Well_Being_Index

April 6, 2025

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
```

[2]:
```
train_data = pd.read_csv('/content/Train_data (1).csv')
test_data = pd.read_csv('/content/Validation (1).csv')
```

[3]:
```
train_data.head()
```

[3]:

	ID	galactic year	galaxy	existence expectancy index	existence expectancy at birth	Gross income per capita	Income Index	Expected years of education (galactic years)
0	10065	1002001	Antlia B	0.624015	56.397241	17649.87156	0.458599	7.857841
1	10087	999000	KKH 11 (ZOAG G135.74-04.53)	0.970048	80.924094	11409.94296	0.757218	15.869798
2	10141	993012	Leo IV Dwarf	0.995540	82.441006	58774.29343	1.032429	17.545117
3	10168	995006	NGC 185	1.004362	75.635986	34960.41911	0.707776	13.578086
4	10201	996004	Grus I	1.050627	83.412540	17073.45121	0.951402	13.518157

```
   Mean years of education (galactic years)  \
0                                  5.196527
1                                 13.065734
2                                 11.399711
3                                       NaN
4                                 11.749071

   Intergalactic Development Index (IDI)  …  \
0                               0.507534  …
1                               0.807108  …
2                               0.973684  …
3                                    NaN  …
4                               0.965452  …

   Intergalactic Development Index (IDI), female  \
0                                            NaN
1                                            NaN
2                                            NaN
3                                            NaN
4                                            NaN

   Intergalactic Development Index (IDI), male  \
0                                          NaN
1                                          NaN
2                                          NaN
3                                          NaN
4                                          NaN

   Gender Development Index (GDI)  \
0                             NaN
1                             NaN
2                             NaN
3                             NaN
4                             NaN

   Intergalactic Development Index (IDI), female, Rank  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   Intergalactic Development Index (IDI), male, Rank  Adjusted net savings   \
0                                                 NaN                    NaN
1                                                 NaN                    NaN
2                                                 NaN                    NaN
```

```
3                                                  NaN                    NaN
4                                                  NaN                    NaN

   Creature Immunodeficiency Disease prevalence, adult (% ages 15-49), total  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                             1.546539
4                                                  NaN

   Private galaxy capital flows (% of GGP)  Gender Inequality Index (GII)  \
0                                      NaN                           NaN
1                                      NaN                           NaN
2                                      NaN                           NaN
3                                      NaN                      0.562809
4                                      NaN                           NaN

   Well-Being Index
0          0.041404
1          0.098777
2          0.200747
3          0.067170
4          0.078351

[5 rows x 81 columns]
```

[4]:
```
test_data.head()
```

[4]:
```
       ID  galactic year                 galaxy  existence expectancy index  \
0  886447        1004004  Andromeda Galaxy (M31)                       0.803915
1  687564        1005006  Andromeda Galaxy (M31)                       0.860011
2  494935        1006009  Andromeda Galaxy (M31)                       0.810644
3  378919        1015056  Andromeda Galaxy (M31)                       0.837170
4  421878        1004004             Andromeda I                       0.749034

   existence expectancy at birth  Gross income per capita  Income Index  \
0                      82.718434                17299.57148      0.691448
1                      73.682279                24971.71631      0.669550
2                      68.456526                15943.82977      0.766118
3                      68.742404                20952.63665      0.757196
4                      72.093220                30068.14043      0.641228

   Expected years of education (galactic years)  \
0                                     16.083635
1                                     12.858577
2                                     14.236676
3                                     14.281498
```

```
4                                    12.510524

   Mean years of education (galactic years)  \
0                                  11.282011
1                                  10.493260
2                                   9.962169
3                                  10.329880
4                                   7.132999

   Intergalactic Development Index (IDI)  …  \
0                                0.715746  …
1                                0.727915  …
2                                0.757072  …
3                                0.759207  …
4                                0.673619  …

   Intergalactic Development Index (IDI), female  \
0                                            NaN
1                                            NaN
2                                            NaN
3                                            NaN
4                                            NaN

   Intergalactic Development Index (IDI), male  \
0                                          NaN
1                                          NaN
2                                          NaN
3                                          NaN
4                                          NaN

   Gender Development Index (GDI)  \
0                             NaN
1                             NaN
2                             NaN
3                             NaN
4                             NaN

   Intergalactic Development Index (IDI), female, Rank  \
0                                                 NaN
1                                                 NaN
2                                                 NaN
3                                                 NaN
4                                                 NaN

   Intergalactic Development Index (IDI), male, Rank  Adjusted net savings   \
0                                                 NaN                    NaN
1                                                 NaN                    NaN
```

```
2                                                   NaN                    NaN
3                                                   NaN                    NaN
4                                                   NaN                    NaN

   Creature Immunodeficiency Disease prevalence, adult (% ages 15-49), total  \
0                                                  NaN
1                                                  NaN
2                                                  NaN
3                                                  NaN
4                                                  NaN

   Private galaxy capital flows (% of GGP)  Gender Inequality Index (GII)  \
0                                      NaN                            NaN
1                                      NaN                            NaN
2                                      NaN                            NaN
3                                24.753738                            NaN
4                                      NaN                            NaN

   Predicted Well-Being Index
0                         NaN
1                         NaN
2                         NaN
3                         NaN
4                         NaN

[5 rows x 81 columns]
```

[5]:
```python
# Ensure all columns are displayed
pd.set_option('display.max_columns', None)

# Calculate percentage of missing values per column
missing_percentage = (train_data.isnull().sum() / len(train_data)) * 100

# Print missing values sorted in descending order
print(missing_percentage.sort_values(ascending=False).head(30))
```

```
Current health expenditure (% of GGP)
90.054892
Interstellar Data Net users, total (% of population)
90.022603
Interstellar phone subscriptions (per 100 people)
89.764288
Respiratory disease incidence (per 100,000 people)
89.635131
Gender Inequality Index (GII)
88.472716
Intergalactic Development Index (IDI), male, Rank
87.794640
```

Intergalactic Development Index (IDI), female, Rank
87.762351
Private galaxy capital flows (% of GGP)
87.762351
Gender Development Index (GDI)
87.504036
Intergalactic Development Index (IDI), male
87.471747
Intergalactic Development Index (IDI), female
87.439458
Adjusted net savings
85.695835
Rural population with access to electricity (%)
85.243784
Intergalactic inbound tourists (thousands)
85.179206
Estimated gross galactic income per capita, male
84.339684
Estimated gross galactic income per capita, female
84.339684
Remittances, inflows (% of GGP)
83.823055
Creature Immunodeficiency Disease prevalence, adult (% ages 15-49), total
83.338715
Domestic credit provided by financial sector (% of GGP)
82.563771
Population with at least some secondary education (% ages 25 and older)
82.208589
Expected years of education, male (galactic years)
82.111721
Expected years of education, female (galactic years)
82.079432
Gross fixed capital formation (% of GGP)
82.014853
Gross enrolment ratio, primary (% of primary under-age population)
81.885696
Share of seats in senate (% held by female)
81.530513
Population with at least some secondary education, male (% ages 25 and older)
81.465935
Population with at least some secondary education, female (% ages 25 and older)
81.465935
Natural resource depletion
81.304488
Mean years of education, male (galactic years)
80.981595
Mean years of education, female (galactic years)
80.917016

