forest_health_data_with_target

December 3, 2024

Dataset Description

The dataset is a comprehensive collection of ecological and environmental measurements focused on tree characteristics and site conditions. Each record in the dataset represents a distinct tree or plot, with the following features:

Plot_ID: A unique identifier for each plot where measurements are taken. This helps in distinguishing between different locations within the study area.

Latitude: The geographical latitude of the plot, measured in degrees. This indicates the north-south position of the plot on the Earth's surface.

Longitude: The geographical longitude of the plot, measured in degrees. This indicates the east-west position of the plot.

DBH (Diameter at Breast Height): The diameter of the tree measured at 1.3 meters (or breast height) above ground level, typically expressed in centimeters. This metric is crucial for assessing tree size and health.

Tree_Height: The total height of the tree from the base to the top, measured in meters. This measurement helps in understanding the growth patterns and ecological role of the tree.

Crown_Width_North_South: The width of the tree's crown measured in the north-south direction, typically in meters. This dimension can indicate the tree's overall health and competitive status in the ecosystem.

Crown_Width_East_West: The width of the tree's crown measured in the east-west direction, also typically in meters. Together with crown width in the north-south direction, it provides a complete view of the tree's canopy size.

Slope: The steepness of the terrain where the tree is located, measured in degrees. This can influence water drainage, soil erosion, and root development.

Elevation: The height of the plot above sea level, measured in meters. Elevation can affect temperature, precipitation, and overall ecosystem dynamics.

Temperature: The average temperature recorded at the plot, measured in degrees Celsius. This factor can influence tree growth, health, and species distribution.

Humidity: The average humidity at the plot, expressed as a percentage. Humidity levels can affect transpiration rates and overall tree health.

Soil_TN (Total Nitrogen): The concentration of total nitrogen in the soil, measured in grams per kilogram (g/kg). Nitrogen is essential for plant growth and development.

Soil_TP (Total Phosphorus): The concentration of total phosphorus in the soil, also measured in grams per kilogram (g/kg). Phosphorus is crucial for energy transfer and photosynthesis.

Soil_AP (Available Phosphorus): The amount of phosphorus readily available to plants in the soil, measured in grams per kilogram (g/kg). This metric helps assess nutrient availability.

Soil_AN (Available Nitrogen): The amount of nitrogen available for plant uptake in the soil, measured in grams per kilogram (g/kg). This reflects soil fertility.

Menhinick_Index: A diversity index that reflects species richness in the area. Higher values indicate greater biodiversity.

Gleason_Index: Another diversity index that accounts for the abundance and richness of species within the community.

Disturbance_Level: A categorical variable indicating the level of ecological disturbance in the area (0: low, 1: medium, 2: high). This can impact the health and stability of the ecosystem.

Fire_Risk_Index: A measure of the likelihood of fire occurrence based on environmental conditions, scored between 0 and 1. This can inform management strategies for fire-prone areas.

Health_Status: A categorical variable indicating the health of the tree, classified as either 'Healthy' or 'Unhealthy.' This is important for understanding the impact of environmental factors on tree vitality.

This file contains a dataset focused on various ecological and environmental measurements related to tree characteristics and site conditions. The dataset includes information on tree dimensions, soil composition, and environmental factors such as temperature and humidity, allowing for comprehensive analysis of the relationships between these variables.

Key Features:

Geographical Information: Latitude and longitude for each plot. Tree Metrics: Diameter at breast height (DBH), tree height, and crown dimensions. Soil Composition: Measurements of total nitrogen (TN), total phosphorus (TP), and available nutrients. Environmental Conditions: Recorded temperature and humidity levels. Ecological Indices: Menhinick and Gleason indices for assessing biodiversity. Disturbance and Fire Risk: Levels of ecological disturbance and fire risk indices. Health Status: Classification of tree health. The dataset is intended for use in ecological research and studies related to forestry, biodiversity, and environmental health.

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

[ ]:
```
df = pd.read_csv('/content/forest_health_data_with_target.csv')
df
```

[ ]:
```
   Plot_ID   Latitude   Longitude        DBH  Tree_Height  \
0        1  24.981605 -117.040695  29.862040    20.835684
1        2  48.028572  -92.066934  28.462986    24.307079
2        3  39.279758  -68.893791  91.094185     9.013101
```

3	4	33.946339	-78.744258	28.706889	19.496475
4	5	16.240746	-73.540720	30.835224	18.008888
..	…	…	…	…	…
995	996	13.663283	-84.013139	87.203097	14.378997
996	997	46.692543	-63.036977	19.940955	11.363233
997	998	15.472745	-125.172939	34.429847	13.048025
998	999	48.009494	-126.006170	32.554326	16.838336
999	1000	27.840231	-110.246905	87.784333	6.518286

	Crown_Width_North_South	Crown_Width_East_West	Slope	Elevation	\
0	6.147963	4.542720	29.171563	212.518419	
1	8.248891	5.260921	7.757386	641.640332	
2	7.841448	8.690927	39.257755	2510.612835	
3	2.385099	4.060039	27.590231	2323.628233	
4	2.343245	8.826847	7.074175	1116.863805	
..	…	…	…	…	
995	9.076576	7.159918	26.088170	892.162899	
996	2.074429	5.528984	30.016659	707.605751	
997	3.950586	7.886340	41.020960	1420.453374	
998	8.341708	5.367616	15.552908	2734.468889	
999	6.375811	2.344435	27.967829	402.992919	

	Temperature	Humidity	Soil_TN	Soil_TP	Soil_AP	Soil_AN	\
0	30.209377	93.086241	0.379904	0.268850	0.328882	0.487287	
1	29.054905	62.028839	0.339583	0.073260	0.044616	0.013501	
2	8.351397	77.992822	0.184392	0.297665	0.124953	0.474088	
3	37.290034	54.883864	0.611194	0.160819	0.387971	0.187495	
4	24.896527	70.402766	0.481858	0.030913	0.266699	0.009995	
..	…	…	…	…	…	…	
995	23.022532	88.612479	0.365851	0.007299	0.470416	0.215047	
996	10.787965	91.800707	0.927440	0.437466	0.347990	0.458267	
997	39.219948	95.736633	0.701670	0.157126	0.434471	0.251479	
998	8.777359	51.799039	0.264605	0.469601	0.186396	0.202424	
999	37.618045	37.371232	0.568444	0.161888	0.026391	0.376443	

