In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
from scipy import stats
import seaborn as sns
import warnings
%matplotlib inline
import types
import pandas as pd
from botocore.client import Config
```

DATA PREPROCESSING

In [2]:
```
df = pd.read_csv("Life Expectancy Data.csv")
df.head()
```

Out[2]:

	Country	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B
0	Afghanistan	2015	Developing	65.0	263.0	62	0.01	71.279624	65.0
1	Afghanistan	2014	Developing	59.9	271.0	64	0.01	73.523582	62.0
2	Afghanistan	2013	Developing	59.9	268.0	66	0.01	73.219243	64.0
3	Afghanistan	2012	Developing	59.5	272.0	69	0.01	78.184215	67.0
4	Afghanistan	2011	Developing	59.2	275.0	71	0.01	7.097109	68.0

5 rows × 22 columns

In [3]:
```
df.shape
```

Out[3]:
```
(2938, 22)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2938 entries, 0 to 2937
Data columns (total 22 columns):
 #   Column                           Non-Null Count  Dtype  
---  ------                           --------------  -----  
 0   Country                          2938 non-null   object 
 1   Year                             2938 non-null   int64  
 2   Status                           2938 non-null   object 
 3   Life expectancy                  2928 non-null   float64
 4   Adult Mortality                  2928 non-null   float64
 5   infant deaths                    2938 non-null   int64  
 6   Alcohol                          2744 non-null   float64
 7   percentage expenditure           2938 non-null   float64
 8   Hepatitis B                      2385 non-null   float64
 9   Measles                          2938 non-null   int64  
 10   BMI                             2904 non-null   float64
 11  under-five deaths                2938 non-null   int64  
 12  Polio                            2919 non-null   float64
 13  Total expenditure                2712 non-null   float64
 14  Diphtheria                       2919 non-null   float64
 15   HIV/AIDS                        2938 non-null   float64
 16  GDP                              2490 non-null   float64
 17  Population                       2286 non-null   float64
 18   thinness  1-19 years            2904 non-null   float64
 19   thinness 5-9 years              2904 non-null   float64
 20  Income composition of resources  2771 non-null   float64
 21  Schooling                        2775 non-null   float64
dtypes: float64(16), int64(4), object(2)
memory usage: 505.1+ KB
```

In [5]:
```
df.describe()
```

Out[5]:

	Year	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatiti
count	2938.000000	2928.000000	2928.000000	2938.000000	2744.000000	2938.000000	2385.000
mean	2007.518720	69.224932	164.796448	30.303948	4.602861	738.251295	80.940
std	4.613841	9.523867	124.292079	117.926501	4.052413	1987.914858	25.070
min	2000.000000	36.300000	1.000000	0.000000	0.010000	0.000000	1.000
25%	2004.000000	63.100000	74.000000	0.000000	0.877500	4.685343	77.000
50%	2008.000000	72.100000	144.000000	3.000000	3.755000	64.912906	92.000
75%	2012.000000	75.700000	228.000000	22.000000	7.702500	441.534144	97.000
max	2015.000000	89.000000	723.000000	1800.000000	17.870000	19479.911610	99.000

In [6]:
```
df.columns
```

Out[6]:
```
Index(['Country', 'Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditur
e',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

In [7]:
```
df = df.drop(['Country'], axis=1)
df.columns
```

Out[7]:
```
Index(['Year', 'Status', 'Life expectancy ', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditur
e',
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling'],
      dtype='object')
```

In [8]:
```
new_df=df.fillna(df.mean())
new_df.isnull().sum()
```

```
C:\Users\kabil\AppData\Local\Temp\ipykernel_18296\2761969969.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only
=None') is deprecated; in a future version this will raise TypeError.  Select
only valid columns before calling the reduction.
  new_df=df.fillna(df.mean())
```

Out[8]:
```
Year                               0
Status                             0
Life expectancy                    0
Adult Mortality                    0
infant deaths                      0
Alcohol                            0
percentage expenditure             0
Hepatitis B                        0
Measles                            0
 BMI                               0
under-five deaths                  0
Polio                              0
Total expenditure                  0
Diphtheria                         0
 HIV/AIDS                          0
GDP                                0
Population                         0
 thinness  1-19 years              0
 thinness 5-9 years                0
Income composition of resources    0
Schooling                          0
dtype: int64
```

