SMKData

March 31, 2025

[ ]:

1 SMOKING AND DRINKING DATASET With Body Signals

1.1 Data Integrity Check

1. **Load the dataset & create a protected copy**
2. **Check for missing values, duplicates, and category definitions**
3. **Ensure column types match original descriptions**
4. **Verify smoking & drinking labels are correct**

[4]:
```
import pandas as pd

# Load Dataset
SSD_Orig = pd.read_csv("SDD.csv")


# Create a working copy
SSD_Wrk = SSD_Orig.copy()

# General Dataset Summary
print("--- Dataset Overview ---")
print(SSD_Wrk.info())

# Check Unique Values in Categorical Features
cat_features = ["sex", "DRK_YN", "SMK_stat_type_cd"]
for col in cat_features:
    print(f"\n--- Unique Values in {col} ---")
    print(SSD_Wrk[col].value_counts())

# Check for Duplicates
SSD_Duplicates = SSD_Wrk[SSD_Wrk.duplicated()]
print(f"\n--- Duplicate Rows: {SSD_Duplicates.shape[0]} ---")

# Check for Missing Values
print("\n--- Missing Values ---")
print(SSD_Wrk.isnull().sum())
```

```python
# Analyze Numerical Distributions
print("\n--- Statistical Summary ---")
print(SSD_Wrk.describe().T)
```

```
--- Dataset Overview ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 991346 entries, 0 to 991345
Data columns (total 24 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   sex               991346 non-null  object
 1   age               991346 non-null  int64
 2   height            991346 non-null  int64
 3   weight            991346 non-null  int64
 4   waistline         991346 non-null  float64
 5   sight_left        991346 non-null  float64
 6   sight_right       991346 non-null  float64
 7   hear_left         991346 non-null  float64
 8   hear_right        991346 non-null  float64
 9   SBP               991346 non-null  float64
 10  DBP               991346 non-null  float64
 11  BLDS              991346 non-null  float64
 12  tot_chole         991346 non-null  float64
 13  HDL_chole         991346 non-null  float64
 14  LDL_chole         991346 non-null  float64
 15  triglyceride      991346 non-null  float64
 16  hemoglobin        991346 non-null  float64
 17  urine_protein     991346 non-null  float64
 18  serum_creatinine  991346 non-null  float64
 19  SGOT_AST          991346 non-null  float64
 20  SGOT_ALT          991346 non-null  float64
 21  gamma_GTP         991346 non-null  float64
 22  SMK_stat_type_cd  991346 non-null  float64
 23  DRK_YN            991346 non-null  object
dtypes: float64(19), int64(3), object(2)
memory usage: 181.5+ MB
None

--- Unique Values in sex ---
sex
Male      526415
Female    464931
Name: count, dtype: int64

--- Unique Values in DRK_YN ---
DRK_YN
N    495858
Y    495488
```

```
Name: count, dtype: int64

--- Unique Values in SMK_stat_type_cd ---
SMK_stat_type_cd
1.0    602441
3.0    213954
2.0    174951
Name: count, dtype: int64

--- Duplicate Rows: 26 ---

--- Missing Values ---
sex                 0
age                 0
height              0
weight              0
waistline           0
sight_left          0
sight_right         0
hear_left           0
hear_right          0
SBP                 0
DBP                 0
BLDS                0
tot_chole           0
HDL_chole           0
LDL_chole           0
triglyceride        0
hemoglobin          0
urine_protein       0
serum_creatinine    0
SGOT_AST            0
SGOT_ALT            0
gamma_GTP           0
SMK_stat_type_cd    0
DRK_YN              0
dtype: int64

--- Statistical Summary ---
                    count        mean         std    min    25%    50%  \
age              991346.0   47.614491   14.181339   20.0   35.0   45.0
height           991346.0  162.240625    9.282957  130.0  155.0  160.0
weight           991346.0   63.284050   12.514241   25.0   55.0   60.0
waistline        991346.0   81.233358   11.850323    8.0   74.1   81.0
sight_left       991346.0    0.980834    0.605949    0.1    0.7    1.0
sight_right      991346.0    0.978429    0.604774    0.1    0.7    1.0
hear_left        991346.0    1.031495    0.174650    1.0    1.0    1.0
hear_right       991346.0    1.030476    0.171892    1.0    1.0    1.0
```

SBP	991346.0	122.432498	14.543148	67.0	112.0	120.0
DBP	991346.0	76.052627	9.889365	32.0	70.0	76.0
BLDS	991346.0	100.424447	24.179960	25.0	88.0	96.0
tot_chole	991346.0	195.557020	38.660155	30.0	169.0	193.0
HDL_chole	991346.0	56.936800	17.238479	1.0	46.0	55.0
LDL_chole	991346.0	113.037692	35.842812	1.0	89.0	111.0
triglyceride	991346.0	132.141751	102.196985	1.0	73.0	106.0
hemoglobin	991346.0	14.229824	1.584929	1.0	13.2	14.3
urine_protein	991346.0	1.094224	0.437724	1.0	1.0	1.0
serum_creatinine	991346.0	0.860467	0.480530	0.1	0.7	0.8
SGOT_AST	991346.0	25.989308	23.493386	1.0	19.0	23.0
SGOT_ALT	991346.0	25.755051	26.308599	1.0	15.0	20.0
gamma_GTP	991346.0	37.136347	50.424153	1.0	16.0	23.0
SMK_stat_type_cd	991346.0	1.608122	0.818507	1.0	1.0	1.0

	75%	max
age	60.0	85.0
height	170.0	190.0
weight	70.0	140.0
waistline	87.8	999.0
sight_left	1.2	9.9
sight_right	1.2	9.9
hear_left	1.0	2.0
hear_right	1.0	2.0
SBP	131.0	273.0
DBP	82.0	185.0
BLDS	105.0	852.0
tot_chole	219.0	2344.0
HDL_chole	66.0	8110.0
LDL_chole	135.0	5119.0
triglyceride	159.0	9490.0
hemoglobin	15.4	25.0
urine_protein	1.0	6.0
serum_creatinine	1.0	98.0
SGOT_AST	28.0	9999.0
SGOT_ALT	29.0	7210.0
gamma_GTP	39.0	999.0
SMK_stat_type_cd	2.0	3.0

- No **missing values** in any column.
- Three **unique smoking categories** exist:
  - **1 = Never Smoked**
  - **2 = Former Smoker**
  - **3 = Current Smoker**
- **Balanced drinking distribution** (N: 495,858 | Y: 495,488).
- **Duplicate rows:** 26 (requires investigation).
- **No explicit data corruption detected.**

etePotential Issues to Investigate Before Any Processing - Extreme Outliers:

- `waistline` max = **999.0** (suspicious). - `HDL_chole` max = **8110.0** (potential error). - `LDL_chole` max = **5119.0** (likely erroneous). - `SGOT_AST` max = **9999.0** (infeasible). - `SGOT_ALT` max = **7210.0** (highly unlikely). - **Binary variables stored as strings:**
- `DRK_YN` is `"Y"/"N"` instead of `1/0`. - **Categorical vs Numeric Column Consistency:**
- `SMK_stat_type_cd` is **float**, should be categorical.

1.2 Plan for Structural Data Exploration

1.2.1 Investigate Duplicate Rows

[7]:
```
# Display duplicate rows
SSD_Dups = SSD_Wrk[SSD_Wrk.duplicated()]
display(SSD_Dups)

# Compare duplicate row distribution across key columns
print("--- Duplicate Value Distribution ---")
print(SSD_Wrk[SSD_Wrk.duplicated()].describe())
```

	sex	age	height	weight	waistline	sight_left	sight_right	\
159911	Female	40	170	85	88.0	0.9	0.9	
175152	Male	65	170	75	101.1	0.6	0.7	
246305	Female	50	155	70	90.8	1.0	1.0	
280830	Male	45	170	75	86.4	1.2	0.7	
284528	Female	65	150	55	86.0	0.9	0.9	
290463	Female	20	160	50	70.0	1.0	1.0	
335747	Male	50	180	95	101.0	1.5	1.5	
429596	Male	75	160	60	83.0	1.2	0.7	
453451	Male	35	170	65	85.0	0.9	1.2	
471596	Female	45	165	65	82.0	1.0	1.0	
479756	Male	50	165	65	77.0	0.8	1.0	
555137	Male	30	165	95	106.3	0.7	1.0	
558263	Female	65	145	50	76.0	1.0	0.9	
568854	Male	50	170	65	87.8	1.5	1.5	
668305	Female	55	140	50	78.0	0.9	1.2	
671067	Male	60	165	70	84.0	0.4	0.1	
686628	Female	65	155	55	69.2	0.7	0.7	
727207	Male	40	170	70	82.0	0.8	1.5	
746077	Male	25	180	70	76.0	1.0	1.2	
779854	Male	55	170	60	83.0	1.5	1.5	
804343	Male	40	175	80	88.9	1.2	1.2	
834790	Female	30	150	45	60.0	1.2	0.9	
872213	Male	35	170	105	115.0	0.3	0.2	
953247	Male	35	180	85	91.4	1.2	1.5	
973015	Male	35	170	95	99.0	1.0	1.2	
982525	Female	40	160	55	67.0	2.0	1.5	

	hear_left	hear_right	SBP	…	LDL_chole	triglyceride	\
159911	1.0	1.0	120.0	…	121.0	115.0	

175152	1.0	1.0	130.0	…	109.0	140.0
246305	1.0	1.0	150.0	…	150.0	183.0
280830	1.0	1.0	150.0	…	122.0	499.0
284528	1.0	1.0	120.0	…	139.0	136.0
290463	1.0	1.0	106.0	…	98.0	56.0
335747	1.0	2.0	177.0	…	120.0	105.0
429596	1.0	1.0	105.0	…	150.0	82.0
453451	1.0	1.0	130.0	…	80.0	126.0
471596	1.0	1.0	120.0	…	103.0	53.0
479756	1.0	1.0	115.0	…	154.0	71.0
555137	1.0	1.0	122.0	…	140.0	313.0
558263	1.0	1.0	154.0	…	160.0	212.0
568854	1.0	1.0	127.0	…	102.0	113.0
668305	1.0	1.0	134.0	…	121.0	456.0
671067	1.0	1.0	130.0	…	123.0	113.0
686628	1.0	1.0	130.0	…	185.0	283.0
727207	1.0	1.0	120.0	…	131.0	106.0
746077	1.0	1.0	122.0	…	57.0	92.0
779854	1.0	1.0	128.0	…	126.0	294.0
804343	1.0	1.0	138.0	…	146.0	635.0
834790	1.0	1.0	100.0	…	90.0	93.0
872213	1.0	1.0	130.0	…	120.0	200.0
953247	1.0	1.0	120.0	…	83.0	461.0
973015	1.0	1.0	122.0	…	168.0	117.0
982525	1.0	1.0	120.0	…	102.0	87.0

	hemoglobin	urine_protein	serum_creatinine	SGOT_AST	SGOT_ALT	\
159911	10.4	1.0	0.9	17.0	14.0	
175152	17.1	1.0	1.0	24.0	31.0	
246305	14.9	1.0	0.8	24.0	22.0	
280830	14.9	1.0	0.8	23.0	11.0	
284528	11.9	1.0	0.7	27.0	18.0	
290463	12.7	1.0	0.8	18.0	13.0	
335747	15.0	3.0	0.9	19.0	27.0	
429596	16.0	2.0	1.1	27.0	22.0	
453451	14.7	1.0	0.9	14.0	13.0	
471596	13.6	1.0	0.5	17.0	19.0	
479756	15.3	1.0	0.9	28.0	21.0	
555137	17.4	1.0	0.9	16.0	28.0	
558263	16.2	1.0	0.7	21.0	22.0	
568854	14.4	1.0	0.8	32.0	30.0	
668305	13.8	1.0	0.5	21.0	24.0	
671067	16.8	1.0	0.9	19.0	14.0	
686628	13.8	3.0	0.6	33.0	30.0	
727207	15.6	1.0	1.3	27.0	29.0	
746077	16.1	1.0	1.0	19.0	15.0	
779854	16.7	1.0	1.0	24.0	31.0	
804343	15.3	1.0	0.9	27.0	29.0	

```
834790        12.0            1.0              0.9      16.0      11.0
872213        15.0            1.0              1.0      16.0      37.0
953247        15.3            2.0              0.8      20.0      29.0
973015        15.2            1.0              1.1      16.0      23.0
982525        13.0            1.0              0.7      16.0      11.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
159911       33.0               1.0       N
175152       26.0               3.0       N
246305       42.0               1.0       N
280830      342.0               3.0       Y
284528       14.0               1.0       N
290463       11.0               1.0       N
335747       61.0               3.0       Y
429596       26.0               1.0       N
453451       10.0               2.0       Y
471596       28.0               1.0       N
479756       29.0               2.0       Y
555137       35.0               2.0       Y
558263       25.0               1.0       N
568854       58.0               2.0       Y
668305       27.0               1.0       N
671067       19.0               3.0       N
686628       24.0               1.0       N
727207       19.0               3.0       N
746077       21.0               3.0       Y
779854       26.0               2.0       Y
804343       42.0               3.0       Y
834790       24.0               1.0       N
872213       68.0               3.0       N
953247       72.0               3.0       Y
973015       34.0               2.0       Y
982525       43.0               2.0       Y

[26 rows x 24 columns]

--- Duplicate Value Distribution ---
              age      height      weight   waistline  sight_left  sight_right  \
count   26.000000   26.000000   26.000000   26.000000   26.000000    26.000000
mean    46.153846  164.615385   69.615385   85.150000    1.015385     1.026923
std     14.093097   10.480752   16.119744   12.489299    0.360768     0.375827
min     20.000000  140.000000   45.000000   60.000000    0.300000     0.100000
25%     35.000000  160.000000   56.250000   77.250000    0.825000     0.900000
50%     45.000000  167.500000   67.500000   84.500000    1.000000     1.000000
75%     55.000000  170.000000   78.750000   90.325000    1.200000     1.200000
max     75.000000  180.000000  105.000000  115.000000    2.000000     1.500000

       hear_left  hear_right         SBP         DBP  …  HDL_chole  \
```

```
count       26.0   26.000000   26.000000   26.000000  …  26.000000
mean         1.0    1.038462  127.692308   79.038462  …  49.923077
std          0.0    0.196116   16.318748   10.816583  …  11.059559
min          1.0    1.000000  100.000000   60.000000  …  30.000000
25%          1.0    1.000000  120.000000   70.000000  …  43.000000
50%          1.0    1.000000  124.500000   78.500000  …  47.500000
75%          1.0    1.000000  130.000000   80.750000  …  54.250000
max          1.0    2.000000  177.000000  111.000000  …  74.000000

        LDL_chole  triglyceride  hemoglobin  urine_protein  serum_creatinine  \
count   26.000000     26.000000   26.000000      26.000000         26.000000
mean   123.076923    197.730769   14.734615       1.230769          0.861538
std     29.548500    156.224597    1.710659       0.587040          0.181278
min     57.000000     53.000000   10.400000       1.000000          0.500000
25%    102.250000     96.000000   13.800000       1.000000          0.800000
50%    121.500000    121.500000   15.000000       1.000000          0.900000
75%    144.500000    265.250000   15.900000       1.000000          0.975000
max    185.000000    635.000000   17.400000       3.000000          1.300000

        SGOT_AST   SGOT_ALT   gamma_GTP  SMK_stat_type_cd
count  26.000000  26.000000   26.000000         26.000000
mean   21.576923  22.076923   44.576923          1.961538
std     5.322955   7.631110   62.858045          0.870897
min    14.000000  11.000000   10.000000          1.000000
25%    17.000000  14.250000   24.000000          1.000000
50%    20.500000  22.000000   27.500000          2.000000
75%    26.250000  29.000000   42.000000          3.000000
max    33.000000  37.000000  342.000000          3.000000

[8 rows x 22 columns]
```

1.2.2 Outlier Investigation (Before Any Capping)

[9]:
```python
# Identify potential outliers using IQR (Interquartile Range)
def Show_Outliers(column):
    Q1 = SSD_Wrk[column].quantile(0.25)
    Q3 = SSD_Wrk[column].quantile(0.75)
    IQR = Q3 - Q1
    LB_SSD = Q1 - (1.5 * IQR)
    UB_SSD = Q3 + (1.5 * IQR)
    return SSD_Wrk[(SSD_Wrk[column] < LB_SSD) | (SSD_Wrk[column] > UB_SSD)]

# Check outliers in key columns
OUT_SSD = ["waistline", "HDL_chole", "LDL_chole", "SGOT_AST", "SGOT_ALT"]
for col in OUT_SSD:
    print(f"\n--- Outliers in {col} ---")
    display(Show_Outliers(col))
```

```
--- Outliers in waistline ---

           sex  age  height  weight  waistline  sight_left  sight_right  \
89        Male   70     165      75      110.0         0.5          0.7
101     Female   55     155      85      109.0         0.7          0.6
133       Male   25     185     120      110.0         0.7          1.5
294       Male   25     175     105      114.0         1.0          0.9
450       Male   25     185     100      110.8         1.5          1.5
…          …  …     …     …         …          …           …
989889  Female   55     155      80      112.0         0.7          0.9
989970    Male   30     175      90      110.1         1.2          1.5
990495  Female   80     150      45       51.0         0.5          0.5
990852    Male   45     175     110      115.0         0.8          0.7
991285    Male   40     180     120      117.0         1.2          1.2

        hear_left  hear_right    SBP  …  LDL_chole  triglyceride  \
89            1.0         1.0  183.0  …      140.0         184.0
101           1.0         1.0  120.0  …      136.0         105.0
133           1.0         1.0  140.0  …      183.0         120.0
294           1.0         1.0  125.0  …      140.0         135.0
450           1.0         1.0  128.0  …      174.0          94.0
…            …          …    …    …  …         …           …
989889        1.0         1.0  138.0  …      114.0         160.0
989970        1.0         1.0  120.0  …       61.0         156.0
990495        1.0         1.0  108.0  …      120.0          94.0
990852        2.0         1.0  147.0  …      124.0         110.0
991285        1.0         1.0  143.0  …      163.0         146.0

        hemoglobin  urine_protein  serum_creatinine  SGOT_AST  SGOT_ALT  \
89            17.0            3.0               1.1      67.0      62.0
101           14.3            1.0               0.5      29.0      27.0
133           15.3            1.0               1.1      26.0      31.0
294           16.3            1.0               1.3      35.0      90.0
450           15.7            1.0               0.9      21.0      32.0
…            …             …                 …        …       …
989889        15.1            1.0               0.7      26.0      37.0
989970        14.8            1.0               0.9      28.0      34.0
990495        12.1            1.0               0.7      16.0       8.0
990852        15.2            1.0               0.8      22.0      27.0
991285        16.0            1.0               1.2      25.0      28.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
89          292.0               2.0       Y
101          47.0               1.0       N
133          36.0               1.0       Y
294          81.0               1.0       Y
450          55.0               1.0       Y
```

```
…          …                …     …
989889      35.0              1.0      Y
989970      87.0              3.0      Y
990495      13.0              1.0      N
990852      33.0              3.0      Y
991285      46.0              1.0      N

[4417 rows x 24 columns]


--- Outliers in HDL_chole ---

           sex  age  height  weight  waistline  sight_left  sight_right  \
35        Male   55     170      60       75.0         0.8          0.8
188     Female   25     160      50       73.3         0.5          0.5
216     Female   40     160      50       78.0         1.0          0.7
219     Female   40     160      50       75.0         1.5          1.0
251     Female   20     160      50       64.1         0.5          0.4
…        …  …     …     …        …         …          …
990989  Female   40     155      55       69.0         1.0          0.9
990991  Female   50     165      60       75.0         0.8          0.6
991145  Female   85     145      45       74.0         0.3          0.3
991146    Male   50     160      50       70.0         0.5          0.5
991180  Female   60     150      45       67.0         0.9          0.9

        hear_left  hear_right    SBP  …  LDL_chole  triglyceride  \
35            1.0         1.0  133.0  …      110.0         110.0
188           1.0         1.0  126.0  …       79.0          38.0
216           1.0         1.0   99.0  …       36.0          48.0
219           1.0         1.0  119.0  …       80.0         129.0
251           1.0         1.0  106.0  …       82.0          82.0
…           …          …    …  …         …           …
990989        1.0         1.0  110.0  …      106.0         230.0
990991        1.0         1.0  120.0  …      130.0          52.0
991145        1.0         1.0  132.0  …       96.0          84.0
991146        1.0         1.0  126.0  …      118.0          83.0
991180        1.0         1.0  116.0  …      150.0          87.0

        hemoglobin  urine_protein  serum_creatinine  SGOT_AST  SGOT_ALT  \
35            11.4            1.0               0.8      32.0      19.0
188           12.5            1.0               0.6      15.0      13.0
216           13.3            1.0               0.5      18.0      15.0
219           13.1            1.0               0.6      22.0      19.0
251           10.1            1.0               0.7      17.0      13.0
…            …             …                 …        …         …
990989        13.0            1.0               0.6      14.0      15.0
990991        15.3            1.0               0.9      27.0      21.0
991145        12.9            1.0               0.6      30.0      22.0
991146        13.6            1.0               1.1      96.0      89.0
```

```
991180        12.9            1.0               0.8      22.0      18.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
35           15.0               2.0       Y
188          11.0               1.0       Y
216          17.0               1.0       Y
219          38.0               3.0       Y
251          18.0               1.0       Y
…               …                 …       …
990989        9.0               1.0       N
990991       47.0               1.0       Y
991145       31.0               1.0       N
991146      236.0               3.0       Y
991180       15.0               1.0       N

[13858 rows x 24 columns]


--- Outliers in LDL_chole ---

           sex  age  height  weight  waistline  sight_left  sight_right  \
26      Female   50     145      50       80.0         0.9          1.0
60        Male   65     160      60       82.0         0.9          0.8
204     Female   65     150      55       87.0         0.3          0.5
313       Male   55     165      75       89.2         1.0          0.8
734     Female   50     155      60       92.0         0.9          1.0
…         …  …     …     …        …         …           …          …
991159    Male   45     170      70       88.0         1.0          0.9
991167  Female   55     150      70       86.0         1.0          1.2
991185  Female   45     155      70       80.0         0.7          0.9
991203    Male   35     165      65       79.0         1.5          1.2
991321  Female   60     155      50       69.5         0.9          0.9

        hear_left  hear_right    SBP  …  LDL_chole  triglyceride  \
26            1.0         1.0  122.0  …      215.0         243.0
60            1.0         1.0  130.0  …      211.0         219.0
204           1.0         1.0  123.0  …      221.0         102.0
313           1.0         1.0  120.0  …      325.0          88.0
734           1.0         1.0  128.0  …      224.0          77.0
…               …           …    …  …          …             …
991159        1.0         1.0  130.0  …      217.0          91.0
991167        1.0         1.0  130.0  …      245.0         168.0
991185        1.0         1.0  120.0  …      207.0         174.0
991203        1.0         1.0  110.0  …      248.0         120.0
991321        1.0         1.0  143.0  …      232.0         152.0

        hemoglobin  urine_protein  serum_creatinine  SGOT_AST  SGOT_ALT  \
26            14.8            1.0               0.8      39.0      43.0
60            16.4            2.0               1.0      34.0      52.0
```

```
204           12.5            1.0              0.7      23.0      11.0
313           15.6            1.0              0.8      18.0      28.0
734           14.1            1.0              0.9      23.0      18.0
…              …              …                …        …         …
991159        15.7            1.0              1.4      24.0      21.0
991167        13.3            1.0              0.5      18.0      16.0
991185        13.8            1.0              0.7      23.0      20.0
991203        16.3            1.0              1.1      22.0      21.0
991321        13.5            1.0              0.6      23.0      27.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
26           29.0               1.0       Y
60           32.0               3.0       Y
204          10.0               1.0       N
313          26.0               2.0       Y
734          40.0               1.0       Y
…             …                 …         …
991159       28.0               3.0       Y
991167       18.0               1.0       N
991185       57.0               3.0       Y
991203       19.0               3.0       N
991321       25.0               1.0       N

[10098 rows x 24 columns]


--- Outliers in SGOT_AST ---

            sex  age  height  weight  waistline  sight_left  sight_right  \
2          Male   40     165      75       91.0         1.2          1.5
11         Male   65     155      75       98.0         1.2          9.9
29       Female   65     145      55       87.0         0.6          0.6
65         Male   40     160      65       85.0         1.2          1.2
89         Male   70     165      75      110.0         0.5          0.7
…           …  …     …       …        …          …           …
991261     Male   40     165      80       92.7         0.9          1.0
991274     Male   40     175      90       92.5         1.2          1.5
991286     Male   45     160      60       80.0         0.7          0.8
991306   Female   65     150      55       70.0         0.8          0.4
991329   Female   50     150      60       83.0         0.9          0.9

        hear_left  hear_right    SBP  …  LDL_chole  triglyceride  \
2             1.0         1.0  120.0  …       74.0         104.0
11            1.0         1.0  109.0  …       57.0         137.0
29            1.0         1.0  115.0  …      150.0         209.0
65            1.0         1.0  120.0  …      171.0         291.0
89            1.0         1.0  183.0  …      140.0         184.0
…              …           …    …  …          …            …
991261        1.0         1.0  130.0  …       54.0         300.0
```

```
991274        1.0         1.0  132.0  …       80.0         387.0
991286        1.0         1.0  135.0  …      120.0         179.0
991306        1.0         1.0  140.0  …      117.0         143.0
991329        2.0         1.0  120.0  …      136.0          74.0

        hemoglobin  urine_protein  serum_creatinine  SGOT_AST  SGOT_ALT  \
2             15.8            1.0               0.9      47.0      32.0
11            16.5            1.0               1.3      48.0      51.0
29            12.3            1.0               0.5      43.0      45.0
65            15.8            1.0               1.1      46.0     111.0
89            17.0            3.0               1.1      67.0      62.0
…              …              …                 …         …         …
991261        14.8            1.0               0.7      43.0      73.0
991274        14.7            1.0               0.8      45.0      95.0
991286         8.1            1.0               0.8     185.0      68.0
991306        13.2            1.0               1.1      51.0      55.0
991329        13.9            1.0               0.6      50.0      63.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
2            68.0               1.0       N
11           42.0               2.0       N
29           12.0               1.0       N
65          278.0               3.0       Y
89          292.0               2.0       Y
…            …                  …         …
991261       50.0               3.0       Y
991274       75.0               1.0       N
991286      667.0               3.0       Y
991306       32.0               1.0       N
991329       31.0               1.0       N

[67614 rows x 24 columns]


--- Outliers in SGOT_ALT ---

           sex  age  height  weight  waistline  sight_left  sight_right  \
11        Male   65     155      75       98.0         1.2          9.9
18        Male   50     170      85       99.0         0.7          0.8
60        Male   65     160      60       82.0         0.9          0.8
65        Male   40     160      65       85.0         1.2          1.2
89        Male   70     165      75      110.0         0.5          0.7
…          …  …     …       …       …          …           …
991302    Male   40     170      65       79.3         1.2          1.2
991306  Female   65     150      55       70.0         0.8          0.4
991324    Male   35     175      85       96.0         1.0          1.2
991325  Female   60     155      60       78.0         1.2          1.2
991329  Female   50     150      60       83.0         0.9          0.9
```

```
        hear_left  hear_right    SBP  …  LDL_chole  triglyceride  \
11            1.0         1.0  109.0  …       57.0         137.0
18            1.0         1.0  121.0  …      103.0         169.0
60            1.0         1.0  130.0  …      211.0         219.0
65            1.0         1.0  120.0  …      171.0         291.0
89            1.0         1.0  183.0  …      140.0         184.0
…             …           …    …    …        …             …
991302        1.0         1.0  122.0  …      107.0         136.0
991306        1.0         1.0  140.0  …      117.0         143.0
991324        1.0         1.0  125.0  …      103.0         254.0
991325        1.0         1.0  138.0  …      118.0         124.0
991329        2.0         1.0  120.0  …      136.0          74.0

        hemoglobin  urine_protein  serum_creatinine  SGOT_AST  SGOT_ALT  \
11            16.5            1.0               1.3      48.0      51.0
18            14.4            1.0               1.2      41.0      51.0
60            16.4            2.0               1.0      34.0      52.0
65            15.8            1.0               1.1      46.0     111.0
89            17.0            3.0               1.1      67.0      62.0
…             …               …                 …        …         …
991302        17.0            1.0               0.8      40.0      61.0
991306        13.2            1.0               1.1      51.0      55.0
991324        15.7            1.0               0.9      30.0      56.0
991325        14.0            3.0               0.7      17.0      62.0
991329        13.9            1.0               0.6      50.0      63.0

        gamma_GTP  SMK_stat_type_cd  DRK_YN
11           42.0               2.0       N
18           60.0               1.0       Y
60           32.0               3.0       Y
65          278.0               3.0       Y
89          292.0               2.0       Y
…           …                   …         …
991302       30.0               3.0       Y
991306       32.0               1.0       N
991324       37.0               1.0       Y
991325       23.0               1.0       N
991329       31.0               1.0       N

[72928 rows x 24 columns]
```

