84.0	16.15	False
4	73580.0	38.0	2.0	1.0	35.9	103.0	2.0	81.0	10.92	False
...	...	...	...	...	...	...	...	...	...	...
2273	83711.0	38.0	2.0	2.0	33.5	100.0	2.0	73.0	6.53	False
2274	83712.0	61.0	1.0	2.0	30.0	93.0	2.0	208.0	13.02	False
2275	83713.0	34.0	1.0	2.0	23.7	103.0	2.0	124.0	21.41	False
2276	83718.0	60.0	2.0	2.0	27.4	90.0	2.0	108.0	4.99	False
2277	83727.0	26.0	1.0	2.0	24.5	108.0	2.0	108.0	3.76	False

2278 rows × 10 columns

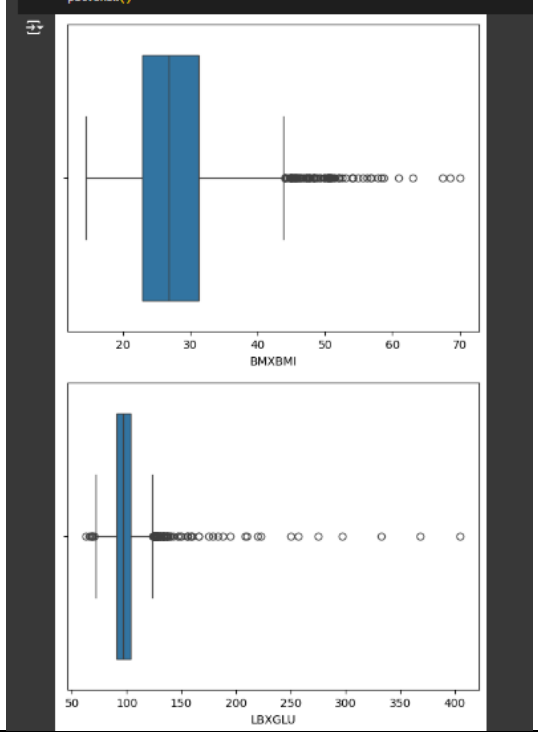
EDA Representations.

I plotted graphs to visualize the data, look for outliers, and look at their relationships.

```
[ ] #Visualization
for i in df.select_dtypes(include='number').columns:
  sns.histplot(data=df, x=i)
  plt.show()
```




```
[ ] #outliers
for i in df.select_dtypes(include='number').columns:
  sns.boxplot(data=df, x=i)
  plt.show()
```



Conclusion

Data wrangling is essential as it prepares the dataset for analysis by effectively cleaning the data, checking for consistencies, transforming and integrating the data as well as conducting exploratory analysis thus enhancing data quality and ensure accurate modeling leading to reliable findings and better understanding of the dataset.