```
dtype: float64
```

[6]:
```
# Check for missing data and identify numeric columns
missing_data = train_data.isnull().mean().sort_values(ascending=False)
numeric_df = train_data.select_dtypes(include=['number'])

# Check how many numeric columns we have and preview missingness
numeric_df.shape, missing_data.head(30)
```

[6]:
```
((3097, 80),
 Current health expenditure (% of GGP)
0.900549
 Interstellar Data Net users, total (% of population)
0.900226
 Interstellar phone subscriptions (per 100 people)
0.897643
 Respiratory disease incidence (per 100,000 people)
0.896351
 Gender Inequality Index (GII)
0.884727
 Intergalactic Development Index (IDI), male, Rank
0.877946
 Intergalactic Development Index (IDI), female, Rank
0.877624
 Private galaxy capital flows (% of GGP)
0.877624
 Gender Development Index (GDI)
0.875040
 Intergalactic Development Index (IDI), male
0.874717
 Intergalactic Development Index (IDI), female
0.874395
 Adjusted net savings
0.856958
 Rural population with access to electricity (%)
0.852438
 Intergalactic inbound tourists (thousands)
0.851792
 Estimated gross galactic income per capita, male
0.843397
 Estimated gross galactic income per capita, female
0.843397
 Remittances, inflows (% of GGP)
0.838231
 Creature Immunodeficiency Disease prevalence, adult (% ages 15-49), total
0.833387
 Domestic credit provided by financial sector (% of GGP)
```

```
0.825638
 Population with at least some secondary education (% ages 25 and older)
0.822086
 Expected years of education, male (galactic years)
0.821117
 Expected years of education, female (galactic years)
0.820794
 Gross fixed capital formation (% of GGP)
0.820149
 Gross enrolment ratio, primary (% of primary under-age population)
0.818857
 Share of seats in senate (% held by female)
0.815305
 Population with at least some secondary education, male (% ages 25 and older)
0.814659
 Population with at least some secondary education, female (% ages 25 and older)
0.814659
 Natural resource depletion
0.813045
 Mean years of education, male (galactic years)
0.809816
 Mean years of education, female (galactic years)
0.809170
 dtype: float64)
```

[7]:
```python
# Ensure all columns are displayed
pd.set_option('display.max_columns', None)

# Calculate percentage of missing values per column
missing_percentage = (test_data.isnull().sum() / len(test_data)) * 100

# Print missing values sorted in descending order
print(missing_percentage.sort_values(ascending=False).head(50))
```

```
Predicted Well-Being Index
100.000000
Creature Immunodeficiency Disease prevalence, adult (% ages 15-49), total
41.805556
Adjusted net savings
35.416667
Gross enrolment ratio, primary (% of primary under-age population)
34.583333
Gender Inequality Index (GII)
33.194444
Private galaxy capital flows (% of GGP)
31.944444
Population with at least some secondary education (% ages 25 and older)
31.666667
```

Population with at least some secondary education, male (% ages 25 and older)
29.861111
Intergalactic Development Index (IDI), female, Rank
29.722222
Intergalactic Development Index (IDI), male, Rank
29.722222
Population with at least some secondary education, female (% ages 25 and older)
29.583333
Gross fixed capital formation (% of GGP)
29.166667
Gender Development Index (GDI)
27.916667
Intergalactic Development Index (IDI), male
27.916667
Intergalactic Development Index (IDI), female
27.916667
Remittances, inflows (% of GGP)
27.638889
Intergalactic inbound tourists (thousands)
26.527778
Domestic credit provided by financial sector (% of GGP)
26.111111
Mean years of education, female (galactic years)
24.861111
Mean years of education, male (galactic years)
24.861111
Share of seats in senate (% held by female)
24.722222
Exports and imports (% of GGP)
24.444444
Expected years of education, male (galactic years)
24.166667
Expected years of education, female (galactic years)
24.166667
Natural resource depletion
24.027778
Current health expenditure (% of GGP)
23.055556
Interstellar Data Net users, total (% of population)
22.638889
Mortality rate, male grown up (per 1,000 people)
22.638889
Mortality rate, female grown up (per 1,000 people)
22.638889
Outer Galaxies direct investment, net inflows (% of GGP)
22.083333
Estimated gross galactic income per capita, female
21.944444

```
Gross galactic product (GGP), total
21.944444
Gross galactic product (GGP) per capita
21.944444
Estimated gross galactic income per capita, male
21.944444
Jungle area (% of total land area)
21.805556
Unemployment, youth (% ages 15â€"24)
21.666667
Total unemployment rate (female to male ratio)
21.666667
Labour force participation rate (% ages 15 and older)
21.666667
Labour force participation rate (% ages 15 and older), female
21.666667
Employment in agriculture (% of total employment)
21.666667
Unemployment, total (% of labour force)
21.666667
Employment to population ratio (% ages 15 and older)
21.666667
Youth unemployment rate (female to male ratio)
21.666667
Vulnerable employment (% of total employment)
21.666667
Employment in services (% of total employment)
21.666667
Labour force participation rate (% ages 15 and older), male
21.666667
Share of employment in nonagriculture, female (% of total employment in
nonagriculture)     21.666667
Rural population with access to electricity (%)
21.388889
Population, ages 65 and older (millions)
21.111111
Maternal mortality ratio (deaths per 100,000 live births)
21.111111
dtype: float64
```

[8]:
```
#drop ID column on both datasets

train_data8 = train_data.drop('ID', axis=1)
test_data8 = test_data.drop('ID', axis=1)


merge_data = pd.concat([train_data8,test_data8])
```

[9]:
```
# Ensure only numeric columns are considered
numeric_cols = train_data8.select_dtypes(include=['number']).columns

# Define a threshold for dropping (e.g., 95% zeros)
zero_threshold = 0.20

# Compute the proportion of zeros in each numeric column (ignoring NaNs)
zero_proportion = (merge_data[numeric_cols] == 0).mean()

# Identify columns to drop based on threshold
columns_to_drop = zero_proportion[zero_proportion > zero_threshold].index

# Drop the identified columns from both datasets
train_data.drop(columns=columns_to_drop, inplace=True, errors='ignore')
test_data.drop(columns=columns_to_drop, inplace=True, errors='ignore')

print(f"Dropped columns: {list(columns_to_drop)}")
```

```
Dropped columns: []
```

[10]:
```
# Drop columns with more than 50% missing values
threshold = 0.5
# Recalculate missing_data using train_data8
missing_data_updated = merge_data.isnull().mean().sort_values(ascending=False)
valid_cols = missing_data_updated[missing_data_updated < threshold].index
clean_df = train_data8[valid_cols]