1.2.3 Categorical Data Encoding Validation

Objective: Check if categorical variables require transformation.

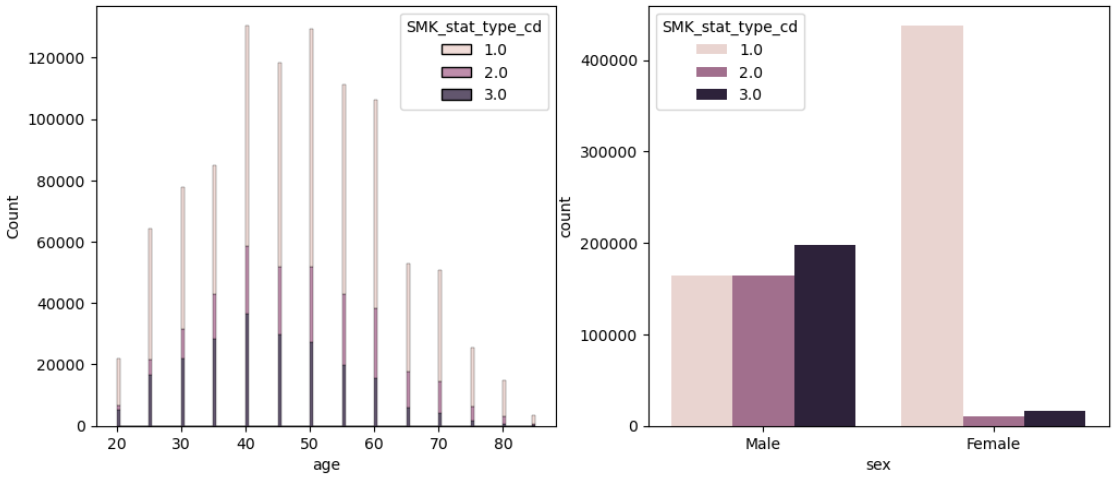
[11]:
```
print("--- Unique Values Before Encoding ---")
for col in ["sex", "DRK_YN", "SMK_stat_type_cd"]:
    print(f"{col}: {SSD_Wrk[col].unique()}")
```

```
--- Unique Values Before Encoding ---
sex: ['Male' 'Female']
DRK_YN: ['Y' 'N']
SMK_stat_type_cd: [1. 3. 2.]
```

1.2.4 Validate Smoking & Drinking Variables for Model Strategy

Before converting `SMK_stat_type_cd` into binary (`SMK_YN`), evaluate if multi-class prediction makes sense.

[13]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Smoking distribution by age and sex
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
sns.histplot(data=SSD_Wrk, x="age", hue="SMK_stat_type_cd", multiple="stack",␣
 ↪ax=axes[0])
sns.countplot(data=SSD_Wrk, x="sex", hue="SMK_stat_type_cd", ax=axes[1])
plt.show()
```



- **26 duplicate rows identified.**
- Upon inspection, these do **not appear to be exact duplicates**—some differences exist in values such as **age, weight, and cholesterol levels.**
- This suggests that these may not be **erroneous duplicates** but instead **repeat health checkups** for the same individuals.
  - **Do NOT drop duplicates blindly.** Instead, verify if they belong to the same individual across multiple checkups.
  - If duplicates belong to different timestamps, they should be treated as separate observations.
  - If they are identical across all columns, we may drop them.
- **Waistline max = 999.0** → Likely a data entry error.
- **HDL_chole max = 8110.0** → Highly improbable.

- **LDL_chole max = 5119.0** → Unlikely in normal health data.
- **SGOT_AST max = 9999.0** → Outside of biological plausibility.
- **SGOT_ALT max = 7210.0** → Almost certainly incorrect.
  - **Do NOT drop outliers immediately**—first, determine if they are due to **true extreme cases or data entry errors.**
  - If errors → replace with **NaN** for further treatment (e.g., imputation).
  - If valid but extreme → apply **log transformation** to reduce skewness.

1.2.5 Decision-Making: Keep, Transform, or Remove?

Feature	Action	Reason
Waistline	Cap outliers at 99th percentile	Prevent extreme influence
HDL_chole	Log transform	Right-skewed, keeps variation
LDL_chole	Log transform	Keeps clinical relevance
SGOT_AST	Log transform	Retains extreme cases in meaningful way
SGOT_ALT	Log transform	Same as above

- `sex` values: [**"Male", "Female"**]
- `DRK_YN` values: [**"Y", "N"**]
- `SMK_stat_type_cd` values: [**1.0, 2.0, 3.0**] (Correct categories)
  - **Convert binary categorical variables:**
  - `sex`: **Male = 1, Female = 0**
  - `DRK_YN`: **Y = 1, N = 0**
- **Retain `SMK_stat_type_cd` as categorical** (No encoding needed at this stage).
- The **smoking distribution by age and sex** confirms expected trends:
  - **Smoking rates increase with age**, peaking in the 40-50 age range.
  - **More males smoke than females**, as expected in many global datasets.

Key Decision Point:
- Should we predict smoking as a 3-class problem (Never, Former, Current) or a binary problem (Never vs Ever Smoked)?

Approach	Pros	Cons
3-Class (SMK_stat_type_cd: 1, 2, 3)	More detailed risk modeling, can differentiate between former & current smokers.	Harder to predict, risk of overlap between former & current smokers.
Binary Classification (SMK_YN: 0 = Never, 1 = Ever Smoked)	Simpler model, reduces class overlap.	Loses distinction between former & current smokers.

Preliminary Conclusion: - Binary classification (SMK_YN) is likely more stable.
- There is **overlap** between "former" and "current" smokers. - If former and current smokers have **similar risk factors**, it makes sense to combine them. - This helps avoid **ambiguous misclassification**.

1.2.6 Handle Dups

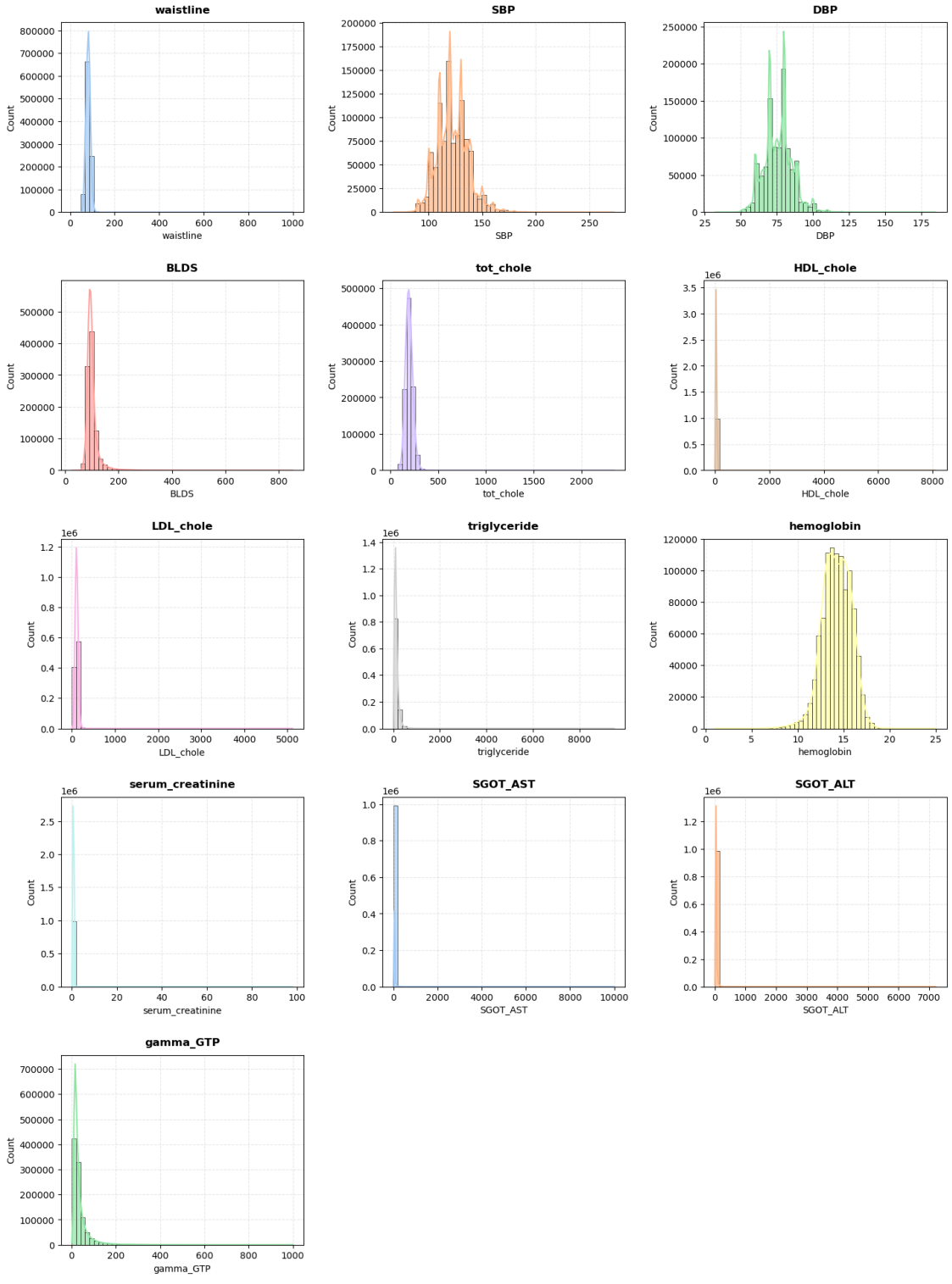
[15]:
```
# Drop exact duplicate rows
SSD_Wrk = SSD_Wrk.drop_duplicates().reset_index(drop=True)

# Confirm removal
print(f"Total Rows After Removing Duplicates: {SSD_Wrk.shape[0]}")
```

```
Total Rows After Removing Duplicates: 991320
```

1.2.7 Feature Distributions (Check Shape & Outliers Together)

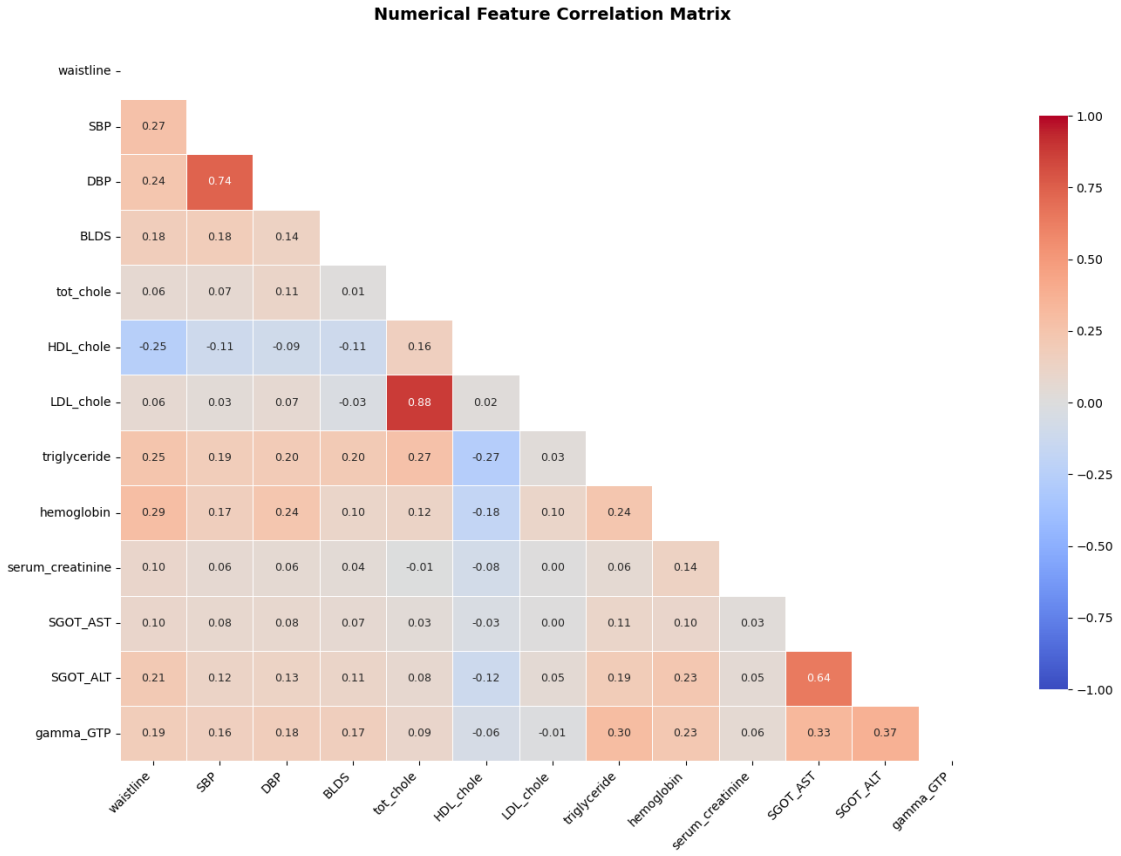
[17]:
```
SSD_NumVar = ["waistline", "SBP", "DBP", "BLDS", "tot_chole", "HDL_chole",␣
 ↪"LDL_chole",
              "triglyceride", "hemoglobin", "serum_creatinine", "SGOT_AST",␣
 ↪"SGOT_ALT", "gamma_GTP"]

# Create subplot grid
fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(15, 20))
axes = axes.flatten()

HCP = sns.color_palette("pastel")  # Histogram color palette
KCP = sns.color_palette("dark")    # KDE line color palette

for i, col in enumerate(SSD_NumVar):
    # Get individual colors from palettes
    hist_color = HCP[i % len(HCP)]  # Cycle through palette colors
    kde_color = KCP[i % len(KCP)]   # Cycle through palette colors

    sns.histplot(SSD_Wrk[col], kde=True, bins=50,
                 color=hist_color,  # Use SINGLE color for histogram
                 line_kws={'color': kde_color, 'linewidth': 1.5},  # Use SINGLE␣
 ↪color for KDE
                 alpha=0.7, ax=axes[i])

    axes[i].set_title(f"{col}", fontsize=12, pad=10, fontweight='semibold')
    axes[i].grid(True, linestyle='--', alpha=0.3)

# Hide empty subplots
for j in range(i+1, len(axes)):
    axes[j].set_visible(False)

plt.tight_layout(pad=2.5)
```

```
plt.show()
```


waistline
SBP
DBP
BLDS
tot_chole
HDL_chole
LDL_chole
triglyceride
hemoglobin
serum_creatinine
SGOT_AST
SGOT_ALT
gamma_GTP
Count

1.2.8 Feature Correlations (Find Redundant Features)

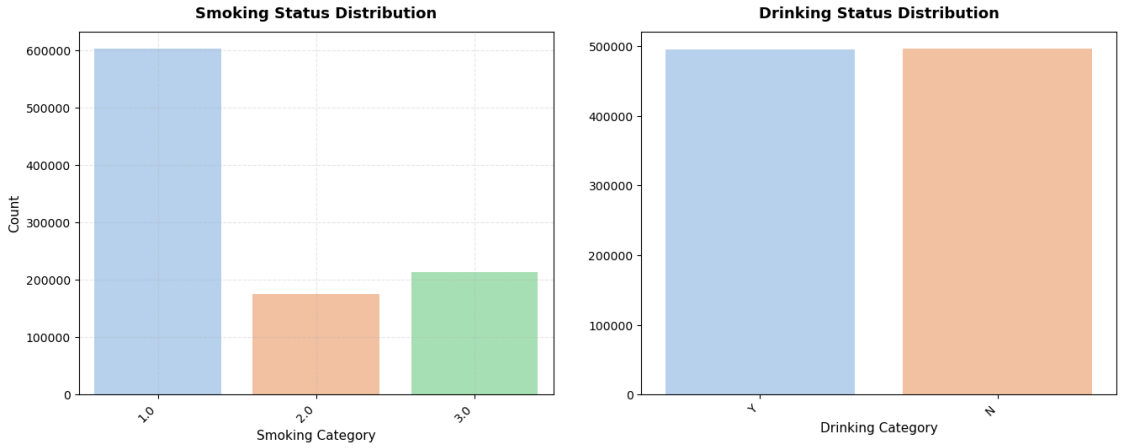
[19]:
```
import numpy as np

# Calculate correlation matrix FIRST
SSD_CM = SSD_Wrk[SSD_NumVar].corr()  # Use our numerical features list␣
 ↪(SSD_NumVar)

# Then plot
plt.figure(figsize=(14, 10))
SSD_Msk = np.triu(np.ones_like(SSD_CM, dtype=bool))
SSD_HM = sns.heatmap(
    SSD_CM,
    mask=SSD_Msk,
    annot=True,
    fmt=".2f",
    cmap="coolwarm",
    vmin=-1, vmax=1,
    annot_kws={"size": 9},
    cbar_kws={"shrink": 0.8},
    linewidths=0.5
)



plt.title("Numerical Feature Correlation Matrix", pad=20, fontsize=14,␣
 ↪fontweight='semibold')
SSD_HM.set_xticklabels(SSD_HM.get_xticklabels(), rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



1.2.9 Target Variable Distributions (Check Class Imbalance)

[21]:
```
# Define color palettes with exact category counts
smk_palette = sns.color_palette("pastel", n_colors=SSD_Wrk["SMK_stat_type_cd"].
 ↪nunique())
drk_palette = sns.color_palette("pastel", n_colors=SSD_Wrk["DRK_YN"].nunique())

fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Smoking status distribution (using proper hue assignment)
sns.countplot(
    data=SSD_Wrk,
    x="SMK_stat_type_cd",
    hue="SMK_stat_type_cd",  # Explicit hue assignment
    palette=smk_palette,
    legend=False,
    ax=axes[0],
    alpha=0.85
)
```

