Obesity Assignment

May 10, 2024

0.0.1 Importing Libraries

[56]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
```

0.0.2 Loading Data

[4]:
```
df=pd.read_csv('ObesityDataSet_raw_and_data_sinthetic.csv')
```

0.0.3 Exploring data

[6]:
```
df.head()
```

[6]:

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	family_history_with_overweight	FAF	TUE	CAEC	MTRANS	NObeyesdad
0	21.0	Female	1.62	64.0	no	no	2.0	3.0	no	no	2.0	yes	0.0	1.0	Sometimes	Public_Transportation	Normal_Weight
1	21.0	Female	1.52	56.0	Sometimes	no	3.0	3.0	yes	yes	3.0	yes	3.0	0.0	Sometimes	Public_Transportation	Normal_Weight
2	23.0	Male	1.80	77.0	Frequently	no	2.0	3.0	no	no	2.0	yes	2.0	1.0	Sometimes	Public_Transportation	Normal_Weight
3	27.0	Male	1.80	87.0	Frequently	no	3.0	3.0	no	no	2.0	no	2.0	0.0	Sometimes	Walking	Overweight_Level_I
4	22.0	Male	1.78	89.8	Sometimes	no	2.0	1.0	no	no	2.0	no	0.0	0.0	Sometimes	Public_Transportation	Overweight_Level_II

```
4  Overweight_Level_II
```

[7]:
```
df.tail()
```

[7]:
```
            Age  Gender    Height      Weight       CALC FAVC  FCVC  NCP SCC  \
2106  20.976842  Female  1.710730  131.408528  Sometimes  yes   3.0  3.0  no
2107  21.982942  Female  1.748584  133.742943  Sometimes  yes   3.0  3.0  no
2108  22.524036  Female  1.752206  133.689352  Sometimes  yes   3.0  3.0  no
2109  24.361936  Female  1.739450  133.346641  Sometimes  yes   3.0  3.0  no
2110  23.664709  Female  1.738836  133.472641  Sometimes  yes   3.0  3.0  no

     SMOKE      CH2O family_history_with_overweight       FAF       TUE  \
2106    no  1.728139                             yes  1.676269  0.906247
2107    no  2.005130                             yes  1.341390  0.599270
2108    no  2.054193                             yes  1.414209  0.646288
2109    no  2.852339                             yes  1.139107  0.586035
2110    no  2.863513                             yes  1.026452  0.714137

           CAEC                 MTRANS        NObeyesdad
2106  Sometimes  Public_Transportation  Obesity_Type_III
2107  Sometimes  Public_Transportation  Obesity_Type_III
2108  Sometimes  Public_Transportation  Obesity_Type_III
2109  Sometimes  Public_Transportation  Obesity_Type_III
2110  Sometimes  Public_Transportation  Obesity_Type_III
```

[8]:
```
df.dtypes
```

[8]:
```
Age                               float64
Gender                             object
Height                            float64
Weight                            float64
CALC                               object
FAVC                               object
FCVC                              float64
NCP                               float64
SCC                                object
SMOKE                              object
CH2O                              float64
family_history_with_overweight     object
FAF                               float64
TUE                               float64
CAEC                               object
MTRANS                             object
NObeyesdad                         object
dtype: object
```

[9]:
```
df.describe
```

[9]:
```
<bound method NDFrame.describe of             Age  Gender    Height      Weight
CALC FAVC  FCVC  NCP  \
0     21.000000  Female  1.620000   64.000000          no   no   2.0  3.0
1     21.000000  Female  1.520000   56.000000   Sometimes   no   3.0  3.0
2     23.000000    Male  1.800000   77.000000  Frequently   no   2.0  3.0
3     27.000000    Male  1.800000   87.000000  Frequently   no   3.0  3.0
4     22.000000    Male  1.780000   89.800000   Sometimes   no   2.0  1.0
…           …     …        …           …           …  …   …  …
2106  20.976842  Female  1.710730  131.408528   Sometimes  yes   3.0  3.0
2107  21.982942  Female  1.748584  133.742943   Sometimes  yes   3.0  3.0
2108  22.524036  Female  1.752206  133.689352   Sometimes  yes   3.0  3.0
2109  24.361936  Female  1.739450  133.346641   Sometimes  yes   3.0  3.0
2110  23.664709  Female  1.738836  133.472641   Sometimes  yes   3.0  3.0