# Drop non-numeric columns and isolate target
numeric_clean_df = clean_df.select_dtypes(include=['number'])

# Separate features and target
features = numeric_clean_df.drop(columns=['Well-Being Index'], errors='ignore')
target = numeric_clean_df['Well-Being Index']

# Impute missing values with median
features_imputed = features.fillna(features.median())

features_imputed.shape, target.shape
```

[10]:
```
((3097, 12), (3097,))
```

[11]:
```
features_imputed.head()
```

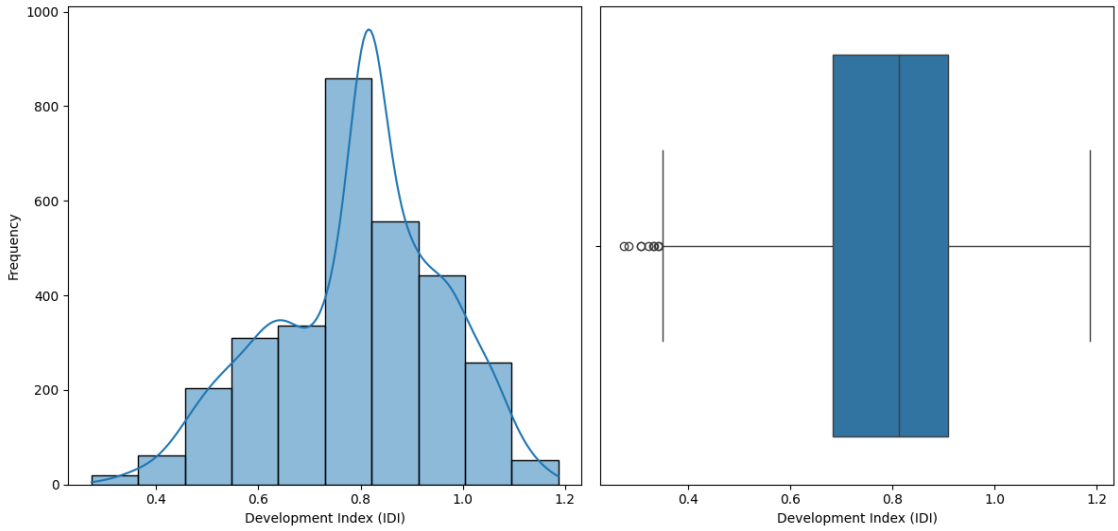
[11]:

	Population using at least basic sanitation services (%) \
0	58.079357
1	109.419112
2	109.419112
3	109.419112

```
4                                    109.419112

   Population using at least basic drinking-water services (%)  \
0                                          33.135967
1                                         107.649896
2                                         107.649896
3                                         107.649896
4                                         107.649896

   Intergalactic Development Index (IDI), Rank  \
0                                    247.196654
1                                    137.299057
2                                     74.709302
3                                    132.204365
4                                    141.210462

   Intergalactic Development Index (IDI)  Education Index  \
0                                0.507534         0.471400
1                                0.807108         0.837559
2                                0.973684         0.890396
3                                0.813372         0.748034
4                                0.965452         0.798000

   Mean years of education (galactic years)  \
0                                  5.196527
1                                 13.065734
2                                 11.399711
3                                 10.129151
4                                 11.749071

   Expected years of education (galactic years)  Gross income per capita  \
0                                      7.857841              17649.87156
1                                     15.869798              11409.94296
2                                     17.545117              58774.29343
3                                     13.578086              34960.41911
4                                     13.518157              17073.45121

   Income Index  existence expectancy index  existence expectancy at birth  \
0      0.458599                    0.624015                       56.397241
1      0.757218                    0.970048                       80.924094
2      1.032429                    0.995540                       82.441006
3      0.707776                    1.004362                       75.635986
4      0.951402                    1.050627                       83.412540

   galactic year
0        1002001
1         999000
```

```
2          993012
3          995006
4          996004
```

[12]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Intergalactic Development Index (IDI)'],␣
 ↪bins=10, kde=True, ax=axes[0])
axes[0].set_title('Distribution of Development Index (IDI)', color='white')
axes[0].set_xlabel('Development Index (IDI)')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Intergalactic Development Index (IDI)'],␣
 ↪ax=axes[1])
axes[1].set_title('Box plot of Development Index (IDI)', color='white')
axes[1].set_xlabel('Development Index (IDI)')
axes[1].set_ylabel('')

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

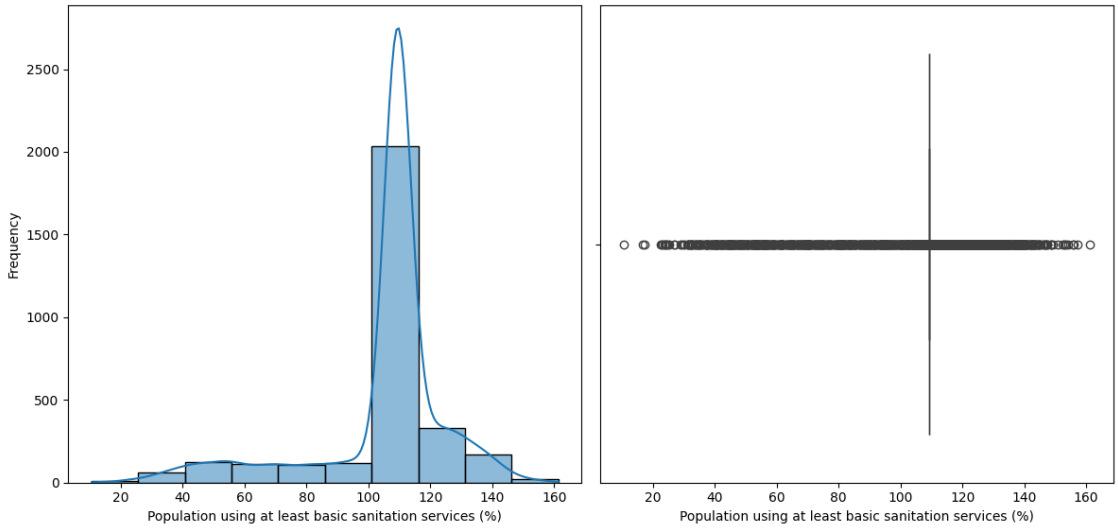


[13]:
```
features_imputed.columns
```

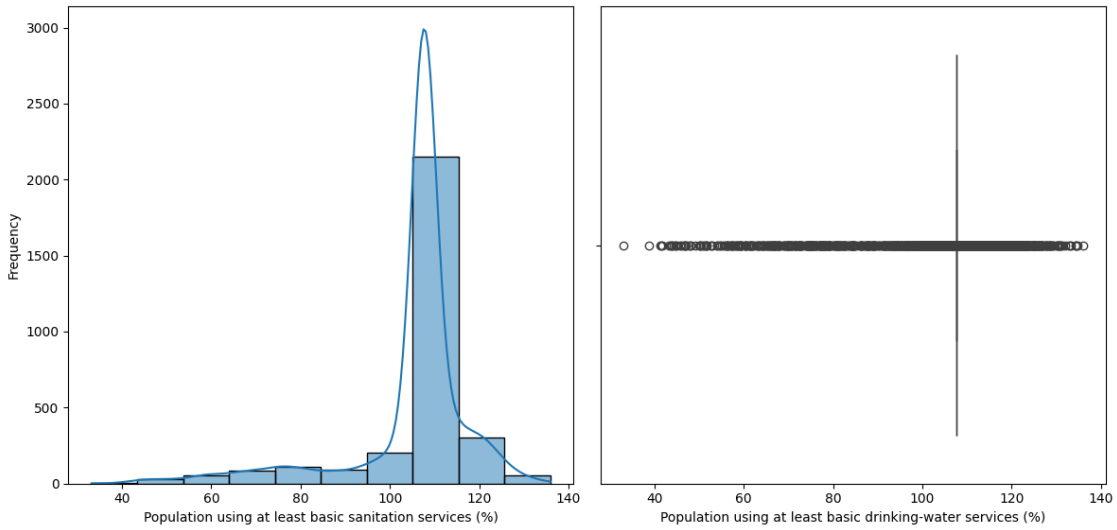
[13]:
```
Index(['Population using at least basic sanitation services (%)',
       'Population using at least basic drinking-water services (%)',
```