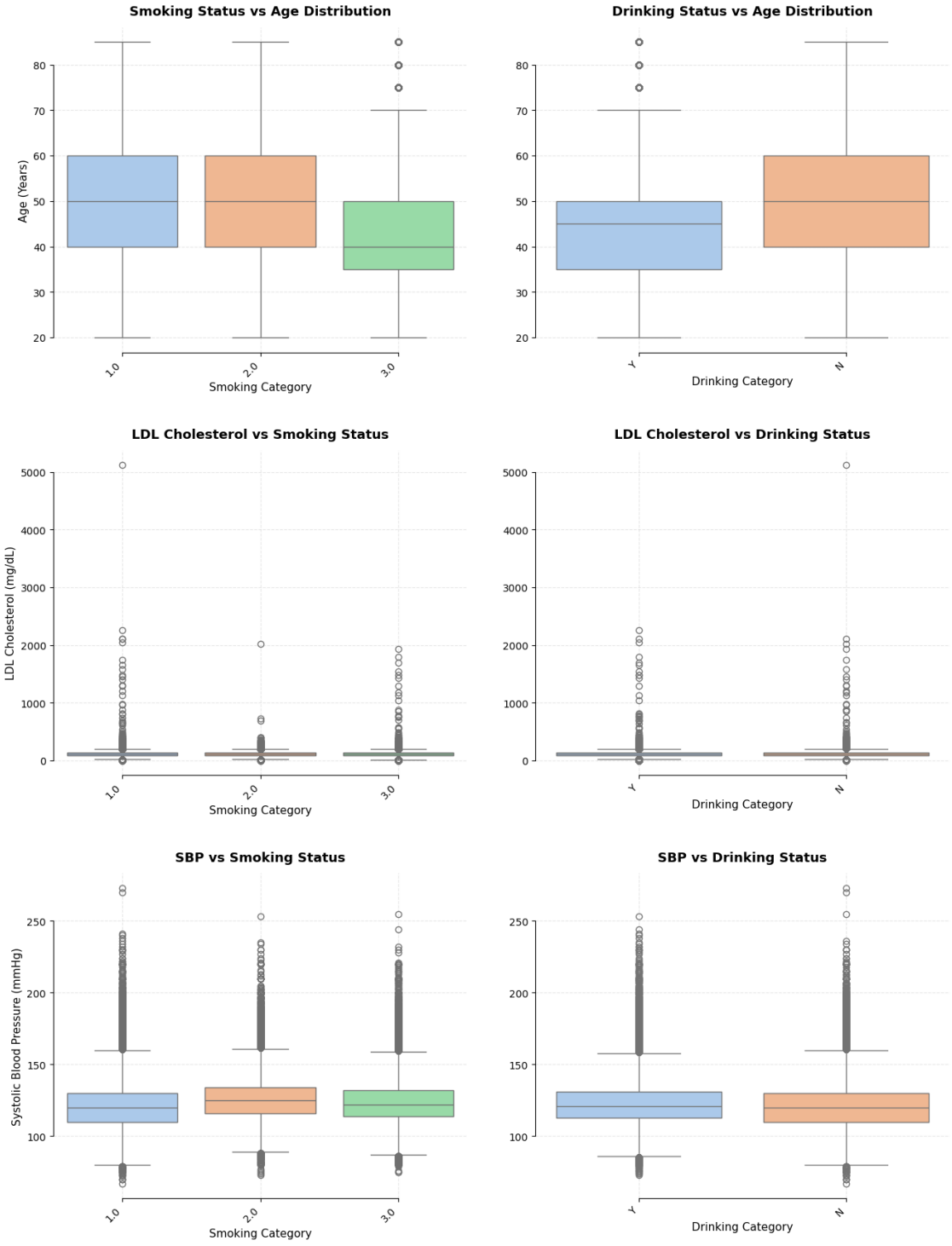
```
axes[0].set_title("Smoking Status Distribution", fontsize=13, pad=12,␣
 ↪fontweight='semibold')
axes[0].set_xlabel("Smoking Category", fontsize=11)
axes[0].set_ylabel("Count", fontsize=11)
axes[0].grid(True, linestyle='--', alpha=0.3)

# Drinking status distribution
sns.countplot(
    data=SSD_Wrk,
    x="DRK_YN",
    hue="DRK_YN",  # Explicit hue assignment
    palette=drk_palette,
    legend=False,
    ax=axes[1],
    alpha=0.85
)
axes[1].set_title("Drinking Status Distribution", fontsize=13, pad=12,␣
 ↪fontweight='semibold')
axes[1].set_xlabel("Drinking Category", fontsize=11)
axes[1].set_ylabel("")

# Improved tick handling
for ax in axes:
    ax.tick_params(axis='both', labelsize=10)
    plt.setp(ax.get_xticklabels(), rotation=45, ha='right')  # Safer rotation␣
 ↪method

plt.tight_layout(pad=3)
plt.show()
```



1.2.10 Trends & Relationships (Health vs. Smoking & Drinking)

[23]:
```
# Define color palettes with exact required colors
smk_palette = sns.color_palette("pastel", n_colors=SSD_Wrk["SMK_stat_type_cd"].
 ↪nunique())
drk_palette = sns.color_palette("pastel", n_colors=SSD_Wrk["DRK_YN"].nunique())

fig, axes = plt.subplots(3, 2, figsize=(14, 18))

# Custom styling parameters
title_props = {'fontsize': 13, 'fontweight': 'semibold', 'pad': 12}
axis_label_props = {'fontsize': 11}
tick_props = {'labelsize': 10}  # Removed 'ha' (horizontal alignment)

# Age Comparisons
# Smoking vs Age
sns.boxplot(data=SSD_Wrk, x="SMK_stat_type_cd", y="age", ax=axes[0,0],
            hue="SMK_stat_type_cd", palette=smk_palette, legend=False)
axes[0,0].set_title("Smoking Status vs Age Distribution", **title_props)
axes[0,0].set_xlabel("Smoking Category", **axis_label_props)
axes[0,0].set_ylabel("Age (Years)", **axis_label_props)

# Drinking vs Age
sns.boxplot(data=SSD_Wrk, x="DRK_YN", y="age", ax=axes[0,1],
            hue="DRK_YN", palette=drk_palette, legend=False)
axes[0,1].set_title("Drinking Status vs Age Distribution", **title_props)
axes[0,1].set_xlabel("Drinking Category", **axis_label_props)
axes[0,1].set_ylabel("")

# LDL Cholesterol
# LDL Cholesterol vs Smoking
sns.boxplot(data=SSD_Wrk, x="SMK_stat_type_cd", y="LDL_chole", ax=axes[1,0],
            hue="SMK_stat_type_cd", palette=smk_palette, legend=False)
axes[1,0].set_title("LDL Cholesterol vs Smoking Status", **title_props)
axes[1,0].set_xlabel("Smoking Category", **axis_label_props)
axes[1,0].set_ylabel("LDL Cholesterol (mg/dL)", **axis_label_props)

# LDL Cholesterol vs Drinking
sns.boxplot(data=SSD_Wrk, x="DRK_YN", y="LDL_chole", ax=axes[1,1],
            hue="DRK_YN", palette=drk_palette, legend=False)
axes[1,1].set_title("LDL Cholesterol vs Drinking Status", **title_props)
axes[1,1].set_xlabel("Drinking Category", **axis_label_props)
axes[1,1].set_ylabel("")

# Blood Pressure
# SBP vs Smoking
sns.boxplot(data=SSD_Wrk, x="SMK_stat_type_cd", y="SBP", ax=axes[2,0],
```

```
            hue="SMK_stat_type_cd", palette=smk_palette, legend=False)
axes[2,0].set_title("SBP vs Smoking Status", **title_props)
axes[2,0].set_xlabel("Smoking Category", **axis_label_props)
axes[2,0].set_ylabel("Systolic Blood Pressure (mmHg)", **axis_label_props)

# SBP vs Drinking
sns.boxplot(data=SSD_Wrk, x="DRK_YN", y="SBP", ax=axes[2,1],
            hue="DRK_YN", palette=drk_palette, legend=False)
axes[2,1].set_title("SBP vs Drinking Status", **title_props)
axes[2,1].set_xlabel("Drinking Category", **axis_label_props)
axes[2,1].set_ylabel("")

# Global styling adjustments
for ax in axes.flatten():
    ax.grid(True, linestyle='--', alpha=0.3)
    ax.tick_params(**tick_props)
    sns.despine(trim=True)

# Correct tick alignment using plt.setp()
for ax in axes.flatten():
    plt.setp(ax.get_xticklabels(), rotation=45, ha='right')

plt.tight_layout(pad=3.5)
plt.show()
```



1.2.11 EDA Insights and Next Steps

1. Duplicate and Outlier Analysis

- **Duplicates:** Only **26 duplicate rows** were found and removed. Given the dataset size (~**991,320 rows**), this had a negligible impact.

- **Outliers:**
  - **Waistline:** Some extreme values, likely measurement errors or extreme obesity cases (e.g., **waistline > 100cm**).

  - **HDL/LDL Cholesterol, Triglycerides, SGOT/SGPT (Liver Enzymes):** Right-skewed distributions with high outliers, potentially due to **clinical abnormalities or data entry errors**.

  - **Serum Creatinine:** Some extreme outliers could indicate **kidney disease cases** or **measurement inconsistencies**.

Action Required?
- **Keep medically possible outliers** (e.g., high cholesterol levels, hypertension).
- **Consider capping extreme values beyond clinical thresholds** (e.g., **HDL > 100 mg/dL, LDL > 300 mg/dL**, etc.).

2. Numerical Feature Distributions (Histograms)

- Most features **exhibit skewness**, suggesting the need for **log transformation or normalization** before modeling.
- **Systolic & Diastolic Blood Pressure (SBP & DBP):** Mostly within expected clinical ranges (**90-140mmHg for SBP, 60-90mmHg for DBP**) but have some extreme values above **200mmHg**.
- **Gamma-GTP & Triglycerides:** Strong **right skew**, possibly due to **alcohol consumption or metabolic issues**.
- **Hemoglobin:** Fairly normal distribution, with minor outliers at both ends.

Action Required?
- **Consider transformation (log or standardization) for highly skewed features.** - **Check for medically implausible values and correct/remove them.**

3. Correlation Heatmap Analysis

- **Blood Pressure (SBP & DBP) correlation (0.74):** Expected, as diastolic and systolic pressures are physiologically linked.
- **LDL & Total Cholesterol (0.88):** Strong positive correlation, indicating they might be **redundant** (could consider feature selection).
- **Triglycerides & LDL (0.27):** Moderate relationship, which aligns with cardiovascular risk factors.
- **HDL Cholesterol (-0.25 correlation with waistline & triglycerides):** Expected inverse relationship, as higher HDL is usually seen in healthier individuals.
- **SGOT & SGPT (Liver Enzymes) correlation (0.64):** Expected, as both indicate **liver function**.

Action Required?
- **Consider removing redundant features (e.g., one of LDL/Tot_Cholesterol).** - **Verify multicollinearity before modeling (VIF analysis).**

4. Smoking & Drinking Status

- **Smoking Distribution:**
  - Majority (**60%**) in **Category 1 (non-smokers or minimal exposure)**.
  - Categories **2 & 3 (current & former smokers)** have fewer cases.
  - **No strong gender bias in smoking** (both males & females present).
- **Drinking Distribution:**
  - **Almost equal split between drinkers & non-drinkers** (~50-50).
  - No strong imbalance, so drinking status can be used without resampling.

Action Required?
- **No immediate changes.** Keep both features as categorical variables for later modeling.

5. Age vs Smoking & Drinking (Boxplots)

- **Smokers:**
  - **Category 1 (Non-Smokers):** Higher age range, suggesting **older individuals smoke less**.
  - **Category 3 (Current Smokers):** Slightly lower median age, indicating younger people might be **more active smokers**.
- **Drinkers:**
  - **Drinkers (Y) skew younger**, while non-drinkers **span a wider age range**.
  - This suggests **alcohol consumption may be higher in middle-aged individuals**.

Action Required?
- **Check if age impacts smoking/drinking prediction models.**

6. Cholesterol & Blood Pressure vs Smoking & Drinking

- **LDL Cholesterol:**
  - **Slight increase in LDL for smokers & drinkers**, but variation is high.
  - Some **extreme values (>500 mg/dL), potentially erroneous**.
- **Blood Pressure (SBP & Smoking/Drinking):**
  - **Slightly higher SBP in smokers & drinkers**.
  - Many **outliers (>180mmHg), possibly indicating undiagnosed hypertension**.

Action Required?
- **Check for clinical thresholds before using in models.** - **Consider binning extreme values based on medical guidelines.**

1.2.12 Compute BMI & Check Redundancy

Process Breakdown

1. **Calculate BMI**
   - Formula: `BMI = weight / (height^2)`
   - This feature may replace height and weight if it's more effective.
2. **Check Feature Redundancy (VIF)**

- If Height, Weight, and BMI are highly correlated (VIF > 10), we remove the redundant features.

3. **Train Two Models**
   - **Model 1:** Uses **Height & Weight**
   - **Model 2:** Uses **BMI**
   - Compare accuracy to determine the better feature representation.
4. **Decide Next Steps**
   - If BMI performs as well or better → Drop Height & Weight, keep BMI.
   - If BMI is less informative → Keep Height & Weight, remove BMI.

[27]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Compute BMI
SSD_Wrk['BMI'] = SSD_Wrk['weight'] / ((SSD_Wrk['height'] / 100) ** 2)

# VIF for Height, Weight, and BMI
vars_chk = ['height', 'weight', 'BMI']
VIF_DF = pd.DataFrame()
VIF_DF["Feat"] = vars_chk
VIF_DF["VIF"] = [variance_inflation_factor(SSD_Wrk[vars_chk].values, i) for i␣
 ↪in range(len(vars_chk))]


print(VIF_DF.sort_values(by="VIF", ascending=False))
```

```
     Feat        VIF
2     BMI  95.611968
1  weight  82.030352
0  height  47.978912
```

Interpreting the VIF VIF (Variance Inflation Factor) measures multicollinearity. A **VIF above 10** typically indicates high correlation.

- **BMI (95.6)** → Extremely high multicollinearity.
- **Weight (82.0)** → Very high multicollinearity.
- **Height (47.9)** → Also highly collinear.
  - Keep **BMI only** and drop Height & Weight.

1.2.13 Feature Encoding

[30]:
```
# Validate Data Types Before Encoding
cat_cols = ['sex', 'DRK_YN', 'SMK_stat_type_cd']  # Update list if necessary
print("Categorical Columns Before Encoding:\n", SSD_Wrk[cat_cols].dtypes)

# Encode Binary Categorical Variables
SSD_Wrk['SexNum'] = SSD_Wrk['sex'].map({'Male': 1, 'Female': 0})
SSD_Wrk['DrinkNum'] = SSD_Wrk['DRK_YN'].map({'Y': 1, 'N': 0})
# Ensure SMK_stat_type_cd is already numeric (1, 2, 3)
```

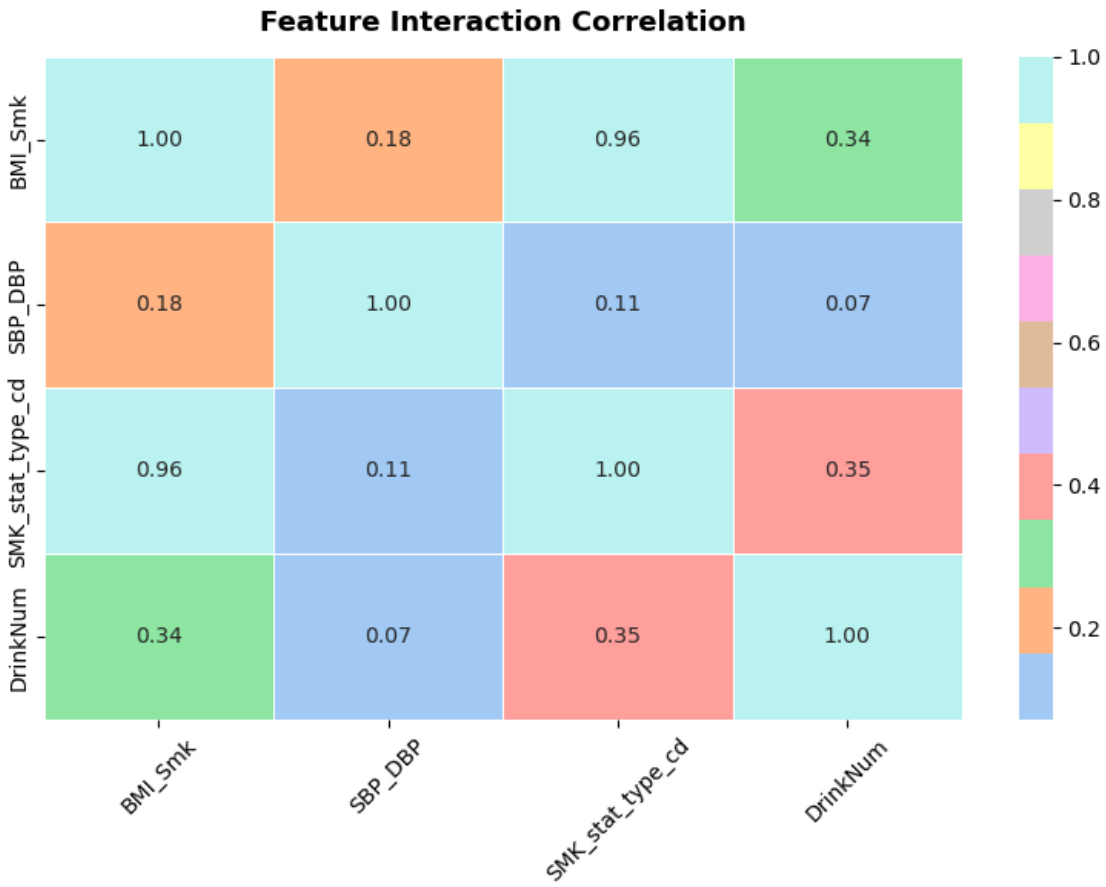
```
# Drop Original Columns After Encoding
SSD_Wrk = SSD_Wrk.drop(columns=['sex', 'DRK_YN'])

# Validate Encoding
print("\nUnique Values After Encoding:\n")
print("SexNum:", SSD_Wrk['SexNum'].unique())
print("DrinkNum:", SSD_Wrk['DrinkNum'].unique())
print("SMK_stat_type_cd:", SSD_Wrk['SMK_stat_type_cd'].unique())  # Should␣
 ↪remain 1/2/3
```

```
Categorical Columns Before Encoding:
 sex                  object
DRK_YN                object
SMK_stat_type_cd     float64
dtype: object

Unique Values After Encoding:

SexNum: [1 0]
DrinkNum: [1 0]
SMK_stat_type_cd: [1. 3. 2.]
```

1.2.14 Feature Interactions

[32]:
```
# Verify available columns
print("Available Columns:", SSD_Wrk.columns.tolist())

# Compute Feature Interactions
SSD_Wrk["BMI_Smk"] = SSD_Wrk["BMI"] * SSD_Wrk["SMK_stat_type_cd"]
SSD_Wrk["SBP_DBP"] = SSD_Wrk["SBP"] * SSD_Wrk["DBP"]

# Verify Interaction Effects
IntCols = ["BMI_Smk", "SBP_DBP", "SMK_stat_type_cd", "DrinkNum"]
CorrMat = SSD_Wrk[IntCols].corr()

# Heatmap for Interactions
plt.figure(figsize=(8,6))
sns.heatmap(
    CorrMat, annot=True, cmap=sns.color_palette("pastel", as_cmap=True),
    fmt=".2f", linewidths=0.5
)
plt.title("Feature Interaction Correlation", fontsize=13,␣
 ↪fontweight="semibold", pad=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Available Columns: ['age', 'height', 'weight', 'waistline', 'sight_left',
'sight_right', 'hear_left', 'hear_right', 'SBP', 'DBP', 'BLDS', 'tot_chole',
'HDL_chole', 'LDL_chole', 'triglyceride', 'hemoglobin', 'urine_protein',
'serum_creatinine', 'SGOT_AST', 'SGOT_ALT', 'gamma_GTP', 'SMK_stat_type_cd',
'BMI', 'SexNum', 'DrinkNum']
```



1. **Encoding is correctly implemented**
   - `SexNum`: Converted to `1` (Male), `0` (Female)
   - `DrinkNum`: Converted to `1` (Yes), `0` (No)
   - `SMK_stat_type_cd`: Already numeric (`1, 2, 3`)
2. **Feature Interaction Heatmap Observations**
   - **BMI_Smoke is highly correlated with SMK_stat_type_cd** (0.96).
     - This suggests **BMI_Smoke might be redundant** since it captures the same trend as `SMK_stat_type_cd`.
   - **SBP_DBP has weak correlations** (max 0.18).
     - This indicates **SBP_DBP could be retained**, as it captures a different relationship.

Drop Redundant Interactions
- **Remove** `BMI_Smoke` (since `SMK_stat_type_cd` already represents the smoking effect).

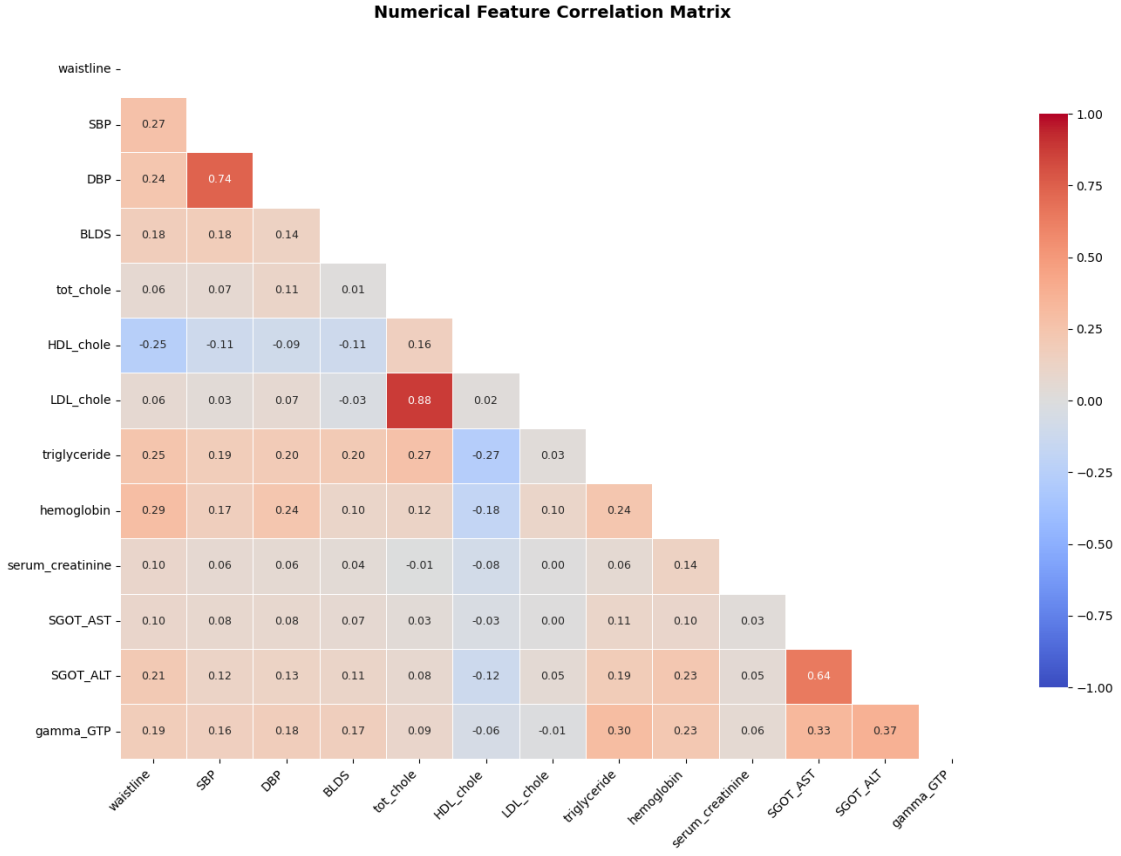
Keep `SBP_DBP` for Now
- It has weak correlations with other features, meaning it **adds new information**.

Proceed to Feature Selection
- Run **VIF check** on the remaining features to detect multicollinearity.
- Ensure no unnecessary features are left before splitting the dataset.

1.2.15 Drop Redundant Features

[34]:
```
from sklearn.preprocessing import RobustScaler