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

           CAEC                 MTRANS           NObeyesdad
0     Sometimes  Public_Transportation        Normal_Weight
1     Sometimes  Public_Transportation        Normal_Weight
2     Sometimes  Public_Transportation        Normal_Weight
3     Sometimes                Walking   Overweight_Level_I
4     Sometimes  Public_Transportation  Overweight_Level_II
…         …                    …                 …
2106  Sometimes  Public_Transportation     Obesity_Type_III
2107  Sometimes  Public_Transportation     Obesity_Type_III
2108  Sometimes  Public_Transportation     Obesity_Type_III
2109  Sometimes  Public_Transportation     Obesity_Type_III
2110  Sometimes  Public_Transportation     Obesity_Type_III

[2111 rows x 17 columns]>
```

0.0.4 Converting categorical variables into numerical

[37]:
```
categorical_df = df.select_dtypes(include=['object'])
label_encoder = LabelEncoder()
for col in categorical_df.columns:
    df[col] = label_encoder.fit_transform(df[col])
```

[39]:
```
df.head()
```

[39]:

	Age	Gender	Height	Weight	CALC	FAVC	FCVC	NCP	SCC	SMOKE	CH2O	\
0	21.0	0	1.62	64.0	3	0	2.0	3.0	0	0	2.0	
1	21.0	0	1.52	56.0	2	0	3.0	3.0	1	1	3.0	
2	23.0	1	1.80	77.0	1	0	2.0	3.0	0	0	2.0	