```
       'Intergalactic Development Index (IDI), Rank',
       'Intergalactic Development Index (IDI)', 'Education Index',
       'Mean years of education (galactic years)',
       'Expected years of education (galactic years)',
       'Gross income per capita', 'Income Index', 'existence expectancy index',
       'existence expectancy at birth', 'galactic year'],
      dtype='object')
```

[14]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Population using at least basic sanitation␣
 ↪services (%)'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Population using at least basic sanitation services (%)',␣
 ↪color='white')
axes[0].set_xlabel('Population using at least basic sanitation services (%)')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Population using at least basic sanitation␣
 ↪services (%)'], ax=axes[1])
axes[1].set_title('Box plot of Population using at least basic sanitation␣
 ↪services (%)', color='white')
axes[1].set_xlabel('Population using at least basic sanitation services (%)')
axes[1].set_ylabel('')

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

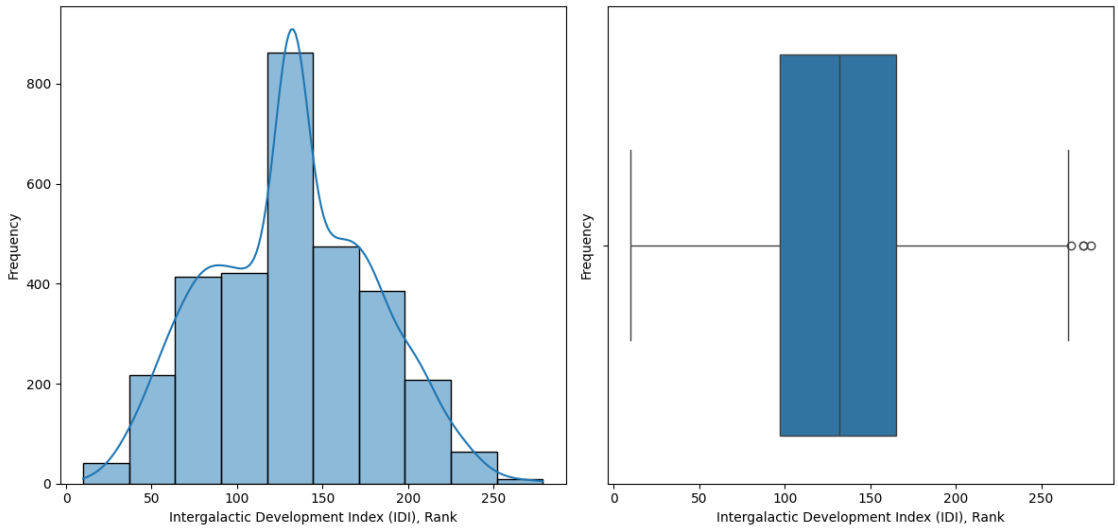


[15]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Population using at least basic drinking-water␣
 ↪services (%)'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Population using at least basic drinking-water services␣
 ↪(%)', color='white')
axes[0].set_xlabel('Population using at least basic sanitation services (%)')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Population using at least basic drinking-water␣
 ↪services (%)'], ax=axes[1])
axes[1].set_title('Population using at least basic drinking-water services␣
 ↪(%)', color='white')
axes[1].set_xlabel('Population using at least basic drinking-water services␣
 ↪(%)')
axes[1].set_ylabel('')

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



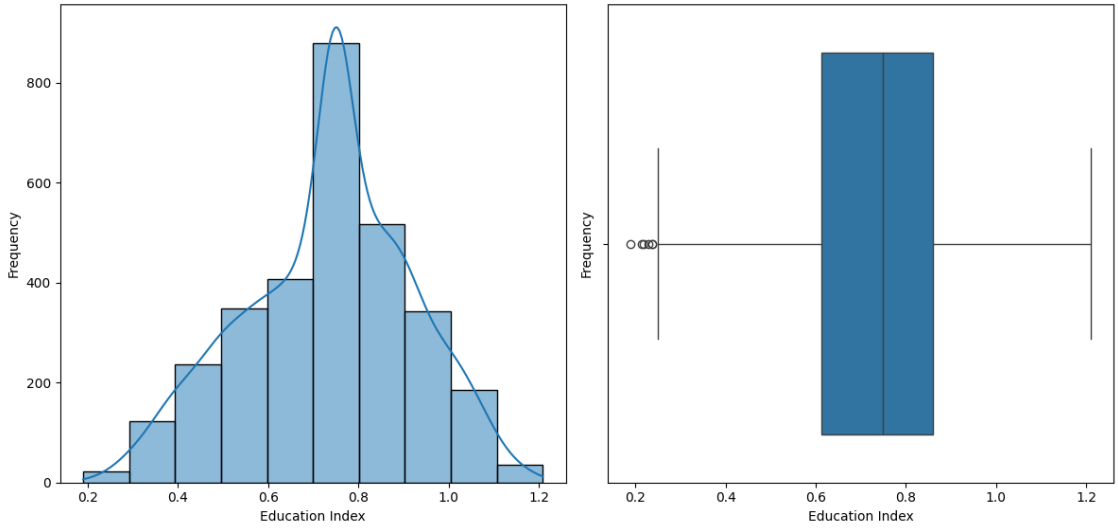
[16]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))
```

```
# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Intergalactic Development Index (IDI), Rank'],␣
 ↪bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Intergalactic Development Index (IDI), Rank')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Intergalactic Development Index (IDI), Rank'],␣
 ↪ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Intergalactic Development Index (IDI), Rank')
axes[1].set_ylabel('Frequency')

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

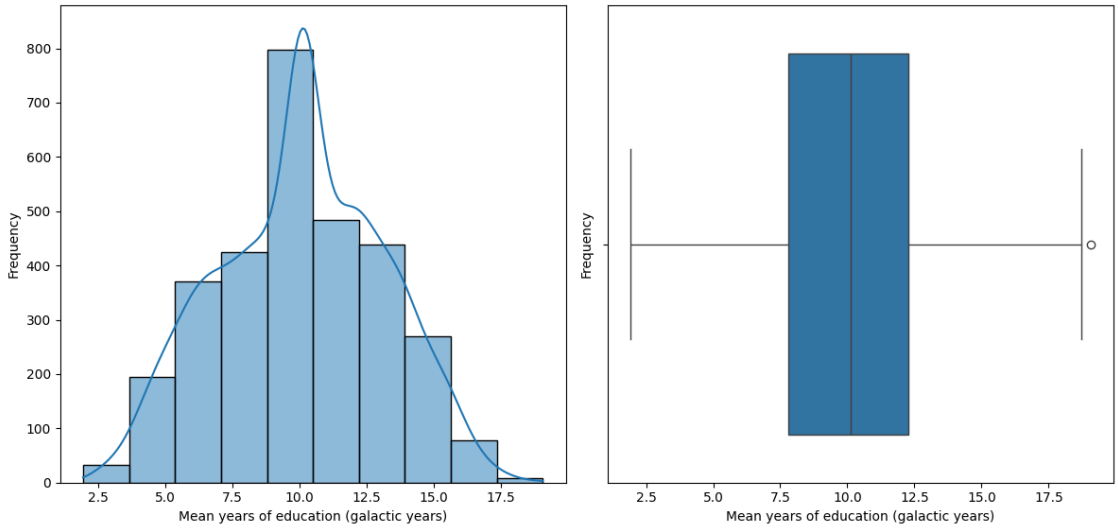


[17]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Education Index'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Education Index')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Education Index'], ax=axes[1])
```