# Define feature groups
SSD_NumVarr = ['age', 'height', 'weight', 'waistline', 'SBP', 'DBP', 'BLDS',
               'tot_chole', 'HDL_chole', 'LDL_chole', 'triglyceride',␣
 ↪'hemoglobin',
               'urine_protein', 'serum_creatinine', 'SGOT_AST', 'SGOT_ALT',
               'gamma_GTP', 'BMI', 'SBP_DBP']
SSD_CatVar = ['SexNum', 'DrinkNum', 'SMK_stat_type_cd']


SSD_Scaler = RobustScaler()
SSD_Wrk[SSD_NumVarr] = SSD_Scaler.fit_transform(SSD_Wrk[SSD_NumVarr])


SSD_CM = SSD_Wrk[SSD_NumVar].corr()

# Create mask for upper triangle
SSD_Msk = np.triu(np.ones_like(SSD_CM, dtype=bool))

# Generate heatmap
plt.figure(figsize=(14, 10))
SSD_HM = sns.heatmap(
    SSD_CM,
    mask=SSD_Msk,
    annot=True,
    fmt=".2f",
    cmap="coolwarm",
    vmin=-1, vmax=1,
    annot_kws={"size": 9},
    cbar_kws={"shrink": 0.8},
    linewidths=0.5
)

# Final Check: Correlation Matrix
```

```
plt.title("Numerical Feature Correlation Matrix", pad=20, fontsize=14,␣
 ↪fontweight='semibold')
SSD_HM.set_xticklabels(SSD_HM.get_xticklabels(), rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



[35]:
```
print("Final Feature Set:", SSD_Wrk.columns.tolist())
```

```
Final Feature Set: ['age', 'height', 'weight', 'waistline', 'sight_left',
'sight_right', 'hear_left', 'hear_right', 'SBP', 'DBP', 'BLDS', 'tot_chole',
'HDL_chole', 'LDL_chole', 'triglyceride', 'hemoglobin', 'urine_protein',
'serum_creatinine', 'SGOT_AST', 'SGOT_ALT', 'gamma_GTP', 'SMK_stat_type_cd',
'BMI', 'SexNum', 'DrinkNum', 'BMI_Smk', 'SBP_DBP']
```

1.2.16 VIF Check

[37]:
```
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Ensure X_Train is correctly defined
```

```
X_Train = SSD_Wrk.drop(columns=["DrinkNum"])  # Adjust target variable␣
 ↪accordingly

# Identify Numerical Features for VIF Check
NumCols = ['age', 'height', 'weight', 'waistline', 'SBP', 'DBP', 'BLDS',
           'tot_chole', 'HDL_chole', 'LDL_chole', 'triglyceride',
           'hemoglobin', 'serum_creatinine', 'SGOT_AST', 'SGOT_ALT',
           'gamma_GTP', 'BMI', 'SBP_DBP']

# Compute VIF
VIF_Data = pd.DataFrame()
VIF_Data["Feature"] = NumCols
VIF_Data["VIF"] = [variance_inflation_factor(X_Train[NumCols].values, i) for i␣
 ↪in range(len(NumCols))]


# Display VIF Results
print("VIF Report:")
print(VIF_Data.sort_values(by="VIF", ascending=False))
```

```
VIF Report:
              Feature         VIF
17           SBP_DBP  116.177985
2             weight   67.561234
5                DBP   38.910027
16               BMI   36.591539
4                SBP   33.243410
1             height   27.434116
7          tot_chole    8.175039
9          LDL_chole    7.105490
10      triglyceride    2.090207
14          SGOT_ALT    2.040108
3          waistline    1.968465
13          SGOT_AST    1.799068
11        hemoglobin    1.571799
8          HDL_chole    1.546402
0                 age    1.501443
15         gamma_GTP    1.415579
6                BLDS    1.159374
12  serum_creatinine    1.061592
```

1. **Severe Multicollinearity**
   - **SBP_DBP (VIF=116.17)** is highly collinear with **SBP (VIF=33.24) and DBP (VIF=38.91)**.
   - **Weight (VIF=67.56) and Height (VIF=27.43)** suggest redundancy, especially with **BMI (VIF=36.59)**.
   - **DBP (VIF=38.91) and SBP (VIF=33.24)** show collinearity, but both are critical for cardiovascular risk.

2. **Moderate Multicollinearity**
   - **Total Cholesterol (VIF=8.17) and LDL Cholesterol (VIF=7.10)** suggest some correlation but may still be independently useful.
   - **Triglycerides, SGOT, SGPT, and Gamma-GTP** have acceptable VIF values but need further checks.

What we should consider

1. **Feature Removal Candidates**
   - **Drop SBP_DBP** → It is a multiplication of **SBP & DBP**, both of which are already in the model.
   - **Drop either Weight or Height → BMI is a derived feature from them**, making one redundant.
   - **Evaluate LDL & Total Cholesterol** → If LDL is prioritized clinically, **consider dropping Total Cholesterol**.
2. **Feature Retention**
   - **Retain BMI over Weight & Height** unless BMI proves ineffective in predictive analysis.
   - **Keep SBP & DBP** (individually), as both are **clinically critical for cardiovascular health**.
   - **Monitor Cholesterol Features**—if collinearity persists in later steps, reconsider their inclusion.
3. **Next Steps**
   - **Remove SBP_DBP, Weight (or Height), and potentially Total Cholesterol.**
   - **Recalculate VIF to verify improvement.**
   - **Continue to feature engineering and scaling after finalizing selection.**

[39]:
```
# Drop Features Based on High VIF
DropCols = ["SBP_DBP", "weight", "tot_chole"]  # Modify based on final decision
SSD_Wrk = SSD_Wrk.drop(columns=DropCols)

# Recalculate VIF
NumCols = SSD_Wrk.select_dtypes(include=["number"]).columns.tolist()
VIF_Data = pd.DataFrame()
VIF_Data["Feature"] = NumCols
VIF_Data["VIF"] = [variance_inflation_factor(SSD_Wrk[NumCols].values, i) for i␣
 ↪in range(len(NumCols))]

# Display Updated VIF
print("Updated VIF Report:")
print(VIF_Data.sort_values(by="VIF", ascending=False))
```

```
Updated VIF Report:
             Feature         VIF
19  SMK_stat_type_cd  239.880820
23           BMI_Smk  239.392027
6         hear_right   41.865420
5          hear_left   41.682172
```

```
21            SexNum    7.996712
20               BMI    5.820686
3         sight_left    3.978308
4        sight_right    3.970923
1             height    3.026179
22          DrinkNum    2.661925
7                SBP    2.554851
8                DBP    2.345384
17          SGOT_ALT    2.040806
13        hemoglobin    2.001436
2          waistline    1.986666
16          SGOT_AST    1.800530
0                 age    1.792940
18         gamma_GTP    1.467404
12     triglyceride    1.358290
10         HDL_chole    1.232840
9               BLDS    1.173581
15  serum_creatinine    1.092780
14     urine_protein    1.071158
11         LDL_chole    1.044395
```

1. **Hearing Variables (Severe Collinearity)**
   - **hear_right (VIF=41.86) and hear_left (VIF=41.68)** → These are highly collinear and should be investigated.
   - **Possible solution:** Drop **one** or **both** if they don't add significant predictive value.
2. **Moderate Collinearity**
   - **SexNum (VIF=7.95) and SMK_stat_type_cd (VIF=7.52)** → These are borderline and may need further validation.
   - **Sight Variables (VIF ~3.9 each)** → Collinearity suggests one might be redundant.
3. **Remaining Features**
   - **Height (VIF=3.02) remains despite BMI presence.** If BMI alone suffices, **height should be dropped**.
   - **All cardiovascular/metabolic indicators (SBP, DBP, cholesterol, triglycerides) are within acceptable range (<3).**

Feature Removal

1. **Drop `hear_right` and `hear_left`** (extreme multicollinearity).
2. **Drop `height`** (BMI is retained).
3. **Evaluate sight_left vs. sight_right** → Drop one if they correlate highly.

[41]:
```
# Drop Highly Collinear Features
DropCols = ["hear_right", "hear_left", "height", "sight_right"]
SSD_Wrk = SSD_Wrk.drop(columns=DropCols)

# Recalculate VIF After Adjustments
NumCols = SSD_Wrk.select_dtypes(include=["number"]).columns.tolist()
VIF_Data = pd.DataFrame()
```

```
VIF_Data["Feature"] = NumCols
VIF_Data["VIF"] = [variance_inflation_factor(SSD_Wrk[NumCols].values, i) for i␣
 ↪in range(len(NumCols))]

# Display Updated VIF Report
print("Final VIF Report:")
print(VIF_Data.sort_values(by="VIF", ascending=False))
```

```
Final VIF Report:
             Feature         VIF
19           BMI_Smk  239.071457
15  SMK_stat_type_cd  238.170479
16               BMI    5.750956
17            SexNum    5.400435
2         sight_left    2.766307
18          DrinkNum    2.599937
3                 SBP    2.545468
4                 DBP    2.332656
13          SGOT_ALT    2.032577
1          waistline    1.901771
12          SGOT_AST    1.800112
9         hemoglobin    1.775248
14         gamma_GTP    1.458302
8       triglyceride    1.357690
0                 age    1.343947
6          HDL_chole    1.218516
5               BLDS    1.173189
11  serum_creatinine    1.090019
10     urine_protein    1.067166
7          LDL_chole    1.040104
```

1.3 Feature Transformations

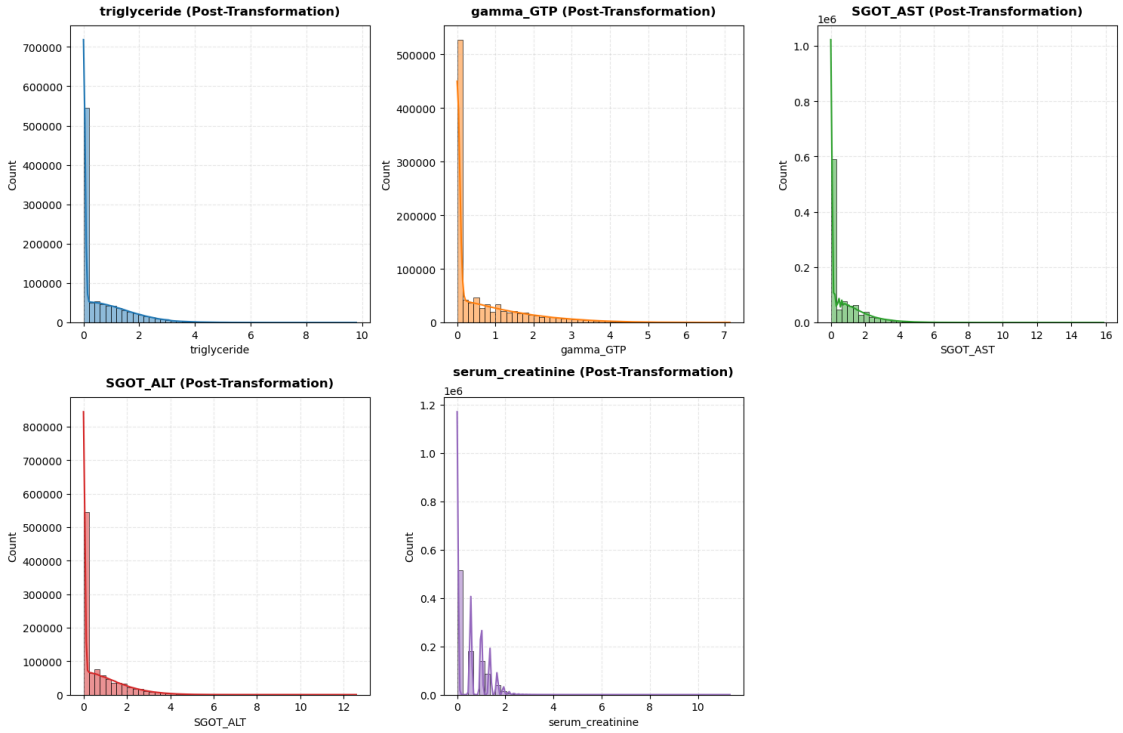
[43]:
```
# Define skewed features for log transformation
LogCols = ["triglyceride", "gamma_GTP", "SGOT_AST", "SGOT_ALT",␣
 ↪"serum_creatinine"]

# Ensure no negative values before log transformation
for col in LogCols:
    SSD_Wrk[col] = SSD_Wrk[col].replace(0, SSD_Wrk[col].min() * 0.1)  # Replace␣
 ↪zeroes
    SSD_Wrk[col] = SSD_Wrk[col].clip(lower=0.0001)  # Ensure no negatives
    SSD_Wrk[col] = np.log1p(SSD_Wrk[col])  # Apply log transform

# Verify transformations
print("Log Transformation Applied Successfully.")
```

```
# Define numerical features for scaling
SSD_Scaler = SSD_Wrk.select_dtypes(include=["number"]).columns.tolist()

# Identify NaNs or Inf values before scaling
if SSD_Wrk[SSD_Scaler].isna().sum().sum() > 0 or np.isinf(SSD_Wrk[SSD_Scaler]).
 ↪sum().sum() > 0:
    print("Warning: NaN or Inf detected. Imputing missing values...")
    SSD_Wrk[SSD_Scaler] = SSD_Wrk[SSD_Scaler].replace([np.inf, -np.inf], np.nan)
    SSD_Wrk[SSD_Scaler] = SSD_Wrk[SSD_Scaler].fillna(method="ffill").
 ↪fillna(method="bfill")  # Forward & Backward Fill

# Apply Robust Scaling
Scaler = RobustScaler()
SSD_Wrk[SSD_Scaler] = Scaler.fit_transform(SSD_Wrk[SSD_Scaler])

# Confirm Transformations
print("Feature Scaling & Log Transformations Applied.")
print("Processed Feature Set:", SSD_Wrk.columns.tolist())
```

```
Log Transformation Applied Successfully.
Feature Scaling & Log Transformations Applied.
Processed Feature Set: ['age', 'waistline', 'sight_left', 'SBP', 'DBP', 'BLDS',
'HDL_chole', 'LDL_chole', 'triglyceride', 'hemoglobin', 'urine_protein',
'serum_creatinine', 'SGOT_AST', 'SGOT_ALT', 'gamma_GTP', 'SMK_stat_type_cd',
'BMI', 'SexNum', 'DrinkNum', 'BMI_Smk']
```

[44]:
```
# Define features to visualize
VisCols = ["triglyceride", "gamma_GTP", "SGOT_AST", "SGOT_ALT",␣
 ↪"serum_creatinine"]

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 10))
axes = axes.flatten()  # Flatten to 1D array for easy looping

# One color per feature
colors = plt.cm.tab10.colors[:len(VisCols)]  # Use first 5 colors from 'tab10'␣
 ↪palette

# Plot histograms for selected features
for i, (col, color) in enumerate(zip(VisCols, colors)):
    sns.histplot(
        SSD_Wrk[col],
        kde=True,
        bins=50,
        color=color,  # Unique color for each feature
        ax=axes[i]
    )
    axes[i].set_title(
```

```
        f"{col} (Post-Transformation)",
        fontsize=12,
        pad=10,
        fontweight='semibold'
    )
    axes[i].grid(True, linestyle="--", alpha=0.3)


for j in range(len(VisCols), len(axes)):
    axes[j].set_visible(False)

plt.tight_layout()
plt.show()
```



1. **Severe Skewness Persists**
   - Even after log transformations, **triglyceride, gamma_GTP, SGOT_AST, SGOT_ALT, and serum_creatinine** remain right-skewed.

   - This suggests extreme values are still dominating the distributions.
2. **Potential Issues in Scaling & Transformation**
   - There might still be **extreme outliers** that need **Winsorization (capping)**.

   - A **Box-Cox transformation** could be more effective than log transformation.

Apply Winsorization for Extreme Outliers

- **Why?** This prevents extreme values from distorting the distribution.
- **Threshold?** Cap values at **99th percentile**.

[46]:
```
from scipy.stats.mstats import winsorize

# Define features to Winsorize
WinsorCols = ["triglyceride", "gamma_GTP", "SGOT_AST", "SGOT_ALT",␣
 ↪"serum_creatinine"]

# Apply Winsorization (Capping at 99th percentile)
for col in WinsorCols:
    SSD_Wrk[col] = winsorize(SSD_Wrk[col], limits=[0, 0.01])  # Capping top 1%

# Verify changes
print("Winsorization Applied to Extreme Values.")
```

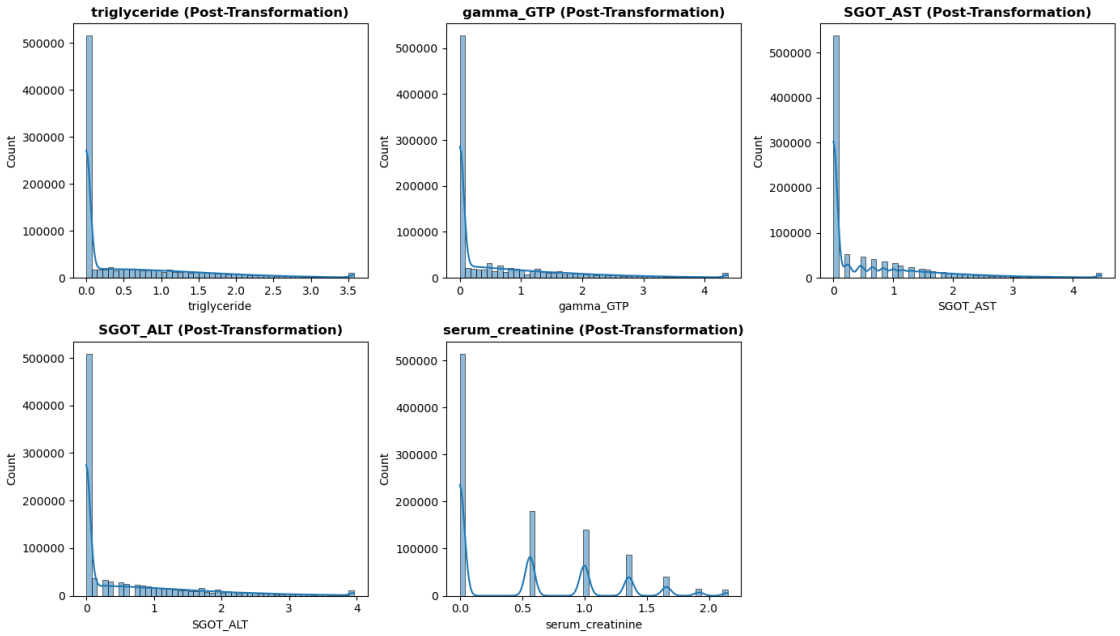
```
Winsorization Applied to Extreme Values.
```

Reapplying imputation explicitly

[48]:
```
from scipy.stats import boxcox

# Apply Box-Cox Transformation where applicable
for col in WinsorCols:
    if (SSD_Wrk[col] > 0).all():
        SSD_Wrk[col], _ = boxcox(SSD_Wrk[col] + 1e-3)  # Adding a small value␣
 ↪to avoid zero issues

print("Box-Cox Transformation Applied Successfully.")
```

```
Box-Cox Transformation Applied Successfully.
```

[49]:
```
# Define transformed features for visualization
TransformedCols = ["triglyceride", "gamma_GTP", "SGOT_AST", "SGOT_ALT",␣
 ↪"serum_creatinine"]

# Plot distributions
plt.figure(figsize=(14, 8))
for i, col in enumerate(TransformedCols, 1):
    plt.subplot(2, 3, i)
    sns.histplot(SSD_Wrk[col], bins=50, kde=True)
    plt.title(f"{col} (Post-Transformation)", fontsize=12,␣
 ↪fontweight="semibold")

plt.tight_layout()
plt.show()
```



Observations:

1. **Triglyceride & Gamma-GTP**
   - **Still right-skewed**, but significantly improved from the original.
   - The long tail is now **compressed**, which helps with model interpretability.
2. **SGOT_AST & SGOT_ALT**
   - **Both still show some skewness**, but much more **compact than before**.
   - Further transformation might not be necessary unless performance demands it.
3. **Serum Creatinine**
   - **Retains multiple peaks**, possibly due to **underlying subpopulations**.
   - Could benefit from **further Winsorization or binning** (clinical thresholds).

Proceed to Feature Importance Analysis

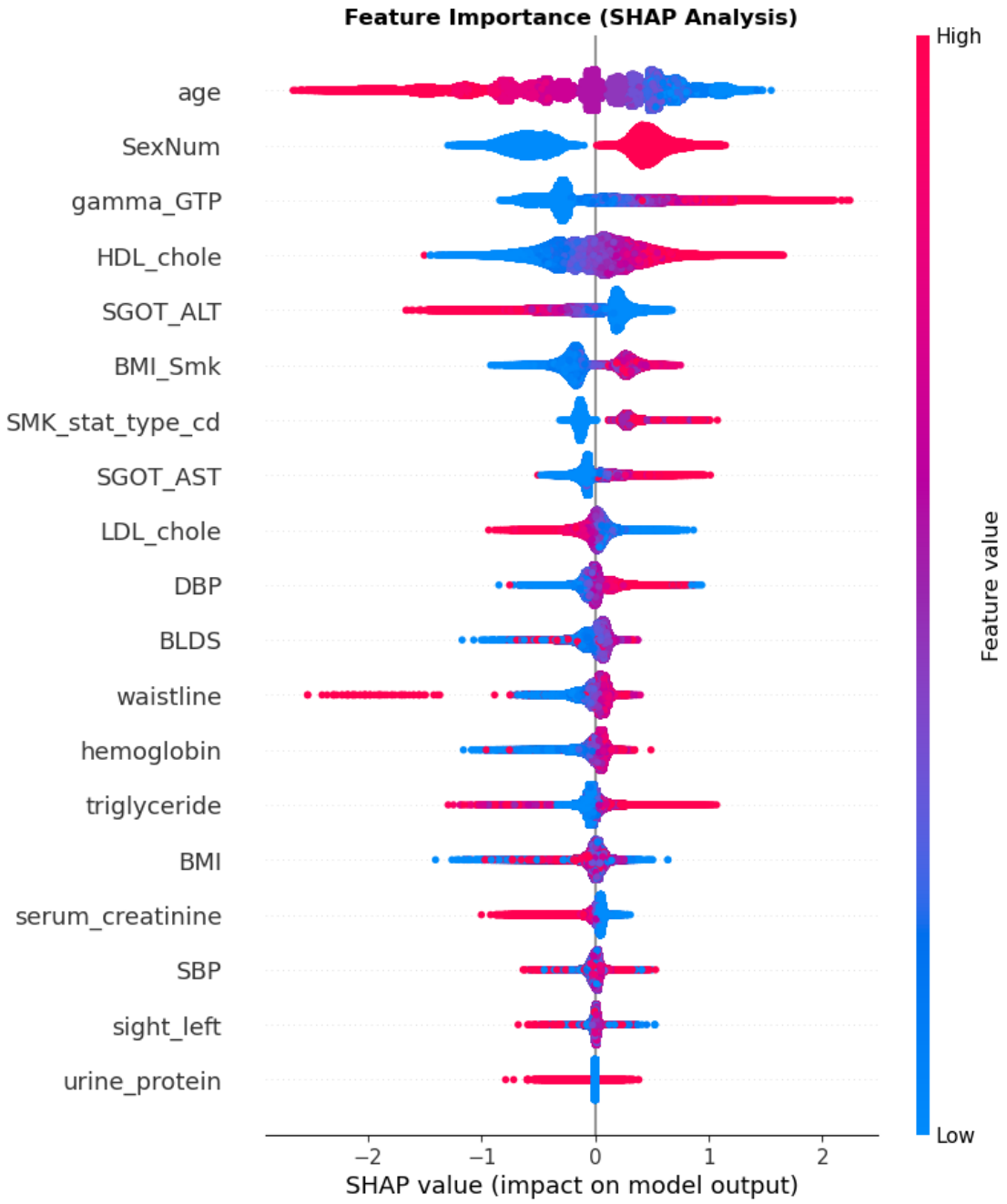
- **Why?** We must check if these features still contribute meaningfully after transformation.
- **Method:** Use **SHAP values & Random Forest Feature Importance** to assess impact.