3	27.0	1	1.80	87.0	1	0	3.0	3.0	0	0	2.0	
4	22.0	1	1.78	89.8	2	0	2.0	1.0	0	0	2.0	

	family_history_with_overweight	FAF	TUE	CAEC	MTRANS	NObeyesdad
0	1	0.0	1.0	2	3	1
1	1	3.0	0.0	2	3	1
2	1	2.0	1.0	2	3	1
3	0	2.0	0.0	2	4	5
4	0	0.0	0.0	2	3	6

0.0.5 Null Values

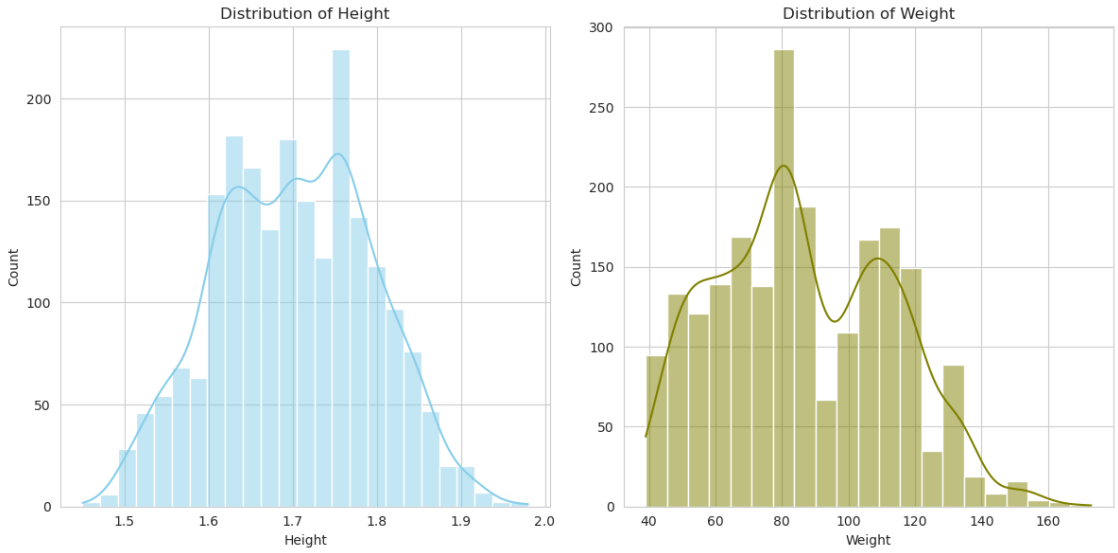
[11]:
```
df.isnull().sum()
```

[11]:
```
Age                                0
Gender                             0
Height                             0
Weight                             0
CALC                               0
FAVC                               0
FCVC                               0
NCP                                0
SCC                                0
SMOKE                              0
CH2O                               0
family_history_with_overweight     0
FAF                                0
TUE                                0
CAEC                               0
MTRANS                             0
NObeyesdad                         0
dtype: int64
```

0.0.6 Histogram for Height and Weight Distribution

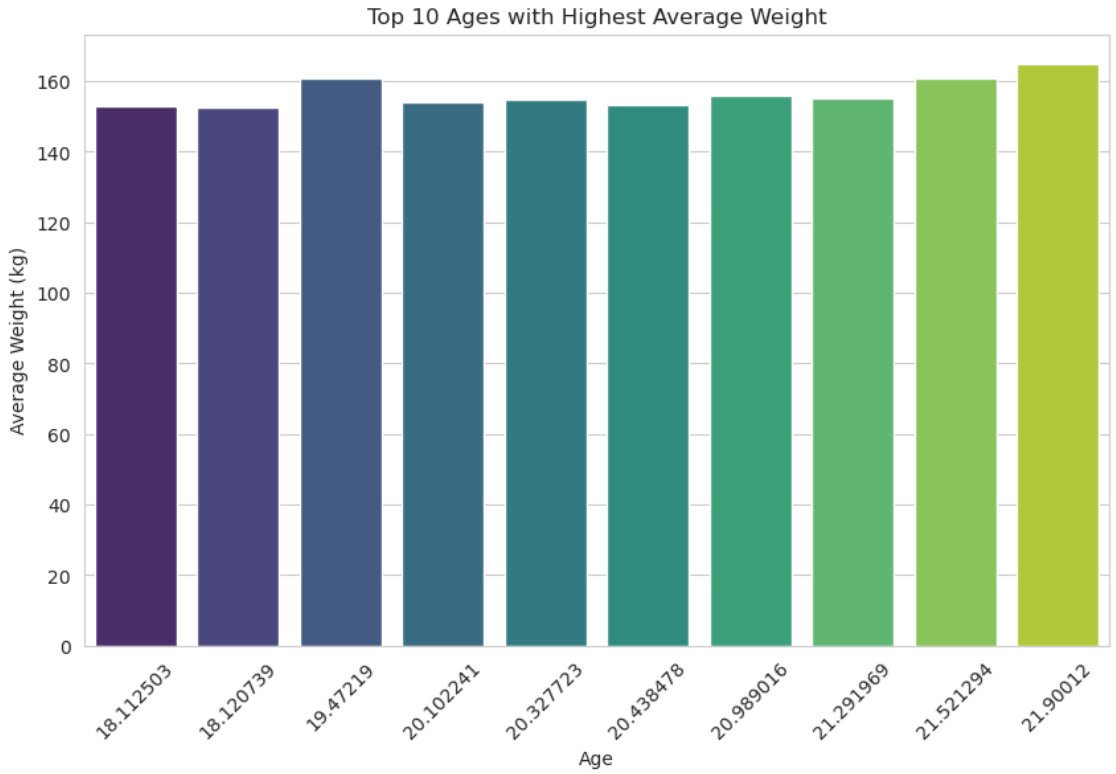
[13]:
```
sns.set_style("whitegrid")
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['Height'], kde=True, color='skyblue')
plt.title('Distribution of Height')