In [9]:
```
new_df.replace(to_replace=['Developing', 'Developed'],
           value= [0, 1],
           inplace=True)
new_df.head()
```

Out[9]:

	Year	Status	Life expectancy	Adult Mortality	infant deaths	Alcohol	percentage expenditure	Hepatitis B	Measles	BMI	..
0	2015	0	65.0	263.0	62	0.01	71.279624	65.0	1154	19.1	.
1	2014	0	59.9	271.0	64	0.01	73.523582	62.0	492	18.6	.
2	2013	0	59.9	268.0	66	0.01	73.219243	64.0	430	18.1	.
3	2012	0	59.5	272.0	69	0.01	78.184215	67.0	2787	17.6	.
4	2011	0	59.2	275.0	71	0.01	7.097109	68.0	3013	17.2	.

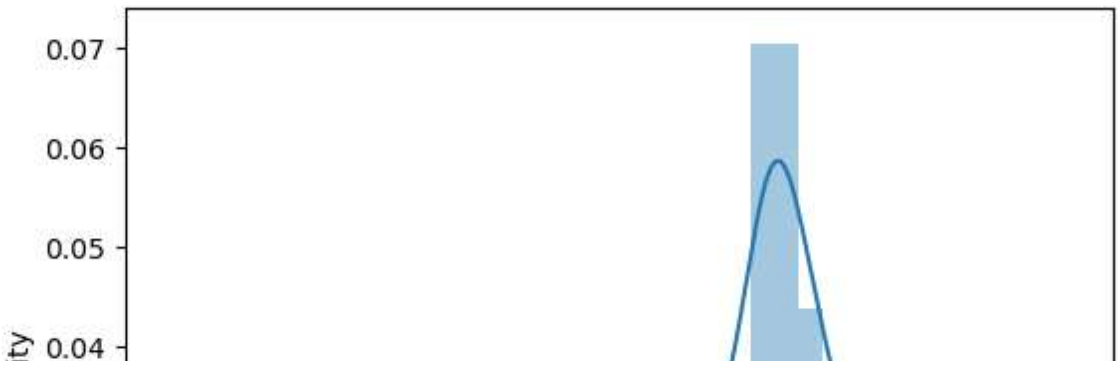
5 rows × 21 columns

In [10]:
```
#sns.pairplot(new_df)
```

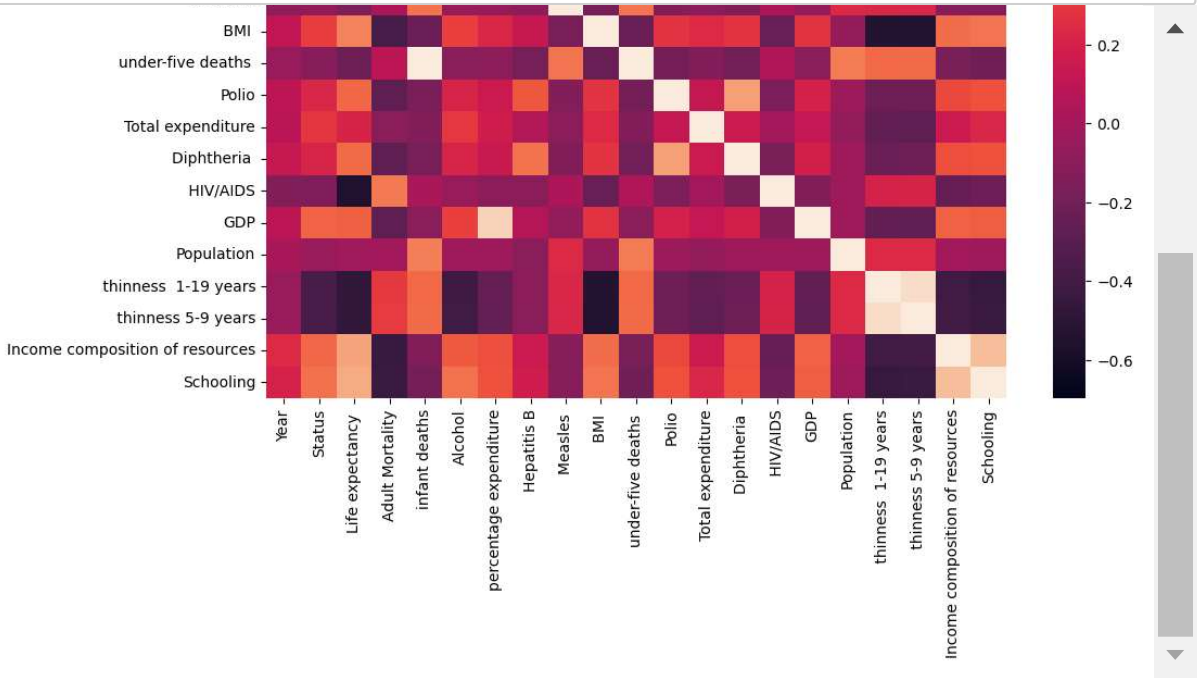
In [11]:
```
sns.distplot(new_df['Life expectancy '])
```

```
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[11]:
```
<AxesSubplot:xlabel='Life expectancy ', ylabel='Density'>
```



In [12]:
```
fig, ax = plt.subplots(figsize=(11, 8))
sns.heatmap(ax=ax, data=new_df.corr())
```



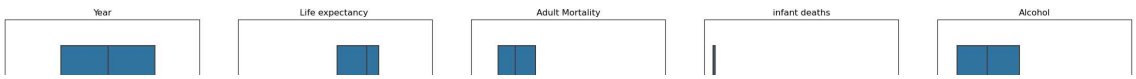
In [13]:
```
columns = {1: 'Year', 2: 'Life expectancy ', 3: 'Adult Mortality', 4: 'infant
        5: 'Alcohol' , 6: 'percentage expenditure', 7: 'Hepatitis B',
       8: 'Measles ', 9: ' BMI ', 10: 'under-five deaths ', 11: 'Polio', 12: '
       13: 'Diphtheria ', 14: ' HIV/AIDS', 15: 'GDP', 16: 'Population',
       17: ' thinness  1-19 years', 18: ' thinness 5-9 years',
       19: 'Income composition of resources', 20: 'Schooling'}

plt.figure(figsize=(28, 30))

for i, column in columns.items():
                    plt.subplot(4,5,i)
                    sns.boxplot(new_df[column], orient='v')
                    plt.title(column)

plt.show()
```

```
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\_core.py:1326: UserWarn
ing: Vertical orientation ignored with only `x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterp
retation.
  warnings.warn(
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\_core.py:1326: UserWarn
ing: Vertical orientation ignored with only `x` specified.
  warnings.warn(single_var_warning.format("Vertical", "x"))
```



In [14]:
```
X = new_df[['Year', 'Status', 'Adult Mortality',
       'infant deaths', 'Alcohol', 'percentage expenditure', 'Hepatitis B',
       'Measles ', ' BMI ', 'under-five deaths ', 'Polio', 'Total expenditure'
       'Diphtheria ', ' HIV/AIDS', 'GDP', 'Population',
       ' thinness  1-19 years', ' thinness 5-9 years',
       'Income composition of resources', 'Schooling']].values
y = new_df['Life expectancy '].values
```

In [15]:
```
from sklearn.model_selection import train_test_split
```

In [16]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, rando
```

In [17]:
```
new_df.shape
```

Out[17]: (2938, 21)

In [18]:
```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

Out[18]:
```
((1762, 20), (1176, 20), (1762,), (1176,))
```

In [19]:
```
print('Training Features Shape:', X_train.shape)
print('Training Labels Shape:', y_train.shape)
print('Testing Features Shape:', X_test.shape)
print('Testing Labels Shape:', y_test.shape)
```

```
Training Features Shape: (1762, 20)
Training Labels Shape: (1762,)
Testing Features Shape: (1176, 20)
Testing Labels Shape: (1176,)
```

LINEAR REGRESSION

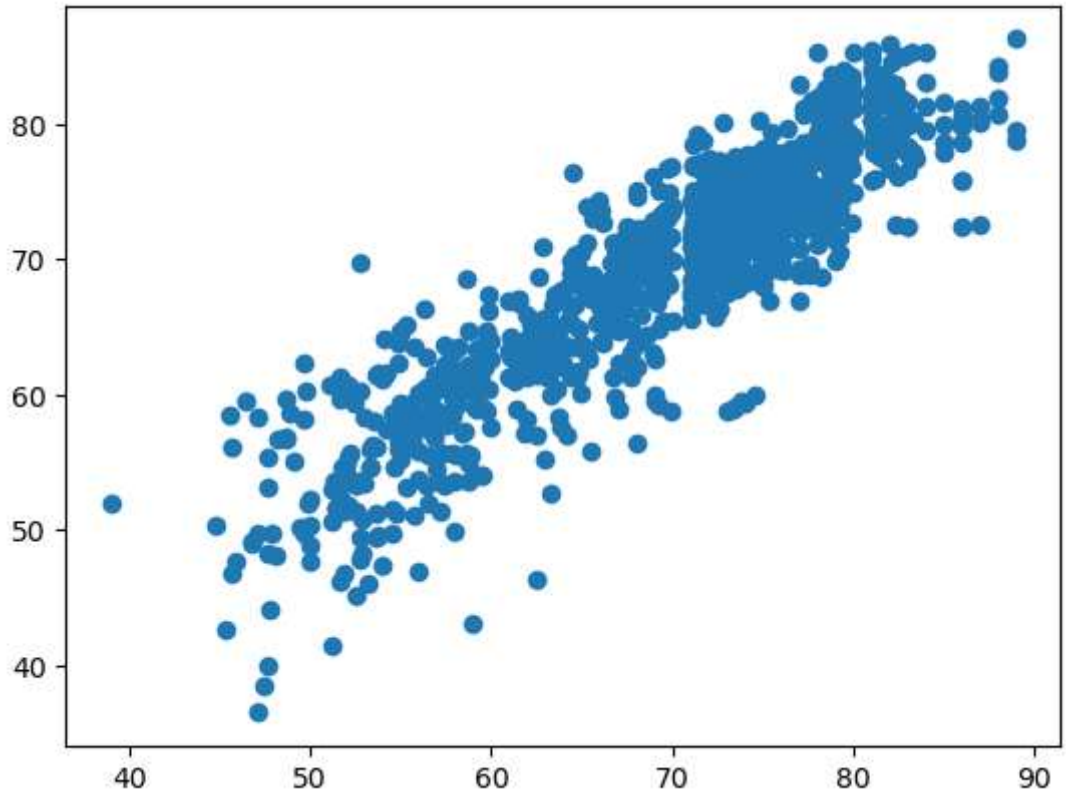
In [108]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
lm = LinearRegression()
lm.fit(X_train,y_train)
lmpredictions = lm.predict(X_test)
```

In [109]:
```
mse = mean_squared_error(y_test,lmpredictions)
mae = mean_absolute_error(y_test,lmpredictions)
r2 = r2_score(y_test, lmpredictions)
print("Mean Squared Error:",mse)
print("Mean Absolute Error:",mae)
print("R2 Square:",r2)
```

```
Mean Squared Error: 16.279182568375546
Mean Absolute Error: 3.0322513121378054
R2 Square: 0.803481416052645
```

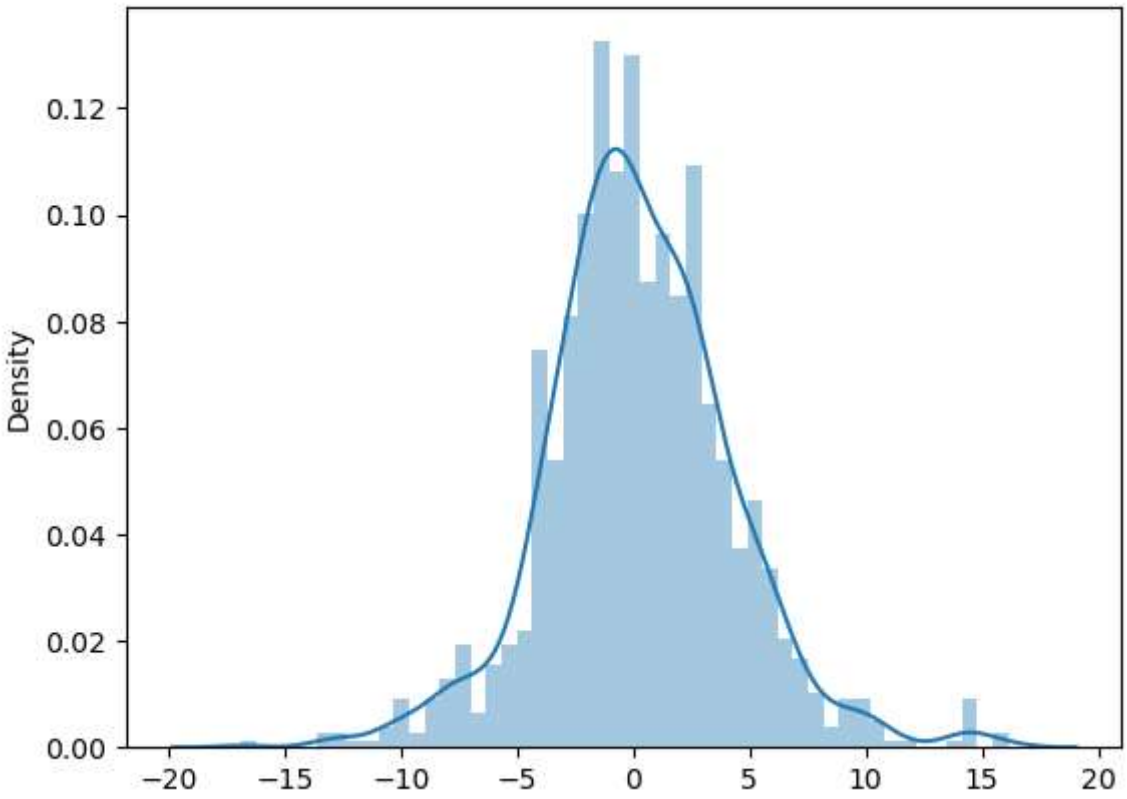
In [110]:
```
plt.scatter(y_test,lmpredictions)
```

Out[110]:
```
<matplotlib.collections.PathCollection at 0x1fa26d67b80>
```



In [111]:
```
sns.distplot((y_test-lmpredictions),bins=50);
```

```
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```



RANDOM FOREST

In [112]:
```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators = 40, random_state = 50)
rf.fit(X_train, y_train)
```

Out[112]:
```
RandomForestRegressor(n_estimators=40, random_state=50)
```

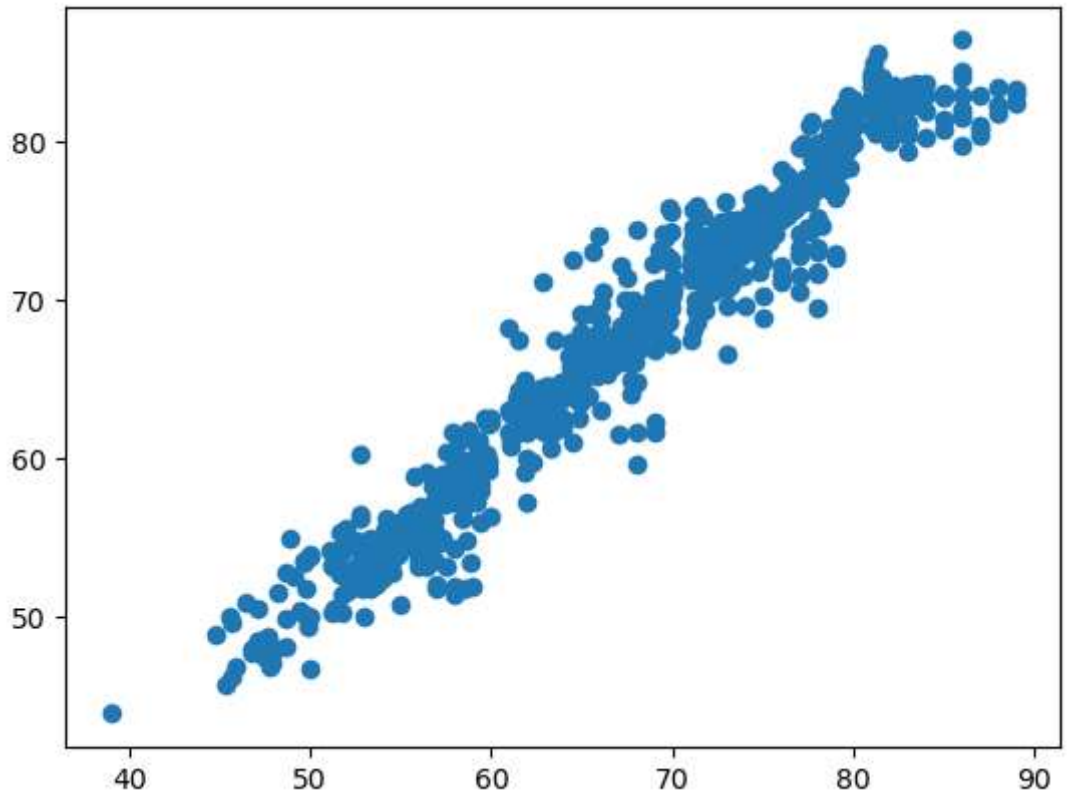
In [113]:
```
rfpredictions= rf.predict(X_test)
```

In [114]:
```
mse = mean_squared_error(y_test,rfpredictions)
mae = mean_absolute_error(y_test,rfpredictions)
r2 = r2_score(y_test, rfpredictions)
print("Mean Squared Error:",mse)
print("Mean Absolute Error:",mae)
print("R2 Square:",r2)
```

```
Mean Squared Error: 3.7236637857805954
Mean Absolute Error: 1.2528619884600023
R2 Square: 0.9550487789418121
```

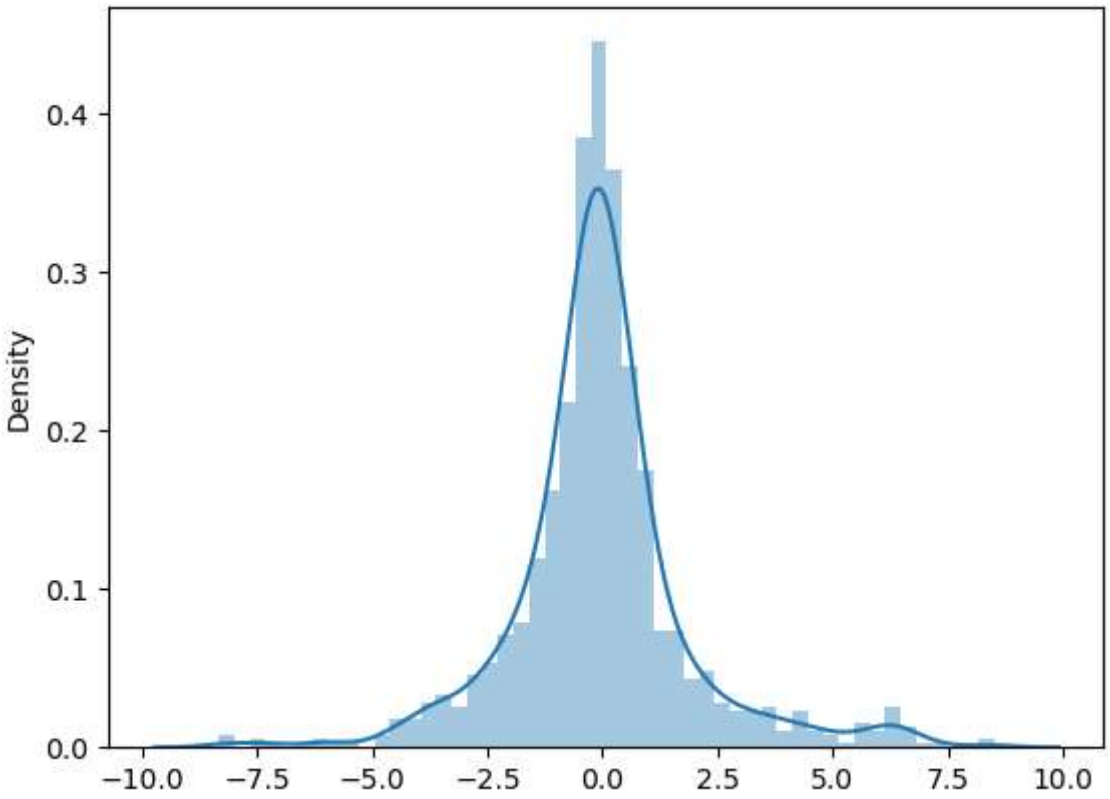
In [115]:
```
plt.scatter(y_test, rfpredictions)
```

Out[115]:
```
<matplotlib.collections.PathCollection at 0x1fa291b6df0>
```



In [116]:
```
sns.distplot((y_test-rfpredictions),bins=50);
```

```
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```



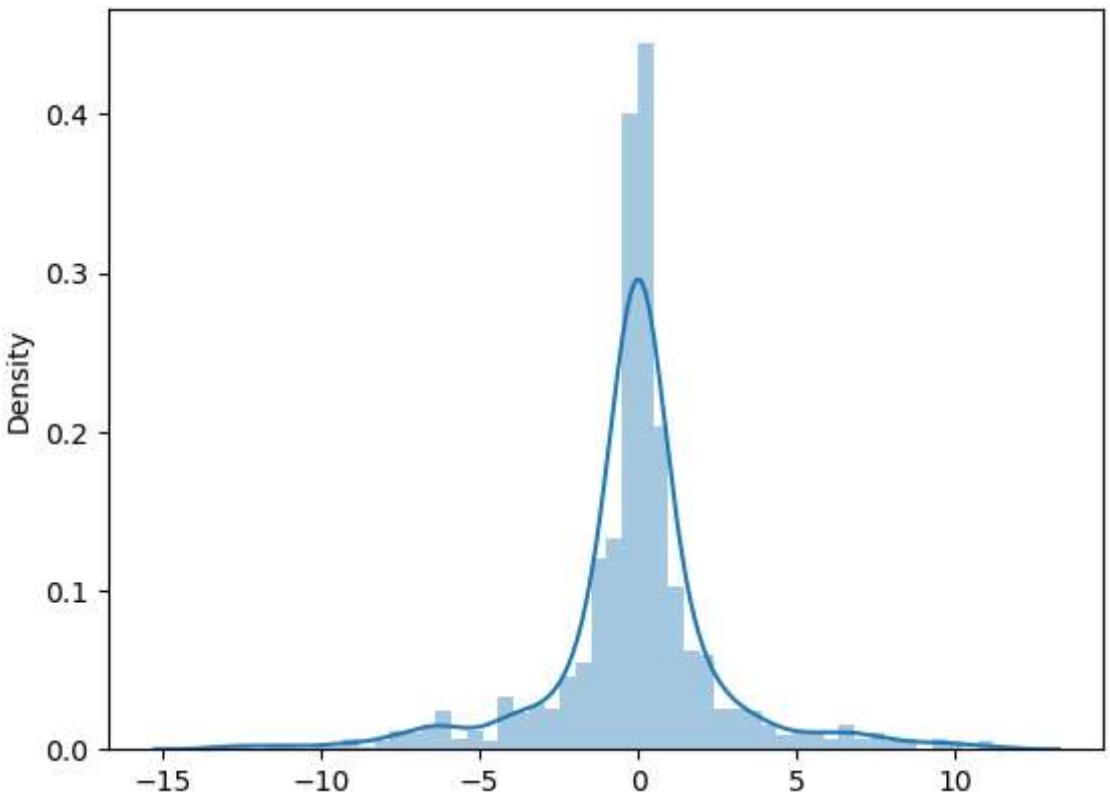
DECISION TREE

In [117]:
```
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
dt = DecisionTreeRegressor()
dt.fit(X_train, y_train)
dt_predictions = dt.predict(X_test)
print(dt_predictions)
mse = mean_squared_error(y_test,dt_predictions)
mae = mean_absolute_error(y_test,dt_predictions)
r2 = r2_score(y_test, dt_predictions)
print("Mean Squared Error:",mse)
print("Mean Absolute Error:",mae)
print("R2 Square:",r2)
```

```
[63.7 54.3 83.5 ... 63.9 72.2 79. ]
Mean Squared Error: 7.88756179489599
Mean Absolute Error: 1.662521432753429
R2 Square: 0.9047831506146139
```

In [118]:
```
sns.distplot((y_test-dt_predictions),bins=50);
```

```
C:\Users\kabil\anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```



RIDGE REGRESSION

In [119]:
```
from sklearn.linear_model import Ridge
ridge = Ridge()
ridge.fit(X_train, y_train)
ridge_predictions = ridge.predict(X_test)
mae = mean_absolute_error(y_test, ridge_predictions)
r2 = r2_score(y_test, ridge_predictions)
mse = mean_squared_error(y_test,ridge_predictions)
print("Mean absolute Error (Ridge Regression):", mae)
print("Mean squared Error (Ridge Regression):",mse)
print("R-squared Score (Ridge Regression):", r2)
```

```
Mean absolute Error (Ridge Regression): 3.0332047740254806
Mean squared Error (Ridge Regression): 16.28419819317254
R-squared Score (Ridge Regression): 0.803420868572538

C:\Users\kabil\anaconda3\lib\site-packages\sklearn\linear_model\_ridge.py:15
7: LinAlgWarning: Ill-conditioned matrix (rcond=4.50792e-18): result may not
be accurate.
  return linalg.solve(A, Xy, sym_pos=True, overwrite_a=True).T
```

LIFE EXPECTANCY PREDICTIONS

In [120]:

```
features = {
    'Year': [2023],
    'Status': [1],
    'Adult Mortality': [200],
    'infant deaths': [100],
    'Alcohol': [10],
    'percentage expenditure': [100],
    'Hepatitis B': [90],
    'Measles ': [50],
    ' BMI ': [25],
    'under-five deaths ': [15],
    'Polio': [95.23],
    'Total expenditure': [7.5],
    'Diphtheria ': [94],
    ' HIV/AIDS': [2.4],
    'GDP': [1500],
    'Population': [1000],
    ' thinness  1-19 years': [7],
    ' thinness 5-9 years': [1.08],
    'Income composition of resources': [3],
    'Schooling': [11]
}
input_df = pd.DataFrame(features)
input_df['Status'].replace(to_replace=['Developing', 'Developed'], value=[0, 1
predicted_life_expectancy = rf.predict(input_df)
predicted_life_expectancy1 = dt.predict(input_df)
predicted_life_expectancy2 = lm.predict(input_df)
predicted_life_expectancy3 = ridge.predict(input_df)
print("Predicted life expectancy using Random Forest:", int(predicted_life_exp
print("Predicted life expectancy using Decision Tree:", int(predicted_life_exp
print("Predicted life expectancy using Linear Regression:", int(predicted_life
print("Predicted life expectancy using Ridge Regression:", int(predicted_life_
```

```
Predicted life expectancy using Random Forest: 66
Predicted life expectancy using Decision Tree: 70
Predicted life expectancy using Linear Regression: 91
Predicted life expectancy using Ridge Regression: 91

C:\Users\kabil\anaconda3\lib\site-packages\sklearn\base.py:443: UserWarning:
X has feature names, but RandomForestRegressor was fitted without feature nam
es
  warnings.warn(
C:\Users\kabil\anaconda3\lib\site-packages\sklearn\base.py:443: UserWarning:
X has feature names, but DecisionTreeRegressor was fitted without feature nam
es
  warnings.warn(
C:\Users\kabil\anaconda3\lib\site-packages\sklearn\base.py:443: UserWarning:
X has feature names, but LinearRegression was fitted without feature names
  warnings.warn(
C:\Users\kabil\anaconda3\lib\site-packages\sklearn\base.py:443: UserWarning:
X has feature names, but Ridge was fitted without feature names
  warnings.warn(
```