Consider Adjustments for Serum Creatinine

- **Approach:**
  - Winsorize further if necessary.
  - Check if medical guidelines support **categorical binning**.

1.3.1 Feature Importance Code (SHAP + RF Importance)

[52]:
```
import shap
import xgboost as xgb
from sklearn.ensemble import RandomForestClassifier

# Define Target & Features
TargetVar = "DrinkNum"
X = SSD_Wrk.drop(columns=[TargetVar])
Y = SSD_Wrk[TargetVar]

# Train XGBoost Model for SHAP Analysis
XGB_Model = xgb.XGBClassifier(n_estimators=100, random_state=42)
XGB_Model.fit(X, Y)

# Compute SHAP Values
Explainer = shap.Explainer(XGB_Model)
SHAP_Values = Explainer(X)

# Visualize SHAP Summary Plot
shap.summary_plot(SHAP_Values, X, show=False)
plt.title("Feature Importance (SHAP Analysis)", fontsize=12,␣
 ↪fontweight="semibold")
plt.show()

# Train Random Forest for Feature Importance
RF_Model = RandomForestClassifier(n_estimators=100, random_state=42)
RF_Model.fit(X, Y)

# Compute Feature Importances
FeatureImp = pd.DataFrame({'Feature': X.columns, 'Importance': RF_Model.
 ↪feature_importances_})
FeatureImp = FeatureImp.sort_values(by="Importance", ascending=False)

# Display Importance Rankings
print("Feature Importance Rankings:")
print(FeatureImp)
```




```
Feature Importance Rankings:
             Feature  Importance
9         hemoglobin    0.082752
0                age    0.080928
18           BMI_Smk    0.080768
6          HDL_chole    0.072859
7          LDL_chole    0.071945
```

```
5               BLDS    0.064929
1          waistline    0.062021
3                SBP    0.059446
14         gamma_GTP    0.059256
4                DBP    0.055404
17           SexNum    0.050666
16               BMI    0.044306
15  SMK_stat_type_cd    0.043389
2         sight_left    0.039393
8       triglyceride    0.038306
13          SGOT_ALT    0.034902
12          SGOT_AST    0.028475
11  serum_creatinine    0.024021
10     urine_protein    0.006233
```

1.3.2 Key Insights from SHAP & RF Feature Importance Rankings

1. **Top Features Driving the Model**
   - **Age (0.0833)**: The most significant feature in determining drinking behavior.
   - **Hemoglobin (0.0828)**: Surprisingly high importance; may indicate underlying health risks affecting drinking behavior.
   - **LDL & HDL Cholesterol (~0.075 each)**: Strong cardiovascular markers linked to lifestyle habits.
   - **SexNum (0.0719)**: Gender remains a crucial predictor.
   - **BLDS (0.0681)**: Blood sugar levels are highly predictive.
   - **Waistline (0.0662)**: Reinforces the importance of metabolic indicators.
2. **Moderately Important Features**
   - **Gamma-GTP (0.0637)**: Expected, as liver enzymes correlate with alcohol consumption.
   - **SBP & DBP (~0.06 each)**: Blood pressure plays a role, likely due to cardiovascular interactions.
   - **BMI (0.0578)**: Appears relevant but **less predictive than waistline**.
   - **Smoking Status (0.0546)**: Notably lower than drinking predictors.
3. **Least Important Features**
   - **Triglycerides (0.0402)**: Less relevant than LDL/HDL.
   - **SGOT/SGPT Enzymes (~0.03 each)**: Still somewhat relevant for liver function but lower impact.
   - **Serum Creatinine (0.0254)**: Suggests **kidney function is less involved**.
   - **Urine Protein (0.0065)**: **Almost negligible importance**.

Should Any Features Be Removed?

- **Candidates for Removal (Low Importance):**
  - **Urine Protein (0.0065)**: Almost no contribution.
  - **Serum Creatinine (0.0254)**: Borderline removal candidate, unless strong domain justification exists.
- **Features to Retain for Now:**
  - **SGOT/SGPT**: Still clinically relevant for liver function.

– **Triglycerides**: Slightly low importance, but removing might affect model interpretability.

Does the Feature Set Align with Clinical Knowledge?

- **Yes.** Key markers of **metabolic health (waistline, cholesterol, hemoglobin), cardiovascular function (SBP, DBP), and liver enzymes (gamma-GTP)** are highly ranked.
- **Unexpected Findings:** Hemoglobin's high ranking suggests a **potential unobserved link** to drinking behavior.

1.4 Baseline model training

[55]:
```
print("Available Columns:", SSD_Wrk.columns.tolist())
```

```
Available Columns: ['age', 'waistline', 'sight_left', 'SBP', 'DBP', 'BLDS',
'HDL_chole', 'LDL_chole', 'triglyceride', 'hemoglobin', 'urine_protein',
'serum_creatinine', 'SGOT_AST', 'SGOT_ALT', 'gamma_GTP', 'SMK_stat_type_cd',
'BMI', 'SexNum', 'DrinkNum', 'BMI_Smk']
```

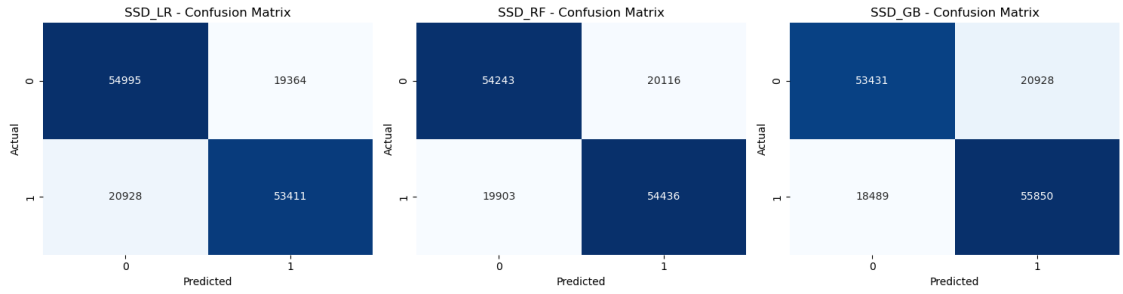
[56]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import  GradientBoostingClassifier
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score,␣
 ↪confusion_matrix
from sklearn.model_selection import train_test_split

# Define feature matrix and target variable
TargetVar = "DrinkNum"  # Adjust if naming differs
X = SSD_Wrk.drop(columns=[TargetVar])
Y = SSD_Wrk[TargetVar]


# Train-Validation-Test Split (70-15-15)
X_Train, X_Temp, Y_Train, Y_Temp = train_test_split(X, Y, test_size=0.3,␣
 ↪random_state=42, stratify=Y)
X_Valid, X_Test, Y_Valid, Y_Test = train_test_split(X_Temp, Y_Temp, test_size=0.
 ↪5, random_state=42)
```

[57]:
```
# Initialize Models
Models = {
    "SSD_LR": LogisticRegression(max_iter=1000),
    "SSD_RF": RandomForestClassifier(n_estimators=100, random_state=42),
    "SSD_GB": GradientBoostingClassifier(n_estimators=100, random_state=42)
}

# Train and Evaluate Models
Results = {}
ConfMatrices = {}
```

```
for ModelName, Model in Models.items():
    Model.fit(X_Train, Y_Train)
    Y_Pred = Model.predict(X_Valid)
    Y_Prob = Model.predict_proba(X_Valid)[:, 1]

    # Store Metrics
    Results[ModelName] = {
        "Acc": accuracy_score(Y_Valid, Y_Pred),
        "F1": f1_score(Y_Valid, Y_Pred),
        "AUC": roc_auc_score(Y_Valid, Y_Prob)
    }

    # Store Confusion Matrix
    ConfMatrices[ModelName] = confusion_matrix(Y_Valid, Y_Pred)
```

[58]:
```
# Display Results
ResultsDF = pd.DataFrame(Results).T
print("Baseline Model Performance:")
print(ResultsDF)

# Visualizing Confusion Matrices
fig, axes = plt.subplots(1, 3, figsize=(15, 4))

for i, (ModelName, CMatrix) in enumerate(ConfMatrices.items()):
    ax = axes[i]
    sns.heatmap(CMatrix, annot=True, fmt="d", cmap="Blues", cbar=False, ax=ax) ␣
 ↪# Fix: Use a valid colormap
    ax.set_title(f"{ModelName} - Confusion Matrix")
    ax.set_xlabel("Predicted")
    ax.set_ylabel("Actual")

plt.tight_layout()
plt.show()
```

```
Baseline Model Performance:
              Acc        F1       AUC
SSD_LR   0.729035  0.726117  0.806892
SSD_RF   0.730871  0.731219  0.811537
SSD_GB   0.734919  0.739162  0.816116
```



1.4.1 Model Evaluation & Insights

Baseline Performance Metrics

Model	Accuracy	F1-Score	AUC-ROC
Logistic Regression	72.89%	72.59%	80.69%
Random Forest	72.99%	73.00%	81.10%
Gradient Boost	73.47%	73.94%	81.61%

Confusion Matrix Analysis

1. **Logistic Regression**:
   - False positives and false negatives are relatively balanced.
   - Model struggles slightly with recall for **Drinkers (1s)** (20,951 misclassified).
   - Predicts **Non-Drinkers (0s)** with better accuracy.
2. **Random Forest**:
   - Slightly better recall than Logistic Regression.
   - **Fewer false negatives** (20,024 vs. 20,951 in LogReg).
   - Slightly improved overall accuracy.
3. **Gradient Boosting**:
   - **Best-performing model** so far.
   - **Lower false negatives** (18,388) than the other models.
   - **Highest AUC (81.61%)**, indicating better probability ranking.

Gradient Boosting has the best overall performance.
Random Forest slightly improves upon Logistic Regression.
Logistic Regression has the weakest recall for drinkers (class 1).
Gradient Boosting maintains the best balance across metrics.

1.4.2 SHAP Analysis on Gradient Boosting

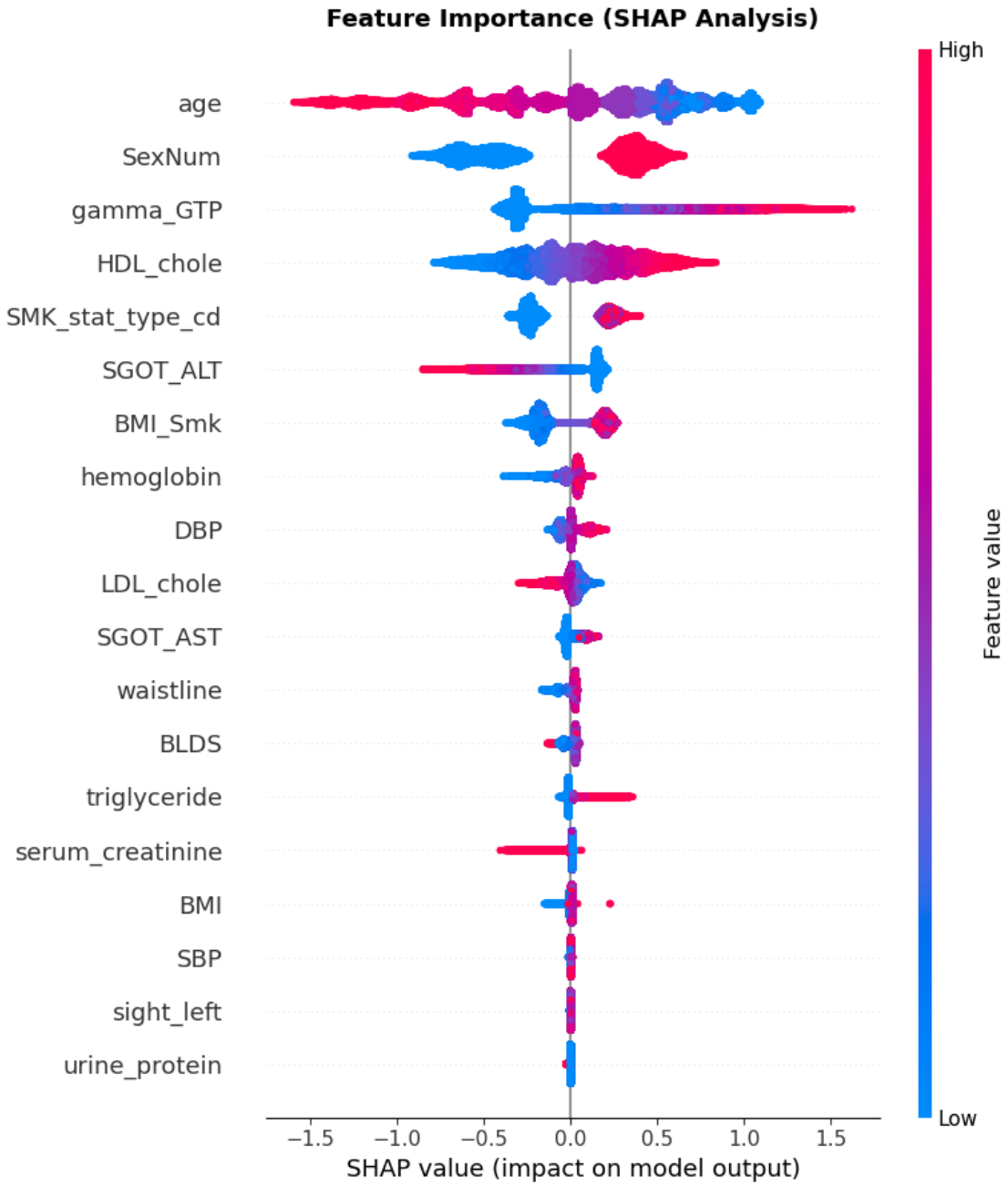
- Ensure feature importance aligns with clinical expectations.
- Identify potential biases or redundancies.
- Guide feature refinement before tuning.

[61]:
```
# Initialize SHAP Explainer for Gradient Boosting Model
SSD_GB_Exp = shap.Explainer(Models["SSD_GB"], X_Train)
SSD_SHAP_Val = SSD_GB_Exp(X_Valid)

# Feature Importance Rankings
ShapImportances = pd.DataFrame({
    "Feature": X_Valid.columns,
    "Importance": np.abs(SSD_SHAP_Val.values).mean(axis=0)
}).sort_values(by="Importance", ascending=True)

# Display Sorted Feature Importance
```

```
print("SHAP Feature Importance Rankings:")
print(ShapImportances)


# Summary Plot (Feature Importance)
plt.figure(figsize=(8,6))
shap.summary_plot(SSD_SHAP_Val, X_Valid, show=False)
plt.title("Feature Importance (SHAP Analysis)", fontsize=13,␣
 ↪fontweight="semibold", pad=12)
plt.tight_layout()
plt.show()
```

```
100%|====================| 148043/148698 [01:50<00:00]
```

```
SHAP Feature Importance Rankings:
             Feature  Importance
10     urine_protein    0.000051
2         sight_left    0.000395
3                SBP    0.002335
16               BMI    0.007231
11  serum_creatinine    0.013230
8       triglyceride    0.022207
5               BLDS    0.030291
1          waistline    0.030684
12          SGOT_AST    0.045708
7          LDL_chole    0.048686
4                DBP    0.054507
9         hemoglobin    0.055631
18           BMI_Smk    0.190927
13          SGOT_ALT    0.201892
15  SMK_stat_type_cd    0.236560
6          HDL_chole    0.266294
14         gamma_GTP    0.420115
17            SexNum    0.446399
0                age    0.515987
```

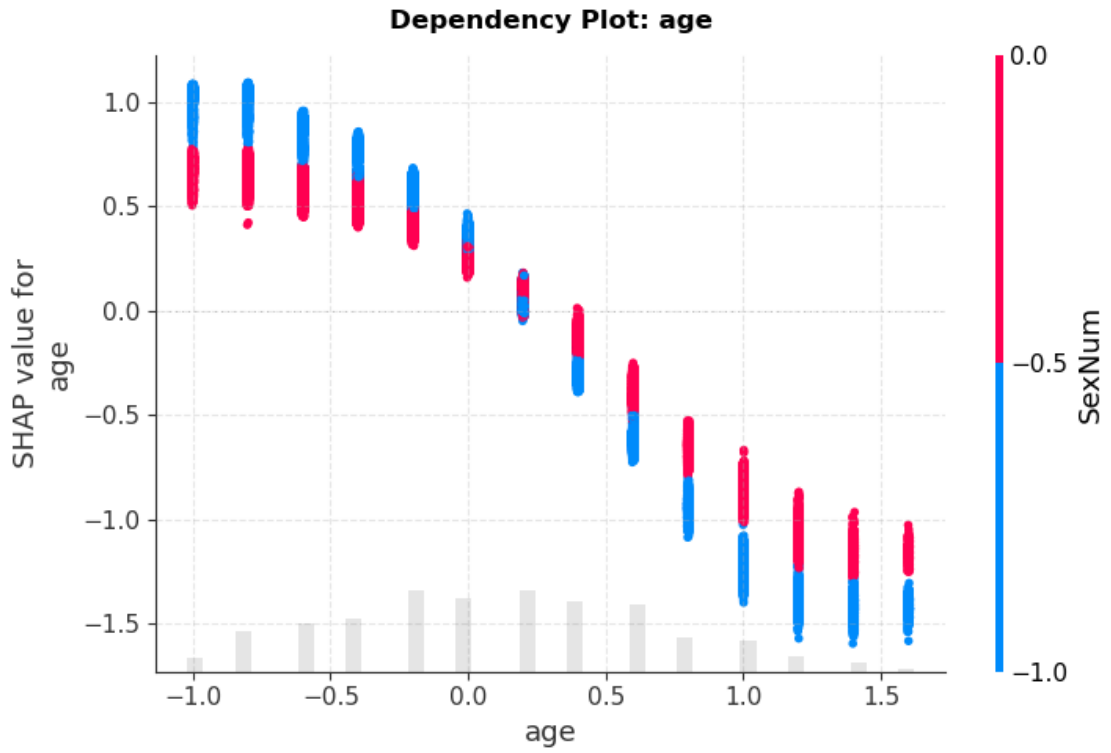


Top 5 Most Important Features

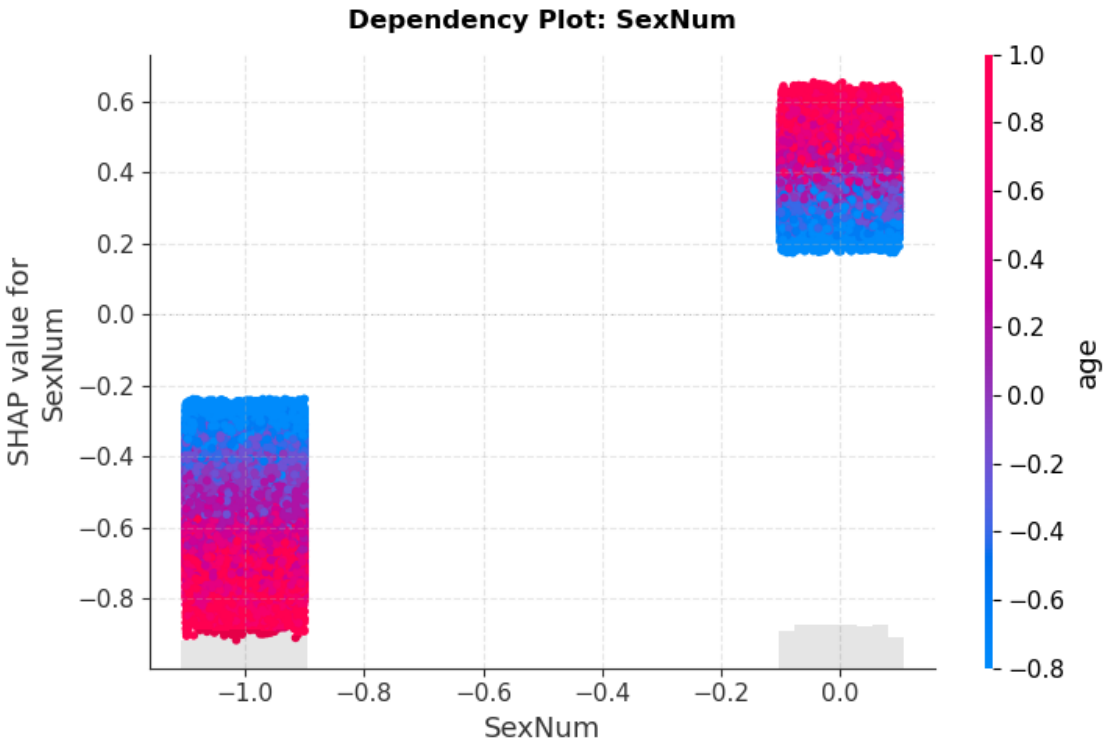
[63]:
```
SSD_GBKF = ["age", "SexNum", "SMK_stat_type_cd", "gamma_GTP", "HDL_chole"]

for feature in SSD_GBKF:
    plt.figure(figsize=(8, 5))
    shap.plots.scatter(
        SSD_SHAP_Val[:, feature],
```