plt.subplot(1, 2, 2)
sns.histplot(df['Weight'], kde=True, color='olive')
plt.title('Distribution of Weight')

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

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is
deprecated and will be removed in a future version. Convert inf values to NaN
before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



0.0.7 Bar plot for Top 10 Ages with Highest Weight

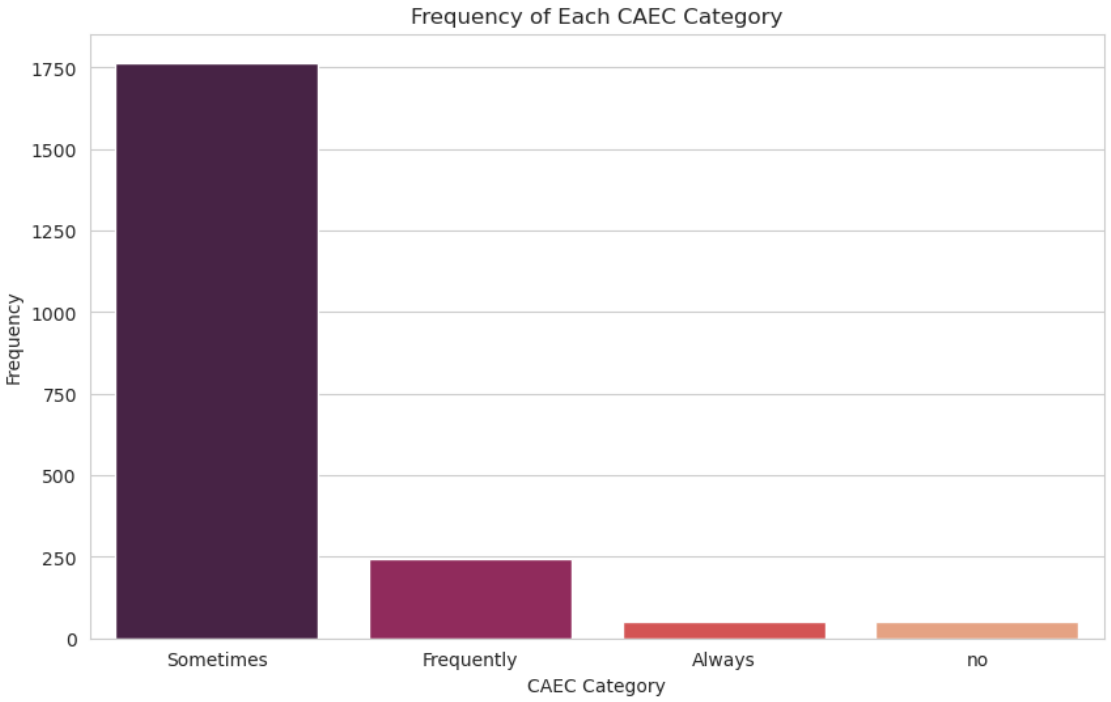
[15]:
```
age_weight_avg = df.groupby('Age')['Weight'].mean().reset_index()
top_ages = age_weight_avg.sort_values(by='Weight', ascending=False).head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x='Age', y='Weight', data=top_ages, palette='viridis')
plt.title('Top 10 Ages with Highest Average Weight')
plt.xlabel('Age')
plt.ylabel('Average Weight (kg)')
plt.xticks(rotation=45)
plt.show()
```



0.0.8 Pie Chart for Distribution of CAEC values

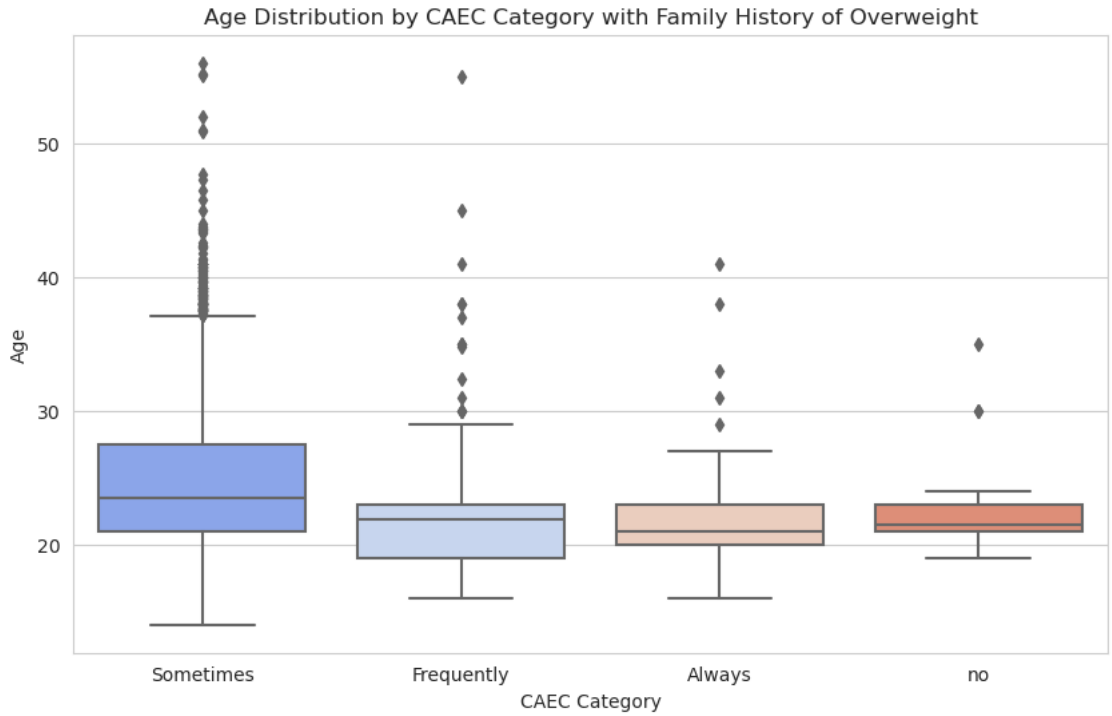
[17]:
```
sns.set_style("whitegrid")
caec_counts = df['CAEC'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=caec_counts.index, y=caec_counts.values, palette='rocket')
plt.title('Frequency of Each CAEC Category')
plt.xlabel('CAEC Category')
plt.ylabel('Frequency')
```

```
plt.show()
```



