```
#Yagmur Parmaksiz Pusula Talent Academy test case task 2
#Data Pre-Processing: Based on the findings from your EDA, clean and
preprocess the data to make it ready for modeling. This includes
handling missing values, encoding  categorical variables, normalizing
or standardizing numerical features, and  addressing any data quality
issues. For example, you can use methods like  SimpleImputer or
KNNImputer to handle missing data, and OneHotEncoder or  LabelEncoder
for categorical data.

from sklearn.impute import KNNImputer
from sklearn.preprocessing import OneHotEncoder, LabelEncoder,
StandardScaler

# 1. Handling Missing Values
# Imputation for numerical features (Kilo, Boy)
imputer = KNNImputer(n_neighbors=5)
train_data[['Kilo', 'Boy']] =
imputer.fit_transform(train_data[['Kilo', 'Boy']])
# Imputation for categorical features (Kan Grubu)
train_data['Kan Grubu'].fillna(train_data['Kan Grubu'].mode()[0],
inplace=True)


# 2. Encoding Categorical Variables
# Using LabelEncoder for binary categories (e.g., Cinsiyet)
label_encoder = LabelEncoder()
train_data['Cinsiyet'] =
label_encoder.fit_transform(train_data['Cinsiyet'])
# OneHotEncoder for nominal categorical variables (e.g., Uyruk, Kan
Grubu, Yan_Etki)
onehot_encoder = OneHotEncoder(sparse=False, drop='first')
categorical_cols = ['Uyruk', 'Kan Grubu', 'Yan_Etki']
# Apply one-hot encoding and convert to DataFrame
encoded_features =
pd.DataFrame(onehot_encoder.fit_transform(train_data[categorical_cols]
),

columns=onehot_encoder.get_feature_names_out(categorical_cols))

# Merge back to original dataset and drop original categorical columns
train_data = pd.concat([train_data, encoded_features], axis=1)
train_data.drop(columns=categorical_cols, inplace=True)

# 3. Standardizing Numerical Features
scaler = StandardScaler()
train_data[['Kilo', 'Boy']] = scaler.fit_transform(train_data[['Kilo',
'Boy']])
# Display the processed data head
train_data.head()
```

KNNImputer is used to fill in missing values in the dataset using the K-nearest neighbors algorithm.

OneHotEncoder is used to convert categorical (non-numerical) variables into a numerical format that machine learning algorithms can understand.

LabelEncoder is used to convert binary categorical variables into numerical labels (0 or 1). It is typically used when there are only two categories.

StandardScaler is used to standardize (normalize) the numerical features so they all have a mean of 0 and a standard deviation of 1.

Summary of task2:

Handling Missing Values:

Numerical features (Kilo and Boy) were imputed using the KNNImputer with 5 nearest neighbors.

The categorical feature Kan Grubu was imputed with the mode (most frequent value).

Encoding Categorical Variables:

Cinsiyet (Gender) was encoded using LabelEncoder as it is a binary categorical feature.

Uyruk (Nationality), Kan Grubu (Blood Type), and Yan_Etki (Side Effects) were one-hot encoded using OneHotEncoder.

Standardizing Numerical Features:

Both Kilo (Weight) and Boy (Height) were standardized using StandardScaler to ensure they have mean 0 and standard deviation 1.