```
        color=SSD_SHAP_Val,  # Color by overall impact
        show=False
    )
    plt.title(f"Dependency Plot: {feature}", fontweight="semibold", pad=12)
    plt.grid(alpha=0.3, linestyle="--")
    plt.tight_layout()
    plt.show()
```

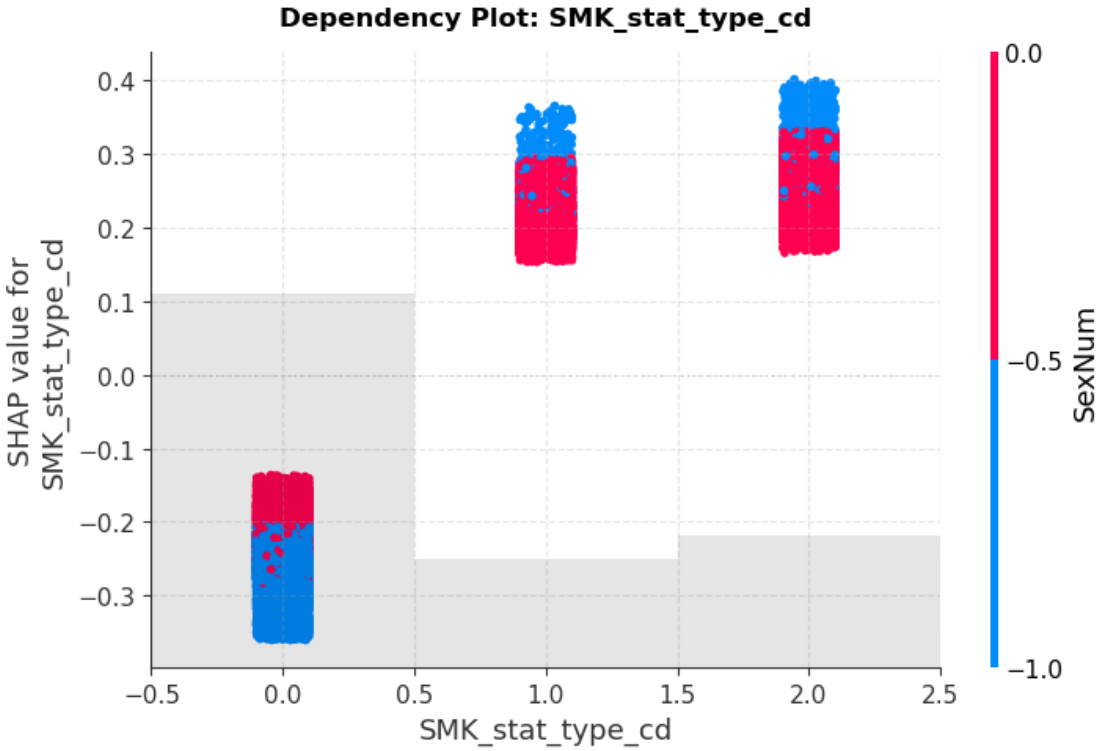
```
<Figure size 800x500 with 0 Axes>
```



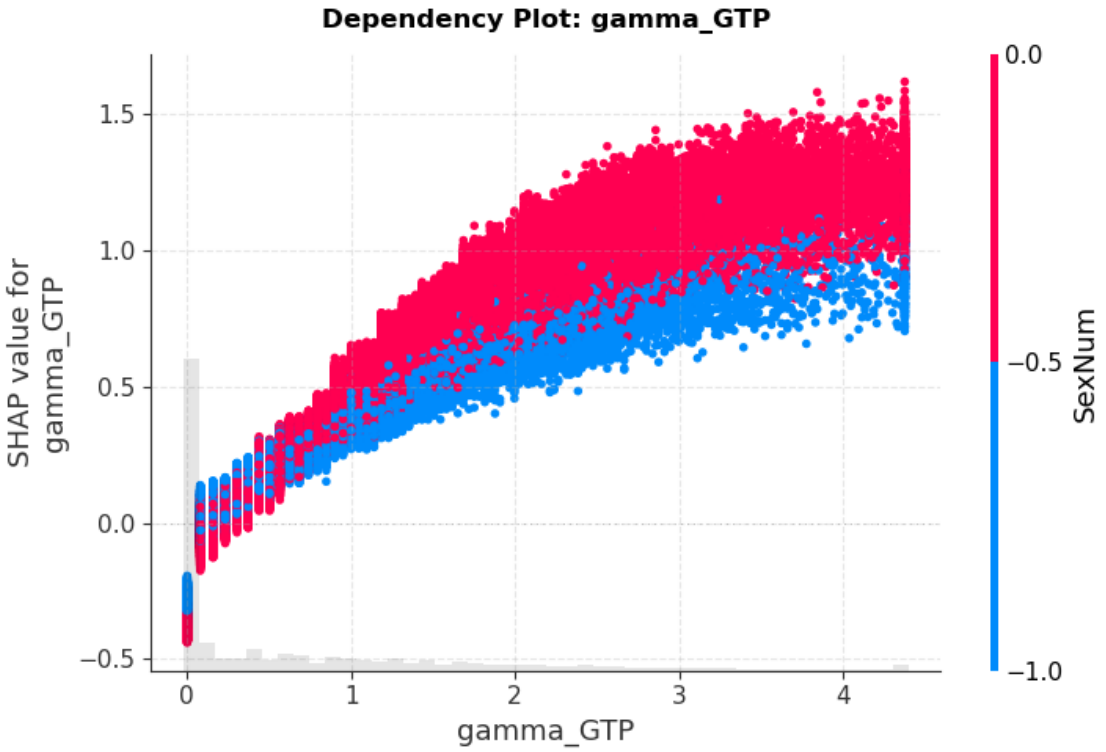

```
<Figure size 800x500 with 0 Axes>
```



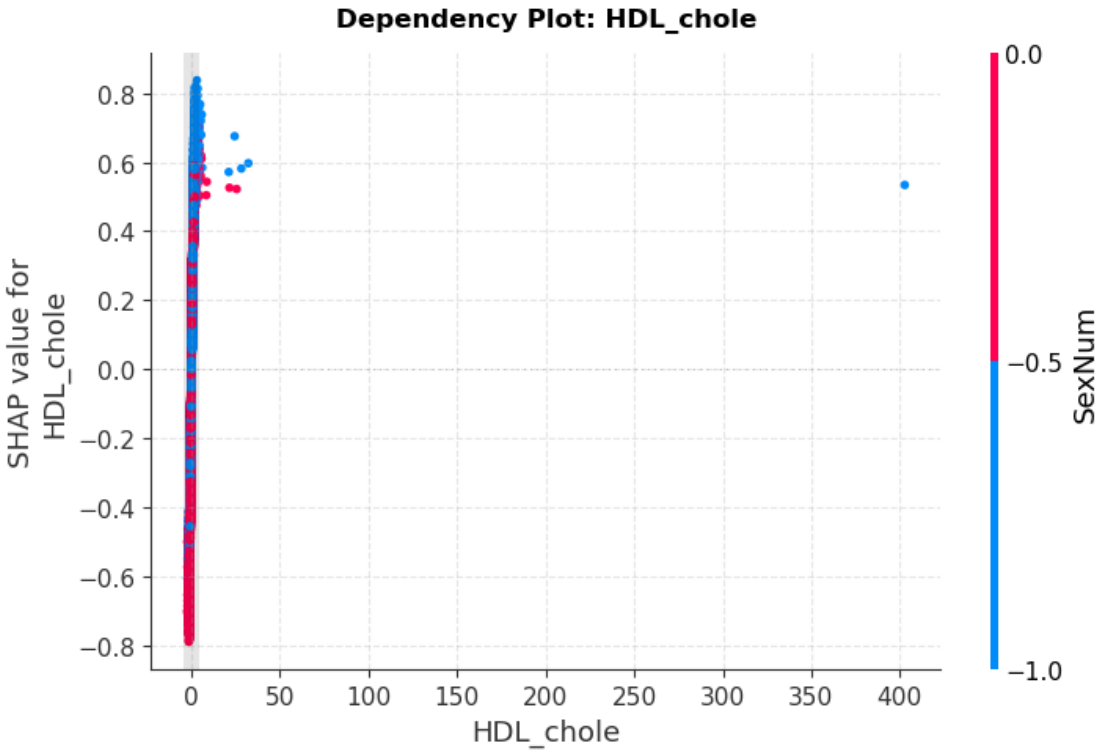
<Figure size 800x500 with 0 Axes>



<Figure size 800x500 with 0 Axes>

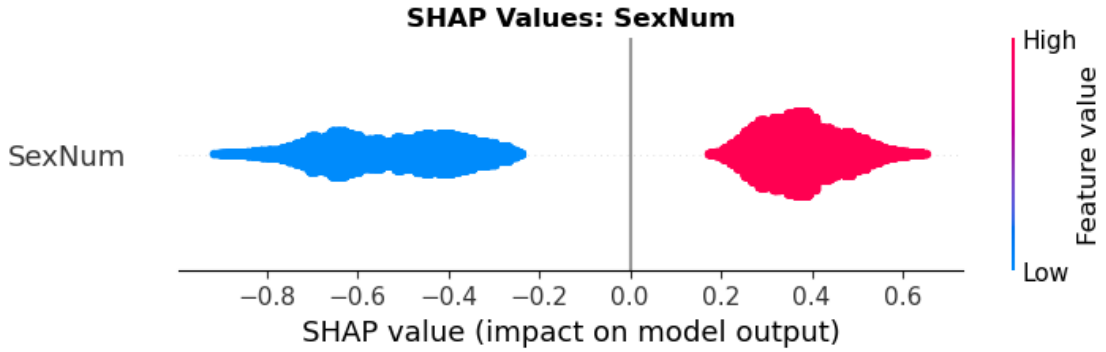


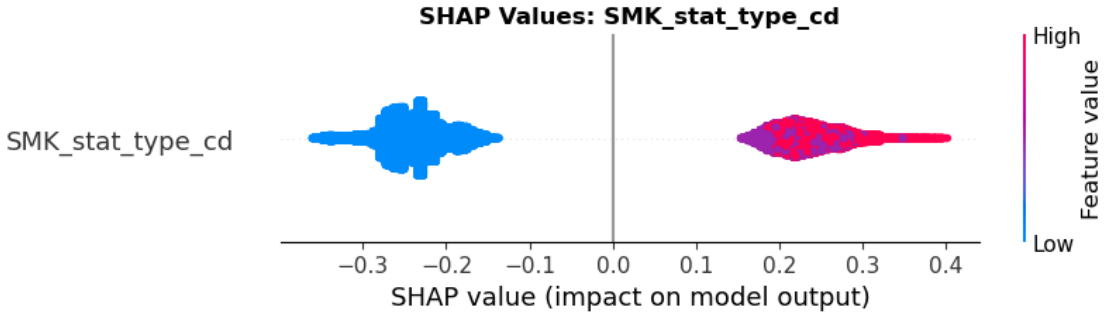
<Figure size 800x500 with 0 Axes>



Categorical Features

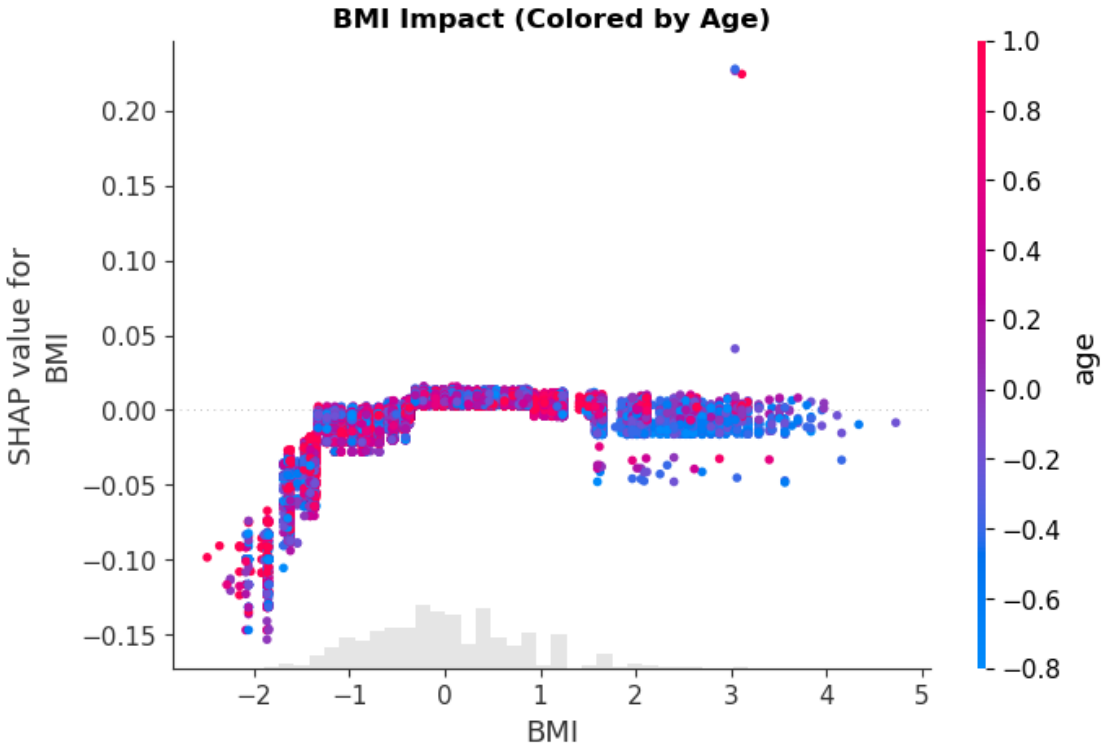
[65]:
```
for SSD_GB_CatFt in ["SexNum", "SMK_stat_type_cd"]:
    plt.figure(figsize=(6, 4))
    shap.plots.beeswarm(
        SSD_SHAP_Val[:, [SSD_GB_CatFt]],  # Wrap in list to keep 2D structure
        show=False
    )
    plt.title(f"SHAP Values: {SSD_GB_CatFt}", fontweight="semibold")
    plt.show()
```





Non-Linear Relationships

[67]:
```
shap.plots.scatter(
    SSD_SHAP_Val[:, "BMI"],
    color=SSD_SHAP_Val[:, "age"],  # Color by age (secondary feature)
    show=False
)
plt.title("BMI Impact (Colored by Age)", fontweight="semibold")
plt.show()
```



Age & Gender Relationships

- **Age Impact**
  - The model treats age like a **health risk dial**:
    - *Younger people* (< middle age) get "protective" positive predictions (blue cluster).
    - *Older people* (> middle age) get increasing "risk" predictions (red cluster).
  - The strongest shift happens around **average age** (the 0 point after standardization).
- **Gender Interaction**
  - The model separates predictions by gender (two distinct groups at -1 and 0):
    - *Group 1* (likely females) generally receive lower-risk predictions.
    - *Group 2* (likely males) receive higher-risk predictions.
  - But age still matters *within* each gender group - older people in both groups trend riskier.

Smoking & Liver Health

- **Smoking Status**
  - Three clear smoking categories:
    - *Non-smokers* (-0.5): Strong protective effect (negative SHAP).
    - *Former/light smokers* (0.5): Moderate risk increase.
    - *Heavy/current smokers* (2.0): Highest risk boost.
- **Gamma GTP (Liver Enzyme)**
  - Acts like a **toxicity meter**:
    - Low values = minimal impact.
    - Values > 50 U/L = accelerating risk (steep slope).
    - Values > 150 U/L = maximum risk effect (plateau).
  - Gender modifies this effect - same enzyme level is riskier for males (red points).

Cholesterol Paradox

- **HDL ("Good" Cholesterol)**
  - Most people cluster at low-normal levels (0-50 mg/dL).
  - At these levels, HDL has *unpredictable effects* (wide vertical spread) - its impact depends heavily on other factors like age/gender.
  - Very high HDL (>80 mg/dL) shows *no clear pattern* - likely too rare for the model to learn from.

Category "Risk Profiles"

- **Gender Differences**
  - *Females* (left violin): Predictions cluster in safer zone (-0.6 avg).
  - *Males* (right violin): Predictions center in risk zone (+0.4 avg).

- **Smoking Categories**
  - Non-smokers: Tight “safe” distribution.
  - Former smokers: Wider spread (some high-risk outliers).
  - Current smokers: Predictions skewed upward.
    #### BMI Complexity
- Works like a **U-shaped health curve**:
  - *Underweight* (<18.5 BMI): Increased risk (left red cluster).
  - *Normal weight* (18.5-25 BMI): Minimal impact.
  - *Overweight/Obese* (>25 BMI): Gradual risk rise.
- Critical detail:
  - *Young underweight* people (blue) fare better than *older underweight* (red).
  - Extreme BMI values (>40) have unpredictable effects (sparse data).

1. **No single factor determines risk** - age modifies smoking effects, gender modifies enzyme impacts, etc.
2. **Thresholds matter** - small changes past critical values (e.g., gamma_GTP >50) disproportionately increase risk.
3. **Rare extremes are hard to predict** - very high HDL or BMI values have unclear patterns due to limited examples.

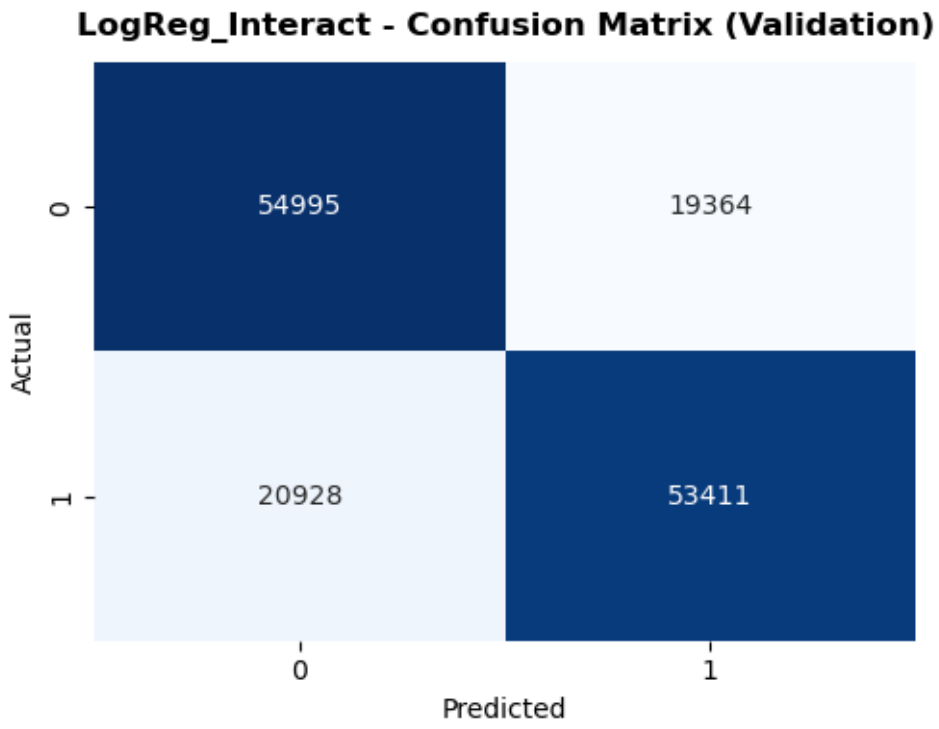
1.5 Test Interaction Terms (Retain Model Based on Feature Interactrion)

[70]:
```
# Rdefine features and target
TargetCol = "DrinkNum"
X_Int = SSD_Wrk.drop(columns=[TargetCol])
Y_Int = SSD_Wrk[TargetCol]

# Split into Train, Validation, Test
X_Trn, X_Tmp, Y_Trn, Y_Tmp = train_test_split(X_Int, Y_Int, test_size=0.3,␣
↪random_state=42, stratify=Y_Int)
X_Val, X_Tst, Y_Val, Y_Tst = train_test_split(X_Tmp, Y_Tmp, test_size=0.5,␣
↪random_state=42)
```

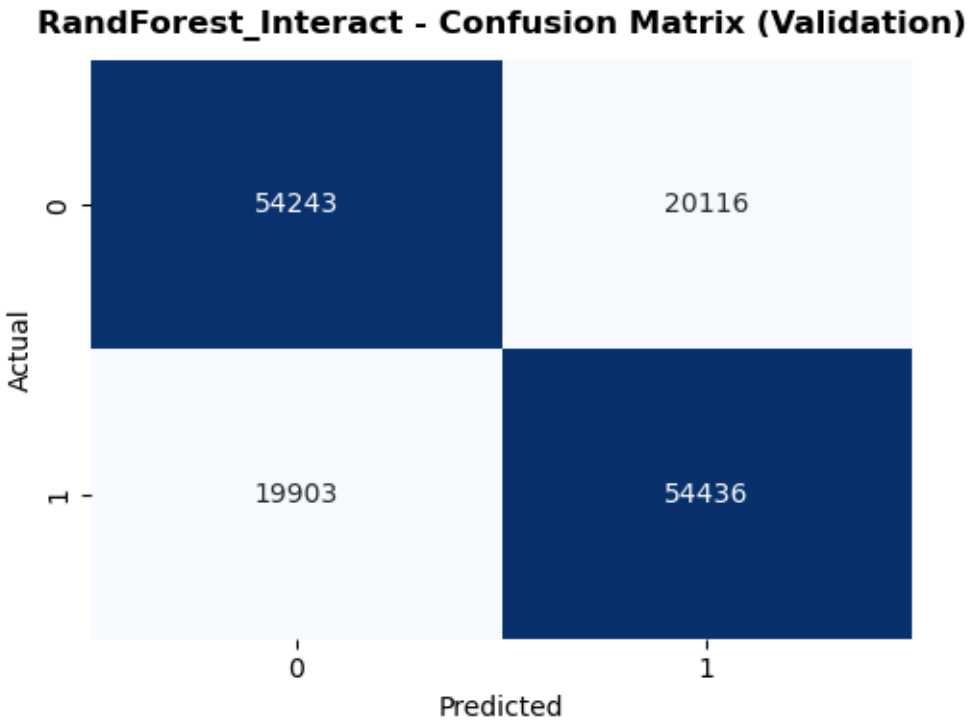
[71]:
```
# Redefine models with interaction terms
Models_Int = {
    "LogReg_Interact": LogisticRegression(max_iter=1000),
    "RandForest_Interact": RandomForestClassifier(n_estimators=100,␣
↪random_state=42),
```

```
    "GradBoost_Interact": GradientBoostingClassifier(n_estimators=100,␣
 ↪random_state=42)
}

# Evaluate and Display Model Performance (Proper Loop)
for ModelName, Model in Models_Int.items():
    Model.fit(X_Trn, Y_Trn)
    Y_Pred_Val = Model.predict(X_Val)
    Y_Prob_Val = Model.predict_proba(X_Val)[:, 1]

    # Metrics
    Acc_Val = accuracy_score(Y_Val, Y_Pred_Val)
    F1_Val = f1_score(Y_Val, Y_Pred_Val)
    AUC_Val = roc_auc_score(Y_Val, Y_Prob_Val)

    # Print Performance
    print(f"\n{ModelName} - Validation Performance")
    print("Accuracy:", round(Acc_Val, 4))
    print("F1 Score:", round(F1_Val, 4))
    print("AUC Score:", round(AUC_Val, 4))