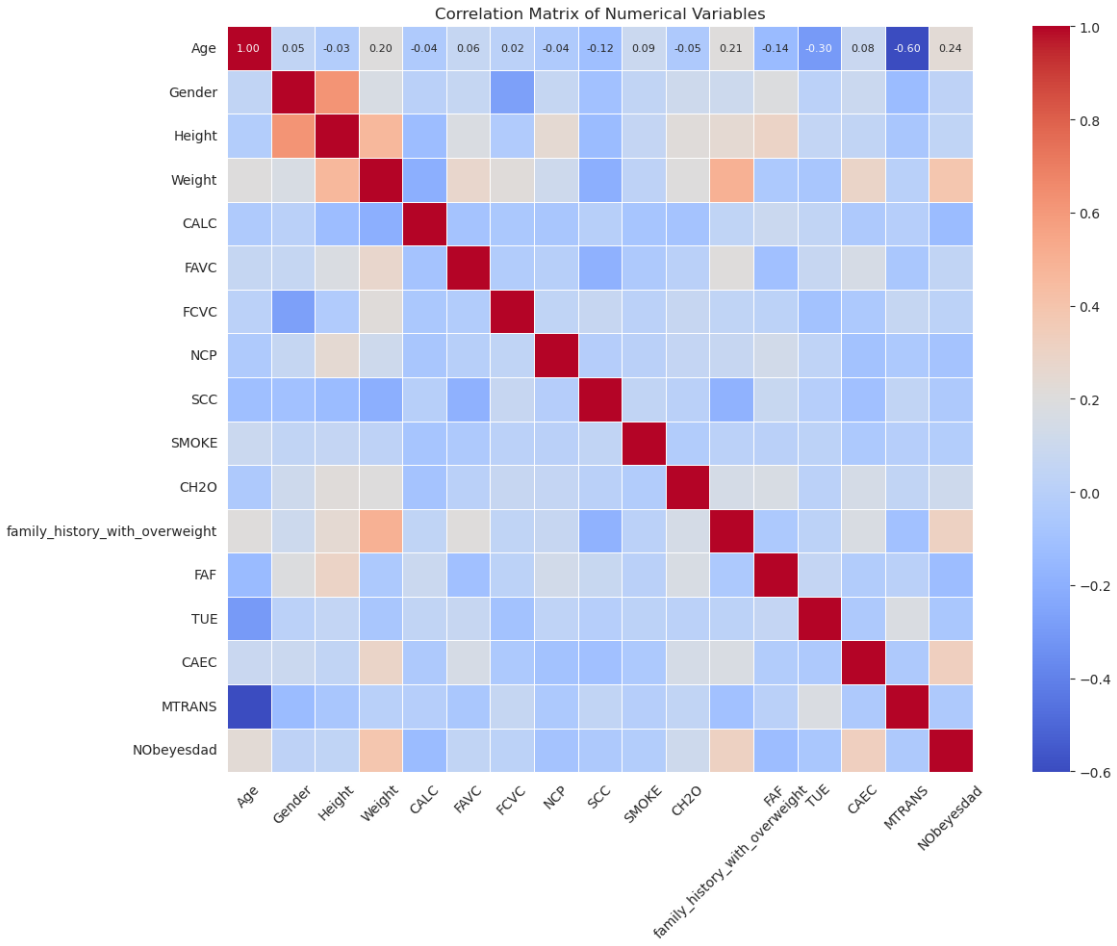
0.0.9 Box plot for Average Ages with Family History with Overweight

[19]:
```
sns.set_style("whitegrid")

overweight_history_df = df[df['family_history_with_overweight'] == 'yes']

plt.figure(figsize=(10, 6))
sns.boxplot(x='CAEC', y='Age', data=overweight_history_df, palette='coolwarm')
plt.title('Age Distribution by CAEC Category with Family History of Overweight')
plt.xlabel('CAEC Category')
plt.ylabel('Age')
plt.show()
```