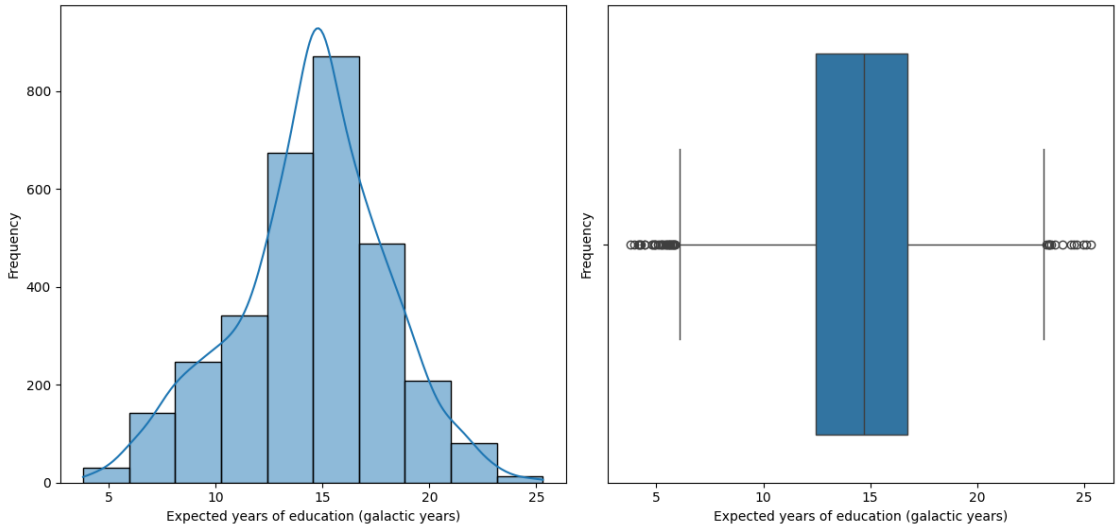
```
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Education Index')
axes[1].set_ylabel('Frequency')

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



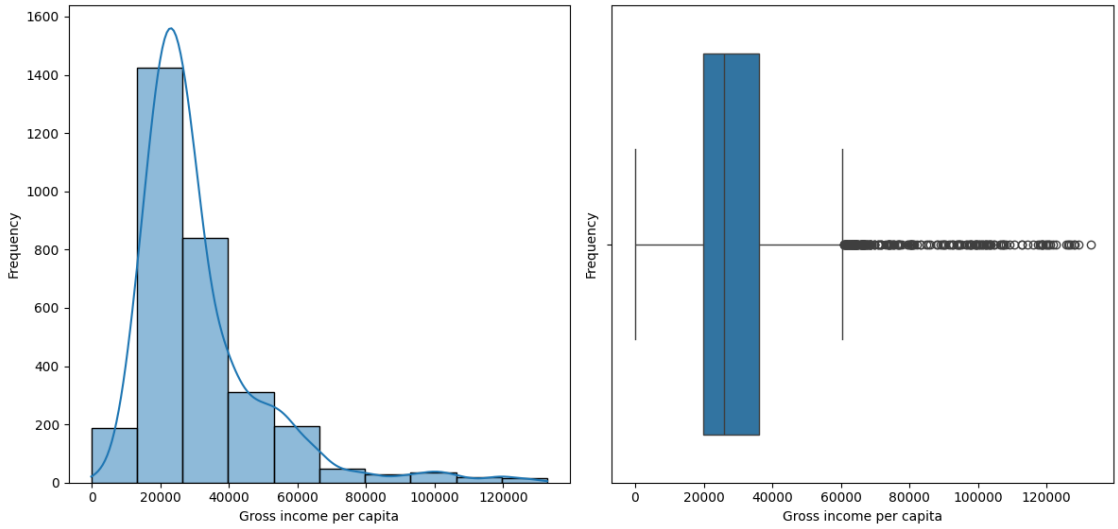
[18]:
```
#Mean years of education (galactic years)

# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Mean years of education (galactic years)'],␣
 ↪bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Mean years of education (galactic years)')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Mean years of education (galactic years)'],␣
 ↪ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Mean years of education (galactic years)')
axes[1].set_ylabel('Frequency')

