final_project_obesity

April 7, 2024

0.0.1 For this project, I used the obesity dataset from Kaggle: https://www.kaggle.com/datasets/aravindpcoder/obesity-or-cvd-risk-classifyregressorcluster

[1]:
```python
import pandas as pd

# reading in the obesity dataset
obesity = pd.read_csv("~/obesity.csv")
obesity
```

[1]:
```
      Gender        Age    Height      Weight family_history_with_overweight  \
0     Female  21.000000  1.620000   64.000000                            yes
1     Female  21.000000  1.520000   56.000000                            yes
2       Male  23.000000  1.800000   77.000000                            yes
3       Male  27.000000  1.800000   87.000000                             no
4       Male  22.000000  1.780000   89.800000                             no
…        …          …         …           …                              …
2106  Female  20.976842  1.710730  131.408528                            yes
2107  Female  21.982942  1.748584  133.742943                            yes
2108  Female  22.524036  1.752206  133.689352                            yes
2109  Female  24.361936  1.739450  133.346641                            yes
2110  Female  23.664709  1.738836  133.472641                            yes

     FAVC  FCVC  NCP       CAEC SMOKE      CH2O  SCC       FAF       TUE  \
0      no   2.0  3.0  Sometimes    no  2.000000   no  0.000000  1.000000
1      no   3.0  3.0  Sometimes   yes  3.000000  yes  3.000000  0.000000
2      no   2.0  3.0  Sometimes    no  2.000000   no  2.000000  1.000000
3      no   3.0  3.0  Sometimes    no  2.000000   no  2.000000  0.000000
4      no   2.0  1.0  Sometimes    no  2.000000   no  0.000000  0.000000
…     …    …    …          …     …         …    …
2106  yes   3.0  3.0  Sometimes    no  1.728139   no  1.676269  0.906247
2107  yes   3.0  3.0  Sometimes    no  2.005130   no  1.341390  0.599270
2108  yes   3.0  3.0  Sometimes    no  2.054193   no  1.414209  0.646288
2109  yes   3.0  3.0  Sometimes    no  2.852339   no  1.139107  0.586035
2110  yes   3.0  3.0  Sometimes    no  2.863513   no  1.026452  0.714137

            CALC                 MTRANS           NObeyesdad
0             no  Public_Transportation        Normal_Weight
```

```
1       Sometimes  Public_Transportation        Normal_Weight
2      Frequently  Public_Transportation        Normal_Weight
3      Frequently                Walking   Overweight_Level_I
4       Sometimes  Public_Transportation  Overweight_Level_II
…             …                      …                    …
2106    Sometimes  Public_Transportation     Obesity_Type_III
2107    Sometimes  Public_Transportation     Obesity_Type_III
2108    Sometimes  Public_Transportation     Obesity_Type_III
2109    Sometimes  Public_Transportation     Obesity_Type_III
2110    Sometimes  Public_Transportation     Obesity_Type_III

[2111 rows x 17 columns]
```

0.1 Step 1: Data Cleaning

[2]:
```
obesity.shape # 2,111 observations and 17 columns
```

[2]:
```
(2111, 17)
```

[3]:
```
# renaming the columns for convenience
obesity.columns = ['gender', 'age', 'height', 'weight', 'overweight_family',
                   'freq_high_calorie_food', 'freq_vegetable',␣
 ↪'num_main_meals', 'consume_food_btwn_meals',
                  'smoke', 'daily_h2o_consumption', 'calorie_monitoring',␣
 ↪'freq_physical_activity',
                  'time_used_technology', 'freq_alcohol_consumption',␣
 ↪'mode_transportation', 'obesity_level']
```

[4]:
```
obesity.head() # checking first few rows of the dataset
```

[4]:
```
   gender   age  height  weight overweight_family freq_high_calorie_food  \
0  Female  21.0    1.62    64.0               yes                     no
1  Female  21.0    1.52    56.0               yes                     no
2    Male  23.0    1.80    77.0               yes                     no
3    Male  27.0    1.80    87.0                no                     no
4    Male  22.0    1.78    89.8                no                     no

   freq_vegetable  num_main_meals consume_food_btwn_meals smoke  \
0             2.0             3.0               Sometimes    no
1             3.0             3.0               Sometimes   yes
2             2.0             3.0               Sometimes    no
3             3.0             3.0               Sometimes    no
4             2.0             1.0               Sometimes    no

   daily_h2o_consumption calorie_monitoring  freq_physical_activity  \
0                    2.0                 no                     0.0
1                    3.0                yes                     3.0
```

```
2                     2.0                 no                      2.0
3                     2.0                 no                      2.0
4                     2.0                 no                      0.0

   time_used_technology freq_alcohol_consumption    mode_transportation  \
0                   1.0                       no  Public_Transportation
1                   0.0                Sometimes  Public_Transportation
2                   1.0               Frequently  Public_Transportation
3                   0.0               Frequently                Walking
4                   0.0                Sometimes  Public_Transportation

         obesity_level
0        Normal_Weight
1        Normal_Weight
2        Normal_Weight
3   Overweight_Level_I
4  Overweight_Level_II
```

[5]:
```
# checking dataset for missing values
obesity.isnull().sum()
# There are no missing values
```

[5]:
```
gender                      0
age                         0
height                      0
weight                      0
overweight_family           0
freq_high_calorie_food      0
freq_vegetable              0
num_main_meals              0
consume_food_btwn_meals     0
smoke                       0
daily_h2o_consumption       0
calorie_monitoring          0
freq_physical_activity      0
time_used_technology        0
freq_alcohol_consumption    0
mode_transportation         0
obesity_level               0
dtype: int64
```

[6]:
```
# checking all of the levels in the obesity_level variable
obesity['obesity_level'].unique()
```

[6]:
```
array(['Normal_Weight', 'Overweight_Level_I', 'Overweight_Level_II',
       'Obesity_Type_I', 'Insufficient_Weight', 'Obesity_Type_II',
       'Obesity_Type_III'], dtype=object)
```

[7]:
```python
# OBESITY LEVEL VARIABLE
# Here, I define a function to map the levels "Normal Weight" and "Insufficient␣
↪Weight" to 0
# and the rest of the levels are mapped to 1
def map_obesity_level(obesity_level):
    if obesity_level in ['Normal_Weight', 'Insufficient_Weight']:
        return 0
    else:
        return 1

# Applying the map_obesity_level function to the 'obesity_level'
# column and create a new column called 'obesity_binary'
obesity['obesity_binary'] = obesity['obesity_level'].apply(map_obesity_level)
```

[8]:
```python
# OVERWEIGHT FAMILY VARIABLE
# Mapping 'yes' to 1 and 'no' to 0
obesity['overweight_family_binary'] = obesity['overweight_family'].map({'yes':␣
↪1, 'no': 0})
obesity
```

[8]:
```
      gender        age    height      weight overweight_family  \
0     Female  21.000000  1.620000   64.000000               yes
1     Female  21.000000  1.520000   56.000000               yes
2       Male  23.000000  1.800000   77.000000               yes
3       Male  27.000000  1.800000   87.000000                no
4       Male  22.000000  1.780000   89.800000                no
…        …          …         …          …                 …
2106  Female  20.976842  1.710730  131.408528               yes
2107  Female  21.982942  1.748584  133.742943               yes
2108  Female  22.524036  1.752206  133.689352               yes
2109  Female  24.361936  1.739450  133.346641               yes
2110  Female  23.664709  1.738836  133.472641               yes

     freq_high_calorie_food  freq_vegetable  num_main_meals  \
0                        no             2.0             3.0
1                        no             3.0             3.0
2                        no             2.0             3.0
3                        no             3.0             3.0
4                        no             2.0             1.0
…                       …              …               …
2106                    yes             3.0             3.0
2107                    yes             3.0             3.0
2108                    yes             3.0             3.0
2109                    yes             3.0             3.0
2110                    yes             3.0             3.0

     consume_food_btwn_meals smoke  daily_h2o_consumption calorie_monitoring  \
```

```
0             Sometimes   no            2.000000             no
1             Sometimes  yes            3.000000            yes
2             Sometimes   no            2.000000             no
3             Sometimes   no            2.000000             no
4             Sometimes   no            2.000000             no
…                     …    …                 …                …
2106          Sometimes   no            1.728139             no
2107          Sometimes   no            2.005130             no
2108          Sometimes   no            2.054193             no
2109          Sometimes   no            2.852339             no
2110          Sometimes   no            2.863513             no

      freq_physical_activity  time_used_technology freq_alcohol_consumption  \
0                   0.000000              1.000000                       no
1                   3.000000              0.000000                Sometimes
2                   2.000000              1.000000               Frequently
3                   2.000000              0.000000               Frequently
4                   0.000000              0.000000                Sometimes
…                        …                    …                        …
2106                1.676269              0.906247                Sometimes
2107                1.341390              0.599270                Sometimes
2108                1.414209              0.646288                Sometimes
2109                1.139107              0.586035                Sometimes
2110                1.026452              0.714137                Sometimes

        mode_transportation        obesity_level  obesity_binary  \
0     Public_Transportation        Normal_Weight               0
1     Public_Transportation        Normal_Weight               0
2     Public_Transportation        Normal_Weight               0
3                   Walking   Overweight_Level_I               1
4     Public_Transportation  Overweight_Level_II               1
…                       …                   …               …
2106  Public_Transportation     Obesity_Type_III               1
2107  Public_Transportation     Obesity_Type_III               1
2108  Public_Transportation     Obesity_Type_III               1
2109  Public_Transportation     Obesity_Type_III               1
2110  Public_Transportation     Obesity_Type_III               1

      overweight_family_binary
0                            1
1                            1
2                            1
3                            0
4                            0
…                          …
2106                         1
2107                         1
```

```
2108                          1
2109                          1
2110                          1

[2111 rows x 19 columns]
```

[9]:
```python
# FREQUENCY HIGH CALORIE FOOD
# Mapping 'yes' to 1 and 'no' to 0
obesity['binary_freq_high_calorie_food'] = obesity['freq_high_calorie_food'].
 ↪map({'yes': 1, 'no': 0})
obesity
```

[9]:
```
      gender        age    height      weight overweight_family  \
0     Female  21.000000  1.620000   64.000000               yes
1     Female  21.000000  1.520000   56.000000               yes
2       Male  23.000000  1.800000   77.000000               yes
3       Male  27.000000  1.800000   87.000000                no
4       Male  22.000000  1.780000   89.800000                no
…        …         …         …          …               …
2106  Female  20.976842  1.710730  131.408528               yes
2107  Female  21.982942  1.748584  133.742943               yes
2108  Female  22.524036  1.752206  133.689352               yes
2109  Female  24.361936  1.739450  133.346641               yes
2110  Female  23.664709  1.738836  133.472641               yes

     freq_high_calorie_food  freq_vegetable  num_main_meals  \
0                        no             2.0             3.0
1                        no             3.0             3.0
2                        no             2.0             3.0
3                        no             3.0             3.0
4                        no             2.0             1.0
…                       …              …              …
2106                    yes             3.0             3.0
2107                    yes             3.0             3.0
2108                    yes             3.0             3.0
2109                    yes             3.0             3.0
2110                    yes             3.0             3.0

     consume_food_btwn_meals smoke  daily_h2o_consumption calorie_monitoring  \
0                  Sometimes    no               2.000000                 no
1                  Sometimes   yes               3.000000                yes
2                  Sometimes    no               2.000000                 no
3                  Sometimes    no               2.000000                 no
4                  Sometimes    no               2.000000                 no
…                        …   …                    …                …
2106               Sometimes    no               1.728139                 no
2107               Sometimes    no               2.005130                 no
```

```
2108              Sometimes    no             2.054193              no
2109              Sometimes    no             2.852339              no
2110              Sometimes    no             2.863513              no

      freq_physical_activity  time_used_technology freq_alcohol_consumption  \
0                   0.000000              1.000000                       no
1                   3.000000              0.000000                Sometimes
2                   2.000000              1.000000               Frequently
3                   2.000000              0.000000               Frequently
4                   0.000000              0.000000                Sometimes
…                        …                     …                        …
2106                1.676269              0.906247                Sometimes
2107                1.341390              0.599270                Sometimes
2108                1.414209              0.646288                Sometimes
2109                1.139107              0.586035                Sometimes
2110                1.026452              0.714137                Sometimes

        mode_transportation        obesity_level  obesity_binary  \
0     Public_Transportation        Normal_Weight               0
1     Public_Transportation        Normal_Weight               0
2     Public_Transportation        Normal_Weight               0
3                   Walking   Overweight_Level_I               1
4     Public_Transportation  Overweight_Level_II               1
…                       …                   …             …
2106  Public_Transportation     Obesity_Type_III               1
2107  Public_Transportation     Obesity_Type_III               1
2108  Public_Transportation     Obesity_Type_III               1
2109  Public_Transportation     Obesity_Type_III               1
2110  Public_Transportation     Obesity_Type_III               1

      overweight_family_binary  binary_freq_high_calorie_food
0                            1                              0
1                            1                              0
2                            1                              0
3                            0                              0
4                            0                              0
…                          …                              …
2106                         1                              1
2107                         1                              1
2108                         1                              1
2109                         1                              1
2110                         1                              1

[2111 rows x 20 columns]
```

[10]:
```
# CALORIE MONITORING
# Mapping 'yes' to 1 and 'no' to 0
```

```
obesity['binary_calorie_monitoring'] = obesity['calorie_monitoring'].map({'yes':
 ↪ 1, 'no': 0})
obesity
```

[10]:

```
      gender        age    height      weight overweight_family  \
0     Female  21.000000  1.620000   64.000000               yes
1     Female  21.000000  1.520000   56.000000               yes
2       Male  23.000000  1.800000   77.000000               yes
3       Male  27.000000  1.800000   87.000000                no
4       Male  22.000000  1.780000   89.800000                no
…        …          …         …          …               …
2106  Female  20.976842  1.710730  131.408528               yes
2107  Female  21.982942  1.748584  133.742943               yes
2108  Female  22.524036  1.752206  133.689352               yes
2109  Female  24.361936  1.739450  133.346641               yes
2110  Female  23.664709  1.738836  133.472641               yes

     freq_high_calorie_food  freq_vegetable  num_main_meals  \
0                        no             2.0             3.0
1                        no             3.0             3.0
2                        no             2.0             3.0
3                        no             3.0             3.0
4                        no             2.0             1.0
…                       …               …               …
2106                    yes             3.0             3.0
2107                    yes             3.0             3.0
2108                    yes             3.0             3.0
2109                    yes             3.0             3.0
2110                    yes             3.0             3.0

     consume_food_btwn_meals smoke  …  calorie_monitoring  \
0                  Sometimes    no  …                  no
1                  Sometimes   yes  …                 yes
2                  Sometimes    no  …                  no
3                  Sometimes    no  …                  no
4                  Sometimes    no  …                  no
…                       …    …  …                 …
2106               Sometimes    no  …                  no
2107               Sometimes    no  …                  no
2108               Sometimes    no  …                  no
2109               Sometimes    no  …                  no
2110               Sometimes    no  …                  no

     freq_physical_activity  time_used_technology  freq_alcohol_consumption  \
0                  0.000000              1.000000                        no
1                  3.000000              0.000000                 Sometimes
2                  2.000000              1.000000                Frequently
```

```
3                 2.000000             0.000000              Frequently
4                 0.000000             0.000000               Sometimes
…                      …                 …                        …
2106              1.676269             0.906247               Sometimes
2107              1.341390             0.599270               Sometimes
2108              1.414209             0.646288               Sometimes
2109              1.139107             0.586035               Sometimes
2110              1.026452             0.714137               Sometimes

        mode_transportation        obesity_level obesity_binary  \
0     Public_Transportation        Normal_Weight              0
1     Public_Transportation        Normal_Weight              0
2     Public_Transportation        Normal_Weight              0
3                   Walking   Overweight_Level_I              1
4     Public_Transportation  Overweight_Level_II              1
…                      …                   …             …
2106  Public_Transportation     Obesity_Type_III              1
2107  Public_Transportation     Obesity_Type_III              1
2108  Public_Transportation     Obesity_Type_III              1
2109  Public_Transportation     Obesity_Type_III              1
2110  Public_Transportation     Obesity_Type_III              1

      overweight_family_binary  binary_freq_high_calorie_food  \
0                            1                              0
1                            1                              0
2                            1                              0
3                            0                              0
4                            0                              0
…                          …                            …
2106                         1                              1
2107                         1                              1
2108                         1                              1
2109                         1                              1
2110                         1                              1

      binary_calorie_monitoring
0                             0
1                             1
2                             0
3                             0
4                             0
…                           …
2106                          0
2107                          0
2108                          0
2109                          0
2110                          0
```

```
[2111 rows x 21 columns]
```

[11]:
```
# SMOKE
# Mapping 'yes' to 1 and 'no' to 0
obesity['binary_smoke'] = obesity['smoke'].map({'yes': 1, 'no': 0})
obesity
```

[11]:
```
      gender        age    height      weight overweight_family  \
0     Female  21.000000  1.620000   64.000000               yes
1     Female  21.000000  1.520000   56.000000               yes
2       Male  23.000000  1.800000   77.000000               yes
3       Male  27.000000  1.800000   87.000000                no
4       Male  22.000000  1.780000   89.800000                no
…        …         …         …         …                 …
2106  Female  20.976842  1.710730  131.408528               yes
2107  Female  21.982942  1.748584  133.742943               yes
2108  Female  22.524036  1.752206  133.689352               yes
2109  Female  24.361936  1.739450  133.346641               yes
2110  Female  23.664709  1.738836  133.472641               yes

     freq_high_calorie_food  freq_vegetable  num_main_meals  \
0                        no             2.0             3.0
1                        no             3.0             3.0
2                        no             2.0             3.0
3                        no             3.0             3.0
4                        no             2.0             1.0
…                       …               …               …
2106                    yes             3.0             3.0
2107                    yes             3.0             3.0
2108                    yes             3.0             3.0
2109                    yes             3.0             3.0
2110                    yes             3.0             3.0

     consume_food_btwn_meals smoke  …  freq_physical_activity  \
0                  Sometimes    no  …                0.000000
1                  Sometimes   yes  …                3.000000
2                  Sometimes    no  …                2.000000
3                  Sometimes    no  …                2.000000
4                  Sometimes    no  …                0.000000
…                       …   …  …                    …
2106               Sometimes    no  …                1.676269
2107               Sometimes    no  …                1.341390
2108               Sometimes    no  …                1.414209
2109               Sometimes    no  …                1.139107
2110               Sometimes    no  …                1.026452
```

```
     time_used_technology  freq_alcohol_consumption    mode_transportation  \
0                1.000000                        no  Public_Transportation
1                0.000000                 Sometimes  Public_Transportation
2                1.000000                Frequently  Public_Transportation
3                0.000000                Frequently                Walking
4                0.000000                 Sometimes  Public_Transportation
…                     …                        …                      …
2106             0.906247                 Sometimes  Public_Transportation
2107             0.599270                 Sometimes  Public_Transportation
2108             0.646288                 Sometimes  Public_Transportation
2109             0.586035                 Sometimes  Public_Transportation
2110             0.714137                 Sometimes  Public_Transportation

            obesity_level obesity_binary overweight_family_binary  \
0           Normal_Weight              0                        1
1           Normal_Weight              0                        1
2           Normal_Weight              0                        1
3      Overweight_Level_I              1                        0
4     Overweight_Level_II              1                        0
…                     …             …                       …
2106     Obesity_Type_III              1                        1
2107     Obesity_Type_III              1                        1
2108     Obesity_Type_III              1                        1
2109     Obesity_Type_III              1                        1
2110     Obesity_Type_III              1                        1

      binary_freq_high_calorie_food  binary_calorie_monitoring  binary_smoke
0                                 0                          0             0
1                                 0                          1             1
2                                 0                          0             0
3                                 0                          0             0
4                                 0                          0             0
…                              …                          …           …
2106                              1                          0             0
2107                              1                          0             0
2108                              1                          0             0
2109                              1                          0             0
2110                              1                          0             0

[2111 rows x 22 columns]
```

[12]:
```
# checking the levels in the consume_food_btwn_meals variable
obesity['consume_food_btwn_meals'].unique()
```

[12]:
```
array(['Sometimes', 'Frequently', 'Always', 'no'], dtype=object)
```

[13]:
```python
# CONSUME FOOD BETWEEN MEALS
# Defining a function to map the categories 'sometimes', 'frequently', and␣
↪'always' to 1
# and everything else to 0
def map_consume_btwn_meals(consumption):
    if consumption in ['Sometimes', 'Frequently', 'Always']:
        return 1
    else:
        return 0

obesity['binary_consume_food_btwn_meals'] = obesity['consume_food_btwn_meals'].
↪apply(map_consume_btwn_meals)
```

[14]:
```python
# checking the levels in the freq_alcohol_consumption variable
obesity['freq_alcohol_consumption'].unique()
```

[14]:
```
array(['no', 'Sometimes', 'Frequently', 'Always'], dtype=object)
```

[15]:
```python
# FREQUENT ALCOHOL CONSUMPTION
# Defining a function to map the categories 'sometimes', 'frequently', and␣
↪'always' to 1
# and everything else to 0
def map_consume_alcohol(consume_alcohol):
    if consume_alcohol in ['Sometimes', 'Frequently', 'Always']:
        return 1
    else:
        return 0

obesity['binary_freq_alcohol_consumption'] =␣
↪obesity['freq_alcohol_consumption'].apply(map_consume_alcohol)
```

[16]:
```python
obesity.columns # these are now all of the variables in the dataset
```

[16]:
```
Index(['gender', 'age', 'height', 'weight', 'overweight_family',
       'freq_high_calorie_food', 'freq_vegetable', 'num_main_meals',
       'consume_food_btwn_meals', 'smoke', 'daily_h2o_consumption',
       'calorie_monitoring', 'freq_physical_activity', 'time_used_technology',
       'freq_alcohol_consumption', 'mode_transportation', 'obesity_level',
       'obesity_binary', 'overweight_family_binary',
       'binary_freq_high_calorie_food', 'binary_calorie_monitoring',
       'binary_smoke', 'binary_consume_food_btwn_meals',
       'binary_freq_alcohol_consumption'],
      dtype='object')
```

[17]:
```python
obesity['is_female'] = (obesity['gender'].str.lower() == 'female').astype(int)
obesity
```

[17]:

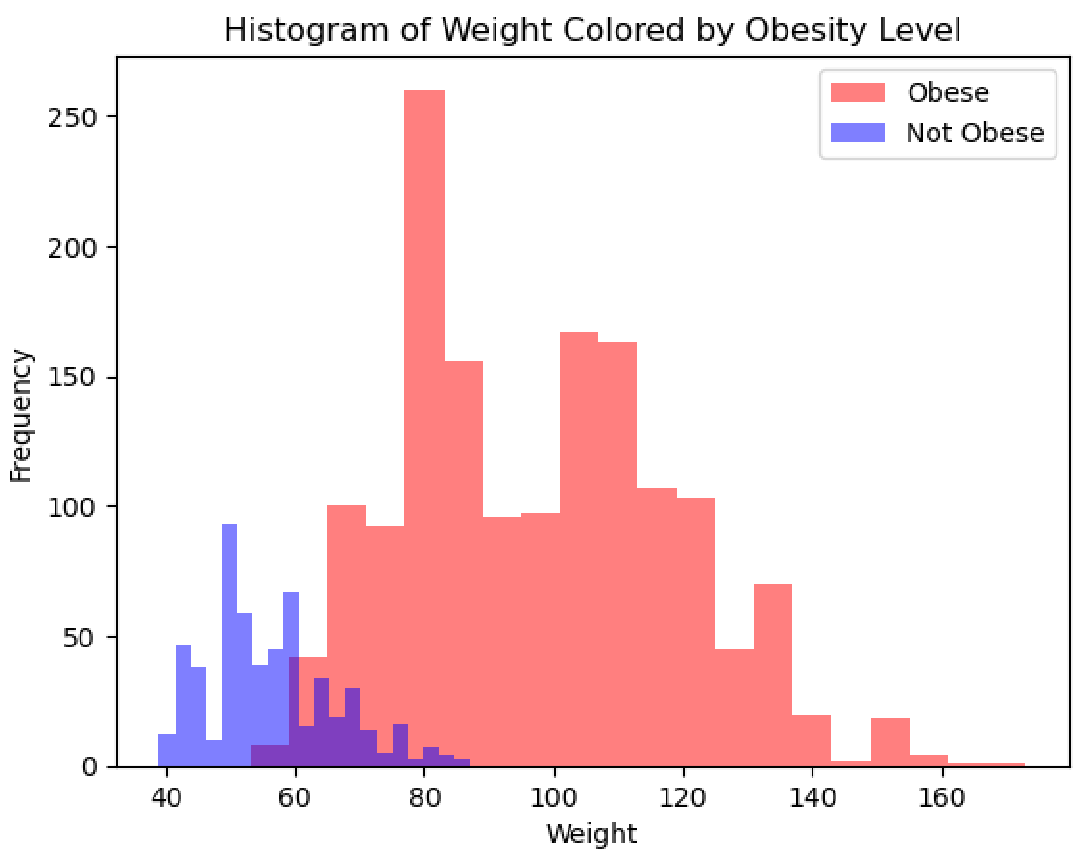
```
      gender        age    height      weight overweight_family  \
0     Female  21.000000  1.620000   64.000000               yes
1     Female  21.000000  1.520000   56.000000               yes
2       Male  23.000000  1.800000   77.000000               yes
3       Male  27.000000  1.800000   87.000000                no
4       Male  22.000000  1.780000   89.800000                no
…        …        …        …        …              …
2106  Female  20.976842  1.710730  131.408528               yes
2107  Female  21.982942  1.748584  133.742943               yes
2108  Female  22.524036  1.752206  133.689352               yes
2109  Female  24.361936  1.739450  133.346641               yes
2110  Female  23.664709  1.738836  133.472641               yes

     freq_high_calorie_food  freq_vegetable  num_main_meals  \
0                        no             2.0             3.0
1                        no             3.0             3.0
2                        no             2.0             3.0
3                        no             3.0             3.0
4                        no             2.0             1.0
…                       …              …              …
2106                    yes             3.0             3.0
2107                    yes             3.0             3.0
2108                    yes             3.0             3.0
2109                    yes             3.0             3.0
2110                    yes             3.0             3.0

     consume_food_btwn_meals smoke  …    mode_transportation  \
0                  Sometimes    no  …  Public_Transportation
1                  Sometimes   yes  …  Public_Transportation
2                  Sometimes    no  …  Public_Transportation
3                  Sometimes    no  …                Walking
4                  Sometimes    no  …  Public_Transportation
…                       …    …  …                    …
2106               Sometimes    no  …  Public_Transportation
2107               Sometimes    no  …  Public_Transportation
2108               Sometimes    no  …  Public_Transportation
2109               Sometimes    no  …  Public_Transportation
2110               Sometimes    no  …  Public_Transportation

             obesity_level  obesity_binary  overweight_family_binary  \
0            Normal_Weight               0                         1
1            Normal_Weight               0                         1
2            Normal_Weight               0                         1
3       Overweight_Level_I               1                         0
4      Overweight_Level_II               1                         0
…                      …              …                        …
2106      Obesity_Type_III               1                         1
```

```
2107     Obesity_Type_III              1                         1
2108     Obesity_Type_III              1                         1
2109     Obesity_Type_III              1                         1
2110     Obesity_Type_III              1                         1

     binary_freq_high_calorie_food binary_calorie_monitoring binary_smoke  \
0                                0                         0            0
1                                0                         1            1
2                                0                         0            0
3                                0                         0            0
4                                0                         0            0
…                              …                       …          …
2106                             1                         0            0
2107                             1                         0            0
2108                             1                         0            0
2109                             1                         0            0
2110                             1                         0            0

      binary_consume_food_btwn_meals  binary_freq_alcohol_consumption  \
0                                  1                                0
1                                  1                                1
2                                  1                                1
3                                  1                                1
4                                  1                                1
…                                …                              …
2106                               1                                1
2107                               1                                1
2108                               1                                1
2109                               1                                1
2110                               1                                1

      is_female
0             1
1             1
2             0
3             0
4             0
…           …
2106          1
2107          1
2108          1
2109          1
2110          1

[2111 rows x 25 columns]
```

[18]:
```python
# dropping the mode of transportation variable
obesity.drop(columns='mode_transportation', inplace=True)
```

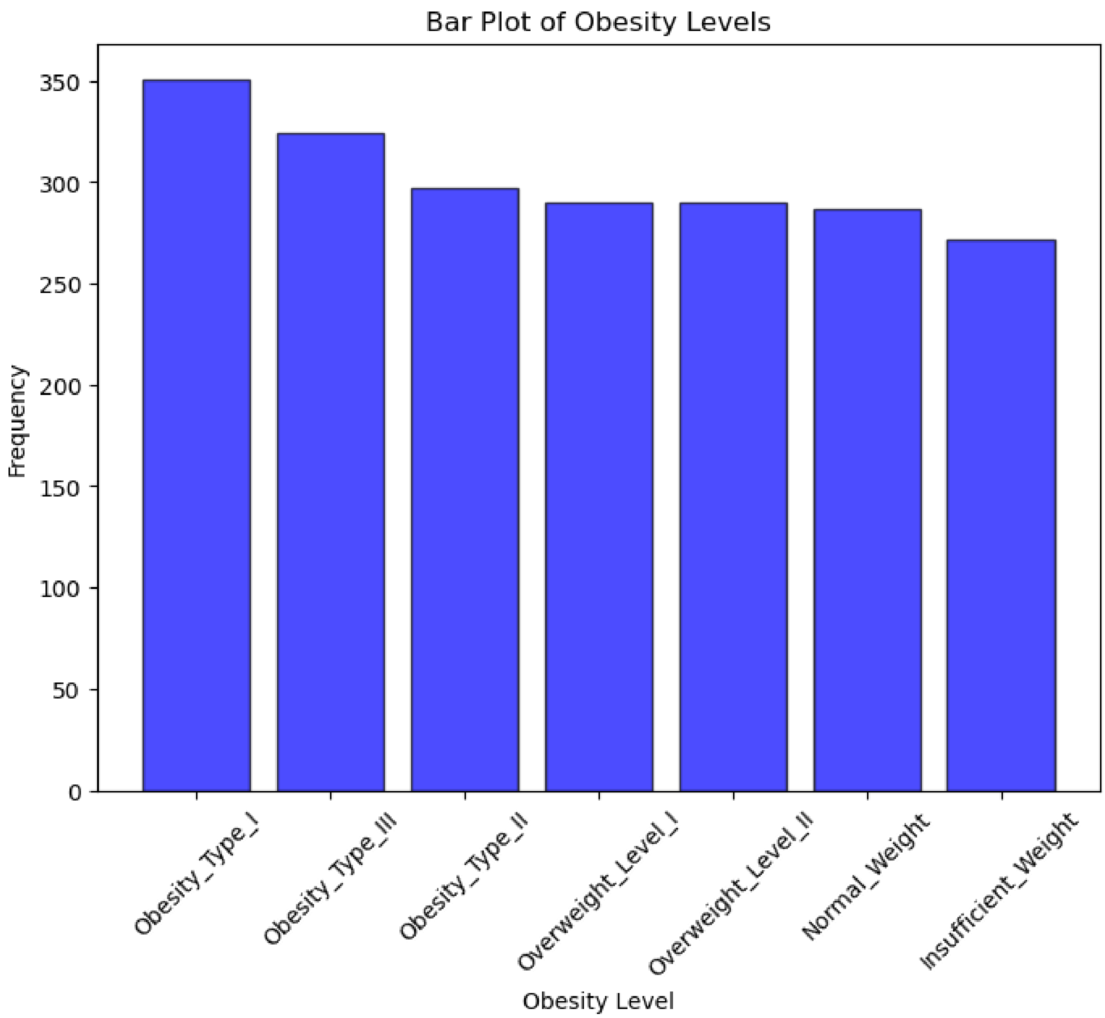
0.2 Step 2: Data Exploration

[19]:
```python
import matplotlib.pyplot as plt

weight_obese = obesity.loc[obesity['obesity_binary'] == 1, 'weight']
weight_not_obese = obesity.loc[obesity['obesity_binary'] == 0, 'weight']

plt.hist(weight_obese, color='red', alpha=0.5, label='Obese', bins=20)
plt.hist(weight_not_obese, color='blue', alpha=0.5, label='Not Obese', bins=20)

plt.xlabel('Weight')
plt.ylabel('Frequency')
plt.title('Histogram of Weight Colored by Obesity Level')

plt.legend()

plt.show()
```



Inspecting the Histogram of Weight Colored by Obesity Level, I notice that there is a little bit of overlap between the "obsese" and "not obese" categories, but they are decently separable.

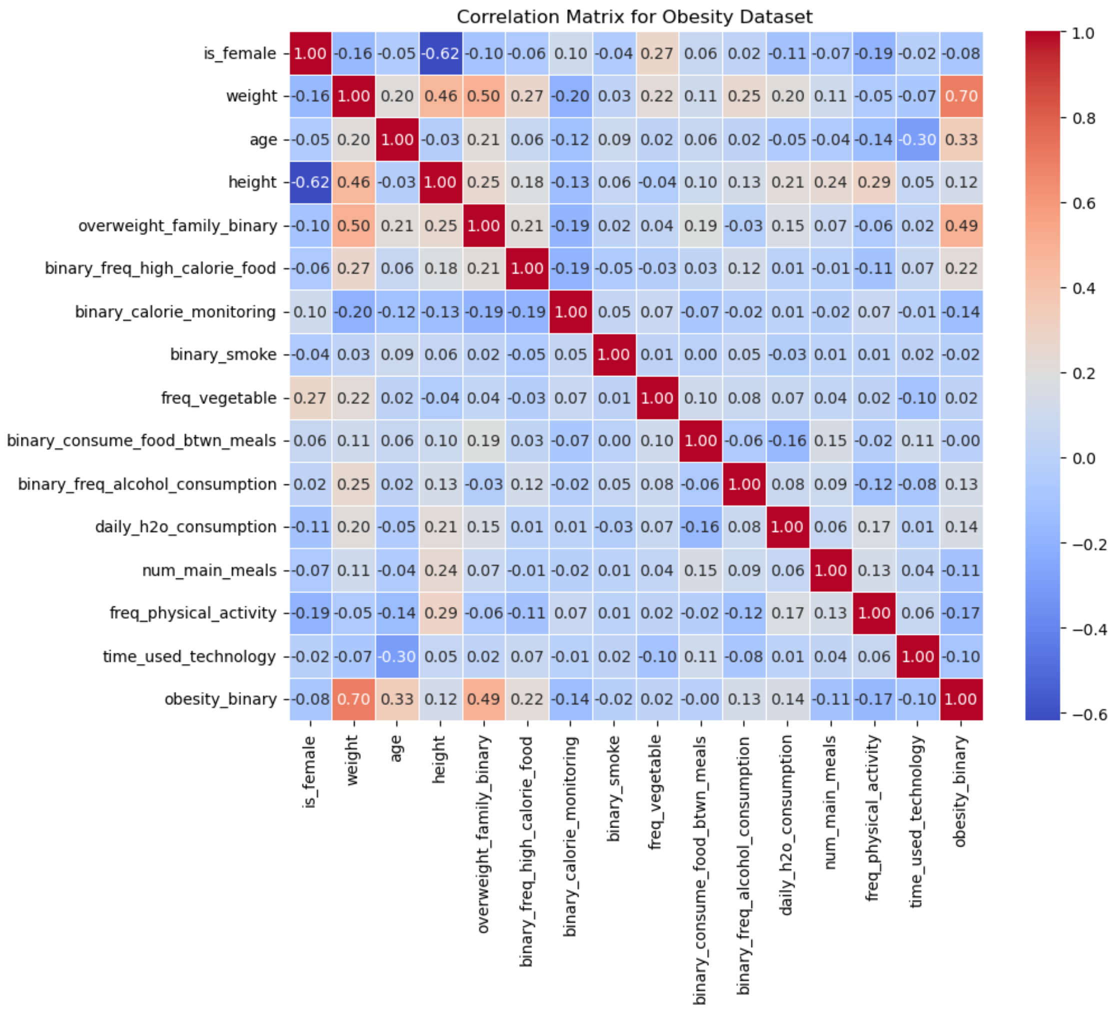
[20]:
```
# Histogram of obesity_level variable
obesity_val_counts = obesity['obesity_level'].value_counts()

plt.figure(figsize=(8, 6))

# creating bar plot
plt.bar(obesity_val_counts.index, obesity_val_counts.values, color='blue',␣
↪edgecolor='black', alpha=0.7)

# adding title
plt.title('Bar Plot of Obesity Levels')

# adding x label
plt.xlabel('Obesity Level')

# adding y label
plt.ylabel('Frequency')

# rotating x ticks
plt.xticks(rotation=45)

# Show plot
plt.show()
```



The bar plot of obesity level reveals that the frequencies of each obesity level are not too different from each other.

[21]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

X = obesity[['is_female','weight','age', 'height','overweight_family_binary',␣
↪'binary_freq_high_calorie_food',
             'binary_calorie_monitoring', 'binary_smoke', 'freq_vegetable',
             'binary_consume_food_btwn_meals',␣
↪'binary_freq_alcohol_consumption',
             'daily_h2o_consumption','num_main_meals',␣
↪'freq_physical_activity', 'time_used_technology']]
y = obesity['obesity_binary']
```

```python
data = pd.concat([X, y], axis=1)


correlation_matrix = data.corr()


plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",␣
 ↪linewidths=.5)
plt.title('Correlation Matrix for Obesity Dataset')
plt.show()
```



The correlation matrix shows the level of correlation between variables in the dataset. Weight and obesity_binary are strongly, positively correlated. Obesity_binary and overweight_family_binary are moderately, positively correlated.

0.3 Step 3: Logistic Regression Model

[22]:
```
# full model: 15 predictors

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix, precision_score,␣
 ↪recall_score, f1_score

X = obesity[['is_female','weight','age', 'height','overweight_family_binary',␣
 ↪'binary_freq_high_calorie_food',
             'binary_calorie_monitoring', 'binary_smoke', 'freq_vegetable',
             'binary_consume_food_btwn_meals',␣
 ↪'binary_freq_alcohol_consumption',
             'daily_h2o_consumption','num_main_meals',␣
 ↪'freq_physical_activity', 'time_used_technology']]
y = obesity['obesity_binary']

# 20% of the data goes into training set and rest for testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state = 43)

# initialize logistic regression model
model = LogisticRegression(max_iter = 1000)

# training the model
model.fit(X_train, y_train)

# predict on the testing data
y_pred = model.predict(X_test)

# Find the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# confusion matrix for precision, recall
# Generating confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Calculating precision and recall
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
```

```python
print("Confusion Matrix:")
print(cm)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

```
Accuracy: 0.9550827423167849
Confusion Matrix:
[[ 89  13]
 [  6 315]]
Precision: 0.9603658536585366
Recall: 0.9813084112149533
F1 Score: 0.9707241910631741
```

[23]:

```python
# Now, using stats models, I can find the pseudo r-squared and p-values
# to see which variables are most statistically significant.
import statsmodels.api as sm
import numpy as np

X = obesity[['is_female','weight','age', 'height','overweight_family_binary',␣
 ↪'binary_freq_high_calorie_food',
             'binary_calorie_monitoring', 'binary_smoke', 'freq_vegetable',
             'binary_consume_food_btwn_meals',␣
 ↪'binary_freq_alcohol_consumption',
             'daily_h2o_consumption','num_main_meals',␣
 ↪'freq_physical_activity', 'time_used_technology']]
y = obesity['obesity_binary']

X_with_const = sm.add_constant(X)

# fitting the model
logit_model = sm.Logit(y, X_with_const)

# getting model results
logit_result = logit_model.fit()

# printing summary
print(logit_result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.010834
         Iterations 16
                           Logit Regression Results
==============================================================================
Dep. Variable:         obesity_binary   No. Observations:                 2111
Model:                          Logit   Df Residuals:                     2095
Method:                           MLE   Df Model:                           15
Date:                Thu, 21 Mar 2024   Pseudo R-squ.:                  0.9813
```

```
Time:                        22:32:28   Log-Likelihood:                -22.870
converged:                       True   LL-Null:                       -1220.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================
===================
                                      coef    std err          z      P>|z|
[0.025      0.975]
--------------------------------------------------------------------------------
-------------------
const                             178.3995     35.808      4.982      0.000
108.218     248.581
is_female                           3.8651      1.717      2.251      0.024
0.500       7.230
weight                              2.7437      0.536      5.115      0.000
1.692       3.795
age                                 0.0758      0.064      1.193      0.233
-0.049       0.200
height                           -221.8823     43.272     -5.128      0.000
-306.695    -137.070
overweight_family_binary           -0.2845      0.988     -0.288      0.774
-2.222       1.653
binary_freq_high_calorie_food       1.8168      1.232      1.475      0.140
-0.598       4.231
binary_calorie_monitoring           2.5950      1.334      1.945      0.052
-0.019       5.209
binary_smoke                       -6.3120     10.318     -0.612      0.541
-26.535      13.911
freq_vegetable                     -0.7555      0.851     -0.888      0.374
-2.423       0.912
binary_consume_food_btwn_meals      0.8899      5.147      0.173      0.863
-9.198      10.978
binary_freq_alcohol_consumption     0.2359      1.143      0.206      0.837
-2.005       2.477
daily_h2o_consumption              -0.6487      0.813     -0.797      0.425
-2.243       0.946
num_main_meals                     -0.7463      0.627     -1.190      0.234
-1.976       0.483
freq_physical_activity             -0.4357      0.467     -0.934      0.350
-1.350       0.479
time_used_technology                0.5279      0.686      0.769      0.442
-0.817       1.873
================================================================================
===================

Possibly complete quasi-separation: A fraction 0.89 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

0.3.1 Suppose we did not have access to weight as a predictor, a.k.a we did not have data on weight. Would we still be able to predict whether or not someone is obese just by looking at the other features?

[24]:
```
# Now we can try Logistic Regression without weight as a feature (14 predictors␣
↪now)

X = obesity[['is_female','age', 'height','overweight_family_binary',␣
↪'binary_freq_high_calorie_food',
             'binary_calorie_monitoring', 'binary_smoke', 'freq_vegetable',
             'binary_consume_food_btwn_meals',␣
↪'binary_freq_alcohol_consumption',
             'daily_h2o_consumption','num_main_meals',␣
↪'freq_physical_activity', 'time_used_technology']]
y = obesity['obesity_binary']

# 20% of the data goes into training set and rest for testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state = 43)

# initialize logistic regression model
model = LogisticRegression(max_iter = 1000)

# training the model
model.fit(X_train, y_train)

# predict on the testing data
y_pred = model.predict(X_test)

# Find the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
# confusion matrix for precision, recall

# Generating confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Calculating precision and recall
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Confusion Matrix:")
print(cm)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

```
#final logistic model coefficients, p-values. correlation matrix
```

```
Accuracy: 0.8534278959810875
Confusion Matrix:
[[ 57  45]
 [ 17 304]]
Precision: 0.8710601719197708
Recall: 0.9470404984423676
F1 Score: 0.9074626865671641
```

[25]:
```
X = obesity[['is_female','age','height','overweight_family_binary',␣
 ↪'binary_freq_high_calorie_food',
            'binary_calorie_monitoring', 'binary_smoke', 'freq_vegetable',
            'binary_consume_food_btwn_meals',␣
 ↪'binary_freq_alcohol_consumption',
            'daily_h2o_consumption','num_main_meals',␣
 ↪'freq_physical_activity', 'time_used_technology']]
y = obesity['obesity_binary']

# Add a constant to the features (X) to account for the intercept term
X_with_const = sm.add_constant(X)

# Fit logistic regression model
logit_model = sm.Logit(y, X_with_const)

# Get the fitted model results
logit_result = logit_model.fit()

# Print model summary containing coefficients and p-values
print(logit_result.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.382022
         Iterations 7
                           Logit Regression Results
==============================================================================
Dep. Variable:         obesity_binary   No. Observations:                 2111
Model:                          Logit   Df Residuals:                     2096
Method:                           MLE   Df Model:                           14
Date:                Thu, 21 Mar 2024   Pseudo R-squ.:                  0.3391
Time:                        22:32:28   Log-Likelihood:                -806.45
converged:                       True   LL-Null:                       -1220.2
Covariance Type:            nonrobust   LLR p-value:                1.439e-167
================================================================================
===================
                                      coef    std err          z      P>|z|
[0.025      0.975]
--------------------------------------------------------------------------------
```

```
-------------------
const                                  -4.4150      1.620     -2.725      0.006
-7.591      -1.239
is_female                              -0.1225      0.169     -0.727      0.467
-0.453       0.208
age                                     0.1672      0.016     10.372      0.000
0.136       0.199
height                                  0.1455      1.010      0.144      0.885
-1.834       2.125
overweight_family_binary                2.6148      0.172     15.210      0.000
2.278       2.952
binary_freq_high_calorie_food           0.7317      0.191      3.830      0.000
0.357       1.106
binary_calorie_monitoring               0.0241      0.279      0.086      0.931
-0.523       0.571
binary_smoke                           -1.0800      0.440     -2.456      0.014
-1.942      -0.218
freq_vegetable                         -0.0942      0.128     -0.737      0.461
-0.345       0.156
binary_consume_food_btwn_meals         -0.6130      0.404     -1.519      0.129
-1.404       0.178
binary_freq_alcohol_consumption         0.7606      0.139      5.464      0.000
0.488       1.033
daily_h2o_consumption                   0.4745      0.113      4.208      0.000
0.253       0.695
num_main_meals                         -0.5013      0.089     -5.658      0.000
-0.675      -0.328
freq_physical_activity                 -0.3051      0.082     -3.742      0.000
-0.465      -0.145
time_used_technology                   -0.1316      0.108     -1.221      0.222
-0.343       0.080
================================================================================
===================
```

0.4 Step 4: Decision Tree Classifier Model

[26]:
```python
# Full model: 15 predictors

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪mean_squared_error
import pandas as pd


# Split the dataset into training and testing sets
```

```
X_train, X_test, y_train, y_test =␣
↪train_test_split(obesity[['is_female','weight','age',␣
↪'height','overweight_family_binary',
                                                                   ␣
↪'binary_freq_high_calorie_food', 'freq_vegetable',
                                                                   ␣
↪'binary_calorie_monitoring', 'binary_smoke',
                                                                   ␣
↪'binary_consume_food_btwn_meals', 'binary_freq_alcohol_consumption',
                                                                   ␣
↪'daily_h2o_consumption','num_main_meals',
                                                                   ␣
↪'freq_physical_activity', 'time_used_technology']],
                                                    obesity['obesity_binary'],␣
↪test_size=0.2, random_state=42)

# Initialize decision tree classifier
tree = DecisionTreeClassifier(random_state=42)

# Train the classifier on the training data
tree.fit(X_train, y_train)

# Predict on the testing data
y_pred = tree.predict(X_test)

# Evaluate the classifier
accuracy = accuracy_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
print("Accuracy:", accuracy)
print("MSE:", mse)

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9716312056737588
MSE: 0.028368794326241134
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.95      0.95       118
           1       0.98      0.98      0.98       305

    accuracy                           0.97       423
   macro avg       0.96      0.96      0.96       423
weighted avg       0.97      0.97      0.97       423
```

```
[27]: # Now trying decision tree without the weight variable (14 predictors now)

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪mean_squared_error
import pandas as pd

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(obesity[['is_female','age',␣
 ↪'height','overweight_family_binary',
                                                                               ␣
 ↪'binary_freq_high_calorie_food', 'freq_vegetable',
                                                                               ␣
 ↪'binary_calorie_monitoring', 'binary_smoke',
                                                                               ␣
 ↪'binary_consume_food_btwn_meals', 'binary_freq_alcohol_consumption',
                                                                               ␣
 ↪'daily_h2o_consumption','num_main_meals',
                                                                               ␣
 ↪'freq_physical_activity', 'time_used_technology']],
                                                    obesity['obesity_binary'],␣
 ↪test_size=0.2, random_state=42)

# Initialize decision tree classifier
tree2 = DecisionTreeClassifier(random_state=42)

# Train the classifier on the training data
tree2.fit(X_train, y_train)

# Predict on the testing data
y_pred = tree2.predict(X_test)

# Evaluate the classifier
accuracy = accuracy_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
print("Accuracy:", accuracy)
print("MSE:", mse)

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.851063829787234
MSE: 0.14893617021276595
Classification Report:
              precision    recall  f1-score   support
```

0	0.74	0.71	0.73	118
1	0.89	0.90	0.90	305
accuracy			0.85	423
macro avg	0.82	0.81	0.81	423
weighted avg	0.85	0.85	0.85	423