0.0.10 Correlation Matrix

[54]:
```
numeric_df = df.select_dtypes(include=[np.number])

correlation_matrix = numeric_df.corr()

plt.figure(figsize=(14, 10))  # Increase figure size for better visibility
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm',␣
↪cbar=True, square=True, linewidths=.5, annot_kws={"size": 8})
plt.title('Correlation Matrix of Numerical Variables')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```



0.0.11 Data Processing

[59]:
```
label_enc_cols = ['Gender', 'SCC', 'SMOKE', 'family_history_with_overweight']
  ↪# example columns
label_encoder = LabelEncoder()
for col in label_enc_cols:
    df[col] = label_encoder.fit_transform(df[col])

# Step 2: Standard Scaling for continuous columns
continuous_cols = ['Age', 'Height', 'Weight', 'FCVC', 'NCP', 'CH2O', 'FAF',
  ↪'TUE']
scaler = StandardScaler()
df[continuous_cols] = scaler.fit_transform(df[continuous_cols])

# Step 3: One-Hot Encoding for nominal categorical columns
one_hot_enc_cols = ['CALC', 'FAVC', 'CAEC', 'MTRANS']
one_hot_encoder = OneHotEncoder(drop='first', sparse=False)
```

```
ct = ColumnTransformer([('one_hot_enc', one_hot_encoder, one_hot_enc_cols)],␣
 ↪remainder='passthrough')
df_encoded = ct.fit_transform(df)

column_names = ct.get_feature_names_out()
df_encoded = pd.DataFrame(df_encoded, columns=column_names)


print(df_encoded.head())
```

```
   one_hot_enc__CALC_1  one_hot_enc__CALC_2  one_hot_enc__CALC_3  \
0                  0.0                  0.0                  1.0
1                  0.0                  1.0                  0.0
2                  1.0                  0.0                  0.0
3                  1.0                  0.0                  0.0
4                  0.0                  1.0                  0.0

   one_hot_enc__FAVC_1  one_hot_enc__CAEC_1  one_hot_enc__CAEC_2  \
0                  0.0                  0.0                  1.0
1                  0.0                  0.0                  1.0
2                  0.0                  0.0                  1.0
3                  0.0                  0.0                  1.0
4                  0.0                  0.0                  1.0

   one_hot_enc__CAEC_3  one_hot_enc__MTRANS_1  one_hot_enc__MTRANS_2  \
0                  0.0                    0.0                    0.0
1                  0.0                    0.0                    0.0
2                  0.0                    0.0                    0.0
3                  0.0                    0.0                    0.0
4                  0.0                    0.0                    0.0

   one_hot_enc__MTRANS_3  …  remainder__Weight  remainder__FCVC  \
0                    1.0  …          -0.862558        -0.785019
1                    1.0  …          -1.168077         1.088342
2                    1.0  …          -0.366090        -0.785019
3                    0.0  …           0.015808         1.088342
4                    1.0  …           0.122740        -0.785019

   remainder__NCP  remainder__SCC  remainder__SMOKE  remainder__CH2O  \
0        0.404153             0.0               0.0        -0.013073
1        0.404153             1.0               1.0         1.618759
2        0.404153             0.0               0.0        -0.013073
3        0.404153             0.0               0.0        -0.013073
4       -2.167023             0.0               0.0        -0.013073

   remainder__family_history_with_overweight  remainder__FAF  remainder__TUE  \
0                                        1.0       -1.188039        0.561997
```

```
1                                    1.0        2.339750       -1.080625
2                                    1.0        1.163820        0.561997
3                                    0.0        1.163820       -1.080625
4                                    0.0       -1.188039       -1.080625

   remainder__NObeyesdad
0                    1.0
1                    1.0
2                    1.0
3                    5.0
4                    6.0

[5 rows x 24 columns]
```

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/sklearn/preprocessing/_encoders.py:972: FutureWarning: `sparse` was
renamed to `sparse_output` in version 1.2 and will be removed in 1.4.
`sparse_output` is ignored unless you leave `sparse` to its default value.
  warnings.warn(
```