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



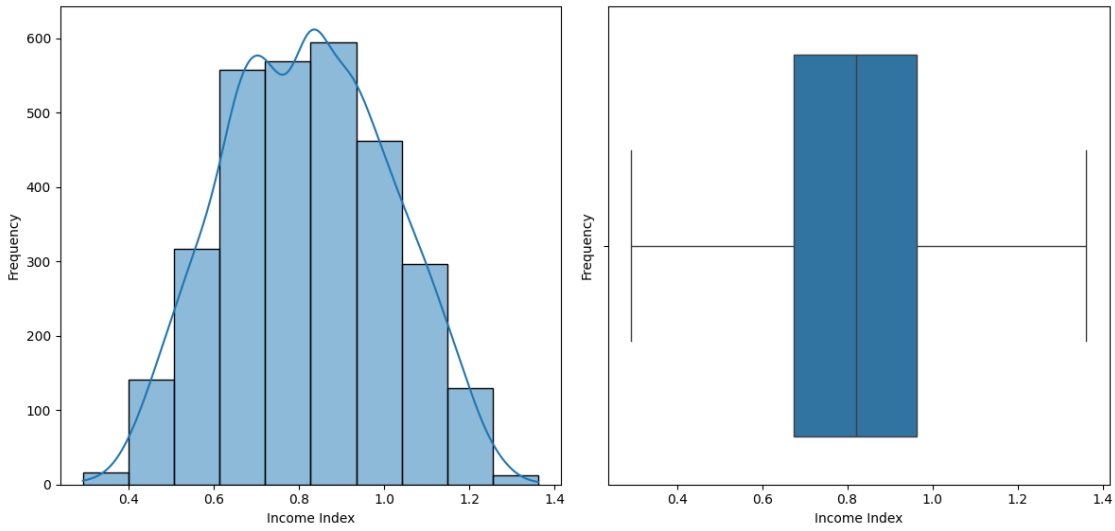
[19]:
```
#Expected years of education (galactic years)

# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Expected years of education (galactic years)'],␣
 ↪bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Expected years of education (galactic years)')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Expected years of education (galactic years)'],␣
 ↪ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Expected years of education (galactic years)')
axes[1].set_ylabel('Frequency')

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



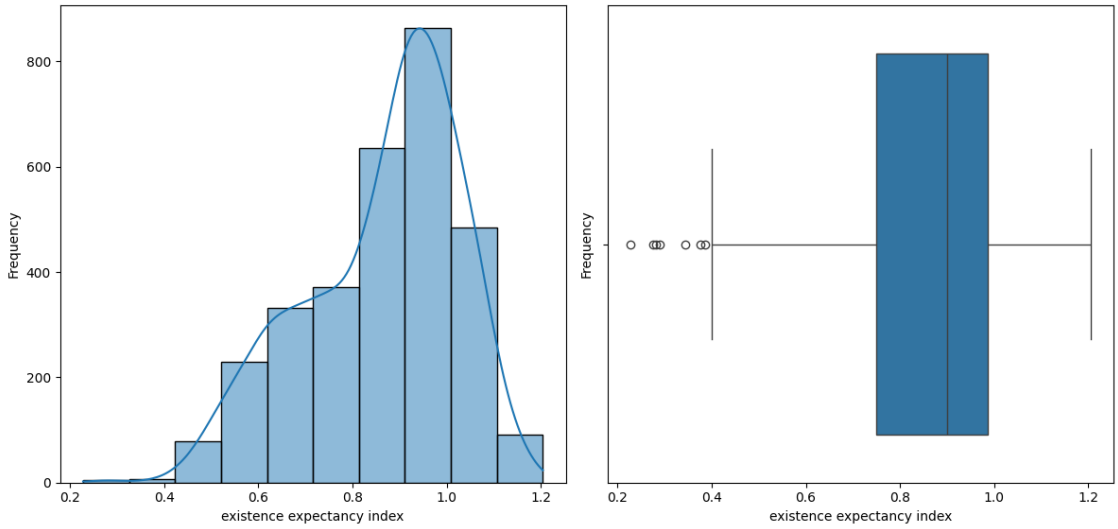
[20]:
```
#Expected years of education (galactic years)

# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Gross income per capita'], bins=10, kde=True,␣
↪ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Gross income per capita')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Gross income per capita'], ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Gross income per capita')
axes[1].set_ylabel('Frequency')

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



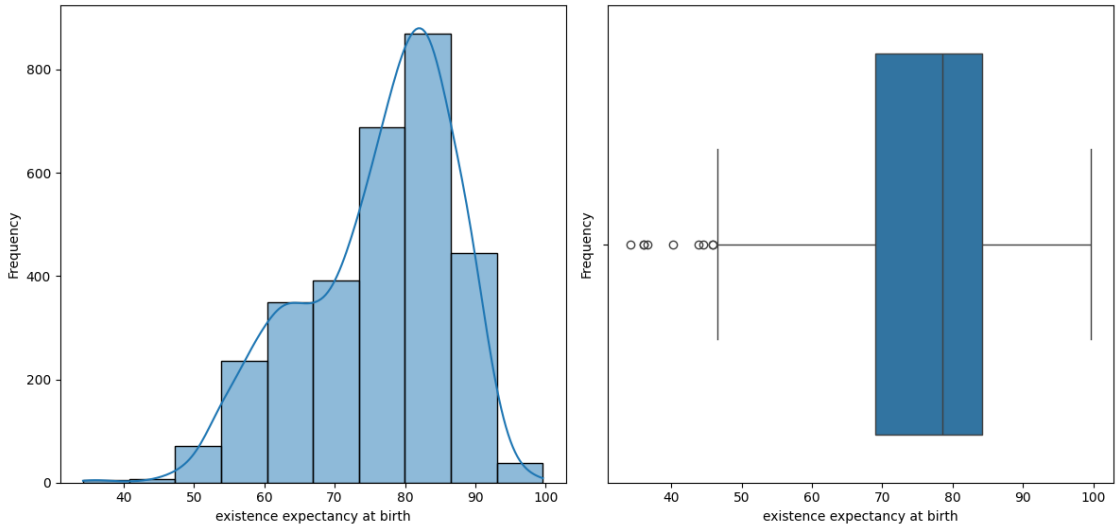
[21]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['Income Index'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('Income Index')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['Income Index'], ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('Income Index')
axes[1].set_ylabel('Frequency')

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



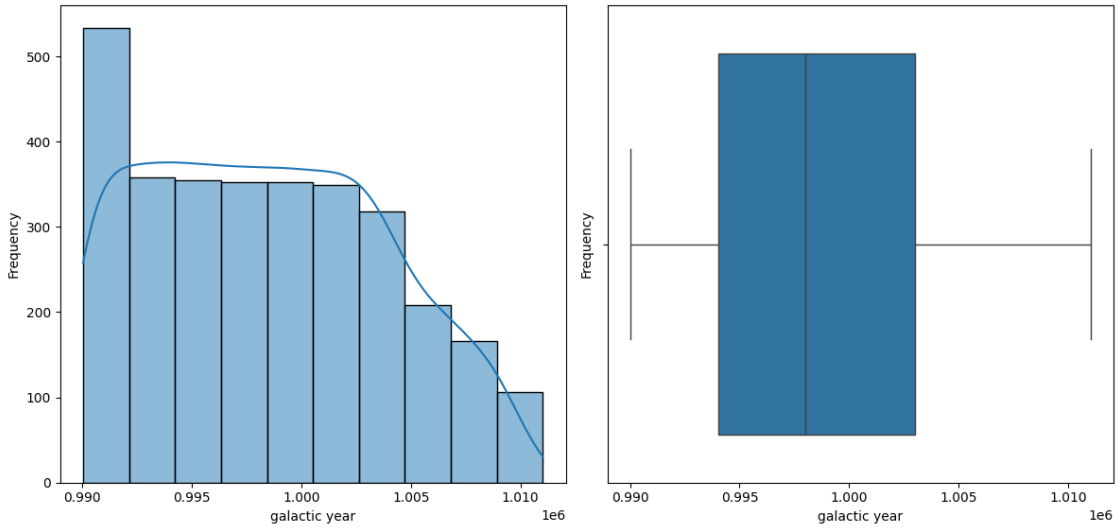
[22]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['existence expectancy index'], bins=10, kde=True,␣
↪ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('existence expectancy index')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['existence expectancy index'], ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('existence expectancy index')
axes[1].set_ylabel('Frequency')