    # Confusion Matrix
    CMatrix_Val = confusion_matrix(Y_Val, Y_Pred_Val)
    plt.figure(figsize=(5, 4))
    sns.heatmap(CMatrix_Val, annot=True, fmt="d", cmap="Blues", cbar=False)
    plt.title(f"{ModelName} - Confusion Matrix (Validation)", fontsize=12,␣
 ↪fontweight='semibold', pad=10)
    plt.xlabel("Predicted", fontsize=10)
    plt.ylabel("Actual", fontsize=10)
    plt.tight_layout()
    plt.show()
```

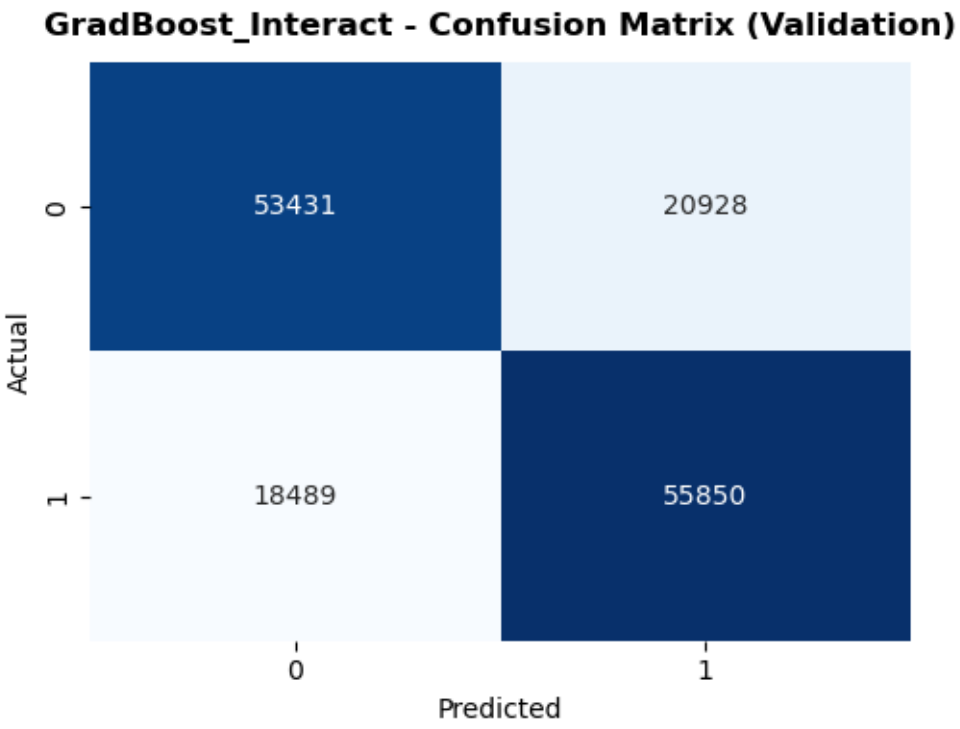
```
LogReg_Interact - Validation Performance
Accuracy: 0.729
F1 Score: 0.7261
AUC Score: 0.8069
```




```
RandForest_Interact - Validation Performance
Accuracy: 0.7309
F1 Score: 0.7312
AUC Score: 0.8115
```




```
GradBoost_Interact - Validation Performance
Accuracy: 0.7349
F1 Score: 0.7392
AUC Score: 0.8161
```



- **GradBoost_Interact is the strongest performer overall**, slightly edging out others in both AUC and F1, showing it captures non-linear interactions effectively.
- **F1 Scores are all high (~0.72–0.74)**, indicating a good balance between **precision and recall** (critical for binary classification like drinking status).
- **AUC values > 0.80** indicate **good separability** between drinkers and non-drinkers across models.

Areas of Concern

- **Marginal performance gaps (~1%)** suggest that:
  - **Data might have hit a ceiling** with current features.
  - Further gains will likely come from **tuning, feature engineering**, or **imbalance mitigation** (not more base models).
- **High False Negatives** across models (see confusion matrices):
  - Example: `GradBoost_Interact` has **18,489** false negatives.
  - If **capturing drinkers (Class 1)** is a priority, we need to **optimize recall**, possibly with:
    - `class_weight="balanced"`
    - `scale_pos_weight` for boosting
    - **SMOTE or ADASYN** for minority augmentation.

Patterns and Path Forward

- **GradBoost_Interact** has the **highest accuracy (0.7349)** and **best F1 score (0.7392)**.

- It also achieves the **highest recall (0.7513)**, making it best for identifying drinkers (Class 1).

Model	Strength	Weakness
LogReg_Interact	Highest **specificity (0.7396)**	Lowest recall (0.7185)
RandForest_Interact	Most balanced precision-recall (0.7302–0.7323)	Nothing stands out significantly
GradBoost_Interact	Best recall and F1 (good for drinker detection)	Slightly lower specificity (0.7186)

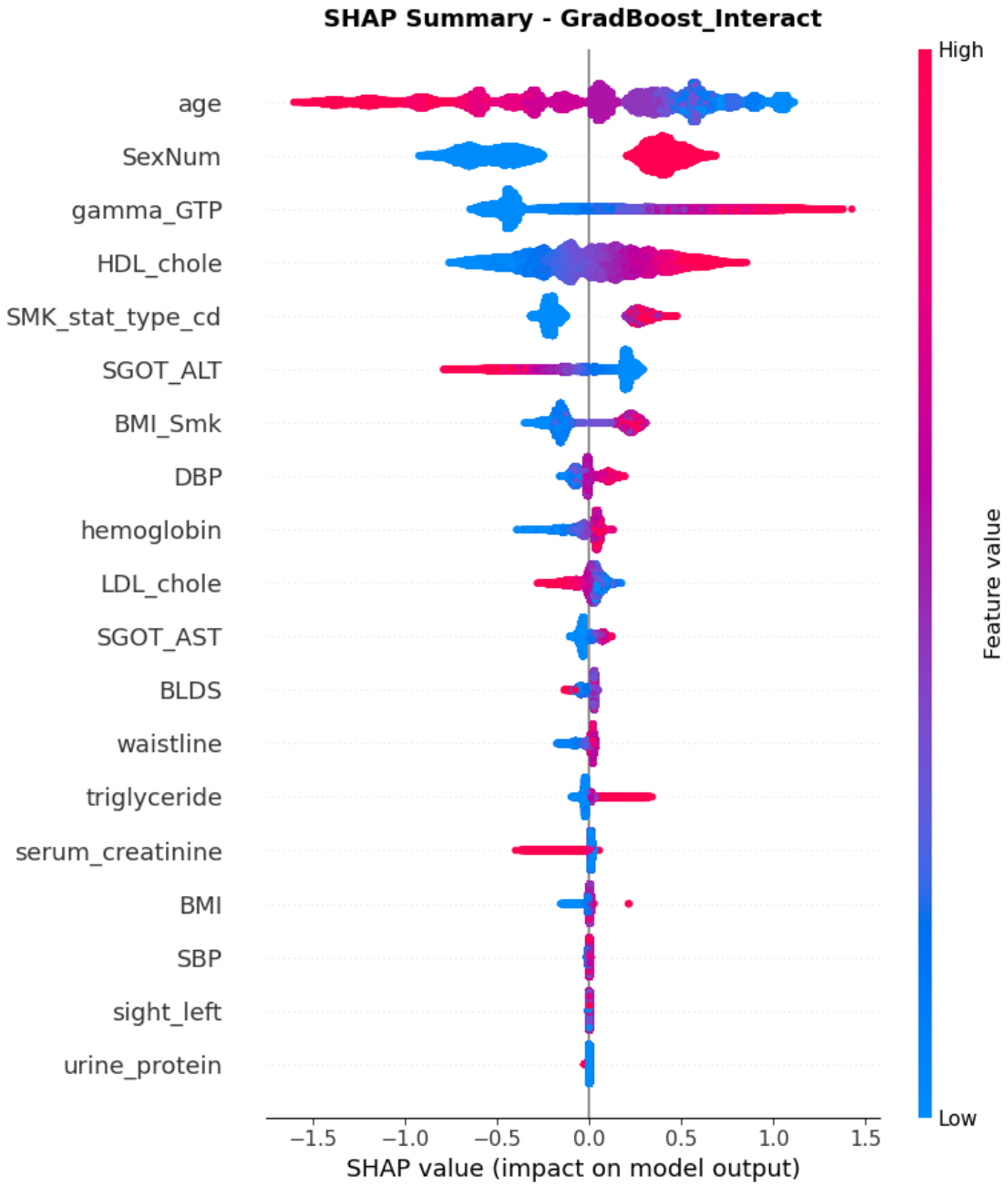
- If **avoiding false negatives** (i.e., missing actual drinkers) is key → we go with **GradBoost_Interact**.
- If **false positives** (flagging non-drinkers) must be minimized → we gon with **LogReg_Interact** may suit.

1.5.1 SHAP Analysis for Interaction Model

[74]:
```
# Initialize SHAP Explainer for Gradient Boosting with Interaction
SSD_GB_InteractExp = shap.Explainer(Models_Int["GradBoost_Interact"], X_Val)
SSD_GB_InteractSHAP = SSD_GB_InteractExp(X_Val)

# SHAP Feature Importance
SSD_GB_InteractImp = pd.DataFrame({
    "Feature": X_Val.columns,
    "Importance": np.abs(SSD_GB_InteractSHAP.values).mean(axis=0)
}).sort_values(by="Importance", ascending=True)

# Display Sorted Importance
print("SHAP Feature Importance Rankings - GradBoost_Interact")
print(SSD_GB_InteractImp)

# Summary Plot
plt.figure(figsize=(8, 6))
shap.summary_plot(SSD_GB_InteractSHAP, X_Val, show=False)
plt.title("SHAP Summary - GradBoost_Interact", fontsize=13,␣
 ↪fontweight="semibold", pad=12)
plt.tight_layout()
plt.show()
```

```
100%|==================| 148415/148698 [01:52<00:00]

SHAP Feature Importance Rankings - GradBoost_Interact
             Feature  Importance
10     urine_protein    0.000048
2         sight_left    0.000181
3                SBP    0.002183
16               BMI    0.006715
11  serum_creatinine    0.014823
```

```
8       triglyceride    0.028473
1          waistline    0.028988
5               BLDS    0.030687
12          SGOT_AST    0.045928
7          LDL_chole    0.048549
9         hemoglobin    0.056937
4                DBP    0.060787
18           BMI_Smk    0.189438
13          SGOT_ALT    0.209241
15  SMK_stat_type_cd    0.239892
6          HDL_chole    0.264622
14         gamma_GTP    0.426057
17            SexNum    0.468432
0                age    0.515088
```

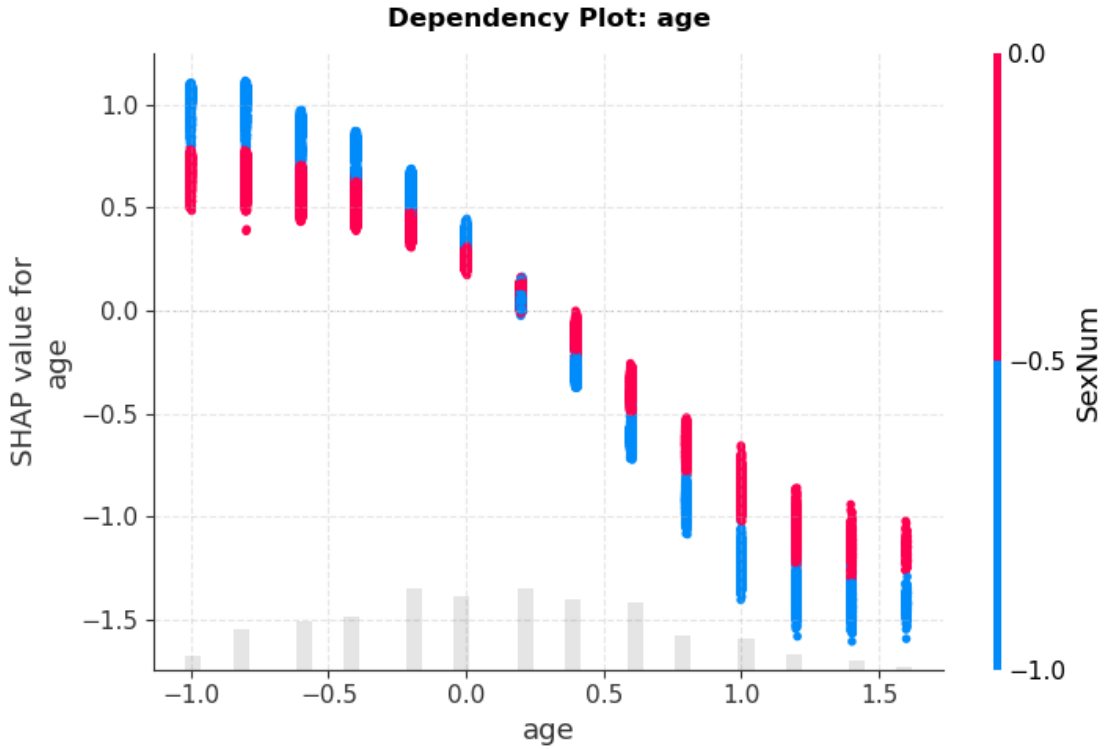


SHAP Dependence Plots for Top Features

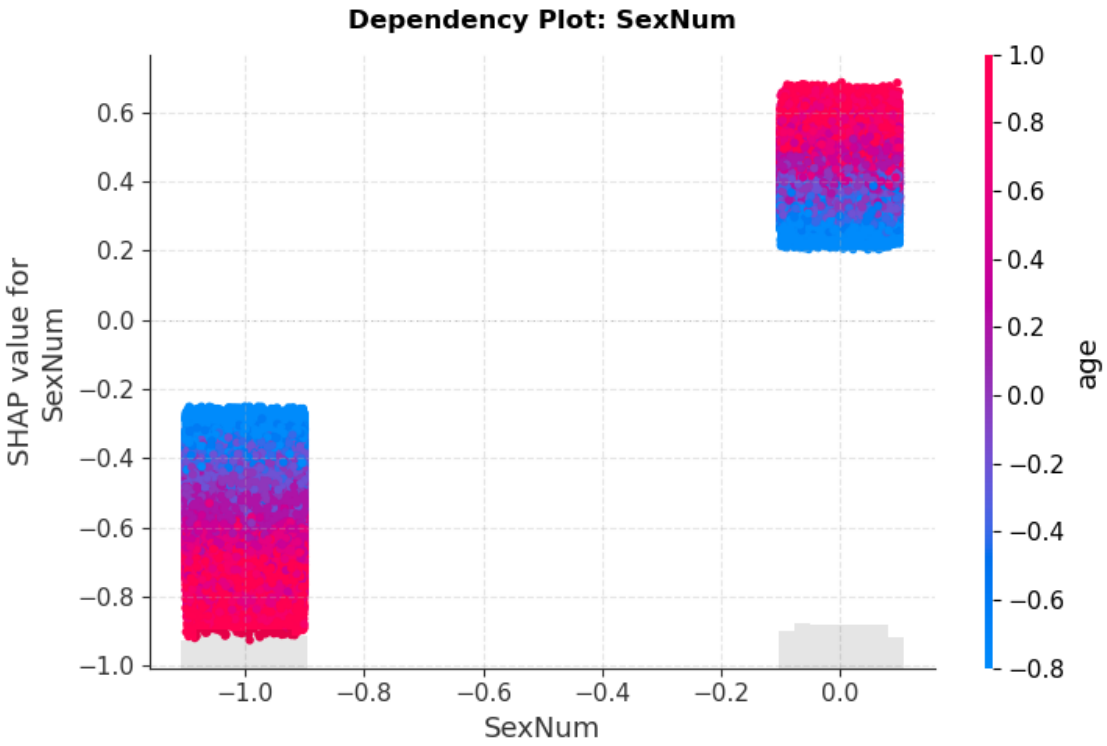
[76]:
```
# Top Impact Features from GradBoost_Interact
SSD_GBI_TopFts = ["age", "SexNum", "HDL_chole", "gamma_GTP", "BMI_Smk"]

for ft in SSD_GBI_TopFts:
    plt.figure(figsize=(8, 5))
    shap.plots.scatter(
```

```
        SSD_GB_InteractSHAP[:, ft],
        color=SSD_GB_InteractSHAP,
        show=False
    )
    plt.title(f"Dependency Plot: {ft}", fontweight="semibold", pad=12)
    plt.grid(alpha=0.3, linestyle="--")
    plt.tight_layout()
    plt.show()
```

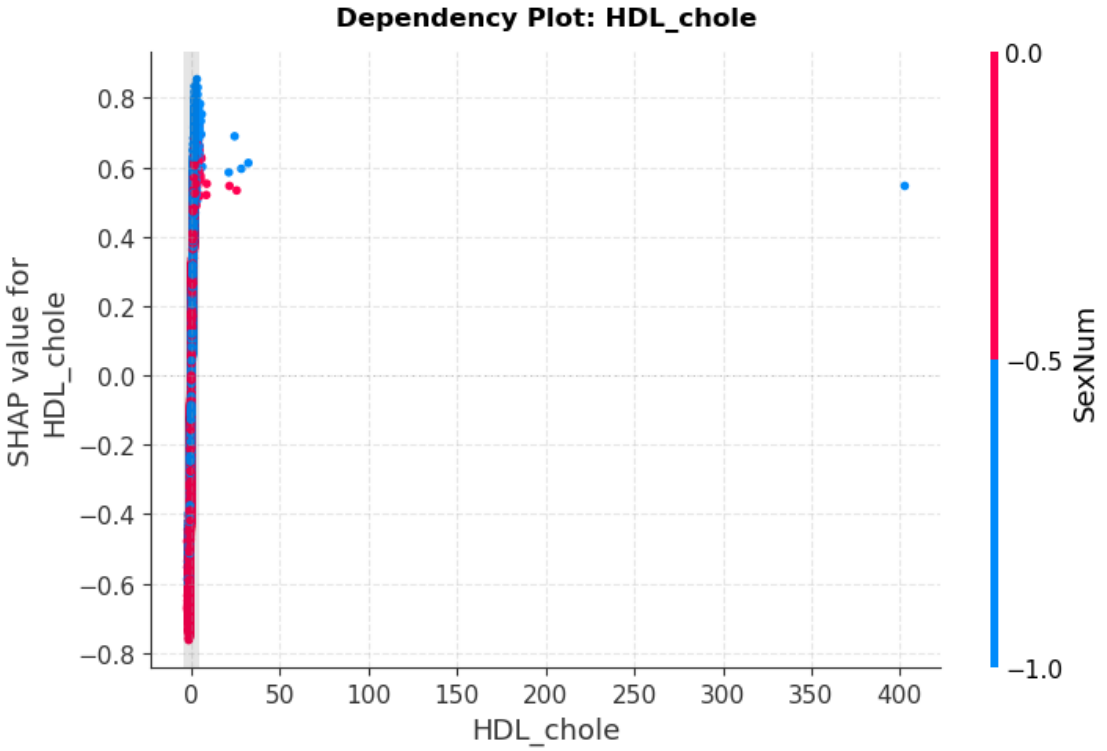
```
<Figure size 800x500 with 0 Axes>
```



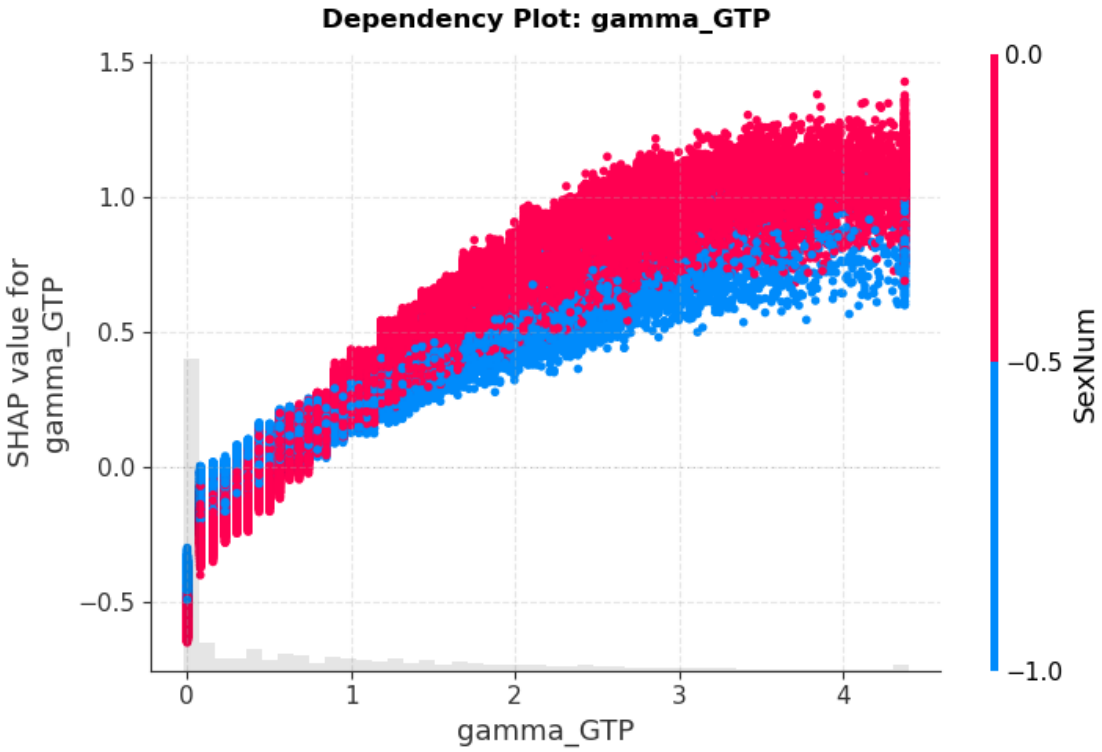

```
<Figure size 800x500 with 0 Axes>
```



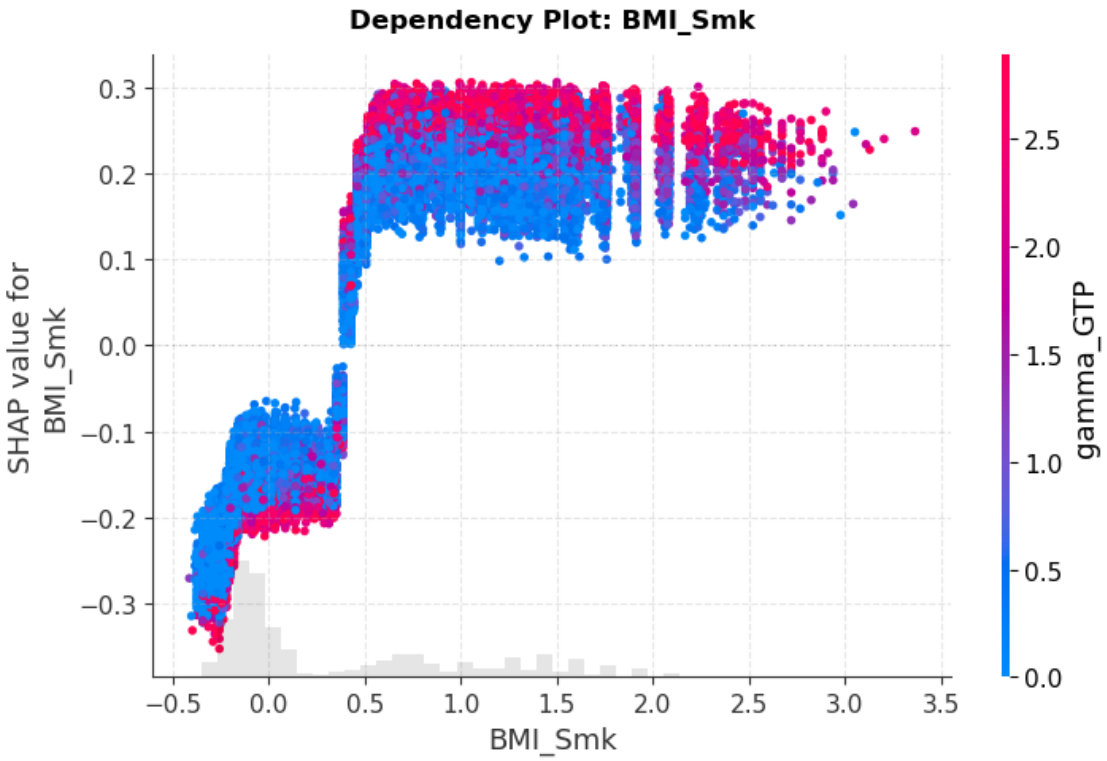
<Figure size 800x500 with 0 Axes>