0.0.12 Train-test Split

[67]:
```
from sklearn.model_selection import train_test_split
X = df_encoded.drop('remainder__NObeyesdad', axis=1)
y = df_encoded['remainder__NObeyesdad']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Outputs the results to verify
print("Training set shape:", X_train.shape)
print("Test set shape:", X_test.shape)
```

```
Training set shape: (1688, 23)
Test set shape: (423, 23)
```

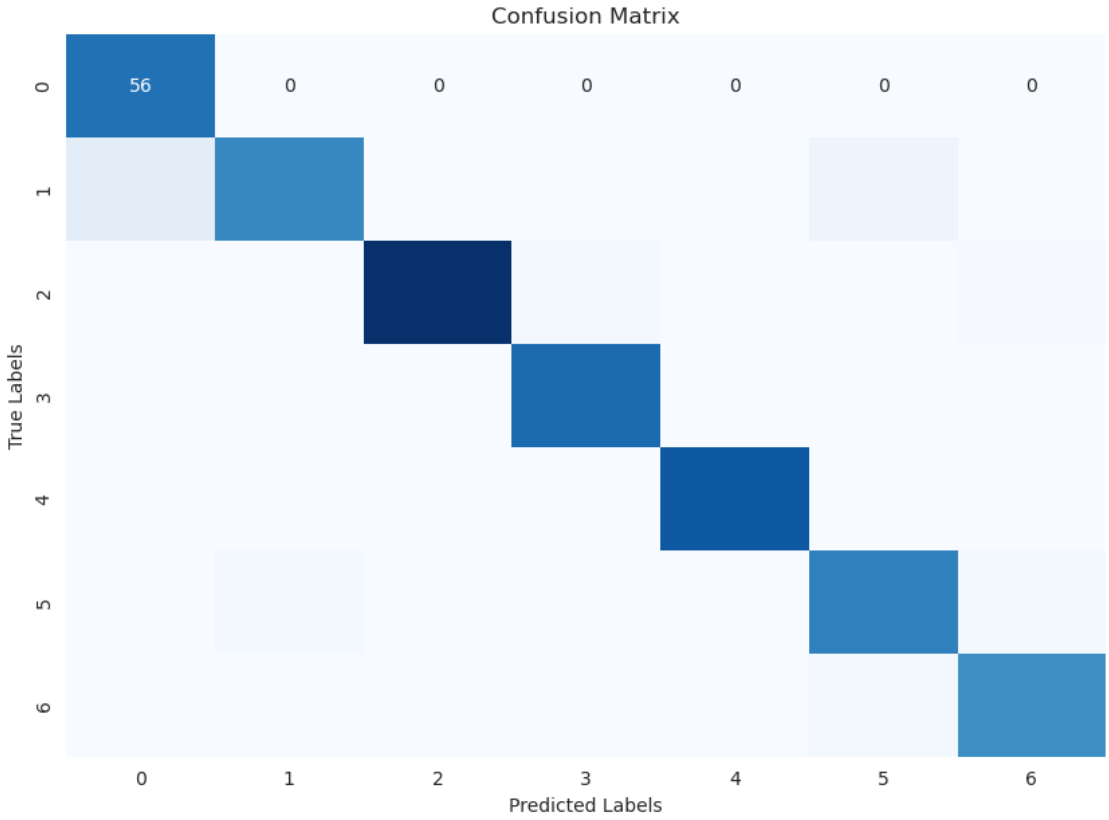
0.0.13 SVM

[73]:
```
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, accuracy_score

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
svm_classifier = SVC(kernel='linear', random_state=42)
svm_classifier.fit(X_train_scaled, y_train)
y_pred = svm_classifier.predict(X_test_scaled)
```

```
print("Accuracy on test set:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy on test set: 0.950354609929078

Classification Report:
              precision    recall  f1-score   support

         0.0       0.88      1.00      0.93        56
         1.0       0.96      0.81      0.88        62
         2.0       1.00      0.96      0.98        78
         3.0       0.97      1.00      0.98        58
         4.0       1.00      1.00      1.00        63
         5.0       0.90      0.93      0.91        56
         6.0       0.94      0.96      0.95        50

    accuracy                           0.95       423
   macro avg       0.95      0.95      0.95       423
weighted avg       0.95      0.95      0.95       423
```

[75]:
```
from sklearn.metrics import confusion_matrix

conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(10, 7))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues', cbar=False)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```


Confusion Matrix
True Labels
Predicted Labels
0
1
2
3
4
5
6
56
0
0
0
0
0
0