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



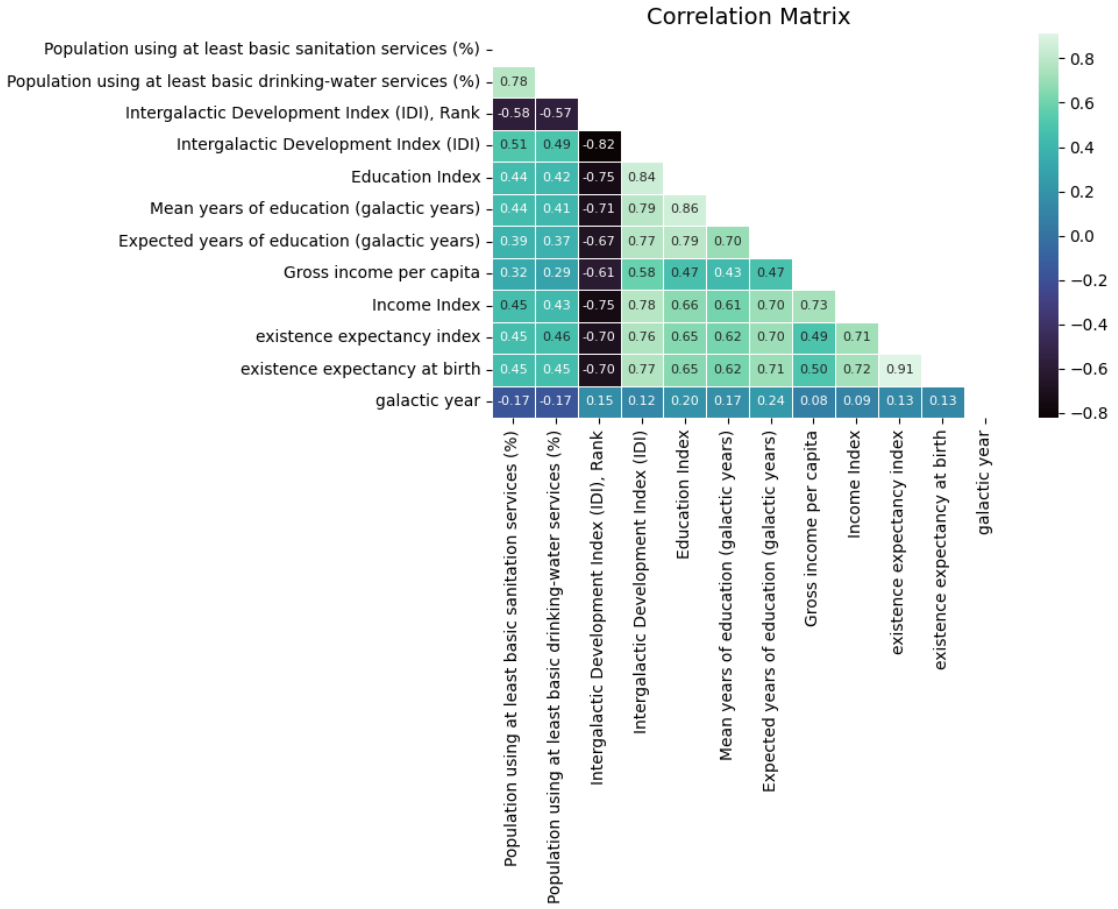
[23]:
```
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['existence expectancy at birth'], bins=10,␣
 ↪kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('existence expectancy at birth')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['existence expectancy at birth'], ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('existence expectancy at birth')
axes[1].set_ylabel('Frequency')

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



[24]:
```
#galactic year

# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot for Development Index (IDI)
sns.histplot(features_imputed['galactic year'], bins=10, kde=True, ax=axes[0])
axes[0].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[0].set_xlabel('galactic year')
axes[0].set_ylabel('Frequency')

# Box plot for Development Index (IDI)
sns.boxplot(x=features_imputed['galactic year'], ax=axes[1])
axes[1].set_title('Intergalactic Development Index (IDI), Rank', color='white')
axes[1].set_xlabel('galactic year')
axes[1].set_ylabel('Frequency')

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



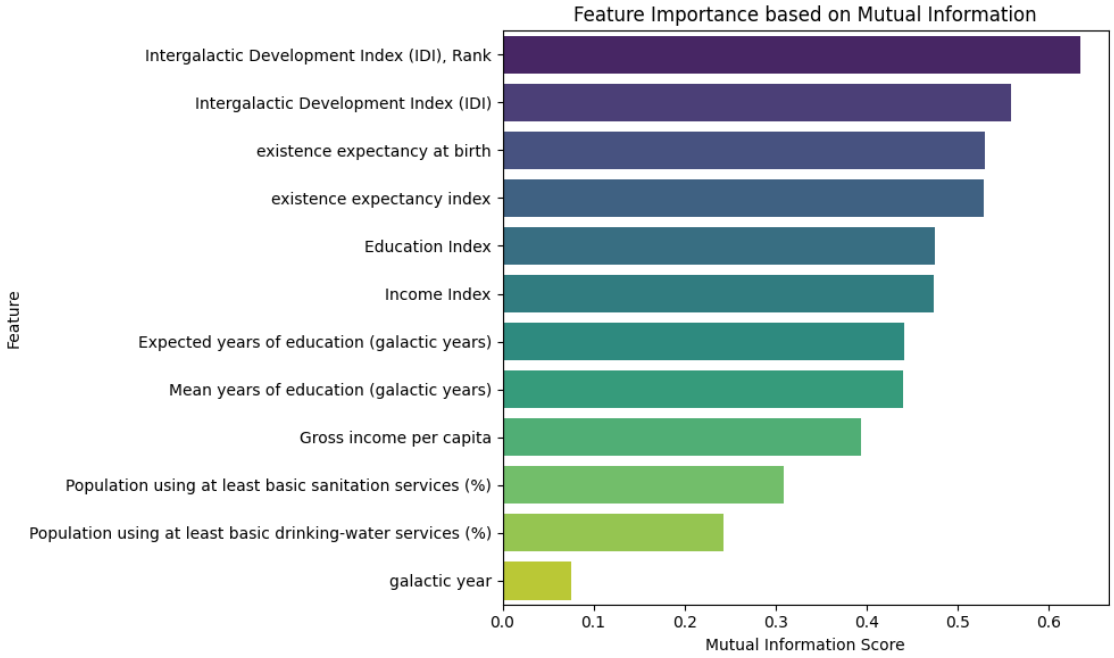
[25]:
```
 # Compute correlation matrix
corr = features_imputed.corr()

    # Plot heatmap
fig, ax = plt.subplots(figsize=(10, 8))
mask = np.triu(np.ones_like(corr, dtype=bool))
sns.heatmap(corr, mask=mask, cmap='mako', annot=True, linewidths=0.5, fmt=".
 ↪2f", annot_kws={"size": 8})
plt.title('Correlation Matrix', fontsize=14)
plt.tight_layout()
plt.show()
```



[30]:
```
# separate featured and target

X = features_imputed
y =  target


from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split

import lightgbm as lgb
#import catboost as cb
import time
import gc

from sklearn.model_selection import KFold
from sklearn.metrics import mean_squared_error, r2_score,␣
 ↪root_mean_squared_error
```

```
from sklearn.feature_selection import mutual_info_regression


X_test,X_train,y_test,y_train = train_test_split(X,y)
```

[31]:
```
# Assuming X and y are your features and target
mutual_info = mutual_info_regression(X, y)
mi_df = pd.DataFrame({'Feature': X.columns, 'Mutual Information': mutual_info})
mi_df = mi_df.sort_values(by='Mutual Information', ascending=False)


# Create the barh plot using seaborn, addressing the FutureWarning
plt.figure(figsize=(10, 6))
sns.barplot(data=mi_df, x='Mutual Information', y='Feature', hue='Feature',␣
 ↪palette='viridis', dodge=False, legend=False)
plt.title('Feature Importance based on Mutual Information')
plt.xlabel('Mutual Information Score')
plt.ylabel('Feature')
plt.tight_layout()
plt.show()
```



[32]:
```
# Train XGBoost model
model = xgb.XGBRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
# Plot top 100 features by gain
fig, ax = plt.subplots(figsize=(10, 4))
xgb.plot_importance(model, max_num_features=100, importance_type='gain', ax=ax)
plt.title("Top 10 Feature Importances (XGBoost)")
plt.show()
```

[37]:
```
# Clean column names


import warnings
warnings.filterwarnings('ignore')


X_train.columns = X_train.columns.str.replace(r'[^\w]', '_', regex=True)
X_test.columns = X_test.columns.str.replace(r'[^\w]', '_', regex=True)

# Now fit the model
model.fit(X_train, y_train)
```

```
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of
testing was 0.000089 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
```

```
[LightGBM] [Info] Total Bins 2828
[LightGBM] [Info] Number of data points in the train set: 775, number of used
features: 12
[LightGBM] [Info] Start training from score 0.085599
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

[37]:
```
LGBMRegressor(metric='rmse', objective='regression', random_state=42)
```

[36]:
```python
# Create and train the LGBM Regressor
model = lgb.LGBMRegressor(
    objective='regression',
    metric='rmse',
    random_state=42,
    n_estimators=100
)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model using RMSE
rmse = root_mean_squared_error(y_test, y_pred)
print(f"RMSE: {rmse}")
```

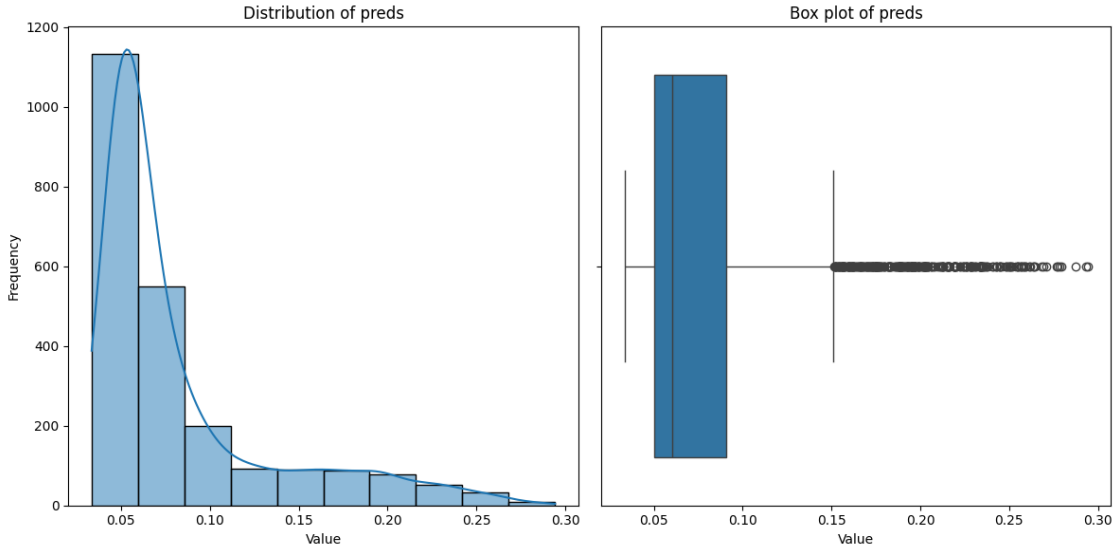
```
[LightGBM] [Info] Auto-choosing col-wise multi-threading, the overhead of
testing was 0.000107 seconds.
You can set `force_col_wise=true` to remove the overhead.
[LightGBM] [Info] Total Bins 2828
[LightGBM] [Info] Number of data points in the train set: 775, number of used
features: 12
[LightGBM] [Info] Start training from score 0.085599
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
```

```
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
[LightGBM] [Warning] No further splits with positive gain, best gain: -inf
RMSE: 0.02724299284891371
```

[39]:

```python
# Adjust the figure and axes creation
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Histogram plot
sns.histplot(y_pred, bins=10, kde=True, ax=axes[0])
axes[0].set_title('Distribution of preds')
axes[0].set_xlabel('Value')
axes[0].set_ylabel('Frequency')
```

```
# Box plot
sns.boxplot(x=y_pred, ax=axes[1])
axes[1].set_title('Box plot of preds')
axes[1].set_xlabel('Value')
axes[1].set_ylabel('')

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



[40]:
```
# Create a DataFrame with actual vs predicted
val_results = pd.DataFrame({
    "Galaxy_ID": X_test.index,
    "Actual_WellBeing": y_test,
    "Predicted_WellBeing": y_pred
})

# Save to CSV
version = 1  # Change this version number as needed
val_results.to_csv(f"validation_results_v{version}.csv", index=False)

# Display the first few rows
val_results.head()
```

[40]:

	Galaxy_ID	Actual_WellBeing	Predicted_WellBeing
1029	1029	0.167935	0.193286
1744	1744	0.050495	0.050803
1778	1778	0.113081	0.117344
1039	1039	0.076878	0.058537

431	431	0.058134	0.053167