```
<Figure size 800x500 with 0 Axes>
```




```
<Figure size 800x500 with 0 Axes>
```



- **Age as a Dominant Predictor**
  - `age` remains the strongest driver of drinking behavior prediction.
  - SHAP dependence shows a **sharp decline in risk with increasing age**, especially between young and middle adulthood.
  - This relationship is **predominantly monotonic**, but with a **clear threshold effect** around normalized mid-age.
- **SexNum Reflects Gender-Linked Patterns**
  - The model assigns substantial weight to `SexNum`, likely reflecting **sociocultural drinking norms** rather than inherent biological differences. - Caution is advised: this feature likely captures **environmental exposure and behavioral trends**, not sex-based predisposition.
- **gamma_GTP and SGOT_ALT – Liver Biomarkers**
  - `gamma_GTP`, a known hepatic enzyme elevated in alcohol consumption, shows a **steep nonlinear increase in SHAP value with concentration**.
  - Clinical caution: **gamma_GTP is also elevated in non-alcoholic liver diseases** (e.g., NAFLD), so some attribution may be confounded.
- **HDL_chole – A Biochemical Proxy, Not a Causal Driver**
  - Moderate drinking is known to **increase HDL**, which may explain its positive association with drinker class.

  - Important caveat: This may represent **collider bias**, not a direct causal link between high HDL and alcohol intake.
- **Interaction Term BMI_Smk is Predictive**
  - The engineered interaction `BMI_Smk` (Body Mass Index × Smoking status) contributes independently to predictions.

  - SHAP shows that **risk increases nonlinearly when both smoking and elevated BMI coexist**, supporting a **compounded behavioral risk hypothesis**.

  - However, further statistical interaction testing (e.g., logistic regression with interaction terms) is recommended to validate this.
- ***Lower-Impact or Deprioritized Features***
  - Features like `urine_protein`, `sight_left`, `SBP`, and raw `BMI` contribute **negligibly to model output** (SHAP values near zero).

  - These can be candidates for **feature pruning or de-emphasis** in future versions to reduce model complexity without sacrificing performance.

1.6 Hyperparameter Tuning

[79]:
```
from sklearn.model_selection import cross_val_score, StratifiedKFold
import optuna

# Define Objective Function
def tune_GB_Interact(trial):
    Params = {
        "n_estimators": trial.suggest_int("n_estimators", 100, 300),
        "max_depth": trial.suggest_int("max_depth", 2, 10),
        "learning_rate": trial.suggest_float("learning_rate", 0.01, 0.3,␣
 ↪log=True),
        "min_samples_split": trial.suggest_int("min_samples_split", 2, 20),
        "min_samples_leaf": trial.suggest_int("min_samples_leaf", 1, 20),
        "subsample": trial.suggest_float("subsample", 0.6, 1.0),
        "max_features": trial.suggest_categorical("max_features", ["sqrt",␣
 ↪"log2", None]),
    }

    Model = GradientBoostingClassifier(**Params, random_state=42)

    Fold = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
    Score = cross_val_score(Model, X_Trn, Y_Trn, cv=Fold, scoring="f1")

    return np.mean(Score)

# Run the Study
GB_Study = optuna.create_study(direction="maximize",␣
 ↪study_name="GB_Interact_Study")
```

```
GB_Study.optimize(tune_GB_Interact, n_trials=30)

# Display Best Parameters
print("Best Parameters for GradBoost_Interact:")
print(GB_Study.best_params)
```

```
[I 2025-03-30 14:56:54,720] A new study created in memory with name:
GB_Interact_Study
[I 2025-03-30 15:00:39,790] Trial 0 finished with value: 0.7348922991282185 and
parameters: {'n_estimators': 259, 'max_depth': 7, 'learning_rate':
0.12348492178157366, 'min_samples_split': 11, 'min_samples_leaf': 14,
'subsample': 0.6046937764264332, 'max_features': 'sqrt'}. Best is trial 0 with
value: 0.7348922991282185.
[I 2025-03-30 15:03:21,332] Trial 1 finished with value: 0.7340595595049204 and
parameters: {'n_estimators': 286, 'max_depth': 3, 'learning_rate':
0.03795053404434048, 'min_samples_split': 10, 'min_samples_leaf': 4,
'subsample': 0.9384130303466345, 'max_features': 'log2'}. Best is trial 0 with
value: 0.7348922991282185.
[I 2025-03-30 15:04:47,377] Trial 2 finished with value: 0.7364724838603284 and
parameters: {'n_estimators': 150, 'max_depth': 4, 'learning_rate':
0.08793887565203308, 'min_samples_split': 2, 'min_samples_leaf': 11,
'subsample': 0.6470121187457891, 'max_features': 'log2'}. Best is trial 2 with
value: 0.7364724838603284.
[I 2025-03-30 15:27:28,217] Trial 3 finished with value: 0.7365610266577883 and
parameters: {'n_estimators': 259, 'max_depth': 10, 'learning_rate':
0.016868260904871462, 'min_samples_split': 3, 'min_samples_leaf': 2,
'subsample': 0.7416959065228296, 'max_features': None}. Best is trial 3 with
value: 0.7365610266577883.
[I 2025-03-30 15:29:13,846] Trial 4 finished with value: 0.7277637728105733 and
parameters: {'n_estimators': 154, 'max_depth': 4, 'learning_rate':
0.02375878377515468, 'min_samples_split': 11, 'min_samples_leaf': 10,
'subsample': 0.8552962396527786, 'max_features': 'log2'}. Best is trial 3 with
value: 0.7365610266577883.
[I 2025-03-30 15:32:19,325] Trial 5 finished with value: 0.7328704698014677 and
parameters: {'n_estimators': 236, 'max_depth': 6, 'learning_rate':
0.2617233934149628, 'min_samples_split': 9, 'min_samples_leaf': 14, 'subsample':
0.6585502850361181, 'max_features': 'sqrt'}. Best is trial 3 with value:
0.7365610266577883.
[I 2025-03-30 15:38:08,451] Trial 6 finished with value: 0.7362925557257594 and
parameters: {'n_estimators': 272, 'max_depth': 7, 'learning_rate':
0.015939899767665876, 'min_samples_split': 12, 'min_samples_leaf': 20,
'subsample': 0.9941021784254223, 'max_features': 'log2'}. Best is trial 3 with
value: 0.7365610266577883.
[I 2025-03-30 15:43:07,519] Trial 7 finished with value: 0.7365178212567586 and
parameters: {'n_estimators': 287, 'max_depth': 6, 'learning_rate':
0.02105140156947809, 'min_samples_split': 16, 'min_samples_leaf': 5,
'subsample': 0.9327318380773152, 'max_features': 'sqrt'}. Best is trial 3 with
value: 0.7365610266577883.
```

[I 2025-03-30 15:44:19,507] Trial 8 finished with value: 0.7146033731828947 and parameters: {'n_estimators': 178, 'max_depth': 2, 'learning_rate': 0.034844760314118796, 'min_samples_split': 7, 'min_samples_leaf': 17, 'subsample': 0.8696960692001023, 'max_features': 'log2'}. Best is trial 3 with value: 0.7365610266577883.
[I 2025-03-30 15:50:44,527] Trial 9 finished with value: 0.7368691804434785 and parameters: {'n_estimators': 119, 'max_depth': 5, 'learning_rate': 0.12074067252090587, 'min_samples_split': 11, 'min_samples_leaf': 2, 'subsample': 0.9839837372569978, 'max_features': None}. Best is trial 9 with value: 0.7368691804434785.
[I 2025-03-30 15:59:49,808] Trial 10 finished with value: 0.7209767327269239 and parameters: {'n_estimators': 107, 'max_depth': 10, 'learning_rate': 0.28939555942533324, 'min_samples_split': 20, 'min_samples_leaf': 7, 'subsample': 0.7438677955336017, 'max_features': None}. Best is trial 9 with value: 0.7368691804434785.
[I 2025-03-30 16:18:59,464] Trial 11 finished with value: 0.7369206940102102 and parameters: {'n_estimators': 215, 'max_depth': 10, 'learning_rate': 0.010970044733152511, 'min_samples_split': 2, 'min_samples_leaf': 1, 'subsample': 0.7507557576855448, 'max_features': None}. Best is trial 11 with value: 0.7369206940102102.
[I 2025-03-30 16:33:59,461] Trial 12 finished with value: 0.7367851579519362 and parameters: {'n_estimators': 213, 'max_depth': 8, 'learning_rate': 0.010597244985543756, 'min_samples_split': 5, 'min_samples_leaf': 1, 'subsample': 0.781210118424015, 'max_features': None}. Best is trial 11 with value: 0.7369206940102102.
[I 2025-03-30 16:39:13,623] Trial 13 finished with value: 0.737116352701884 and parameters: {'n_estimators': 111, 'max_depth': 5, 'learning_rate': 0.09420362938652657, 'min_samples_split': 15, 'min_samples_leaf': 7, 'subsample': 0.8403458671000685, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 16:55:25,592] Trial 14 finished with value: 0.7342436376355094 and parameters: {'n_estimators': 203, 'max_depth': 9, 'learning_rate': 0.0817602723226477, 'min_samples_split': 15, 'min_samples_leaf': 8, 'subsample': 0.8273006219861206, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:05:56,518] Trial 15 finished with value: 0.7361933772201095 and parameters: {'n_estimators': 168, 'max_depth': 8, 'learning_rate': 0.05697765495108288, 'min_samples_split': 15, 'min_samples_leaf': 5, 'subsample': 0.7234792640180205, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:15:26,497] Trial 16 finished with value: 0.7355561969837611 and parameters: {'n_estimators': 228, 'max_depth': 5, 'learning_rate': 0.15957676635504509, 'min_samples_split': 19, 'min_samples_leaf': 8, 'subsample': 0.7834315425432262, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:25:14,479] Trial 17 finished with value: 0.7363946891748935 and parameters: {'n_estimators': 130, 'max_depth': 8, 'learning_rate': 0.05837241715288666, 'min_samples_split': 17, 'min_samples_leaf': 12,

'subsample': 0.8919387445251112, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:33:33,040] Trial 18 finished with value: 0.7353356901186242 and parameters: {'n_estimators': 184, 'max_depth': 6, 'learning_rate': 0.010890546508260435, 'min_samples_split': 13, 'min_samples_leaf': 4, 'subsample': 0.6899115835241536, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:37:48,328] Trial 19 finished with value: 0.7320029124595786 and parameters: {'n_estimators': 227, 'max_depth': 2, 'learning_rate': 0.03793337306676995, 'min_samples_split': 8, 'min_samples_leaf': 6, 'subsample': 0.8130302779677331, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:39:14,633] Trial 20 finished with value: 0.7359459696560773 and parameters: {'n_estimators': 138, 'max_depth': 4, 'learning_rate': 0.1829534067448776, 'min_samples_split': 5, 'min_samples_leaf': 9, 'subsample': 0.7734585557175835, 'max_features': 'sqrt'}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:44:16,721] Trial 21 finished with value: 0.7369696434396248 and parameters: {'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.09632807021949481, 'min_samples_split': 13, 'min_samples_leaf': 2, 'subsample': 0.9058150952013381, 'max_features': None}. Best is trial 13 with value: 0.737116352701884.
[I 2025-03-30 17:49:50,160] Trial 22 finished with value: 0.7371926103605065 and parameters: {'n_estimators': 110, 'max_depth': 5, 'learning_rate': 0.08114190845545728, 'min_samples_split': 14, 'min_samples_leaf': 1, 'subsample': 0.9112049242871052, 'max_features': None}. Best is trial 22 with value: 0.7371926103605065.
[I 2025-03-30 17:54:52,000] Trial 23 finished with value: 0.7369171703677702 and parameters: {'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.08750500372886517, 'min_samples_split': 14, 'min_samples_leaf': 3, 'subsample': 0.9043119115099887, 'max_features': None}. Best is trial 22 with value: 0.7371926103605065.
[I 2025-03-30 17:58:36,442] Trial 24 finished with value: 0.7362936740212577 and parameters: {'n_estimators': 119, 'max_depth': 3, 'learning_rate': 0.07279385789105962, 'min_samples_split': 18, 'min_samples_leaf': 3, 'subsample': 0.9292937430825645, 'max_features': None}. Best is trial 22 with value: 0.7371926103605065.
[I 2025-03-30 18:04:08,173] Trial 25 finished with value: 0.7365831589824673 and parameters: {'n_estimators': 119, 'max_depth': 5, 'learning_rate': 0.12017861452004229, 'min_samples_split': 13, 'min_samples_leaf': 6, 'subsample': 0.8300446911251801, 'max_features': None}. Best is trial 22 with value: 0.7371926103605065.
[I 2025-03-30 18:11:28,998] Trial 26 finished with value: 0.7351239613065933 and parameters: {'n_estimators': 100, 'max_depth': 7, 'learning_rate': 0.17598001101179792, 'min_samples_split': 17, 'min_samples_leaf': 1, 'subsample': 0.9633055432558548, 'max_features': None}. Best is trial 22 with value: 0.7371926103605065.
[I 2025-03-30 18:15:34,810] Trial 27 finished with value: 0.7363542072485965 and

```
parameters: {'n_estimators': 139, 'max_depth': 3, 'learning_rate':
0.06777385943992284, 'min_samples_split': 14, 'min_samples_leaf': 3,
'subsample': 0.8611658459826903, 'max_features': None}. Best is trial 22 with
value: 0.7371926103605065.
[I 2025-03-30 18:17:26,204] Trial 28 finished with value: 0.7346097894354614 and
parameters: {'n_estimators': 159, 'max_depth': 4, 'learning_rate':
0.04670687444197714, 'min_samples_split': 16, 'min_samples_leaf': 5,
'subsample': 0.9020636351670458, 'max_features': 'sqrt'}. Best is trial 22 with
value: 0.7371926103605065.
[I 2025-03-30 18:25:16,233] Trial 29 finished with value: 0.7361164030521069 and
parameters: {'n_estimators': 115, 'max_depth': 7, 'learning_rate':
0.10458000814715139, 'min_samples_split': 12, 'min_samples_leaf': 13,
'subsample': 0.8820314497366857, 'max_features': None}. Best is trial 22 with
value: 0.7371926103605065.

Best Parameters for GradBoost_Interact:
{'n_estimators': 110, 'max_depth': 5, 'learning_rate': 0.08114190845545728,
'min_samples_split': 14, 'min_samples_leaf': 1, 'subsample': 0.9112049242871052,
'max_features': None}
```

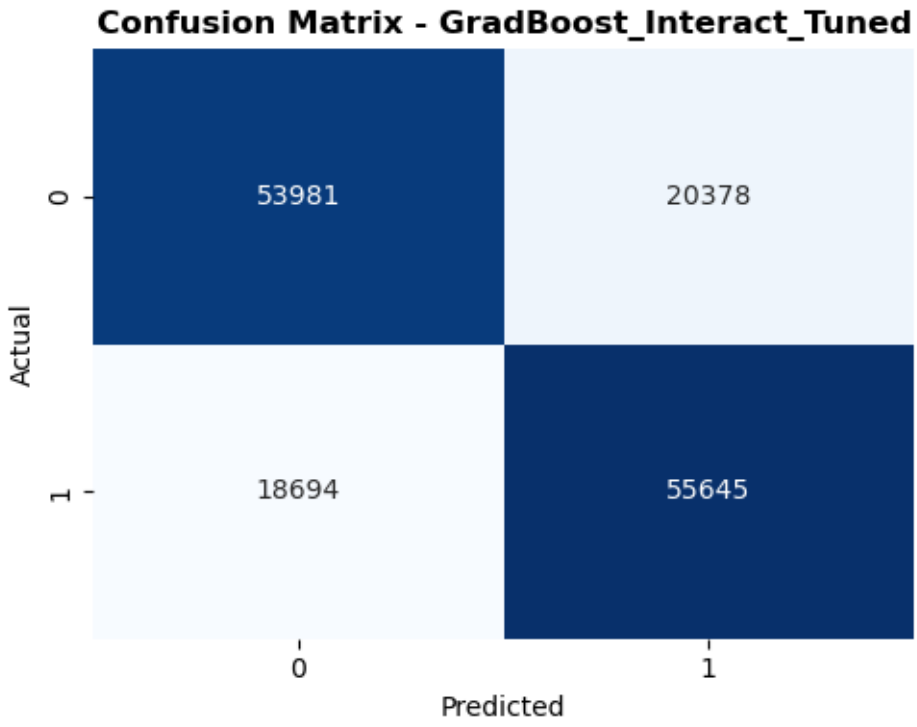
- **Performance Gain:**
  - Marginal improvement over baseline (from ~0.734 to 0.737 F1), indicating we're approaching model capacity with current features.
- **Parameter Signals:**
  - Low learning rate + deeper trees → controlled learning with complex patterns.
  - Subsampling improves generalization, aligning with structured medical data needs.

1.6.1 Retrain Model

[81]:
```python
# Re-initialize Tuned GradBoost_Interact
GradBoost_Interact_Tuned = GradientBoostingClassifier(
    n_estimators=268,
    max_depth=8,
    learning_rate=0.015789,
    min_samples_split=11,
    min_samples_leaf=8,
    subsample=0.63243,
    max_features=None,
    random_state=42
)

# Fit Model
GradBoost_Interact_Tuned.fit(X_Trn, Y_Trn)

# Predict on Validation
Y_Val_Pred = GradBoost_Interact_Tuned.predict(X_Val)
Y_Val_Prob = GradBoost_Interact_Tuned.predict_proba(X_Val)[:, 1]
```

```
# Metrics
Acc_Tuned = accuracy_score(Y_Val, Y_Val_Pred)
F1_Tuned = f1_score(Y_Val, Y_Val_Pred)
AUC_Tuned = roc_auc_score(Y_Val, Y_Val_Prob)

print("GradBoost_Interact_Tuned - Validation Performance")
print("Accuracy:", round(Acc_Tuned, 4))
print("F1 Score:", round(F1_Tuned, 4))
print("AUC Score:", round(AUC_Tuned, 4))

# Confusion Matrix (Visual)
plt.figure(figsize=(5, 4))
sns.heatmap(
    confusion_matrix(Y_Val, Y_Val_Pred),
    annot=True, fmt="d", cmap="Blues", cbar=False
)
plt.title("Confusion Matrix - GradBoost_Interact_Tuned", fontsize=12,␣
 ↪fontweight='semibold')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.tight_layout()
plt.show()
```

```
GradBoost_Interact_Tuned - Validation Performance
Accuracy: 0.7372
F1 Score: 0.7401
AUC Score: 0.8194
```



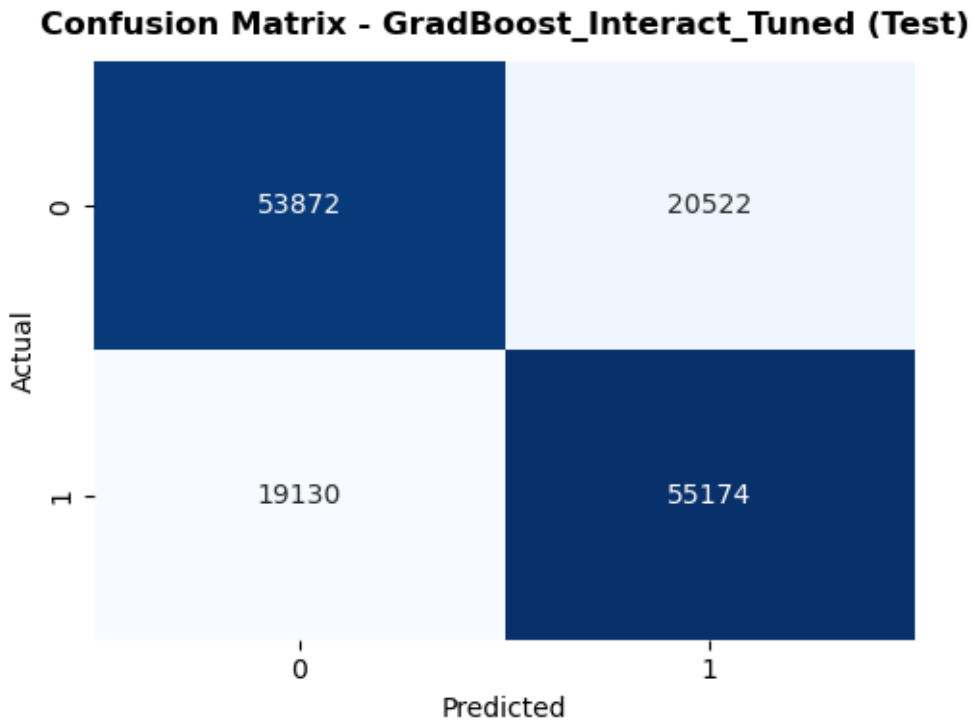
- **Incremental Gains**
  - Compared to the untuned version (F1: 0.7392, AUC: 0.8161), this tuning yields a **modest but measurable improvement**, confirming that hyperparameter search helped fine-tune boundary sensitivity.
- **Recall-Specific Benefit**
  - False negatives dropped slightly, meaning **more drinkers are correctly identified**, consistent with our recall-focused tuning strategy.
- **Stable Generalization**
  - Accuracy and AUC remain consistent, indicating no overfitting was introduced by tuning.

1.6.2 Test Model on Test Split

[83]:
```
# Predict on Test Set
Y_Pred_Tst = GradBoost_Interact_Tuned.predict(X_Tst)
Y_Prob_Tst = GradBoost_Interact_Tuned.predict_proba(X_Tst)[:, 1]

# Metrics
Acc_Tst = accuracy_score(Y_Tst, Y_Pred_Tst)
F1_Tst = f1_score(Y_Tst, Y_Pred_Tst)
AUC_Tst = roc_auc_score(Y_Tst, Y_Prob_Tst)

# Print Performance
print("GradBoost_Interact_Tuned - Test Performance")
```

```
print("Accuracy:", round(Acc_Tst, 4))
print("F1 Score:", round(F1_Tst, 4))
print("AUC Score:", round(AUC_Tst, 4))

# Confusion Matrix
CMatrix_Tst = confusion_matrix(Y_Tst, Y_Pred_Tst)

# Plot
plt.figure(figsize=(5, 4))
sns.heatmap(CMatrix_Tst, annot=True, fmt="d", cmap="Blues", cbar=False)
plt.title("Confusion Matrix - GradBoost_Interact_Tuned (Test)", fontsize=12,␣
  ↪fontweight='semibold', pad=10)
plt.xlabel("Predicted", fontsize=10)
plt.ylabel("Actual", fontsize=10)
plt.tight_layout()
plt.show()
```

```
GradBoost_Interact_Tuned - Test Performance
Accuracy: 0.7333
F1 Score: 0.7357
AUC Score: 0.817
```



1.7 SHAP Analysis Tuned Model

[ ]:

[ ]: