

ML-22112028

May 3, 2024

0.1 Problem Statement:

Problem Statement: Predictive Modeling for Human Development Index (HDI) of Indian States and Union Territories.

Description: Develop a machine learning model to predict the Human Development Index (HDI) of Indian states and union territories based on various socio-economic indicators such as health index, educational index, income index, life expectancy, expected years of schooling, mean years of schooling, and gross national income per capita.

Dataset: The dataset contains socio-economic indicators for each Indian state and union territory, including: - Subnational HDI - Health index - Educational index - Income index - Life expectancy - Expected years of schooling - Mean years of schooling - Log Gross National Income per capita

Objective: Build a regression model to accurately predict the HDI of Indian states and union territories based on the provided socio-economic indicators. Evaluate the model's performance using appropriate metrics such as mean absolute error, mean squared error, and R-squared score.

Tasks: 1. Data Exploration: Analyze the dataset to understand the distribution and relationships between variables. 2. Data Preprocessing: Handle missing values, scale numerical features, and encode categorical variables if present. 3. Model Selection: Experiment with various regression algorithms such as Linear regression, KNN Regressor, Decision Tree regression, Dandom Forest regression, and Support Vector regression. 4. Model Evaluation: Evaluate the performance of each model using cross-validation, GridSearchCV and appropriate evaluation metrics. 5. Hyperparameter Tuning: Fine-tune the hyperparameters of the best-performing model using techniques like grid search or random search. 6. Feature Selection: Identify the most relevant features for predicting HDI using statistical tests or feature importance techniques.(Random Forest has been used.)

Expected Outcome: A robust predictive model that accurately predicts the Human Development Index of Indian states and union territories, providing valuable insights for policymakers and stakeholders to prioritize socio-economic development initiatives.

~Siddharth R Bhardwaj Register No: 22112028

```
[1]: ## Importing the required Libraries...
import pandas as pd
import numpy as np
from bs4 import BeautifulSoup
import requests
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler
import matplotlib.pyplot as plt
```

```
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn import set_config
set_config(display='diagram')
from sklearn.model_selection import GridSearchCV
```

```
[2]: # Define the URL of the Wikipedia page to scrape
url = 'https://en.wikipedia.org/wiki/
↳List_of_Indian_states_and_union_territories_by_Human_Development_Index'
```

```
[3]: # Send an HTTP request to the URL and get the HTML content of the page
response = requests.get(url)
html_content = response.text

# Create a BeautifulSoup object to parse the HTML content
soup = BeautifulSoup(response.text, 'html')
soup.find('table')
```

```
[3]: <table class="sidebar nomobile nowraplinks"><tbody><tr><th class="sidebar-
title"><a href="/wiki/States_and_union_territories_of_India" title="States and
union territories of India">States and union <br/> territories of India</a>
<br/> ordered by</th></tr><tr><td class="sidebar-image"><figure class="mw-
halighn-center" typeof="mw:File"><a class="mw-file-description"
href="/wiki/File:Flag_of_India.svg"></a><figcaption></figcaption></figure></td></tr><tr><td
class="sidebar-content hlist">
<ul><li><a href="/wiki/List_of_states_and_union_territories_of_India_by_area"
title="List of states and union territories of India by area">Area</a></li>
<li><a href="/wiki/List_of_states_and_union_territories_of_India_by_population"
title="List of states and union territories of India by
population">Population</a></li>
<li><a href="/wiki/List_of_Indian_states_and_union_territories_by_GDP"
title="List of Indian states and union territories by GDP">GDP</a> (<a
href="/wiki/List_of_Indian_states_and_union_territories_by_GDP_per_capita"
title="List of Indian states and union territories by GDP per capita">per
capita</a></li></ul>
<hr/>
<ul><li><a href="/wiki/ISO_3166-2:IN" title="ISO
```

3166-2:IN">Abbreviations

- Access to safe drinking water
- Availability of toilets
- Capitals
- Child nutrition
- Crime rate
- Ease of doing business
- Electricity penetration
- Exports
- Fertility rate
- Forest cover
- Highest point
- HDI
- Home ownership
- Household size
- Human trafficking
- Infant mortality rate
- Institutional delivery
- Life expectancy at

birth

Literacy rate

Media exposure

Number of vehicles

Number of voters

Open defecation

Origin of name

Past population

Places of worship

Poverty rate

Power capacity

Safety of women

School enrollment rate

Sex ratio

Suicide rate

Tax revenues

TV ownership

<a

```

href="/wiki/List_of_states_and_union_territories_of_India_by_transport_network"
title="List of states and union territories of India by transport
network">Transport network</a></li>
<li><a href="/wiki/Indian_states_ranking_by_underweight_people" title="Indian
states ranking by underweight people">Underweight people</a></li>
<li><a
href="/wiki/List_of_states_and_union_territories_of_India_by_unemployment_rate"
title="List of states and union territories of India by unemployment
rate">Unemployment rate</a></li>
<li><a href="/wiki/List_of_states_and_union_territories_of_India_by_vaccination_
coverage" title="List of states and union territories of India by vaccination
coverage">Vaccination coverage</a></li>
<li><a href="/wiki/List_of_Indian_states_by_wildlife_population" title="List of
Indian states by wildlife population">Wildlife population</a></li></ul></td>
</tr><tr><td class="sidebar-navbar"><link href="mw-
data:TemplateStyles:r1129693374" rel="mw-deduplicated-inline-style"/><style
data-mw-deduplicate="TemplateStyles:r1063604349">.mw-parser-output
.navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output
.navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-
boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-
space:nowrap;line-height:inherit}.mw-parser-output .navbar-
brackets::before{margin-right:-0.125em;content:"[ "}.mw-parser-output .navbar-
brackets::after{margin-left:-0.125em;content:" ]"}.mw-parser-output .navbar
li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output
.navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-
variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-
parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output
.navbar-ct-mini{font-size:114%;margin:0 4em}</style><div class="navbar
plainlinks hlist navbar-mini"><ul><li class="nv-view"><a
href="/wiki/Template:Lists_of_states_and_territories_of_India"
title="Template:Lists of states and territories of India"><abbr title="View this
template">v</abbr></a></li><li class="nv-talk"><a
href="/wiki/Template_talk:Lists_of_states_and_territories_of_India"
title="Template talk:Lists of states and territories of India"><abbr
title="Discuss this template">t</abbr></a></li><li class="nv-edit"><a
href="/wiki/Special:EditPage/Template:Lists_of_states_and_territories_of_India"
title="Special:EditPage/Template:Lists of states and territories of India"><abbr
title="Edit this
template">e</abbr></a></li></ul></div></td></tr></tbody></table>

```

```

[4]: # Find the table within the HTML content using BeautifulSoup
soup.find_all('table')[6]

```

```

[4]: <table class="wikitable sortable">
<caption>Custom set of indicators (2021)<sup class="reference"
id="cite_ref-21"><a href="#cite_note-21">[21]</a></sup>
</caption>

```

```

<tbody><tr>
<th>
</th>
<th>Subnational HDI
</th>
<th>Health index
</th>
<th>Educational index
</th>
<th>Income index
</th>
<th>Life expectancy
</th>
<th>Life expectancy females
</th>
<th>Life expectancy males
</th>
<th>Expected years schooling
</th>
<th>Expected years schooling girls
</th>
<th>Expected years schooling boys
</th>
<th>Mean years schooling
</th>
<th>Mean years schooling females
</th>
<th>Mean years schooling males
</th>
<th>Log Gross National Income per capita
</th>
<th>Log Gross National Income per capita females
</th>
<th>Log Gross National Income per capita males
</th></tr>
<tr>
<td>Total
</td>
<td>0.633
</td>
<td>0.727
</td>
<td>0.552
</td>
<td>0.633
</td>
<td>67.24

```

```

</td>
<td>68.89
</td>
<td>65.76
</td>
<td>11.88
</td>
<td>11.94
</td>
<td>11.81
</td>
<td>6.655
</td>
<td>6.252
</td>
<td>7.229
</td>
<td>8.793
</td>
<td>7.731
</td>
<td>9.272
</td></tr>
<tr>
<td><a href="/wiki/Andaman_and_Nicobar_Islands" title="Andaman and Nicobar
Islands">Andaman and Nicobar Islands</a>
</td>
<td>0.706
</td>
<td>0.820
</td>
<td>0.607
</td>
<td>0.706
</td>
<td>73.31
</td>
<td>76.22
</td>
<td>71.50
</td>
<td>12.38
</td>
<td>12.78
</td>
<td>11.98
</td>

```

<td>7.900
</td>
<td>8.424
</td>
<td>7.712
</td>
<td>9.278
</td>
<td>8.157
</td>
<td>9.783
</td></tr>
<tr>
<td>Andhra Pradesh
</td>
<td>0.630
</td>
<td>0.734
</td>
<td>0.517
</td>
<td>0.660
</td>
<td>67.69
</td>
<td>69.44
</td>
<td>66.19
</td>
<td>11.95
</td>
<td>12.04
</td>
<td>11.87
</td>
<td>5.546
</td>
<td>5.135
</td>
<td>6.120
</td>
<td>8.973
</td>
<td>7.888
</td>
<td>9.461
</td></tr>


```

<tr>
<td><a href="/wiki/Arunachal_Pradesh" title="Arunachal Pradesh">Arunachal
Pradesh</a>
</td>
<td>0.665
</td>
<td>0.767
</td>
<td>0.575
</td>
<td>0.665
</td>
<td>69.87
</td>
<td>72.08
</td>
<td>68.27
</td>
<td>13.20
</td>
<td>13.32
</td>
<td>13.11
</td>
<td>6.257
</td>
<td>6.118
</td>
<td>6.556
</td>
<td>9.010
</td>
<td>7.921
</td>
<td>9.500
</td></tr>
<tr>
<td><a href="/wiki/Assam" title="Assam">Assam</a>
</td>
<td>0.597
</td>
<td>0.714
</td>
<td>0.530
</td>
<td>0.564
</td>

```

```

<td>66.38
</td>
<td>67.83
</td>
<td>64.92
</td>
<td>11.35
</td>
<td>11.70
</td>
<td>11.04
</td>
<td>6.446
</td>
<td>6.616
</td>
<td>6.499
</td>
<td>8.336
</td>
<td>7.329
</td>
<td>8.790
</td></tr>
<tr>
<td><a href="/wiki/Bihar" title="Bihar">Bihar</a>
</td>
<td>0.571
</td>
<td>0.712
</td>
<td>0.480
</td>
<td>0.544
</td>
<td>66.30
</td>
<td>67.73
</td>
<td>64.84
</td>
<td>11.48
</td>
<td>11.36
</td>
<td>11.73
</td>

```

```

<td>4.845
</td>
<td>3.775
</td>
<td>6.026
</td>
<td>8.204
</td>
<td>7.212
</td>
<td>8.650
</td></tr>
<tr>
<td><a href="/wiki/Chandigarh" title="Chandigarh">Chandigarh</a>
</td>
<td>0.744
</td>
<td>0.780
</td>
<td>0.704
</td>
<td>0.751
</td>
<td>70.71
</td>
<td>73.11
</td>
<td>69.07
</td>
<td>12.95
</td>
<td>13.19
</td>
<td>12.77
</td>
<td>10.32
</td>
<td>11.16
</td>
<td>9.957
</td>
<td>9.575
</td>
<td>8.418
</td>
<td>10.10
</td></tr>

```

```

<tr>
<td><a href="/wiki/Chhattisgarh" title="Chhattisgarh">Chhattisgarh</a>
</td>
<td>0.605
</td>
<td>0.689
</td>
<td>0.528
</td>
<td>0.609
</td>
<td>64.76
</td>
<td>65.83
</td>
<td>63.33
</td>
<td>11.85
</td>
<td>12.30
</td>
<td>11.42
</td>
<td>5.975
</td>
<td>5.293
</td>
<td>6.727
</td>
<td>8.635
</td>
<td>7.592
</td>
<td>9.105
</td></tr>
<tr>
<td><a href="/wiki/Dadra_and_Nagar_Haveli" title="Dadra and Nagar Haveli">Dadra
and Nagar Haveli</a>
</td>
<td>0.620
</td>
<td>0.766
</td>
<td>0.507
</td>
<td>0.613
</td>

```

<td>69.76
</td>
<td>71.95
</td>
<td>68.17
</td>
<td>10.39
</td>
<td>10.94
</td>
<td>9.974
</td>
<td>6.550
</td>
<td>5.620
</td>
<td>7.229
</td>
<td>8.665
</td>
<td>7.618
</td>
<td>9.137
</td></tr>
<tr>
<td>Daman and Diu
</td>
<td>0.661
</td>
<td>0.772
</td>
<td>0.554
</td>
<td>0.675
</td>
<td>70.21
</td>
<td>72.49
</td>
<td>68.59
</td>
<td>10.60
</td>
<td>11.05
</td>
<td>10.21
</td>

```

<td>7.785
</td>
<td>7.538
</td>
<td>8.100
</td>
<td>9.070
</td>
<td>7.974
</td>
<td>9.564
</td></tr>
<tr>
<td><a href="/wiki/Goa" title="Goa">Goa</a>
</td>
<td>0.751
</td>
<td>0.809
</td>
<td>0.696
</td>
<td>0.752
</td>
<td>72.59
</td>
<td>75.36
</td>
<td>70.83
</td>
<td>13.48
</td>
<td>13.67
</td>
<td>13.29
</td>
<td>9.648
</td>
<td>10.43
</td>
<td>9.398
</td>
<td>9.585
</td>
<td>8.427
</td>
<td>10.11
</td></tr>

```

```

<tr>
<td><a href="/wiki/Gujarat" title="Gujarat">Gujarat</a>
</td>
<td>0.638
</td>
<td>0.745
</td>
<td>0.519
</td>
<td>0.669
</td>
<td>68.44
</td>
<td>70.35
</td>
<td>66.91
</td>
<td>10.62
</td>
<td>10.37
</td>
<td>10.78
</td>
<td>6.728
</td>
<td>6.239
</td>
<td>7.324
</td>
<td>9.037
</td>
<td>7.945
</td>
<td>9.529
</td></tr>
<tr>
<td><a href="/wiki/Haryana" title="Haryana">Haryana</a>
</td>
<td>0.691
</td>
<td>0.756
</td>
<td>0.613
</td>
<td>0.713
</td>
<td>69.12

```

```

</td>
<td>71.18
</td>
<td>67.57
</td>
<td>12.90
</td>
<td>13.12
</td>
<td>12.65
</td>
<td>7.639
</td>
<td>7.126
</td>
<td>8.261
</td>
<td>9.324
</td>
<td>8.198
</td>
<td>9.832
</td></tr>
<tr>
<td><a href="/wiki/Himachal_Pradesh" title="Himachal Pradesh">Himachal
Pradesh</a>
</td>
<td>0.703
</td>
<td>0.757
</td>
<td>0.649
</td>
<td>0.709
</td>
<td>69.19
</td>
<td>71.26
</td>
<td>67.63
</td>
<td>13.28
</td>
<td>13.84
</td>
<td>12.71
</td>

```



```

<td>8.403
</td>
<td>8.473
</td>
<td>8.716
</td>
<td>9.299
</td>
<td>8.175
</td>
<td>9.804
</td></tr>
<tr>
<td><a href="/wiki/Jammu_and_Kashmir_(union_territory)" title="Jammu and Kashmir
(union territory)">Jammu and Kashmir</a>
</td>
<td>0.699
</td>
<td>0.762
</td>
<td>0.644
</td>
<td>0.696
</td>
<td>69.56
</td>
<td>71.71
</td>
<td>67.98
</td>
<td>14.24
</td>
<td>14.43
</td>
<td>14.04
</td>
<td>7.450
</td>
<td>6.907
</td>
<td>8.034
</td>
<td>9.214
</td>
<td>8.100
</td>
<td>9.715

```

```

</td></tr>
<tr>
<td><a href="/wiki/Jharkhand" title="Jharkhand">Jharkhand</a>
</td>
<td>0.589
</td>
<td>0.715
</td>
<td>0.512
</td>
<td>0.557
</td>
<td>66.49
</td>
<td>67.97
</td>
<td>65.03
</td>
<td>11.77
</td>
<td>11.91
</td>
<td>11.69
</td>
<td>5.556
</td>
<td>4.613
</td>
<td>6.585
</td>
<td>8.295
</td>
<td>7.293
</td>
<td>8.747
</td></tr>
<tr>
<td><a href="/wiki/Karnataka" title="Karnataka">Karnataka</a>
</td>
<td>0.667
</td>
<td>0.777
</td>
<td>0.567
</td>
<td>0.673
</td>

```

```

<td>70.50
</td>
<td>72.84
</td>
<td>68.87
</td>
<td>12.16
</td>
<td>12.34
</td>
<td>11.97
</td>
<td>6.896
</td>
<td>6.676
</td>
<td>7.333
</td>
<td>9.057
</td>
<td>7.963
</td>
<td>9.550
</td></tr>
<tr>
<td><a href="/wiki/Kerala" title="Kerala">Kerala</a>
</td>
<td>0.752
</td>
<td>0.834
</td>
<td>0.713
</td>
<td>0.716
</td>
<td>74.23
</td>
<td>77.32
</td>
<td>72.35
</td>
<td>14.20
</td>
<td>14.77
</td>
<td>13.65
</td>

```

<td>9.543
</td>
<td>10.88
</td>
<td>8.820
</td>
<td>9.344
</td>
<td>8.215
</td>
<td>9.852
</td></tr>
<tr>
<td>Lakshadweep
</td>
<td>0.715
</td>
<td>0.785
</td>
<td>0.649
</td>
<td>0.718
</td>
<td>71.01
</td>
<td>73.46
</td>
<td>69.35
</td>
<td>12.83
</td>
<td>13.35
</td>
<td>12.39
</td>
<td>8.779
</td>
<td>9.408
</td>
<td>8.726
</td>
<td>9.356
</td>
<td>8.225
</td>
<td>9.865
</td></tr>

```

<tr>
<td><a href="/wiki/Madhya_Pradesh" title="Madhya Pradesh">Madhya Pradesh</a>
</td>
<td>0.596
</td>
<td>0.693
</td>
<td>0.509
</td>
<td>0.600
</td>
<td>65.05
</td>
<td>66.19
</td>
<td>63.62
</td>
<td>11.31
</td>
<td>11.24
</td>
<td>11.35
</td>
<td>5.839
</td>
<td>5.062
</td>
<td>6.603
</td>
<td>8.577
</td>
<td>7.541
</td>
<td>9.044
</td></tr>
<tr>
<td><a href="/wiki/Maharashtra" title="Maharashtra">Maharashtra</a>
</td>
<td>0.688
</td>
<td>0.779
</td>
<td>0.620
</td>
<td>0.676
</td>
<td>70.60

```

```

</td>
<td>72.97
</td>
<td>68.97
</td>
<td>12.75
</td>
<td>12.71
</td>
<td>12.73
</td>
<td>7.963
</td>
<td>7.793
</td>
<td>8.356
</td>
<td>9.077
</td>
<td>7.980
</td>
<td>9.571
</td></tr>
<tr>
<td><a href="/wiki/Manipur" title="Manipur">Manipur</a>
</td>
<td>0.678
</td>
<td>0.783
</td>
<td>0.656
</td>
<td>0.606
</td>
<td>70.87
</td>
<td>73.29
</td>
<td>69.21
</td>
<td>13.15
</td>
<td>13.19
</td>
<td>13.13
</td>
<td>8.727

```

```

</td>
<td>8.898
</td>
<td>8.953
</td>
<td>8.619
</td>
<td>7.578
</td>
<td>9.088
</td></tr>
<tr>
<td><a href="/wiki/Meghalaya" title="Meghalaya">Meghalaya</a>
</td>
<td>0.643
</td>
<td>0.753
</td>
<td>0.572
</td>
<td>0.616
</td>
<td>68.93
</td>
<td>70.94
</td>
<td>67.38
</td>
<td>12.67
</td>
<td>13.38
</td>
<td>11.98
</td>
<td>6.613
</td>
<td>7.429
</td>
<td>6.196
</td>
<td>8.684
</td>
<td>7.635
</td>
<td>9.157
</td></tr>
<tr>

```

Mizoram	0.688
0.725	0.636
0.705	67.15
68.77	65.66
12.64	12.71
12.51	8.549
9.347	8.205
9.271	8.150
9.775	9.775
Nagaland	0.670
0.767	0.614
0.639	69.84


```

<td>72.05
</td>
<td>68.25
</td>
<td>12.70
</td>
<td>13.14
</td>
<td>12.27
</td>
<td>7.838
</td>
<td>8.352
</td>
<td>7.665
</td>
<td>8.835
</td>
<td>7.767
</td>
<td>9.315
</td></tr>
<tr>
<td><a href="/wiki/New_Delhi" title="New Delhi">New Delhi</a>
</td>
<td>0.730
</td>
<td>0.775
</td>
<td>0.684
</td>
<td>0.733
</td>
<td>70.35
</td>
<td>72.66
</td>
<td>68.73
</td>
<td>13.30
</td>
<td>13.84
</td>
<td>12.80
</td>
<td>9.428
</td>

```

```

<td>9.711
</td>
<td>9.436
</td>
<td>9.460
</td>
<td>8.317
</td>
<td>9.975
</td></tr>
<tr>
<td><a href="/wiki/Odisha" title="Odisha">Orissa</a>
</td>
<td>0.597
</td>
<td>0.717
</td>
<td>0.505
</td>
<td>0.587
</td>
<td>66.62
</td>
<td>68.13
</td>
<td>65.16
</td>
<td>11.07
</td>
<td>11.09
</td>
<td>11.09
</td>
<td>5.912
</td>
<td>5.535
</td>
<td>6.461
</td>
<td>8.492
</td>
<td>7.466
</td>
<td>8.954
</td></tr>
<tr>
<td><a href="/wiki/Puducherry_(union_territory)" title="Puducherry (union

```

```

territory)">Puducherry</a>
</td>
<td>0.726
</td>
<td>0.794
</td>
<td>0.664
</td>
<td>0.724
</td>
<td>71.60
</td>
<td>74.17
</td>
<td>69.90
</td>
<td>12.99
</td>
<td>13.69
</td>
<td>12.33
</td>
<td>9.104
</td>
<td>9.420
</td>
<td>9.313
</td>
<td>9.399
</td>
<td>8.263
</td>
<td>9.910
</td></tr>
<tr>
<td><a href="/wiki/Punjab,_India" title="Punjab, India">Punjab</a>
</td>
<td>0.694
</td>
<td>0.765
</td>
<td>0.598
</td>
<td>0.729
</td>
<td>69.73
</td>

```

```

<td>71.92
</td>
<td>68.14
</td>
<td>12.34
</td>
<td>12.94
</td>
<td>11.79
</td>
<td>7.670
</td>
<td>8.157
</td>
<td>7.522
</td>
<td>9.432
</td>
<td>8.292
</td>
<td>9.945
</td></tr>
<tr>
<td><a href="/wiki/Rajasthan" title="Rajasthan">Rajasthan</a>
</td>
<td>0.638
</td>
<td>0.725
</td>
<td>0.543
</td>
<td>0.660
</td>
<td>67.13
</td>
<td>68.75
</td>
<td>65.65
</td>
<td>12.62
</td>
<td>12.49
</td>
<td>12.78
</td>
<td>5.766
</td>

```

<td>4.467
</td>
<td>7.011
</td>
<td>8.973
</td>
<td>7.889
</td>
<td>9.461
</td></tr>
<tr>
<td>Sikkim
</td>
<td>0.702
</td>
<td>0.785
</td>
<td>0.644
</td>
<td>0.683
</td>
<td>71.04
</td>
<td>73.50
</td>
<td>69.38
</td>
<td>13.89
</td>
<td>14.37
</td>
<td>13.39
</td>
<td>7.754
</td>
<td>8.352
</td>
<td>7.504
</td>
<td>9.128
</td>
<td>8.025
</td>
<td>9.625
</td></tr>
<tr>
<td>Tamil Nadu

```

</td>
<td>0.686
</td>
<td>0.791
</td>
<td>0.608
</td>
<td>0.671
</td>
<td>71.40
</td>
<td>73.93
</td>
<td>69.71
</td>
<td>12.68
</td>
<td>13.14
</td>
<td>12.23
</td>
<td>7.691
</td>
<td>7.915
</td>
<td>7.854
</td>
<td>9.047
</td>
<td>7.954
</td>
<td>9.539
</td></tr>
<tr>
<td><a href="/wiki/Telangana" title="Telangana">Telangana</a>
</td>
<td>0.647
</td>
<td>0.748
</td>
<td>0.542
</td>
<td>0.667
</td>
<td>68.59
</td>
<td>70.53

```

```

</td>
<td>67.06
</td>
<td>12.47
</td>
<td>12.65
</td>
<td>12.30
</td>
<td>5.870
</td>
<td>5.208
</td>
<td>6.698
</td>
<td>9.022
</td>
<td>7.932
</td>
<td>9.513
</td></tr>
<tr>
<td><a href="/wiki/Tripura" title="Tripura">Tripura</a>
</td>
<td>0.629
</td>
<td>0.773
</td>
<td>0.549
</td>
<td>0.586
</td>
<td>70.25
</td>
<td>72.54
</td>
<td>68.63
</td>
<td>11.84
</td>
<td>11.69
</td>
<td>12.08
</td>
<td>6.603
</td>
<td>6.701

```

```

</td>
<td>6.746
</td>
<td>8.486
</td>
<td>7.461
</td>
<td>8.948
</td></tr>
<tr>
<td><a href="/wiki/Uttar_Pradesh" title="Uttar Pradesh">Uttar Pradesh</a>
</td>
<td>0.592
</td>
<td>0.667
</td>
<td>0.524
</td>
<td>0.591
</td>
<td>63.39
</td>
<td>64.12
</td>
<td>61.96
</td>
<td>11.22
</td>
<td>11.11
</td>
<td>11.34
</td>
<td>6.383
</td>
<td>5.535
</td>
<td>7.314
</td>
<td>8.519
</td>
<td>7.490
</td>
<td>8.983
</td></tr>
<tr>
<td><a href="/wiki/Uttarakhand" title="Uttarakhand">Uttarakhand</a>
</td>

```


<td>0.672
</td>
<td>0.733
</td>
<td>0.609
</td>
<td>0.678
</td>
<td>67.66
</td>
<td>69.40
</td>
<td>66.16
</td>
<td>12.50
</td>
<td>12.85
</td>
<td>12.16
</td>
<td>7.869
</td>
<td>7.478
</td>
<td>8.498
</td>
<td>9.093
</td>
<td>7.994
</td>
<td>9.588
</td></tr>
<tr>
<td>West Bengal
</td>
<td>0.624
</td>
<td>0.761
</td>
<td>0.534
</td>
<td>0.598
</td>
<td>69.48
</td>
<td>71.61
</td>

```

<td>67.90
</td>
<td>11.77
</td>
<td>12.27
</td>
<td>11.30
</td>
<td>6.226
</td>
<td>6.179
</td>
<td>6.480
</td>
<td>8.567
</td>
<td>7.532
</td>
<td>9.033
</td></tr></tbody></table>

```

```

[5]: ## Looking at the above result and mentioning the "class"
soup.find('table', class_='wikitable sortable')

# Find the 7th table on the page (index 6)
table = soup.find_all('table')[6]

```

```

[6]: # Finding all the table header elements (th) within the table
states_titles = table.find_all('th')
states_titles

```

```

[6]: [<th>
</th>,
<th>Subnational HDI
</th>,
<th>Health index
</th>,
<th>Educational index
</th>,
<th>Income index
</th>,
<th>Life expectancy
</th>,
<th>Life expectancy females
</th>,
<th>Life expectancy males
</th>,

```

```

<th>Expected years schooling
</th>,
<th>Expected years schooling girls
</th>,
<th>Expected years schooling boys
</th>,
<th>Mean years schooling
</th>,
<th>Mean years schooling females
</th>,
<th>Mean years schooling males
</th>,
<th>Log Gross National Income per capita
</th>,
<th>Log Gross National Income per capita females
</th>,
<th>Log Gross National Income per capita males
</th>]

```

```

[7]: # Extracting the text from each table header element and strip any leading or
      ↪trailing whitespace
states_table_titles = [title.text.strip() for title in states_titles]
print(states_table_titles)

```

```

['', 'Subnational HDI', 'Health index', 'Educational index', 'Income index',
'Life expectancy', 'Life expectancy females', 'Life expectancy males', 'Expected
years schooling', 'Expected years schooling girls', 'Expected years schooling
boys', 'Mean years schooling', 'Mean years schooling females', 'Mean years
schooling males', 'Log Gross National Income per capita', 'Log Gross National
Income per capita females', 'Log Gross National Income per capita males']

```

```

[8]: # Creating an empty DataFrame with column names extracted from the table titles
import pandas as pd
df = pd.DataFrame(columns = states_table_titles)
df

```

[8]: Empty DataFrame

```

Columns: [], Subnational HDI, Health index, Educational index, Income index, Life
expectancy, Life expectancy females, Life expectancy males, Expected years
schooling, Expected years schooling girls, Expected years schooling boys, Mean
years schooling, Mean years schooling females, Mean years schooling males, Log
Gross National Income per capita, Log Gross National Income per capita females,
Log Gross National Income per capita males]
Index: []

```

```

[9]: soup = BeautifulSoup(html_content, 'html.parser')
tables = soup.find_all('table')

```

```
[10]: # Assigning the 7th table to the variable table_16 and reading the HTML table
      ↪ into a DataFrame using pandas read_html function,
      # and then selecting the first table (index 0) from the list of tables returned.
      if len(tables) >= 7:
          table_16 = tables[6]
          df = pd.read_html(str(table_16))[0] # Adjusted index to 0 to get the first
          ↪ table
          print(df)
      else:
          print("Error: There are fewer than 7 tables on the page.")
          # If there are fewer than 7 tables, print an error message
```

	Unnamed: 0	Subnational HDI	Health index \
0	Total	0.633	0.727
1	Andaman and Nicobar Islands	0.706	0.820
2	Andhra Pradesh	0.630	0.734
3	Arunachal Pradesh	0.665	0.767
4	Assam	0.597	0.714
5	Bihar	0.571	0.712
6	Chandigarh	0.744	0.780
7	Chhattisgarh	0.605	0.689
8	Dadra and Nagar Haveli	0.620	0.766
9	Daman and Diu	0.661	0.772
10	Goa	0.751	0.809
11	Gujarat	0.638	0.745
12	Haryana	0.691	0.756
13	Himachal Pradesh	0.703	0.757
14	Jammu and Kashmir	0.699	0.762
15	Jharkhand	0.589	0.715
16	Karnataka	0.667	0.777
17	Kerala	0.752	0.834
18	Lakshadweep	0.715	0.785
19	Madhya Pradesh	0.596	0.693
20	Maharashtra	0.688	0.779
21	Manipur	0.678	0.783
22	Meghalaya	0.643	0.753
23	Mizoram	0.688	0.725
24	Nagaland	0.670	0.767
25	New Delhi	0.730	0.775
26	Orissa	0.597	0.717
27	Puducherry	0.726	0.794
28	Punjab	0.694	0.765
29	Rajasthan	0.638	0.725
30	Sikkim	0.702	0.785
31	Tamil Nadu	0.686	0.791
32	Telangana	0.647	0.748
33	Tripura	0.629	0.773

34	Uttar Pradesh	0.592	0.667
35	Uttarakhand	0.672	0.733
36	West Bengal	0.624	0.761

	Educational index	Income index	Life expectancy	Life expectancy females \
0	0.552	0.633	67.24	68.89
1	0.607	0.706	73.31	76.22
2	0.517	0.660	67.69	69.44
3	0.575	0.665	69.87	72.08
4	0.530	0.564	66.38	67.83
5	0.480	0.544	66.30	67.73
6	0.704	0.751	70.71	73.11
7	0.528	0.609	64.76	65.83
8	0.507	0.613	69.76	71.95
9	0.554	0.675	70.21	72.49
10	0.696	0.752	72.59	75.36
11	0.519	0.669	68.44	70.35
12	0.613	0.713	69.12	71.18
13	0.649	0.709	69.19	71.26
14	0.644	0.696	69.56	71.71
15	0.512	0.557	66.49	67.97
16	0.567	0.673	70.50	72.84
17	0.713	0.716	74.23	77.32
18	0.649	0.718	71.01	73.46
19	0.509	0.600	65.05	66.19
20	0.620	0.676	70.60	72.97
21	0.656	0.606	70.87	73.29
22	0.572	0.616	68.93	70.94
23	0.636	0.705	67.15	68.77
24	0.614	0.639	69.84	72.05
25	0.684	0.733	70.35	72.66
26	0.505	0.587	66.62	68.13
27	0.664	0.724	71.60	74.17
28	0.598	0.729	69.73	71.92
29	0.543	0.660	67.13	68.75
30	0.644	0.683	71.04	73.50
31	0.608	0.671	71.40	73.93
32	0.542	0.667	68.59	70.53
33	0.549	0.586	70.25	72.54
34	0.524	0.591	63.39	64.12
35	0.609	0.678	67.66	69.40
36	0.534	0.598	69.48	71.61

	Life expectancy males	Expected years schooling \
0	65.76	11.88
1	71.50	12.38
2	66.19	11.95
3	68.27	13.20

4	64.92	11.35
5	64.84	11.48
6	69.07	12.95
7	63.33	11.85
8	68.17	10.39
9	68.59	10.60
10	70.83	13.48
11	66.91	10.62
12	67.57	12.90
13	67.63	13.28
14	67.98	14.24
15	65.03	11.77
16	68.87	12.16
17	72.35	14.20
18	69.35	12.83
19	63.62	11.31
20	68.97	12.75
21	69.21	13.15
22	67.38	12.67
23	65.66	12.64
24	68.25	12.70
25	68.73	13.30
26	65.16	11.07
27	69.90	12.99
28	68.14	12.34
29	65.65	12.62
30	69.38	13.89
31	69.71	12.68
32	67.06	12.47
33	68.63	11.84
34	61.96	11.22
35	66.16	12.50
36	67.90	11.77

	Expected years schooling girls	Expected years schooling boys \
0	11.94	11.810
1	12.78	11.980
2	12.04	11.870
3	13.32	13.110
4	11.70	11.040
5	11.36	11.730
6	13.19	12.770
7	12.30	11.420
8	10.94	9.974
9	11.05	10.210
10	13.67	13.290
11	10.37	10.780
12	13.12	12.650

13	13.84	12.710
14	14.43	14.040
15	11.91	11.690
16	12.34	11.970
17	14.77	13.650
18	13.35	12.390
19	11.24	11.350
20	12.71	12.730
21	13.19	13.130
22	13.38	11.980
23	12.71	12.510
24	13.14	12.270
25	13.84	12.800
26	11.09	11.090
27	13.69	12.330
28	12.94	11.790
29	12.49	12.780
30	14.37	13.390
31	13.14	12.230
32	12.65	12.300
33	11.69	12.080
34	11.11	11.340
35	12.85	12.160
36	12.27	11.300

	Mean years schooling	Mean years schooling females \
0	6.655	6.252
1	7.900	8.424
2	5.546	5.135
3	6.257	6.118
4	6.446	6.616
5	4.845	3.775
6	10.320	11.160
7	5.975	5.293
8	6.550	5.620
9	7.785	7.538
10	9.648	10.430
11	6.728	6.239
12	7.639	7.126
13	8.403	8.473
14	7.450	6.907
15	5.556	4.613
16	6.896	6.676
17	9.543	10.880
18	8.779	9.408
19	5.839	5.062
20	7.963	7.793
21	8.727	8.898

22	6.613	7.429
23	8.549	9.347
24	7.838	8.352
25	9.428	9.711
26	5.912	5.535
27	9.104	9.420
28	7.670	8.157
29	5.766	4.467
30	7.754	8.352
31	7.691	7.915
32	5.870	5.208
33	6.603	6.701
34	6.383	5.535
35	7.869	7.478
36	6.226	6.179

	Mean years schooling males	Log Gross National Income per capita \
0	7.229	8.793
1	7.712	9.278
2	6.120	8.973
3	6.556	9.010
4	6.499	8.336
5	6.026	8.204
6	9.957	9.575
7	6.727	8.635
8	7.229	8.665
9	8.100	9.070
10	9.398	9.585
11	7.324	9.037
12	8.261	9.324
13	8.716	9.299
14	8.034	9.214
15	6.585	8.295
16	7.333	9.057
17	8.820	9.344
18	8.726	9.356
19	6.603	8.577
20	8.356	9.077
21	8.953	8.619
22	6.196	8.684
23	8.205	9.271
24	7.665	8.835
25	9.436	9.460
26	6.461	8.492
27	9.313	9.399
28	7.522	9.432
29	7.011	8.973
30	7.504	9.128

31	7.854	9.047
32	6.698	9.022
33	6.746	8.486
34	7.314	8.519
35	8.498	9.093
36	6.480	8.567

	Log Gross National Income per capita females \
0	7.731
1	8.157
2	7.888
3	7.921
4	7.329
5	7.212
6	8.418
7	7.592
8	7.618
9	7.974
10	8.427
11	7.945
12	8.198
13	8.175
14	8.100
15	7.293
16	7.963
17	8.215
18	8.225
19	7.541
20	7.980
21	7.578
22	7.635
23	8.150
24	7.767
25	8.317
26	7.466
27	8.263
28	8.292
29	7.889
30	8.025
31	7.954
32	7.932
33	7.461
34	7.490
35	7.994
36	7.532

	Log Gross National Income per capita males
0	9.272

1	9.783
2	9.461
3	9.500
4	8.790
5	8.650
6	10.100
7	9.105
8	9.137
9	9.564
10	10.110
11	9.529
12	9.832
13	9.804
14	9.715
15	8.747
16	9.550
17	9.852
18	9.865
19	9.044
20	9.571
21	9.088
22	9.157
23	9.775
24	9.315
25	9.975
26	8.954
27	9.910
28	9.945
29	9.461
30	9.625
31	9.539
32	9.513
33	8.948
34	8.983
35	9.588
36	9.033

```
[11]: df
```

```
[11]:
```

	Unnamed: 0	Subnational HDI	Health index \
0	Total	0.633	0.727
1	Andaman and Nicobar Islands	0.706	0.820
2	Andhra Pradesh	0.630	0.734
3	Arunachal Pradesh	0.665	0.767
4	Assam	0.597	0.714
5	Bihar	0.571	0.712
6	Chandigarh	0.744	0.780

7	Chhattisgarh	0.605	0.689
8	Dadra and Nagar Haveli	0.620	0.766
9	Daman and Diu	0.661	0.772
10	Goa	0.751	0.809
11	Gujarat	0.638	0.745
12	Haryana	0.691	0.756
13	Himachal Pradesh	0.703	0.757
14	Jammu and Kashmir	0.699	0.762
15	Jharkhand	0.589	0.715
16	Karnataka	0.667	0.777
17	Kerala	0.752	0.834
18	Lakshadweep	0.715	0.785
19	Madhya Pradesh	0.596	0.693
20	Maharashtra	0.688	0.779
21	Manipur	0.678	0.783
22	Meghalaya	0.643	0.753
23	Mizoram	0.688	0.725
24	Nagaland	0.670	0.767
25	New Delhi	0.730	0.775
26	Orissa	0.597	0.717
27	Puducherry	0.726	0.794
28	Punjab	0.694	0.765
29	Rajasthan	0.638	0.725
30	Sikkim	0.702	0.785
31	Tamil Nadu	0.686	0.791
32	Telangana	0.647	0.748
33	Tripura	0.629	0.773
34	Uttar Pradesh	0.592	0.667
35	Uttarakhand	0.672	0.733
36	West Bengal	0.624	0.761

	Educational index	Income index	Life expectancy	Life expectancy females \
0	0.552	0.633	67.24	68.89
1	0.607	0.706	73.31	76.22
2	0.517	0.660	67.69	69.44
3	0.575	0.665	69.87	72.08
4	0.530	0.564	66.38	67.83
5	0.480	0.544	66.30	67.73
6	0.704	0.751	70.71	73.11
7	0.528	0.609	64.76	65.83
8	0.507	0.613	69.76	71.95
9	0.554	0.675	70.21	72.49
10	0.696	0.752	72.59	75.36
11	0.519	0.669	68.44	70.35
12	0.613	0.713	69.12	71.18
13	0.649	0.709	69.19	71.26
14	0.644	0.696	69.56	71.71

15	0.512	0.557	66.49	67.97
16	0.567	0.673	70.50	72.84
17	0.713	0.716	74.23	77.32
18	0.649	0.718	71.01	73.46
19	0.509	0.600	65.05	66.19
20	0.620	0.676	70.60	72.97
21	0.656	0.606	70.87	73.29
22	0.572	0.616	68.93	70.94
23	0.636	0.705	67.15	68.77
24	0.614	0.639	69.84	72.05
25	0.684	0.733	70.35	72.66
26	0.505	0.587	66.62	68.13
27	0.664	0.724	71.60	74.17
28	0.598	0.729	69.73	71.92
29	0.543	0.660	67.13	68.75
30	0.644	0.683	71.04	73.50
31	0.608	0.671	71.40	73.93
32	0.542	0.667	68.59	70.53
33	0.549	0.586	70.25	72.54
34	0.524	0.591	63.39	64.12
35	0.609	0.678	67.66	69.40
36	0.534	0.598	69.48	71.61

	Life expectancy males	Expected years schooling \
0	65.76	11.88
1	71.50	12.38
2	66.19	11.95
3	68.27	13.20
4	64.92	11.35
5	64.84	11.48
6	69.07	12.95
7	63.33	11.85
8	68.17	10.39
9	68.59	10.60
10	70.83	13.48
11	66.91	10.62
12	67.57	12.90
13	67.63	13.28
14	67.98	14.24
15	65.03	11.77
16	68.87	12.16
17	72.35	14.20
18	69.35	12.83
19	63.62	11.31
20	68.97	12.75
21	69.21	13.15
22	67.38	12.67

23	65.66	12.64
24	68.25	12.70
25	68.73	13.30
26	65.16	11.07
27	69.90	12.99
28	68.14	12.34
29	65.65	12.62
30	69.38	13.89
31	69.71	12.68
32	67.06	12.47
33	68.63	11.84
34	61.96	11.22
35	66.16	12.50
36	67.90	11.77

	Expected years schooling girls	Expected years schooling boys \
0	11.94	11.810
1	12.78	11.980
2	12.04	11.870
3	13.32	13.110
4	11.70	11.040
5	11.36	11.730
6	13.19	12.770
7	12.30	11.420
8	10.94	9.974
9	11.05	10.210
10	13.67	13.290
11	10.37	10.780
12	13.12	12.650
13	13.84	12.710
14	14.43	14.040
15	11.91	11.690
16	12.34	11.970
17	14.77	13.650
18	13.35	12.390
19	11.24	11.350
20	12.71	12.730
21	13.19	13.130
22	13.38	11.980
23	12.71	12.510
24	13.14	12.270
25	13.84	12.800
26	11.09	11.090
27	13.69	12.330
28	12.94	11.790
29	12.49	12.780
30	14.37	13.390

31	13.14	12.230
32	12.65	12.300
33	11.69	12.080
34	11.11	11.340
35	12.85	12.160
36	12.27	11.300

	Mean years schooling	Mean years schooling females \
0	6.655	6.252
1	7.900	8.424
2	5.546	5.135
3	6.257	6.118
4	6.446	6.616
5	4.845	3.775
6	10.320	11.160
7	5.975	5.293
8	6.550	5.620
9	7.785	7.538
10	9.648	10.430
11	6.728	6.239
12	7.639	7.126
13	8.403	8.473
14	7.450	6.907
15	5.556	4.613
16	6.896	6.676
17	9.543	10.880
18	8.779	9.408
19	5.839	5.062
20	7.963	7.793
21	8.727	8.898
22	6.613	7.429
23	8.549	9.347
24	7.838	8.352
25	9.428	9.711
26	5.912	5.535
27	9.104	9.420
28	7.670	8.157
29	5.766	4.467
30	7.754	8.352
31	7.691	7.915
32	5.870	5.208
33	6.603	6.701
34	6.383	5.535
35	7.869	7.478
36	6.226	6.179

Mean years schooling males Log Gross National Income per capita \

0	7.229	8.793
1	7.712	9.278
2	6.120	8.973
3	6.556	9.010
4	6.499	8.336
5	6.026	8.204
6	9.957	9.575
7	6.727	8.635
8	7.229	8.665
9	8.100	9.070
10	9.398	9.585
11	7.324	9.037
12	8.261	9.324
13	8.716	9.299
14	8.034	9.214
15	6.585	8.295
16	7.333	9.057
17	8.820	9.344
18	8.726	9.356
19	6.603	8.577
20	8.356	9.077
21	8.953	8.619
22	6.196	8.684
23	8.205	9.271
24	7.665	8.835
25	9.436	9.460
26	6.461	8.492
27	9.313	9.399
28	7.522	9.432
29	7.011	8.973
30	7.504	9.128
31	7.854	9.047
32	6.698	9.022
33	6.746	8.486
34	7.314	8.519
35	8.498	9.093
36	6.480	8.567

Log Gross National Income per capita females \

0	7.731
1	8.157
2	7.888
3	7.921
4	7.329
5	7.212
6	8.418
7	7.592

8	7.618
9	7.974
10	8.427
11	7.945
12	8.198
13	8.175
14	8.100
15	7.293
16	7.963
17	8.215
18	8.225
19	7.541
20	7.980
21	7.578
22	7.635
23	8.150
24	7.767
25	8.317
26	7.466
27	8.263
28	8.292
29	7.889
30	8.025
31	7.954
32	7.932
33	7.461
34	7.490
35	7.994
36	7.532

Log Gross National Income per capita males

0	9.272
1	9.783
2	9.461
3	9.500
4	8.790
5	8.650
6	10.100
7	9.105
8	9.137
9	9.564
10	10.110
11	9.529
12	9.832
13	9.804
14	9.715
15	8.747

16	9.550
17	9.852
18	9.865
19	9.044
20	9.571
21	9.088
22	9.157
23	9.775
24	9.315
25	9.975
26	8.954
27	9.910
28	9.945
29	9.461
30	9.625
31	9.539
32	9.513
33	8.948
34	8.983
35	9.588
36	9.033

```
[12]: ## Saving the dataset into a csv file so it stays saved for future use.
df.to_csv("wiki.csv")
```

0.2 — Loading the Dataset —

```
[13]: df = pd.read_csv("wiki.csv")
```

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

```
[14]: Unnamed: 0.1      Unnamed: 0  Subnational HDI  Health index \
0          0          Total          0.633      0.727
1          1  Andaman and Nicobar Islands      0.706      0.820
2          2          Andhra Pradesh          0.630      0.734
3          3          Arunachal Pradesh      0.665      0.767
4          4          Assam          0.597      0.714

      Educational index  Income index  Life expectancy  Life expectancy females \
0          0.552      0.633      67.24      68.89
1          0.607      0.706      73.31      76.22
2          0.517      0.660      67.69      69.44
3          0.575      0.665      69.87      72.08
4          0.530      0.564      66.38      67.83

      Life expectancy males  Expected years schooling \
0          65.76      11.88
```

1	71.50	12.38
2	66.19	11.95
3	68.27	13.20
4	64.92	11.35

	Expected years schooling girls	Expected years schooling boys \
0	11.94	11.81
1	12.78	11.98
2	12.04	11.87
3	13.32	13.11
4	11.70	11.04

	Mean years schooling	Mean years schooling females \
0	6.655	6.252
1	7.900	8.424
2	5.546	5.135
3	6.257	6.118
4	6.446	6.616

	Mean years schooling males	Log Gross National Income per capita \
0	7.229	8.793
1	7.712	9.278
2	6.120	8.973
3	6.556	9.010
4	6.499	8.336

	Log Gross National Income per capita females \
0	7.731
1	8.157
2	7.888
3	7.921
4	7.329

	Log Gross National Income per capita males
0	9.272
1	9.783
2	9.461
3	9.500
4	8.790

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

```
[15]: Unnamed: 0.1 Unnamed: 0 Subnational HDI Health index \
32 32 Telangana 0.647 0.748
33 33 Tripura 0.629 0.773
34 34 Uttar Pradesh 0.592 0.667
35 35 Uttarakhand 0.672 0.733
```

36	36	West Bengal	0.624	0.761	
	Educational index	Income index	Life expectancy	Life expectancy females	\
32	0.542	0.667	68.59	70.53	
33	0.549	0.586	70.25	72.54	
34	0.524	0.591	63.39	64.12	
35	0.609	0.678	67.66	69.40	
36	0.534	0.598	69.48	71.61	
	Life expectancy males	Expected years schooling			\
32	67.06	12.47			
33	68.63	11.84			
34	61.96	11.22			
35	66.16	12.50			
36	67.90	11.77			
	Expected years schooling girls	Expected years schooling boys			\
32	12.65	12.30			
33	11.69	12.08			
34	11.11	11.34			
35	12.85	12.16			
36	12.27	11.30			
	Mean years schooling	Mean years schooling females			\
32	5.870	5.208			
33	6.603	6.701			
34	6.383	5.535			
35	7.869	7.478			
36	6.226	6.179			
	Mean years schooling males	Log Gross National Income per capita			\
32	6.698	9.022			
33	6.746	8.486			
34	7.314	8.519			
35	8.498	9.093			
36	6.480	8.567			
	Log Gross National Income per capita females				\
32	7.932				
33	7.461				
34	7.490				
35	7.994				
36	7.532				
	Log Gross National Income per capita males				
32	9.513				
33	8.948				

34	8.983
35	9.588
36	9.033

```
[16]: ## Looking for the missing values
df.isna().sum()
```

```
[16]: Unnamed: 0.1      0
      Unnamed: 0      0
      Subnational HDI      0
      Health index      0
      Educational index      0
      Income index      0
      Life expectancy      0
      Life expectancy females      0
      Life expectancy males      0
      Expected years schooling      0
      Expected years schooling girls      0
      Expected years schooling boys      0
      Mean years schooling      0
      Mean years schooling females      0
      Mean years schooling males      0
      Log Gross National Income per capita      0
      Log Gross National Income per capita females      0
      Log Gross National Income per capita males      0
      dtype: int64
```

```
[17]: ## Dropping the redundant Columns
```

```
[18]: df.drop(["Unnamed: 0.1", "Unnamed: 0"], axis = 1, inplace = True)
```

```
[19]: df.head(2)
```

```
[19]: Subnational HDI  Health index  Educational index  Income index  \
0      0.633      0.727      0.552      0.633
1      0.706      0.820      0.607      0.706

      Life expectancy  Life expectancy females  Life expectancy males  \
0      67.24      68.89      65.76
1      73.31      76.22      71.50

      Expected years schooling  Expected years schooling girls  \
0      11.88      11.94
1      12.38      12.78

      Expected years schooling boys  Mean years schooling  \
0      11.81      6.655
```

```

1              11.98              7.900

    Mean years schooling females  Mean years schooling males \
0              6.252              7.229
1              8.424              7.712

    Log Gross National Income per capita \
0              8.793
1              9.278

    Log Gross National Income per capita females \
0              7.731
1              8.157

    Log Gross National Income per capita males
0              9.272
1              9.783

```

```
[20]: ### As we can see there are no missing values in the dataset, we can proceed
      ↪with the next step.
```

```
[21]: y = df["Subnational HDI"]
      y.head(4)
```

```
[21]: 0    0.633
      1    0.706
      2    0.630
      3    0.665
      Name: Subnational HDI, dtype: float64
```

0.3 Scaling

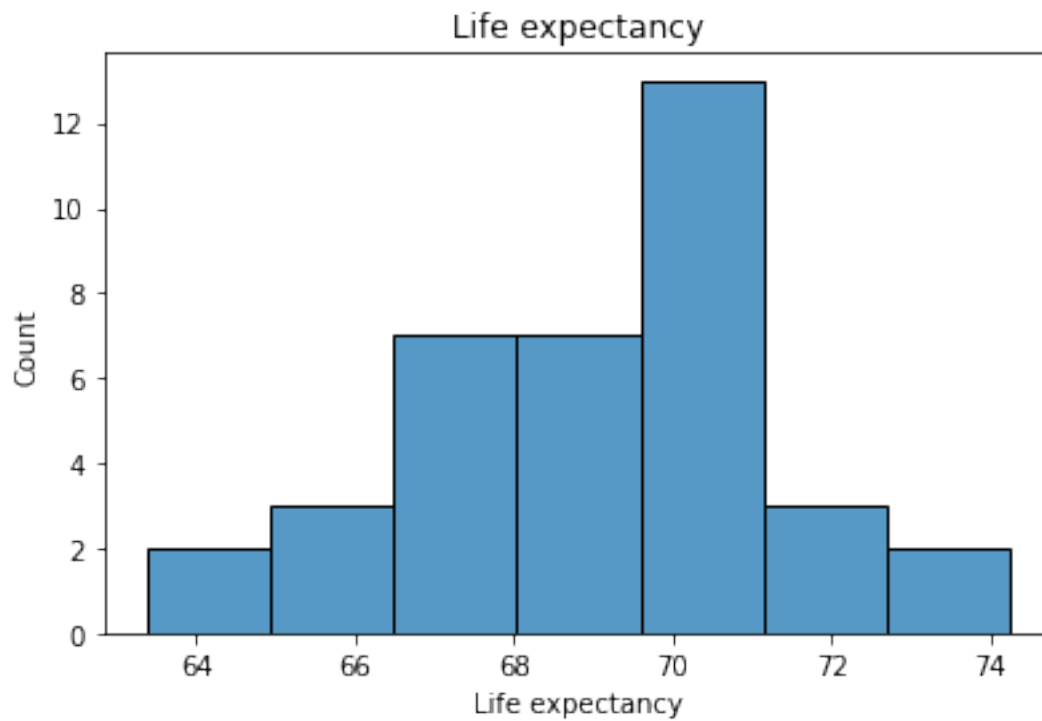
Standardization was likely chosen for your dataset because it ensures consistent scales across features, aiding algorithms like SVM and KNN. It mitigates the influence of outliers and maintains interpretability. Standardized features facilitate model convergence, improving performance, especially in algorithms sensitive to feature scales.

```
[22]: from sklearn.preprocessing import StandardScaler
```

0.3.1 Checking the distributions beforehand to confirm, that standardisation is the best method of Scaling.

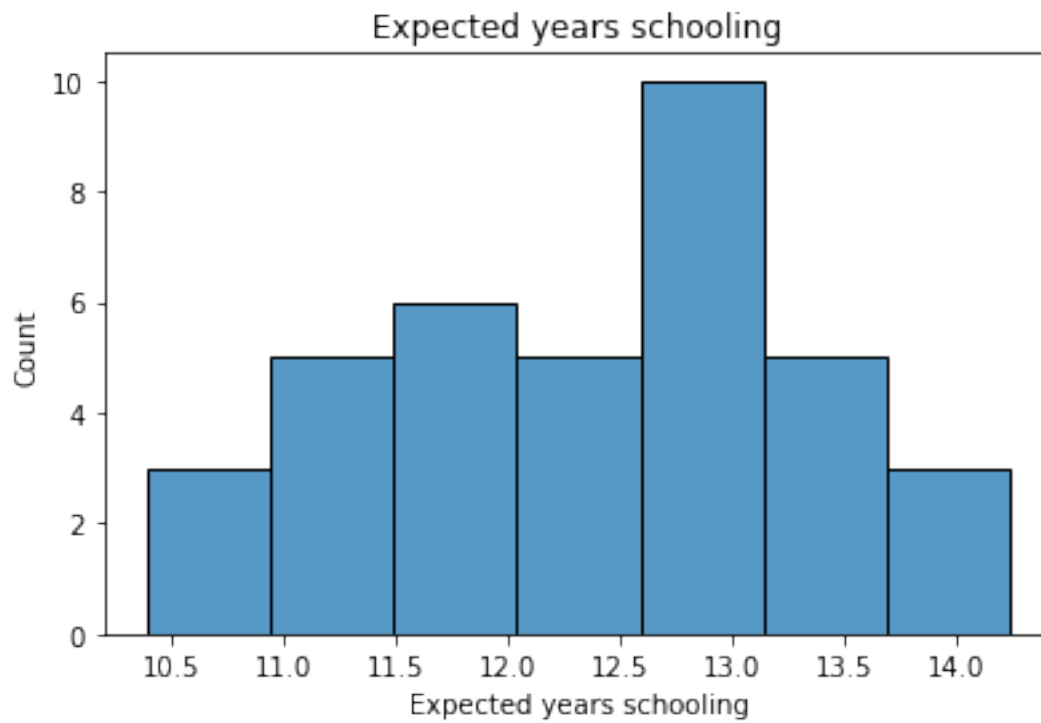
```
[23]: plt.figure(figsize = (14,4))
      plt.subplot(121)
      sns.histplot(df['Life expectancy'])
      plt.title('Life expectancy')
```

```
[23]: Text(0.5, 1.0, 'Life expectancy')
```



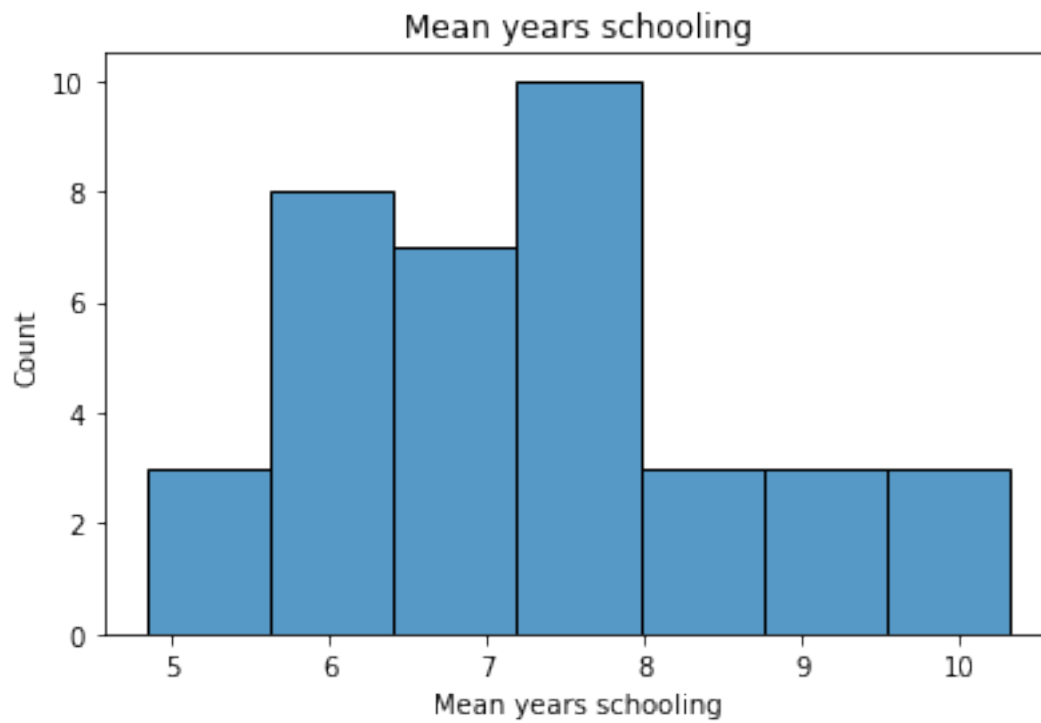
```
[24]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Expected years schooling'])  
plt.title('Expected years schooling')
```

```
[24]: Text(0.5, 1.0, 'Expected years schooling')
```



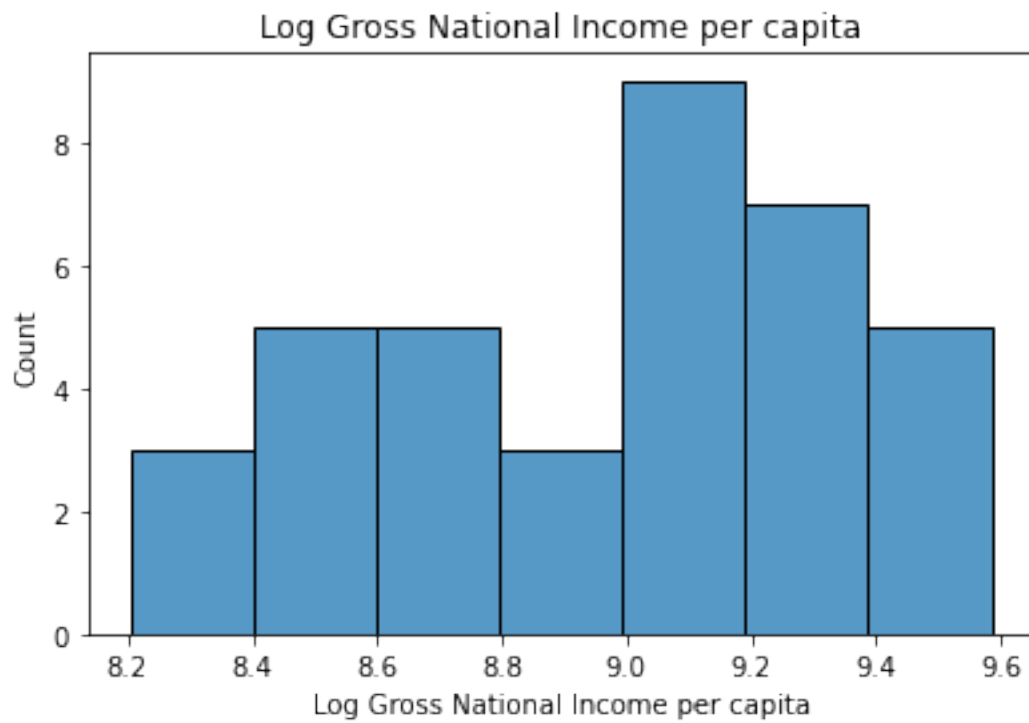
```
[25]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Mean years schooling'])  
plt.title('Mean years schooling')
```

```
[25]: Text(0.5, 1.0, 'Mean years schooling')
```



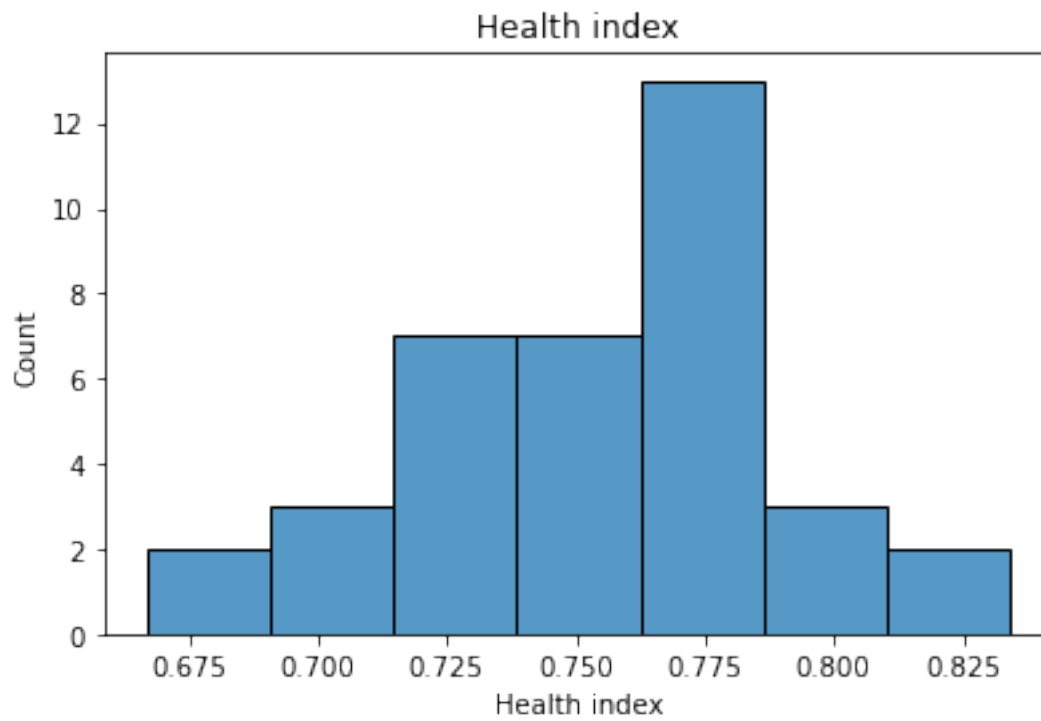
```
[26]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Log Gross National Income per capita'])  
plt.title('Log Gross National Income per capita')
```

```
[26]: Text(0.5, 1.0, 'Log Gross National Income per capita')
```

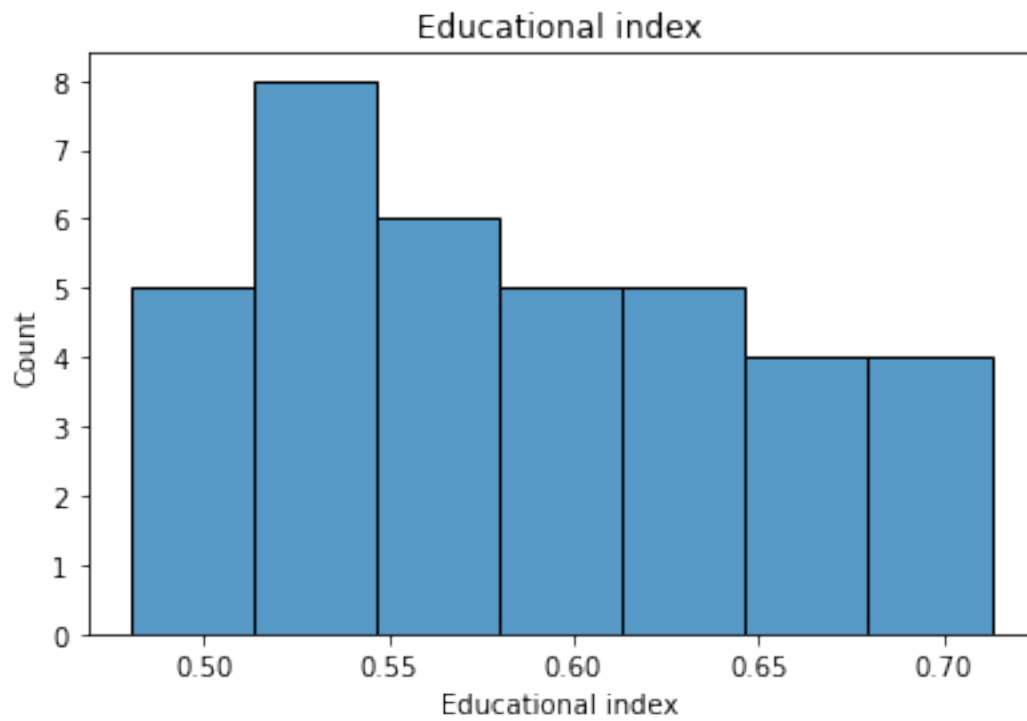
```
[27]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Health index'])  
plt.title('Health index')
```

```
[27]: Text(0.5, 1.0, 'Health index')
```



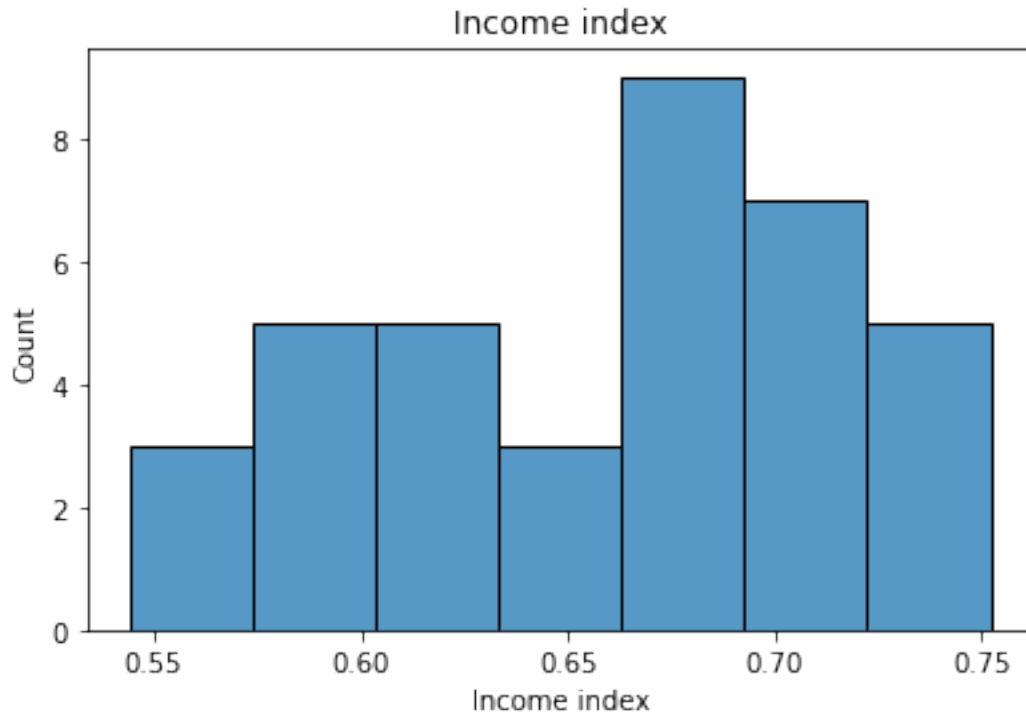
```
[28]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Educational index'])  
plt.title('Educational index')
```

```
[28]: Text(0.5, 1.0, 'Educational index')
```



```
[29]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Income index'])  
plt.title('Income index')
```

```
[29]: Text(0.5, 1.0, 'Income index')
```

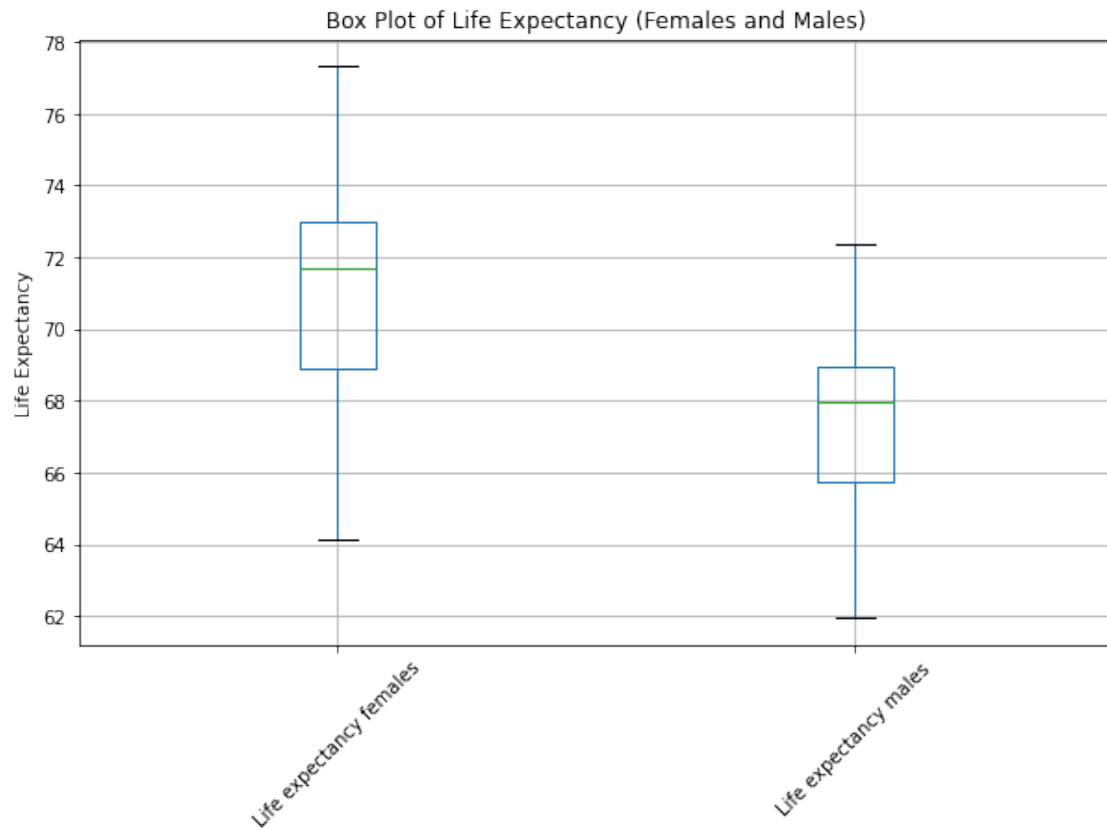


Checking for the outliers, if there are any datapoints outside the boxes, but they all fall inside the boxes

```
[30]: import matplotlib.pyplot as plt

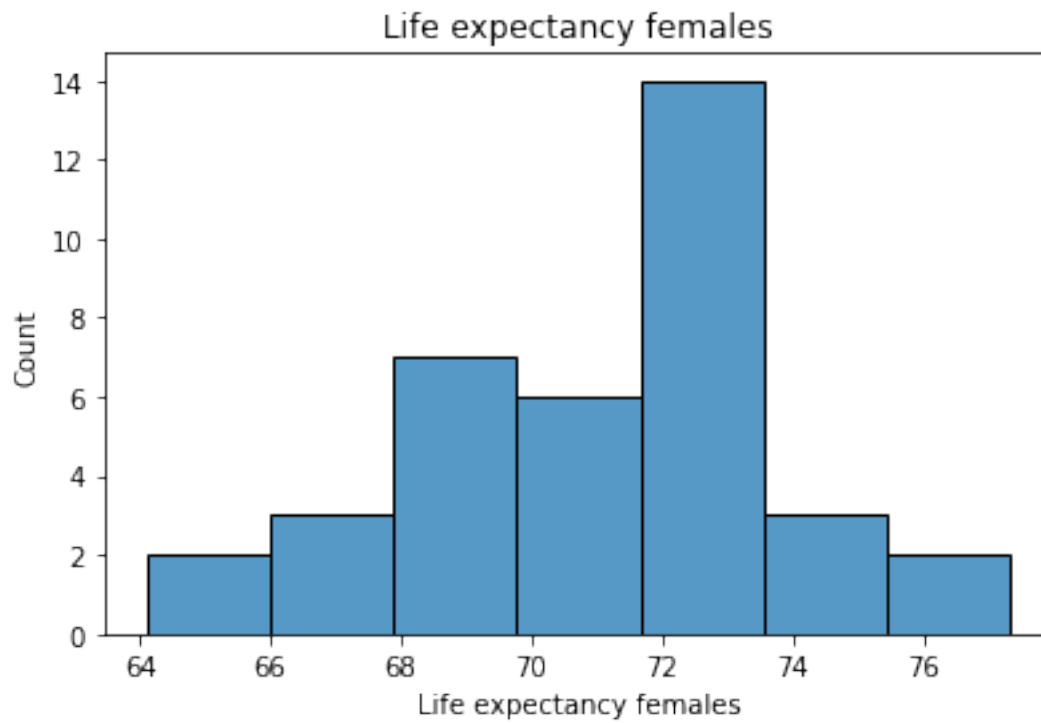
# Select the columns of interest
columns_of_interest = ['Life expectancy females', 'Life expectancy males']

# Create a box plot for each column
plt.figure(figsize=(10, 6))
df[columns_of_interest].boxplot()
plt.title('Box Plot of Life Expectancy (Females and Males)')
plt.ylabel('Life Expectancy')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



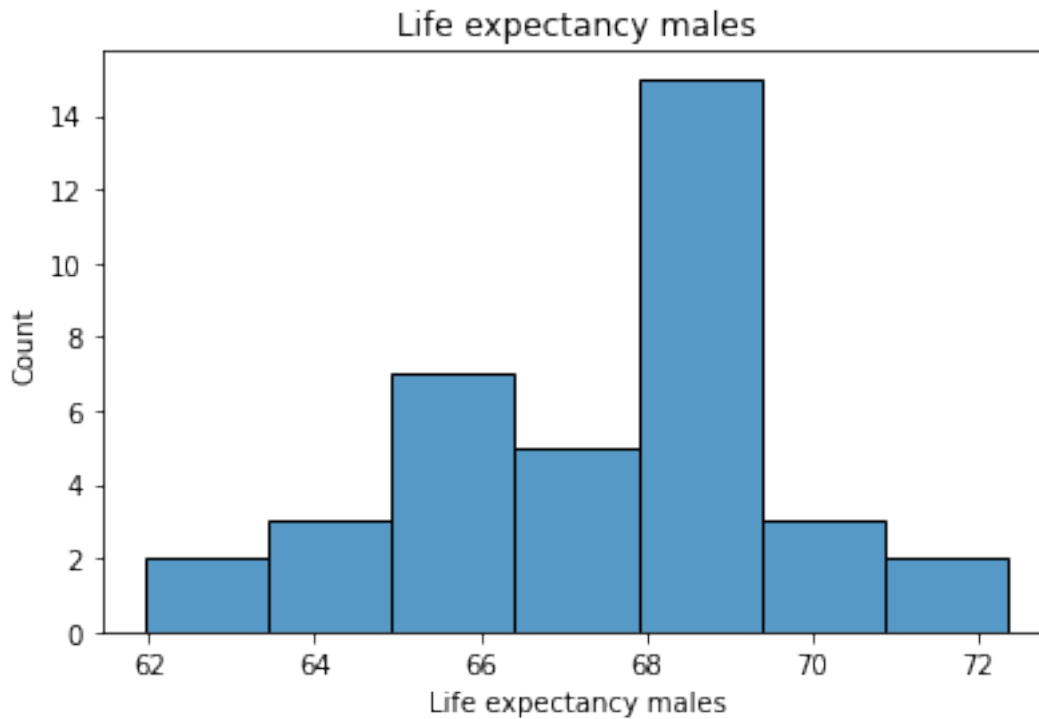
```
[31]: plt.figure(figsize = (14,4))
plt.subplot(121)
sns.histplot(df['Life expectancy females'])
plt.title('Life expectancy females')
```

```
[31]: Text(0.5, 1.0, 'Life expectancy females')
```



```
[32]: plt.figure(figsize = (14,4))  
plt.subplot(121)  
sns.histplot(df['Life expectancy males'])  
plt.title('Life expectancy males')
```

```
[32]: Text(0.5, 1.0, 'Life expectancy males')
```



```
[33]: from sklearn.preprocessing import StandardScaler

# Define the columns for StandardScaler
continuous_cols = ['Life expectancy', 'Life expectancy females', 'Life_
↳ expectancy males',
                    'Expected years schooling', 'Expected years schooling_
↳ girls', 'Expected years schooling boys',
                    'Mean years schooling', 'Mean years schooling females',_
↳ 'Mean years schooling males',
                    'Log Gross National Income per capita', 'Log Gross National_
↳ Income per capita females',
                    'Log Gross National Income per capita males']

# Initialize the StandardScaler
scaler = StandardScaler()

# Apply scaling to continuous numerical features
df[continuous_cols] = scaler.fit_transform(df[continuous_cols])
```

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

```
[34]: Subnational HDI  Health index  Educational index  Income index  \
0          0.633          0.727          0.552          0.633
```

1	0.706	0.820	0.607	0.706
2	0.630	0.734	0.517	0.660
3	0.665	0.767	0.575	0.665
4	0.597	0.714	0.530	0.564

	Life expectancy	Life expectancy females	Life expectancy males	\
0	-0.796963	-0.791650	-0.788925	
1	1.791092	1.777802	1.768728	
2	-0.605097	-0.598854	-0.597324	
3	0.324386	0.326570	0.329491	
4	-1.163640	-1.163222	-1.163215	

	Expected years schooling	Expected years schooling girls	\
0	-0.511052	-0.643727	
1	0.018313	0.152098	
2	-0.436941	-0.548986	
3	0.886472	0.663700	
4	-1.072179	-0.871106	

	Expected years schooling boys	Mean years schooling	\
0	-0.356537	-0.498189	
1	-0.164433	0.438852	
2	-0.288736	-1.332871	
3	1.112492	-0.797741	
4	-1.226654	-0.655492	

	Mean years schooling females	Mean years schooling males	\
0	-0.539697	-0.380575	
1	0.635766	0.082284	
2	-1.144206	-1.443330	
3	-0.612217	-1.025511	
4	-0.342705	-1.080134	

	Log Gross National Income per capita	\
0	-0.464284	
1	0.839561	
2	0.019618	
3	0.119086	
4	-1.692855	

	Log Gross National Income per capita females	\
0	-0.462957	
1	0.840101	
2	0.017278	
3	0.118219	
4	-1.692603	


```

    Log Gross National Income per capita males
0          -0.463218
1           0.838902
2           0.018388
3           0.117767
4          -1.691441

```

0.4 Splitting and Model Training

```
[35]: X = df.drop("Subnational HDI", axis = 1)
      X.head(2)
```

```
[35]:
Health index  Educational index  Income index  Life expectancy \
0          0.727              0.552          0.633          -0.796963
1          0.820              0.607          0.706          1.791092

Life expectancy females  Life expectancy males  Expected years schooling \
0          -0.791650              -0.788925          -0.511052
1           1.777802              1.768728           0.018313

Expected years schooling girls  Expected years schooling boys \
0          -0.643727              -0.356537
1           0.152098              -0.164433

Mean years schooling  Mean years schooling females \
0          -0.498189              -0.539697
1           0.438852              0.635766

Mean years schooling males  Log Gross National Income per capita \
0          -0.380575              -0.464284
1           0.082284              0.839561

Log Gross National Income per capita females \
0          -0.462957
1           0.840101

Log Gross National Income per capita males
0          -0.463218
1           0.838902

```

```
[36]: y.head()
```

```
[36]:
0    0.633
1    0.706
2    0.630
3    0.665
4    0.597

```

Name: Subnational HDI, dtype: float64

```
[37]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.
      ↪3,random_state=42)
```

0.5 Linear Regression

```
[38]: model = LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
      ↪normalize=False)
      model.fit(X_train, y_train)
```

```
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
      warnings.warn(
```

```
[38]: LinearRegression(normalize=False)
```

```
[39]: y_pred = model.predict(X_test)

      print('MAE : ', mean_absolute_error(y_test,y_pred))
      print('MSE : ', mean_squared_error(y_test,y_pred))
      print('R2 Score : ', r2_score(y_test,y_pred))
```

```
MAE : 0.0006594361862145385
MSE : 6.645191969534528e-07
R2 Score : 0.9997864136021881
```

0.6 Cross Validation

```
[40]: # cross validation using cross_val_score
      from sklearn.model_selection import cross_val_score
      cross_val_score(model, X_train, y_train, cv=5, scoring='r2')
```

```
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
      warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
```

parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
```

```
[40]: array([0.99707241, 0.99964995, 0.9984072 , 0.99395801, 0.99960314])
```

```
[41]: # cross validation using cross_val_score
from sklearn.model_selection import cross_val_score
cross_val_score(model, X_train, y_train, cv=5, scoring='r2').mean()
```

```
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
```

deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
```

[41]: 0.9977381406299909

0.7 GridSearchCV

```
[42]: # Define the parameter grid
param_grid = {
    "fit_intercept": [True, False],
    "normalize": [True, False],
    "copy_X": [True, False],
    "n_jobs": [None, -1] # Number of CPU cores to use during the computation.
    ↪ None means 1 and -1 means using all processors.
}

# Initialize the linear regression model
model = LinearRegression()

# Initialize GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=5, scoring='r2')

# Perform grid search
grid_search.fit(X_train, y_train)

# Get the best parameters and best score
best_params = grid_search.best_params_
best_score = grid_search.best_score_

print("Best Parameters:", best_params)
```

```
print("Best R^2 Score:", best_score)
```

```
/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was  
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}  
model.fit(X, y, **kwargs)
```

```
warnings.warn(  
/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was  
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}  
model.fit(X, y, **kwargs)
```

```
warnings.warn(  
/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was  
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
```

parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```


If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
```

packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use sklearn.preprocessing.StandardScaler instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit

parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
```

/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was deprecated in version 1.0 and will be removed in 1.2.

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a sample_weight parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
```

```

use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

```

```

from sklearn.pipeline import make_pipeline

```

```

model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())

```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```

kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:

```

```

from sklearn.pipeline import make_pipeline

```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
```


packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was deprecated in version 1.0 and will be removed in 1.2.

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
```

parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
```

deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
```

```
Best Parameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None,
'normalize': False}
```

```
Best R2 Score: 0.9977381406299909
```

```
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit

parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
```

If you wish to scale the data, use Pipeline with a StandardScaler in a preprocessing stage. To reproduce the previous behavior:

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
```

parameter to its default value to silence this warning. The default behavior of this estimator is to not do any normalization. If normalization is needed please use `sklearn.preprocessing.StandardScaler` instead.

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```

```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:141: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2.
If you wish to scale the data, use Pipeline with a StandardScaler in a
preprocessing stage. To reproduce the previous behavior:
```



```
from sklearn.pipeline import make_pipeline
```

```
model = make_pipeline(StandardScaler(with_mean=False), LinearRegression())
```

If you wish to pass a `sample_weight` parameter, you need to pass it as a fit parameter to each step of the pipeline as follows:

```
kwargs = {s[0] + '__sample_weight': sample_weight for s in model.steps}
model.fit(X, y, **kwargs)
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
```

```
packages/sklearn/linear_model/_base.py:148: FutureWarning: 'normalize' was
deprecated in version 1.0 and will be removed in 1.2. Please leave the normalize
parameter to its default value to silence this warning. The default behavior of
this estimator is to not do any normalization. If normalization is needed please
use sklearn.preprocessing.StandardScaler instead.
warnings.warn(
```

0.8 Result for the Cross-Validation

-> Best Parameters: {'copy_X': True, 'fit_intercept': True, 'n_jobs': None, 'normalize': False}

-> Best R² Score: 0.9977381406299909

[]:

0.9 KNN Regressor

```
[43]: from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

[44]: X.head()

```
[44]: Health index Educational index Income index Life expectancy \
0      0.727      0.552      0.633      -0.796963
1      0.820      0.607      0.706      1.791092
2      0.734      0.517      0.660      -0.605097
3      0.767      0.575      0.665      0.324386
4      0.714      0.530      0.564      -1.163640

Life expectancy females Life expectancy males Expected years schooling \
0      -0.791650      -0.788925      -0.511052
1      1.777802      1.768728      0.018313
2      -0.598854      -0.597324      -0.436941
3      0.326570      0.329491      0.886472
4      -1.163222      -1.163215      -1.072179

Expected years schooling girls Expected years schooling boys \
0      -0.643727      -0.356537
1      0.152098      -0.164433
2      -0.548986      -0.288736
3      0.663700      1.112492
4      -0.871106      -1.226654

Mean years schooling Mean years schooling females \
0      -0.498189      -0.539697
1      0.438852      0.635766
```

2	-1.332871	-1.144206
3	-0.797741	-0.612217
4	-0.655492	-0.342705

	Mean years schooling males	Log Gross National Income per capita \
0	-0.380575	-0.464284
1	0.082284	0.839561
2	-1.443330	0.019618
3	-1.025511	0.119086
4	-1.080134	-1.692855

	Log Gross National Income per capita females \
0	-0.462957
1	0.840101
2	0.017278
3	0.118219
4	-1.692603

	Log Gross National Income per capita males
0	-0.463218
1	0.838902
2	0.018388
3	0.117767
4	-1.691441

```
[45]: y.head()
```

```
[45]: 0    0.633
      1    0.706
      2    0.630
      3    0.665
      4    0.597
      Name: Subnational HDI, dtype: float64
```

```
[46]: # knn = KNeighborsRegressor(n_neighbors=3)
      # Initialize the KNN regressor with the best hyperparameters
      knn = KNeighborsRegressor(n_neighbors=5, weights='distance', metric =
      ↪ 'minkowski')

      # Train the model on the entire dataset
      knn.fit(X, y)
```

```
[46]: KNeighborsRegressor(weights='distance')
```

```
[47]: y_pred = knn.predict(X_test)

      # Calculate mean squared error
```

```

mse = mean_squared_error(y_test, y_pred)

# Calculate R^2 score
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R^2 Score:", r2)

```

Mean Squared Error: 0.0
R^2 Score: 1.0

```

[48]: from sklearn.neighbors import KNeighborsRegressor
      from sklearn.metrics import mean_squared_error

      # Initialize an empty list to store MSE scores
      mse_scores = []

      # Loop through different values of k
      for k in range(1, 20):
          # Initialize KNN regressor with k neighbors
          knn_regressor = KNeighborsRegressor(n_neighbors=k)

          # Fit the model on the training data
          knn_regressor.fit(X_train, y_train)

          # Make predictions on the testing data
          y_pred = knn_regressor.predict(X_test)

          # Calculate mean squared error
          mse = mean_squared_error(y_test, y_pred)

          # Append the MSE score to the list
          mse_scores.append(mse)

      # Find the best value of k with the lowest MSE
      best_k = mse_scores.index(min(mse_scores)) + 1
      best_mse = min(mse_scores)

      print("Best value of k:", best_k)
      print("Lowest MSE:", best_mse)

```

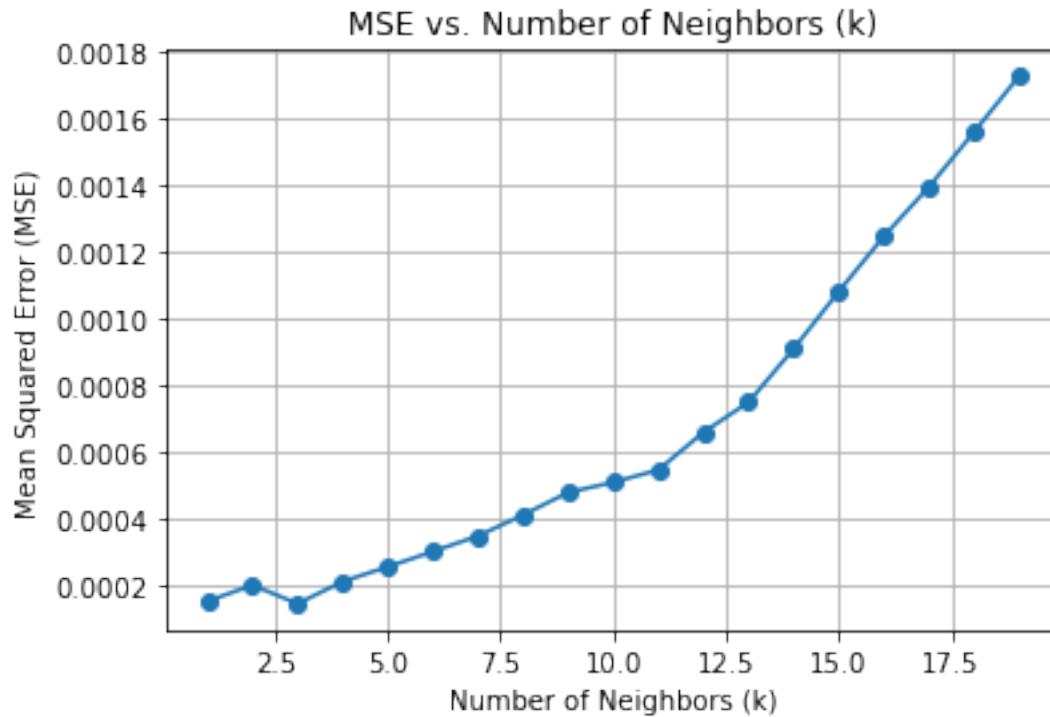
Best value of k: 3
Lowest MSE: 0.000143888888888888942

```

[49]: import matplotlib.pyplot as plt
      # Plot the MSE scores against the values of k
      plt.plot(range(1, 20), mse_scores, marker='o')

```

```
plt.xlabel('Number of Neighbors (k)')
plt.ylabel('Mean Squared Error (MSE)')
plt.title('MSE vs. Number of Neighbors (k)')
plt.grid(True)
plt.show()
## Changing the value of k = 3
```



0.10 Cross Validation

```
[50]: from sklearn.model_selection import cross_val_score
cross_val_score(knn, X_train, y_train, cv=5, scoring='r2')
```

```
[50]: array([0.98257868, 0.88720638, 0.89878448, 0.79035762, 0.99212754])
```

```
[51]: from sklearn.model_selection import cross_val_score
cross_val_score(knn, X_train, y_train, cv=5, scoring='r2').mean()
```

```
[51]: 0.9102109401922472
```

0.11 GridSearchCV

```
[52]: param_grid = {
    "n_neighbors": [5,10,15,20,25,30],
    "weights": ['uniform','distance'],
    "metric":['minkowski','manhattan','euclidean']
}

from sklearn.model_selection import GridSearchCV
grid = GridSearchCV(knn, param_grid, cv=5, scoring='r2')
grid.fit(X_train, y_train)
```

/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring failed. The score on this train-test partition for these parameters will be set to nan. Details:

Traceback (most recent call last):

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py", line 761, in _score
 scores = scorer(estimator, X_test, y_test)

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 216, in __call__
 return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 258, in _score
 y_pred = method_caller(estimator, "predict", X)

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
 return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_regression.py", line 229, in predict
 neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_base.py", line 727, in kneighbors
 raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 25

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring failed. The score on this train-test partition for these parameters will be set to nan. Details:

Traceback (most recent call last):

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py", line 761, in _score
 scores = scorer(estimator, X_test, y_test)

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring

```

failed. The score on this train-test partition for these parameters will be set to nan. Details:

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py", line 761, in _score  
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 216, in __call__  
    return self._score(  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 258, in _score  
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call  
    return getattr(estimator, method)(*args, **kwargs)  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_regression.py", line 229, in predict  
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_base.py", line 727, in kneighbors  
    raise ValueError(  
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors  
= 25
```

```
warnings.warn(  
/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring  
failed. The score on this train-test partition for these parameters will be set  
to nan. Details:
```

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py", line 761, in _score  
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 216, in __call__  
    return self._score(  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 258, in _score  
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call  
    return getattr(estimator, method)(*args, **kwargs)  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_regression.py", line 229, in predict  
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_base.py", line 727, in kneighbors  
    raise ValueError(  
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors  
= 25
```



```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
```

```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

```

Traceback (most recent call last):

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
```

```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring

```


failed. The score on this train-test partition for these parameters will be set to nan. Details:

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
```

ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 30

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring failed. The score on this train-test partition for these parameters will be set to nan. Details:
```

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
```

```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

```

Traceback (most recent call last):

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score

```

```

    scores = scorer(estimator, X_test, y_test)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__

```

```

    return self._score(

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score

```

```

    y_pred = method_caller(estimator, "predict", X)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call

```

```

    return getattr(estimator, method)(*args, **kwargs)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict

```

```

    neigh_dist, neigh_ind = self.kneighbors(X)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors

```

```

    raise ValueError(

```

```

ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

```

Traceback (most recent call last):

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score

```

```

    scores = scorer(estimator, X_test, y_test)

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__

```

```

    return self._score(

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

```

Traceback (most recent call last):

```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25

```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 25
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
```

```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
```

```
Traceback (most recent call last):
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)

```



```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring

```

failed. The score on this train-test partition for these parameters will be set to nan. Details:

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py", line 761, in _score  
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 216, in __call__  
    return self._score(  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 258, in _score  
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call  
    return getattr(estimator, method)(*args, **kwargs)  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_regression.py", line 229, in predict  
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_base.py", line 727, in kneighbors  
    raise ValueError(  
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors  
= 30
```

```
warnings.warn(  
/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring  
failed. The score on this train-test partition for these parameters will be set  
to nan. Details:
```

Traceback (most recent call last):

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/model_selection/_validation.py", line 761, in _score  
    scores = scorer(estimator, X_test, y_test)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 216, in __call__  
    return self._score(  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 258, in _score  
    y_pred = method_caller(estimator, "predict", X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call  
    return getattr(estimator, method)(*args, **kwargs)  
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_regression.py", line 229, in predict  
    neigh_dist, neigh_ind = self.kneighbors(X)
```

```
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-  
packages/sklearn/neighbors/_base.py", line 727, in kneighbors  
    raise ValueError(  
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors  
= 30
```

```
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors = 30
```

```
warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:
Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
```

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:770: UserWarning: Scoring
failed. The score on this train-test partition for these parameters will be set
to nan. Details:

```

Traceback (most recent call last):

```

File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 761, in _score
    scores = scorer(estimator, X_test, y_test)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 216, in __call__
    return self._score(
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 258, in _score
    y_pred = method_caller(estimator, "predict", X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/metrics/_scorer.py", line 68, in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 229, in predict
    neigh_dist, neigh_ind = self.kneighbors(X)
File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 727, in kneighbors
    raise ValueError(
ValueError: Expected n_neighbors <= n_samples, but n_samples = 20, n_neighbors
= 30

```

```

warnings.warn(
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py:372: FitFailedWarning:
60 fits failed out of a total of 180.
The score on these train-test partitions for these parameters will be set to
nan.
If these failures are not expected, you can try to debug them by setting
error_score='raise'.

```

Below are more details about the failures:

60 fits failed with the following error:

```

Traceback (most recent call last):
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_validation.py", line 680, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_regression.py", line 213, in fit
    return self._fit(X, y)
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 437, in _fit
    self._check_algorithm_metric()
  File "/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/neighbors/_base.py", line 374, in _check_algorithm_metric
    raise ValueError(
ValueError: Metric 'manhattan' not valid. Use
sorted(sklearn.neighbors.VALID_METRICS['brute']) to get valid options. Metric
can also be a callable function.

    warnings.warn(some_fits_failed_message, FitFailedWarning)
/Users/sid24082/opt/anaconda3/lib/python3.9/site-
packages/sklearn/model_selection/_search.py:969: UserWarning: One or more of the
test scores are non-finite: [ 0.88544519  0.91021094  0.71729282  0.82324351
0.28342464  0.60931974
-0.42376105  0.31231414          nan          nan          nan          nan
          nan          nan          nan          nan          nan          nan
          nan          nan          nan          nan          nan          nan
 0.88544519  0.91021094  0.71729282  0.82324351  0.28342464  0.60931974
-0.42376105  0.31231414          nan          nan          nan          nan]
    warnings.warn(

```

```

[52]: GridSearchCV(cv=5, estimator=KNeighborsRegressor(weights='distance'),
    param_grid={'metric': ['minkowski', 'manhattan', 'euclidean'],
                'n_neighbors': [5, 10, 15, 20, 25, 30],
                'weights': ['uniform', 'distance']},
    scoring='r2')

```

```

[53]: grid.best_score_

```

```

[53]: 0.9102109401922472

```

```

[54]: grid.best_params_

```

```

[54]: {'metric': 'minkowski', 'n_neighbors': 5, 'weights': 'distance'}

```

0.12 Decision Tree

```

[118]: from sklearn.tree import DecisionTreeRegressor

```

```
[123]: modeld = DecisionTreeRegressor(max_depth=10, criterion = 'absolute_error')
```

```
[124]: modeld.fit(X_train, y_train)
```

```
[124]: DecisionTreeRegressor(criterion='absolute_error', max_depth=10)
```

```
[125]: # Assuming you have trained the model and have new data X_test for prediction  
  
# Make predictions on the test data  
y_pred = modeld.predict(X_test)
```

```
[126]: mse = mean_squared_error(y_test, y_pred)  
r2 = r2_score(y_test, y_pred)  
mae = mean_absolute_error(y_test, y_pred)  
  
print("Mean Absolute Error (MAE):", mae)  
print("Mean Squared Error (MSE):", mse)  
print("R^2 Score:", r2)
```

Mean Absolute Error (MAE): 0.015749999999999997

Mean Squared Error (MSE): 0.00042124999999999999

R^2 Score: 0.8646039565286294

0.13 Cross-Validation

```
[127]: # cross validation using cross_val_score  
from sklearn.model_selection import cross_val_score  
cross_val_score(modeld, X_train, y_train, cv=5, scoring='r2')
```

```
[127]: array([0.78390795, 0.71610676, 0.5513093 , 0.85564608, 0.78061252])
```

```
[128]: # cross validation using cross_val_score  
from sklearn.model_selection import cross_val_score  
cross_val_score(modeld, X_train, y_train, cv=5, scoring='r2').mean()
```

```
[128]: 0.7537497611859567
```

```
[129]: from sklearn.model_selection import GridSearchCV  
# Define the parameter grid  
param_grid = {  
    'max_depth': [2,4,6,8,10,12], # Maximum depth of the tree  
    'criterion': ['squared_error', 'friedman_mse', 'absolute_error', 'poisson']  
}  
grid = GridSearchCV(modeld, param_grid, cv=5, scoring='r2')  
grid.fit(X_train, y_train)  
# Get the best hyperparameters and the best MSE score  
print(grid.best_params_)
```

```
print(grid.best_score_)
```

```
{'criterion': 'squared_error', 'max_depth': 4}  
0.755266119462917
```

0.13.1 Results

→ Best Hyperparameters: {'criterion': 'squared_error', 'max_depth': 4}

→ Best Mean Squared Error:0.755266119462917

[]:

0.14 Random Forest Regressor

```
[63]: from sklearn.ensemble import RandomForestRegressor
```

```
[64]: rfr = RandomForestRegressor(oob_score=True, max_depth = 8, criterion =  
    ↪ "absolute_error")
```

```
[65]: rfr.fit(X_train, y_train)
```

```
[65]: RandomForestRegressor(criterion='absolute_error', max_depth=8, oob_score=True)
```

```
[66]: y_pred = rfr.predict(X_test)
```

```
[67]: mse = mean_squared_error(y_test, y_pred)  
r2 = r2_score(y_test, y_pred)  
mae = mean_absolute_error(y_test, y_pred)  
  
print("Mean Absolute Error (MAE):", mae)  
print("Mean Squared Error (MSE):", mse)  
print("R^2 Score:", r2)
```

```
Mean Absolute Error (MAE): 0.010796666666666663  
Mean Squared Error (MSE): 0.00015670824999999999  
R^2 Score: 0.9496316272300952
```

```
[68]: print("The oob score is: ", rfr.oob_score_)
```

```
The oob score is: 0.8854104016989504
```

0.15 Cross-Validation

```
[69]: # cross validation using cross_val_score  
from sklearn.model_selection import cross_val_score  
cross_val_score(rfr, X_train, y_train, cv=5, scoring='r2')
```

```
[69]: array([0.97222674, 0.84535981, 0.86317557, 0.81560636, 0.9320805 ])
```

```
[70]: from sklearn.model_selection import cross_val_score
print("The mean cross validation score is: ", cross_val_score(rfr, X_train,
    ↪ y_train, cv=5, scoring='r2').mean())
```

The mean cross validation score is: 0.9007516340608875

0.16 GridSearchCV

```
[71]: params_grid = {
    "criterion" : ['squared_error', 'absolute_error', 'friedman_mse',
    ↪ 'poisson'],
    'max_depth': [2,4,6,8,10,12]}
```

```
[72]: from sklearn.model_selection import GridSearchCV

rf_grid = GridSearchCV(estimator = rfr,
    param_grid = params_grid,
    cv = 5)
```

```
[73]: rf_grid.fit(X_train,y_train)
```

```
[73]: GridSearchCV(cv=5,
    estimator=RandomForestRegressor(criterion='absolute_error',
    max_depth=8, oob_score=True),
    param_grid={'criterion': ['squared_error', 'absolute_error',
    'friedman_mse', 'poisson'],
    'max_depth': [2, 4, 6, 8, 10, 12]})
```

```
[74]: rf_grid.best_params_
```

```
[74]: {'criterion': 'absolute_error', 'max_depth': 12}
```

```
[75]: rf_grid.best_score_
```

```
[75]: 0.8994339731856098
```

0.17 FEATURE SELECTION USING RANDOM FOREST (Important Feature)

```
[76]: imp = rfr.feature_importances_
```

```
[77]: imp
```

```
[77]: array([0.03427649, 0.35426624, 0.04080619, 0.03457072, 0.04925842,
    0.04181693, 0.02839054, 0.05822956, 0.02051933, 0.08863148,
```


0.05270288, 0.0602689 , 0.04531732, 0.03668215, 0.05426284])

```
[78]: imp.max()
```

```
[78]: 0.35426623938818
```

```
[79]: ## This code lists out the top 7 importance values of the features present in  
↳ the dataset.  
max_7_features = np.argsort(imp)[-7:]  
  
# Get the maximum 5 values  
max_7_values = imp[max_7_features]  
  
print("Top 7 maximum values:")  
for value in max_7_values:  
    print(value)
```

```
Top 7 maximum values:  
0.04925842107752124  
0.05270288346013007  
0.054262839975781664  
0.05822955596039054  
0.060268904883991635  
0.08863148392037379  
0.35426623938818
```

```
[80]: ## This code gives us the information about the feature number along with it's  
↳ feature importance  
max_7_features = np.argsort(imp)[-7:]  
  
print("Top 7 maximum feature importances:")  
for feature_index in max_7_features:  
    print(f"Feature's Index in the dataset: {feature_index}, Importance:   
    ↳ {imp[feature_index]}")
```

```
Top 7 maximum feature importances:  
Feature's Index in the dataset: 4, Importance: 0.04925842107752124  
Feature's Index in the dataset: 10, Importance: 0.05270288346013007  
Feature's Index in the dataset: 14, Importance: 0.054262839975781664  
Feature's Index in the dataset: 7, Importance: 0.05822955596039054  
Feature's Index in the dataset: 11, Importance: 0.060268904883991635  
Feature's Index in the dataset: 9, Importance: 0.08863148392037379  
Feature's Index in the dataset: 1, Importance: 0.35426623938818
```

0.18 Support Vector Regressor

```
[81]: from sklearn.svm import SVR

[82]: model = SVR()

[83]: model.fit(X_train, y_train)

[83]: SVR()

[84]: y_pred = model.predict(X_test)

[85]: print('MAE',mean_absolute_error(y_test,y_pred))
      print('MSE',mean_squared_error(y_test,y_pred))
      print('R2 Score',r2_score(y_test,y_pred))
```

```
MAE 0.04791666666666667
MSE 0.0031265833333333336
R2 Score -0.004930594461395543
```

0.19 CROSS VALIDATION

```
[86]: # cross validation using cross_val_score
      from sklearn.model_selection import cross_val_score
      cross_val_score(model, X_train, y_train, cv=5, scoring='r2')

[86]: array([-0.24105546, -0.01207552, -0.57201156, -0.61038613, -0.24672057])

[87]: # cross validation using cross_val_score
      from sklearn.model_selection import cross_val_score
      print("The Cross Validation Score is : ", cross_val_score(model, X_train, y_train, cv=5, scoring='r2').mean())
```

```
The Cross Validation Score is : -0.33644984819779516
```

0.20 GridSearchCV

```
[88]: param_grid = {
      "kernel": ['linear', 'poly', 'rbf', 'sigmoid'],
      "degree": [0,1,2,3,4],
      "gamma": ['scale', 'auto'],
      "max_iter": [-1,1,2,3]
      }

[89]: from sklearn.model_selection import GridSearchCV

      svr_grid = GridSearchCV(model,
                              param_grid,
```

```
cv = 5)
svr_grid.fit(X_train, y_train)
```

```
[89]: GridSearchCV(cv=5, estimator=SVR(),
                  param_grid={'degree': [0, 1, 2, 3, 4], 'gamma': ['scale', 'auto'],
                              'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
                              'max_iter': [-1, 1, 2, 3]})
```

```
[90]: svr_grid.best_params_
```

```
[90]: {'degree': 0, 'gamma': 'scale', 'kernel': 'linear', 'max_iter': -1}
```

```
[91]: svr_grid.best_score_
```

```
[91]: -0.33644984819779516
```

1 Result:

Among the regression models, Linear Regression demonstrated exceptional performance, boasting a remarkably low Mean Absolute Error (MAE) of 0.000659 and Mean Squared Error (MSE) of 6.645e-07. Its R-squared score stood at an impressive 0.9998, indicating an almost perfect fit to the data. Cross-validation reaffirmed its superiority, yielding a mean R-squared score of 0.9977. These results underscore Linear Regression's effectiveness in accurately predicting the Human Development Index (HDI) of Indian states and union territories based on socio-economic indicators.

1.1 The END.....