	Menhinick_Index	Gleason_Index	Disturbance_Level	Fire_Risk_Index	\
0	0.682938	2.998681	0.004402	0.812948	
1	0.723696	3.986987	0.000330	0.678542	
2	2.129934	3.250667	0.472263	0.889075	
3	1.717352	1.333210	0.029294	0.449336	
4	2.476038	1.742321	0.974533	0.893890	
..	…	…	…	…	
995	2.127209	1.030366	0.317082	0.621856	
996	0.944496	3.290682	0.536056	0.760532	
997	0.628099	3.868748	0.661191	0.725692	
998	2.207404	3.980321	0.703485	0.692031	
999	1.127698	4.619908	0.136755	0.821540	

```
    Health_Status
0         Healthy
1    Very Healthy
2         Healthy
3       Unhealthy
4       Unhealthy
..            …
995     Unhealthy
996       Healthy
997       Healthy
998  Very Healthy
999       Healthy

[1000 rows x 20 columns]
```

[ ]:

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Plot_ID                 1000 non-null   int64
 1   Latitude                1000 non-null   float64
 2   Longitude               1000 non-null   float64
 3   DBH                     1000 non-null   float64
 4   Tree_Height             1000 non-null   float64
 5   Crown_Width_North_South 1000 non-null   float64
 6   Crown_Width_East_West   1000 non-null   float64
 7   Slope                   1000 non-null   float64
 8   Elevation               1000 non-null   float64
 9   Temperature             1000 non-null   float64
 10  Humidity                1000 non-null   float64
 11  Soil_TN                 1000 non-null   float64
 12  Soil_TP                 1000 non-null   float64
 13  Soil_AP                 1000 non-null   float64
 14  Soil_AN                 1000 non-null   float64
 15  Menhinick_Index         1000 non-null   float64
 16  Gleason_Index           1000 non-null   float64
 17  Disturbance_Level       1000 non-null   float64
 18  Fire_Risk_Index         1000 non-null   float64
 19  Health_Status           1000 non-null   object
dtypes: float64(18), int64(1), object(1)
memory usage: 156.4+ KB
```

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

[ ]:

	Plot_ID	Latitude	Longitude	DBH	Tree_Height
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	29.610262	-94.508789	52.728544	15.730501
std	288.819436	11.685494	20.453293	27.614049	8.021702
min	1.000000	10.185281	-129.774722	5.001105	2.018295
25%	250.750000	19.438931	-113.124801	29.828343	8.773222
50%	500.500000	29.872295	-93.688627	52.558322	15.559820
75%	750.250000	39.772784	-76.767446	77.114835	22.651143
max	1000.000000	49.988707	-60.041039	99.792981	29.987616

	Crown_Width_North_South	Crown_Width_East_West	Slope
count	1000.000000	1000.000000	1000.000000
mean	5.446948	5.486180	22.198898
std	2.581289	2.602753	13.038014
min	1.000276	1.055654	0.064275
25%	3.204766	3.244420	10.809975
50%	5.451383	5.413625	21.808936
75%	7.659941	7.658666	34.040896
max	9.979745	9.994153	44.975731

	Elevation	Temperature	Humidity	Soil_TN	Soil_TP
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	1498.874791	22.027384	59.743599	0.510635	0.255100
std	826.251755	9.878208	22.572259	0.283588	0.146605
min	100.698914	5.008503	20.004226	0.010934	0.005078
25%	784.368948	13.728430	41.131530	0.260105	0.130452
50%	1503.573023	21.754533	59.614944	0.511302	0.249754
75%	2171.952127	30.056674	78.897379	0.759135	0.387961
max	2996.823629	39.860447	99.960415	0.999676	0.499671

	Soil_AP	Soil_AN	Menhinick_Index	Gleason_Index
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.251220	0.249344	1.762232	2.963965
std	0.142471	0.145486	0.724376	1.163286
min	0.005596	0.005660	0.503300	1.001239
25%	0.127690	0.121242	1.136698	1.947451
50%	0.247471	0.243803	1.752412	2.969374
75%	0.377836	0.377283	2.421229	3.987144
max	0.499356	0.499428	2.999513	4.999699

	Disturbance_Level	Fire_Risk_Index
count	1000.000000	1000.000000
mean	0.512124	0.509207
std	0.287952	0.281458
min	0.000252	0.000854

```
25%             0.270524         0.277123
50%             0.523023         0.516489
75%             0.750194         0.746163
max             0.999150         0.997163
```

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

[ ]:
```
Plot_ID                    0
Latitude                   0
Longitude                  0
DBH                        0
Tree_Height                0
Crown_Width_North_South    0
Crown_Width_East_West      0
Slope                      0
Elevation                  0
Temperature                0
Humidity                   0
Soil_TN                    0
Soil_TP                    0
Soil_AP                    0
Soil_AN                    0
Menhinick_Index            0
Gleason_Index              0
Disturbance_Level          0
Fire_Risk_Index            0
Health_Status              0
dtype: int64
```

[ ]:
```
df.shape
```

[ ]:
```
(1000, 20)
```

[ ]:
```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['Health_Status'] = labelencoder.fit_transform(df['Health_Status'])
df
```

[ ]:
```
     Plot_ID   Latitude   Longitude        DBH  Tree_Height  \
0          1  24.981605 -117.040695  29.862040    20.835684
1          2  48.028572  -92.066934  28.462986    24.307079
2          3  39.279758  -68.893791  91.094185     9.013101
3          4  33.946339  -78.744258  28.706889    19.496475
4          5  16.240746  -73.540720  30.835224    18.008888
..       …          …          …          …          …
995      996  13.663283  -84.013139  87.203097    14.378997
996      997  46.692543  -63.036977  19.940955    11.363233
```

997	998	15.472745	-125.172939	34.429847	13.048025
998	999	48.009494	-126.006170	32.554326	16.838336
999	1000	27.840231	-110.246905	87.784333	6.518286

	Crown_Width_North_South	Crown_Width_East_West	Slope	Elevation	\
0	6.147963	4.542720	29.171563	212.518419	
1	8.248891	5.260921	7.757386	641.640332	
2	7.841448	8.690927	39.257755	2510.612835	
3	2.385099	4.060039	27.590231	2323.628233	
4	2.343245	8.826847	7.074175	1116.863805	
..	…	…	…	…	
995	9.076576	7.159918	26.088170	892.162899	
996	2.074429	5.528984	30.016659	707.605751	
997	3.950586	7.886340	41.020960	1420.453374	
998	8.341708	5.367616	15.552908	2734.468889	
999	6.375811	2.344435	27.967829	402.992919	

	Temperature	Humidity	Soil_TN	Soil_TP	Soil_AP	Soil_AN	\
0	30.209377	93.086241	0.379904	0.268850	0.328882	0.487287	
1	29.054905	62.028839	0.339583	0.073260	0.044616	0.013501	
2	8.351397	77.992822	0.184392	0.297665	0.124953	0.474088	
3	37.290034	54.883864	0.611194	0.160819	0.387971	0.187495	
4	24.896527	70.402766	0.481858	0.030913	0.266699	0.009995	
..	…	…	…	…	…	…	
995	23.022532	88.612479	0.365851	0.007299	0.470416	0.215047	
996	10.787965	91.800707	0.927440	0.437466	0.347990	0.458267	
997	39.219948	95.736633	0.701670	0.157126	0.434471	0.251479	
998	8.777359	51.799039	0.264605	0.469601	0.186396	0.202424	
999	37.618045	37.371232	0.568444	0.161888	0.026391	0.376443	

	Menhinick_Index	Gleason_Index	Disturbance_Level	Fire_Risk_Index	\
0	0.682938	2.998681	0.004402	0.812948	
1	0.723696	3.986987	0.000330	0.678542	
2	2.129934	3.250667	0.472263	0.889075	
3	1.717352	1.333210	0.029294	0.449336	
4	2.476038	1.742321	0.974533	0.893890	
..	…	…	…	…	
995	2.127209	1.030366	0.317082	0.621856	
996	0.944496	3.290682	0.536056	0.760532	
997	0.628099	3.868748	0.661191	0.725692	
998	2.207404	3.980321	0.703485	0.692031	
999	1.127698	4.619908	0.136755	0.821540	

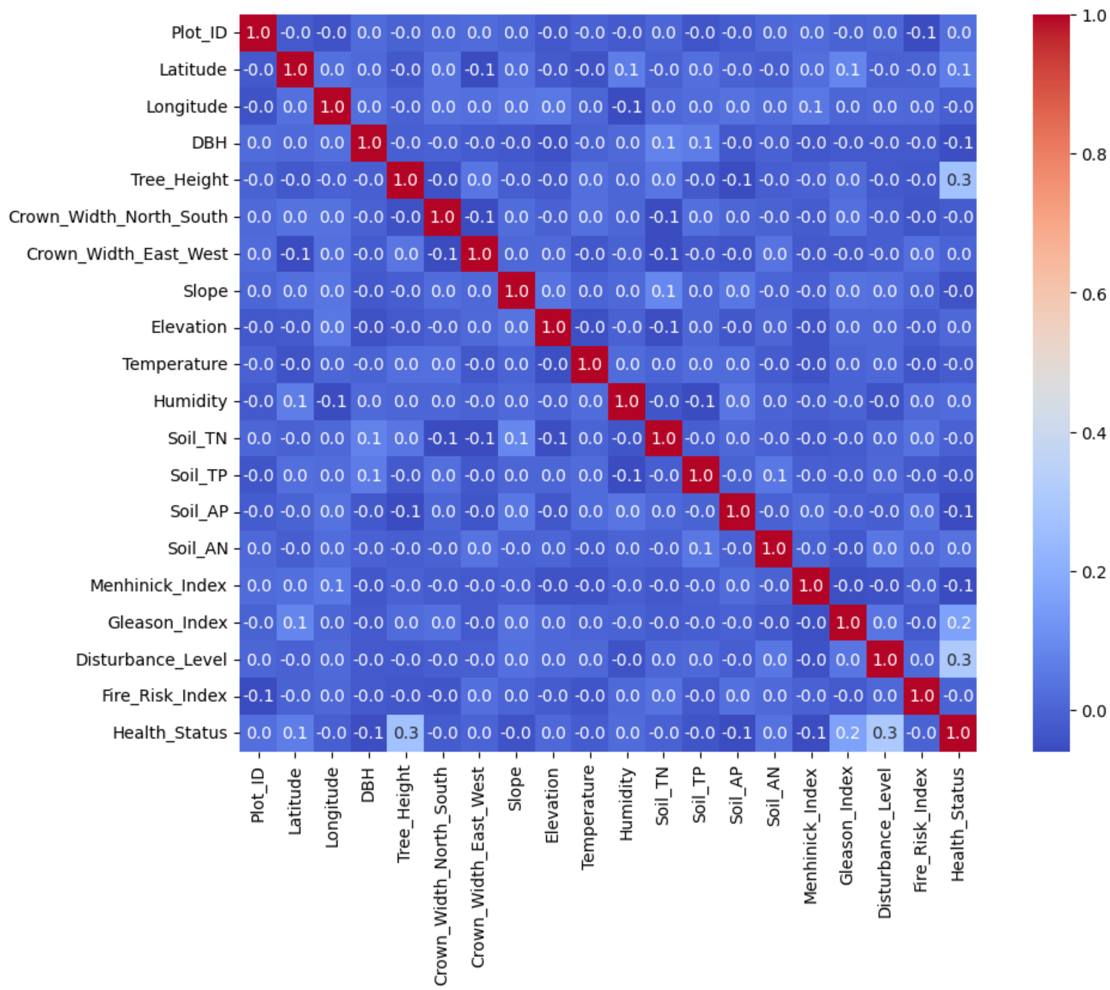
	Health_Status
0	0
1	3
2	0

```
3                 2
4                 2
..              …
995               2
996               0
997               0
998               3
999               0

[1000 rows x 20 columns]
```

[ ]:

```python
corr_matrix = df.corr()
plt.figure(figsize=(12,8))
sns.heatmap(corr_matrix,annot=True, fmt='0.01f',␣
 ↪cmap='coolwarm',annot_kws=None, linewidths=0, robust=True,square=True)
plt.show()
```



[ ]:
```
corr_matrix
```

[ ]:
```
                         Plot_ID  Latitude  Longitude       DBH  Tree_Height  \
Plot_ID                 1.000000 -0.024495  -0.038855  0.015767    -0.003548
Latitude               -0.024495  1.000000   0.029310  0.014518    -0.029424
Longitude              -0.038855  0.029310   1.000000  0.027262    -0.005791
DBH                     0.015767  0.014518   0.027262  1.000000    -0.013560
Tree_Height            -0.003548 -0.029424  -0.005791 -0.013560     1.000000
Crown_Width_North_South  0.013645  0.034785   0.032911 -0.008858    -0.044812
Crown_Width_East_West   0.016340 -0.059440   0.029994 -0.020353     0.039959
Slope                   0.006027  0.027269   0.042327 -0.028299    -0.017133
Elevation              -0.028854 -0.025491   0.049960 -0.046775    -0.022849
Temperature            -0.006546 -0.040870   0.003279 -0.005227     0.025397
Humidity               -0.023253  0.060702  -0.065495  0.013119     0.004490
Soil_TN                 0.005536 -0.005791   0.008880  0.076806     0.043190
Soil_TP                -0.037982  0.024781   0.020570  0.060070    -0.022730
Soil_AP                -0.019029 -0.005228   0.035344 -0.027398    -0.079880
Soil_AN                 0.008283 -0.017100   0.022119 -0.005860    -0.016807
Menhinick_Index         0.015657  0.025585   0.050224 -0.032187    -0.009280
Gleason_Index          -0.002757  0.082400   0.005469 -0.026336     0.018715
Disturbance_Level       0.004502 -0.006398   0.013049 -0.022349    -0.022732
Fire_Risk_Index        -0.057152 -0.003098   0.010811 -0.022301    -0.030735
Health_Status           0.026929  0.061662  -0.010078 -0.054794     0.267919

                         Crown_Width_North_South  Crown_Width_East_West  \
Plot_ID                                 0.013645               0.016340
Latitude                                0.034785              -0.059440
Longitude                               0.032911               0.029994
DBH                                    -0.008858              -0.020353
Tree_Height                            -0.044812               0.039959
Crown_Width_North_South                 1.000000              -0.055864
Crown_Width_East_West                  -0.055864               1.000000
Slope                                   0.020511               0.007983
Elevation                              -0.008889               0.012288
Temperature                             0.028328              -0.031942
Humidity                                0.005123              -0.002803
Soil_TN                                -0.057163              -0.055644
Soil_TP                                 0.025343              -0.026626
Soil_AP                                 0.003647              -0.037322
Soil_AN                                -0.004964               0.039061
Menhinick_Index                        -0.019592              -0.023508
Gleason_Index                           0.029000              -0.024188
Disturbance_Level                      -0.017836              -0.011534
Fire_Risk_Index                        -0.038124               0.026544
Health_Status                          -0.019718               0.001414

                            Slope  Elevation  Temperature  Humidity   Soil_TN  \
```

Plot_ID	0.006027	-0.028854	-0.006546	-0.023253	0.005536
Latitude	0.027269	-0.025491	-0.040870	0.060702	-0.005791
Longitude	0.042327	0.049960	0.003279	-0.065495	0.008880
DBH	-0.028299	-0.046775	-0.005227	0.013119	0.076806
Tree_Height	-0.017133	-0.022849	0.025397	0.004490	0.043190
Crown_Width_North_South	0.020511	-0.008889	0.028328	0.005123	-0.057163
Crown_Width_East_West	0.007983	0.012288	-0.031942	-0.002803	-0.055644
Slope	1.000000	0.038155	0.001169	0.013190	0.086294
Elevation	0.038155	1.000000	-0.049307	-0.008831	-0.051893
Temperature	0.001169	-0.049307	1.000000	0.017458	0.001202
Humidity	0.013190	-0.008831	0.017458	1.000000	-0.036207
Soil_TN	0.086294	-0.051893	0.001202	-0.036207	1.000000
Soil_TP	0.006264	0.002730	0.018661	-0.083376	-0.015680
Soil_AP	0.045217	-0.032696	0.016049	0.039948	0.009976
Soil_AN	-0.000348	0.005226	-0.020689	0.002025	-0.008883
Menhinick_Index	-0.012213	-0.042175	-0.039130	-0.014058	-0.026393
Gleason_Index	0.032052	0.009770	0.001327	-0.004267	-0.019694
Disturbance_Level	0.029232	0.008705	0.007319	-0.040440	0.004372
Fire_Risk_Index	0.000398	-0.021846	-0.038563	0.010919	0.032654
Health_Status	-0.040577	0.010535	-0.021022	0.015195	-0.009481

	Soil_TP	Soil_AP	Soil_AN	Menhinick_Index \
Plot_ID	-0.037982	-0.019029	0.008283	0.015657
Latitude	0.024781	-0.005228	-0.017100	0.025585
Longitude	0.020570	0.035344	0.022119	0.050224
DBH	0.060070	-0.027398	-0.005860	-0.032187
Tree_Height	-0.022730	-0.079880	-0.016807	-0.009280
Crown_Width_North_South	0.025343	0.003647	-0.004964	-0.019592
Crown_Width_East_West	-0.026626	-0.037322	0.039061	-0.023508
Slope	0.006264	0.045217	-0.000348	-0.012213
Elevation	0.002730	-0.032696	0.005226	-0.042175
Temperature	0.018661	0.016049	-0.020689	-0.039130
Humidity	-0.083376	0.039948	0.002025	-0.014058
Soil_TN	-0.015680	0.009976	-0.008883	-0.026393
Soil_TP	1.000000	-0.004282	0.051053	-0.010134
Soil_AP	-0.004282	1.000000	-0.007827	0.015391
Soil_AN	0.051053	-0.007827	1.000000	-0.005071
Menhinick_Index	-0.010134	0.015391	-0.005071	1.000000
Gleason_Index	-0.014953	-0.009035	-0.028123	-0.048760
Disturbance_Level	0.009845	-0.017563	0.047247	-0.044280
Fire_Risk_Index	-0.023741	0.031064	0.013869	-0.008298
Health_Status	-0.037896	-0.059820	0.032414	-0.056643

	Gleason_Index	Disturbance_Level	Fire_Risk_Index \
Plot_ID	-0.002757	0.004502	-0.057152
Latitude	0.082400	-0.006398	-0.003098
Longitude	0.005469	0.013049	0.010811

```
DBH                        -0.026336            -0.022349        -0.022301
Tree_Height                 0.018715            -0.022732        -0.030735
Crown_Width_North_South     0.029000            -0.017836        -0.038124
Crown_Width_East_West      -0.024188            -0.011534         0.026544
Slope                       0.032052             0.029232         0.000398
Elevation                   0.009770             0.008705        -0.021846
Temperature                 0.001327             0.007319        -0.038563
Humidity                   -0.004267            -0.040440         0.010919
Soil_TN                    -0.019694             0.004372         0.032654
Soil_TP                    -0.014953             0.009845        -0.023741
Soil_AP                    -0.009035            -0.017563         0.031064
Soil_AN                    -0.028123             0.047247         0.013869
Menhinick_Index            -0.048760            -0.044280        -0.008298
Gleason_Index               1.000000             0.041623        -0.022479
Disturbance_Level           0.041623             1.000000         0.008205
Fire_Risk_Index            -0.022479             0.008205         1.000000
Health_Status               0.163639             0.300717        -0.002316

                         Health_Status
Plot_ID                       0.026929
Latitude                      0.061662
Longitude                    -0.010078
DBH                          -0.054794
Tree_Height                   0.267919
Crown_Width_North_South      -0.019718
Crown_Width_East_West         0.001414
Slope                        -0.040577
Elevation                     0.010535
Temperature                  -0.021022
Humidity                      0.015195
Soil_TN                      -0.009481
Soil_TP                      -0.037896
Soil_AP                      -0.059820
Soil_AN                       0.032414
Menhinick_Index              -0.056643
Gleason_Index                 0.163639
Disturbance_Level             0.300717
Fire_Risk_Index              -0.002316
Health_Status                 1.000000
```

[ ]:

```
sor = df.corr()['Health_Status']
sor = sor.sort_values(ascending=False)
sor
```

[ ]:

```
Health_Status              1.000000
Disturbance_Level          0.300717
Tree_Height                0.267919
```

```
Gleason_Index              0.163639
Latitude                   0.061662
Soil_AN                    0.032414
Plot_ID                    0.026929
Humidity                   0.015195
Elevation                  0.010535
Crown_Width_East_West      0.001414
Fire_Risk_Index           -0.002316
Soil_TN                   -0.009481
Longitude                 -0.010078
Crown_Width_North_South   -0.019718
Temperature               -0.021022
Soil_TP                   -0.037896
Slope                     -0.040577
DBH                       -0.054794
Menhinick_Index           -0.056643
Soil_AP                   -0.059820
Name: Health_Status, dtype: float64
```

[ ]:
```
df.drop(columns=['Latitude'], inplace=True)
df.drop(columns=['Soil_AN'], inplace=True)
df.drop(columns=['Plot_ID'], inplace=True)
df.drop(columns=['Elevation'], inplace=True)
df.drop(columns=['Crown_Width_East_West'], inplace=True)
df.drop(columns=['Fire_Risk_Index'], inplace=True)
df.drop(columns=['Soil_TN'], inplace=True)
df.drop(columns=['Longitude'], inplace=True)
df.drop(columns=['Crown_Width_North_South'], inplace=True)
df.drop(columns=['Temperature'], inplace=True)
df.drop(columns=['Soil_TP'], inplace=True)
df.drop(columns=['Slope'], inplace=True)
df.drop(columns=['DBH'], inplace=True)
df.drop(columns=['Menhinick_Index'], inplace=True)
df.drop(columns=['Soil_AP'], inplace=True)
df.drop(columns=['Humidity'], inplace=True)
```

[ ]:
```
df
```

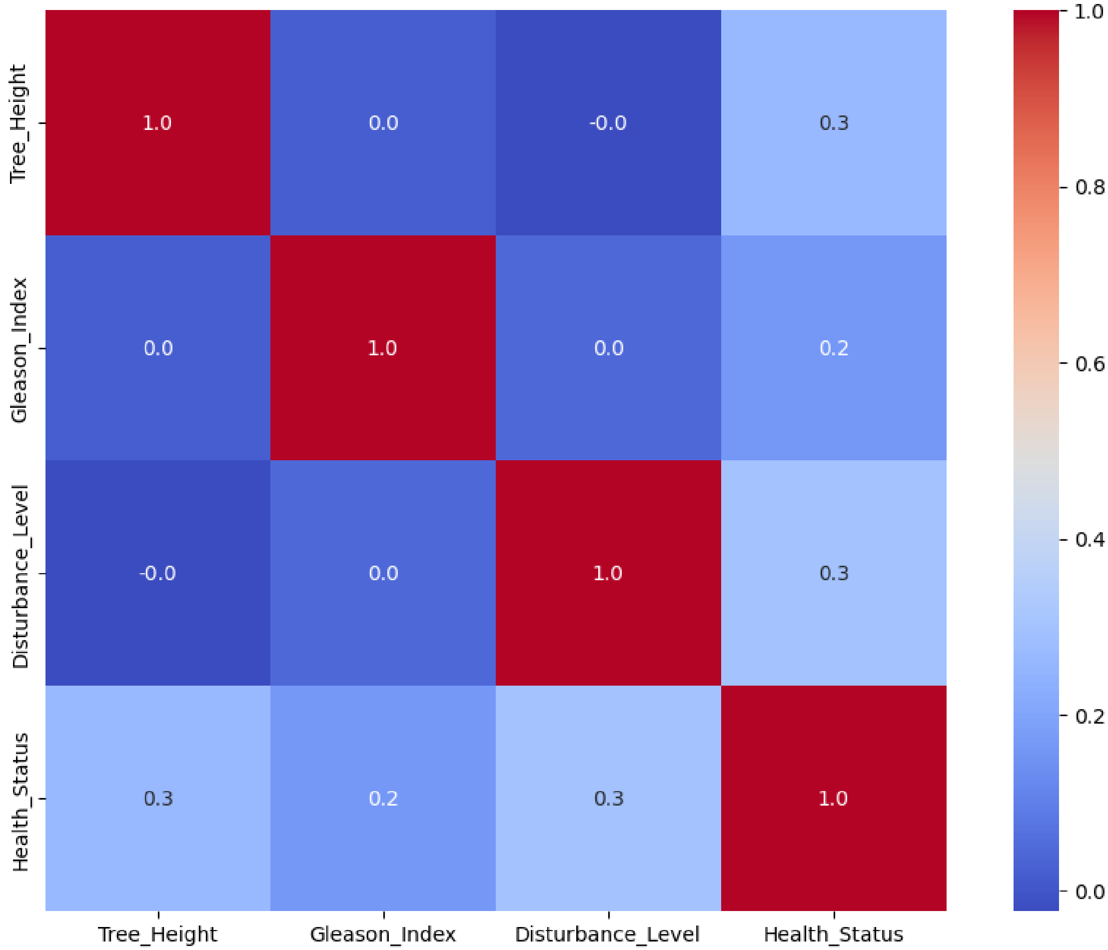
[ ]:

	Tree_Height	Gleason_Index	Disturbance_Level	Health_Status
0	20.835684	2.998681	0.004402	0
1	24.307079	3.986987	0.000330	3
2	9.013101	3.250667	0.472263	0
3	19.496475	1.333210	0.029294	2
4	18.008888	1.742321	0.974533	2
..	…	…	…	…
995	14.378997	1.030366	0.317082	2
996	11.363233	3.290682	0.536056	0

```
997    13.048025       3.868748            0.661191              0
998    16.838336       3.980321            0.703485              3
999     6.518286       4.619908            0.136755              0

[1000 rows x 4 columns]
```

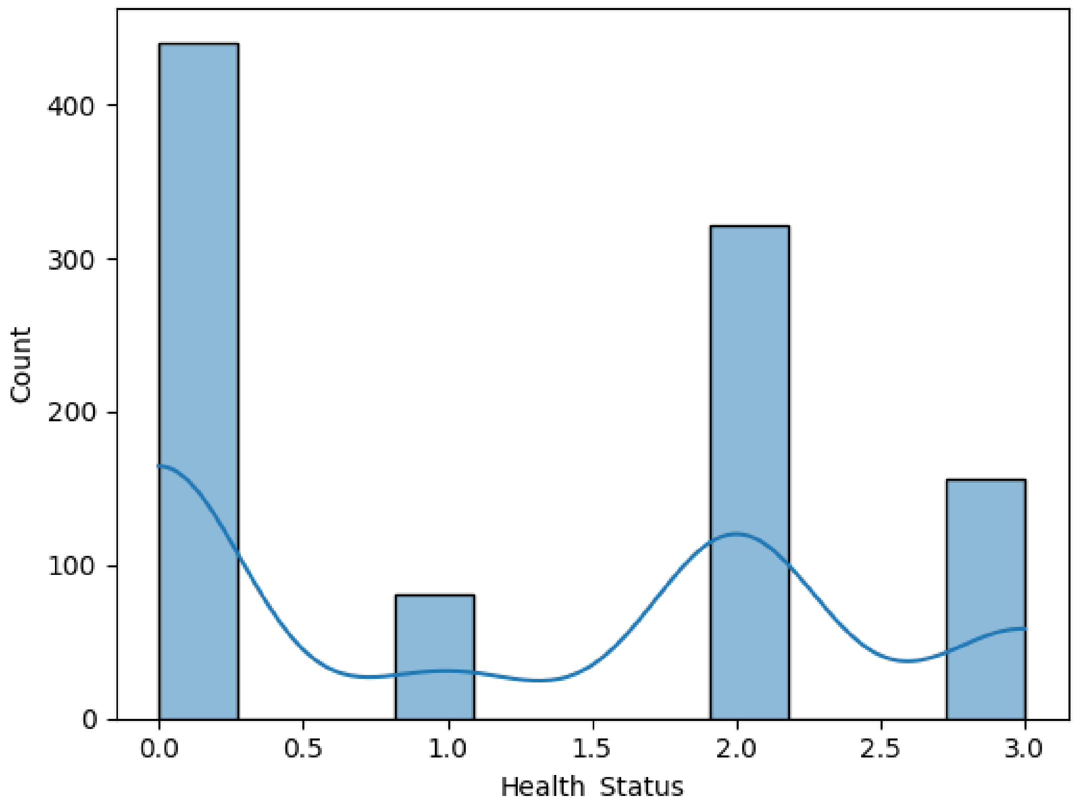
```
[ ]: corr_matrix = df.corr()
     plt.figure(figsize=(12,8))
     sns.heatmap(corr_matrix,annot=True, fmt='0.01f',
       cmap='coolwarm',annot_kws=None, linewidths=0, robust=True,square=True)
     plt.show()
```



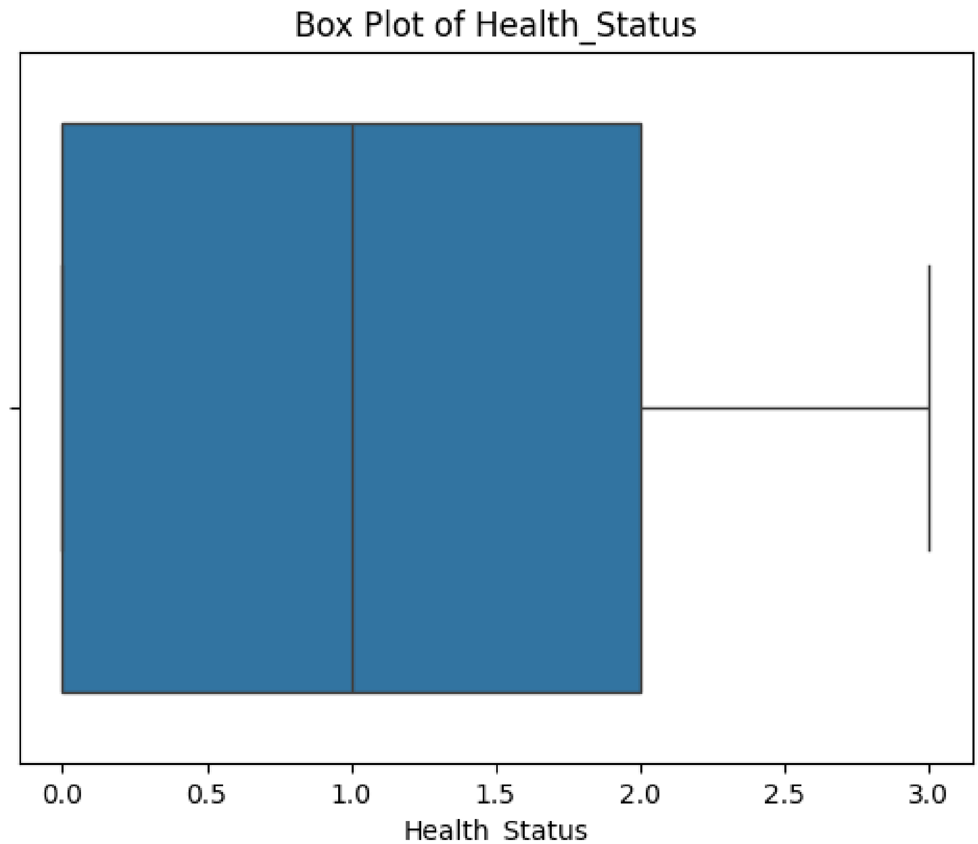

```
[ ]: sor = df.corr()['Health_Status']
     sor = sor.sort_values(ascending=False)
     sor
```

[ ]: Health_Status 1.000000
Disturbance_Level 0.300717
Tree_Height 0.267919
Gleason_Index 0.163639
Name: Health_Status, dtype: float64

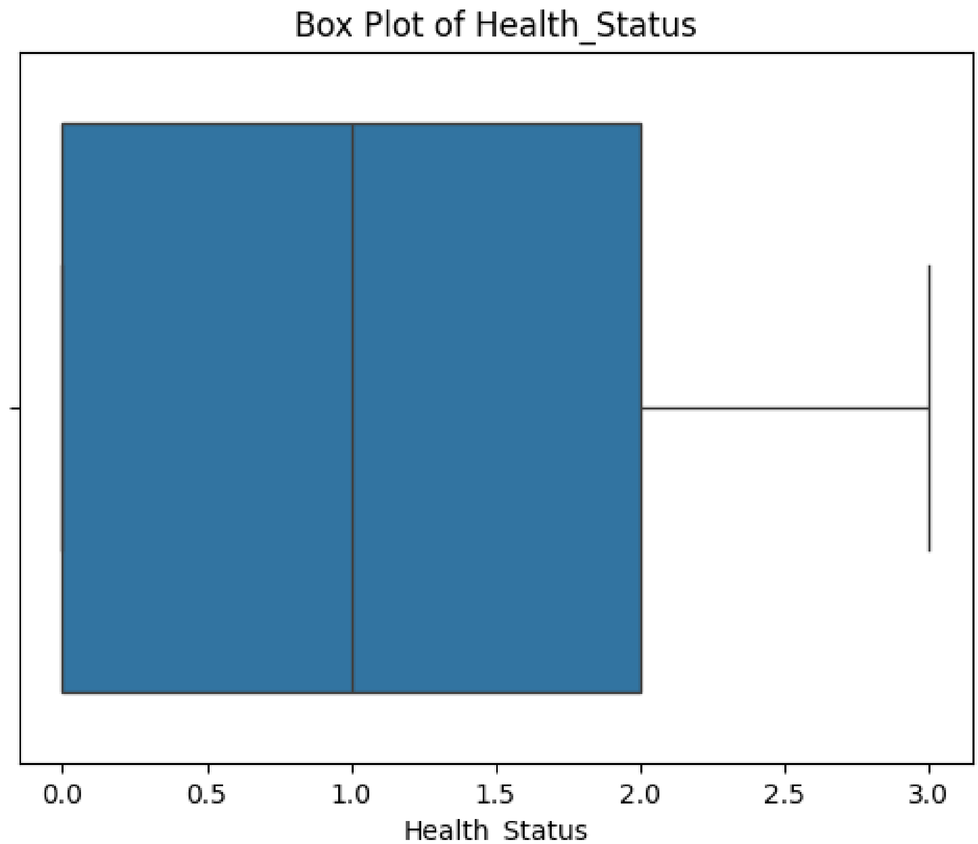
[ ]:
```
sns.histplot(df['Health_Status'], kde=True)
plt.show()
```



[ ]:
```
sns.boxplot(x=df['Health_Status'])
plt.title('Box Plot of Health_Status')
plt.show()
```

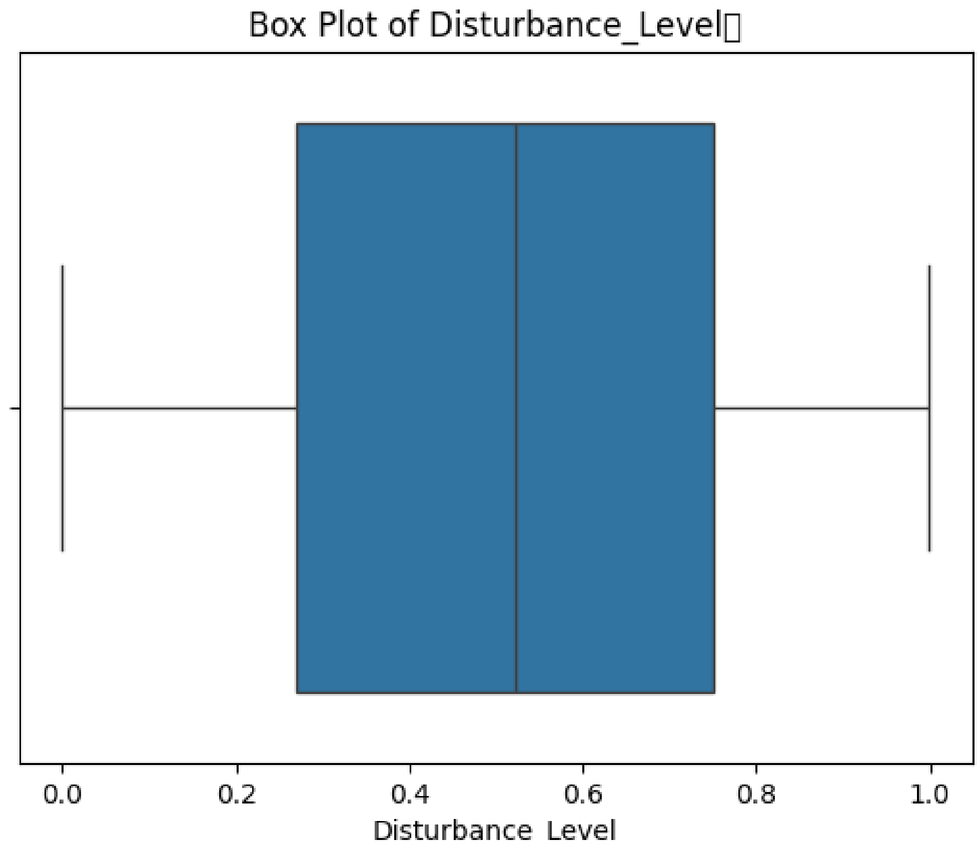


[ ]:
```
sns.boxplot(x=df['Health_Status'])
plt.title('Box Plot of Health_Status')
plt.show()
```

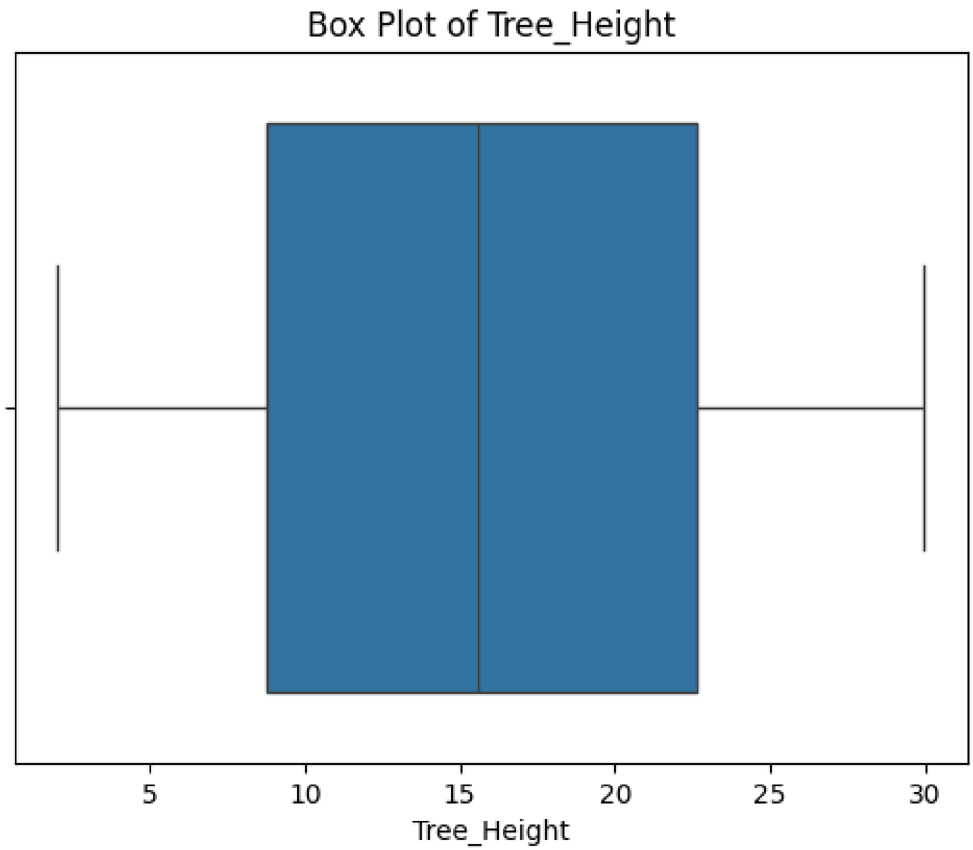


[ ]:
```
sns.boxplot(x=df['Disturbance_Level'])
plt.title('Box Plot of Disturbance_Level        ')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151:
UserWarning: Glyph 9 (   ) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```

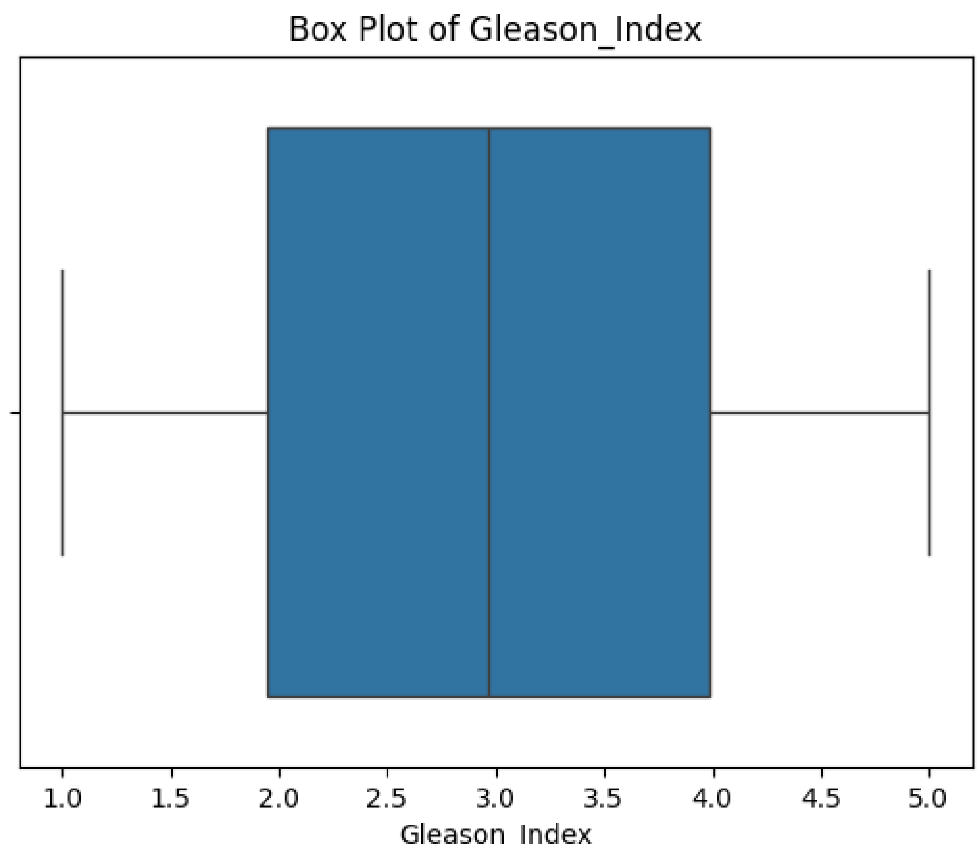


[ ]:
```
sns.boxplot(x=df['Tree_Height'])
plt.title('Box Plot of Tree_Height')
plt.show()
```



[ ]:

```
sns.boxplot(x=df['Gleason_Index'])
plt.title('Box Plot of Gleason_Index')
plt.show()
```



[ ]:
```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

[ ]:
```python
independet_feature = ['Disturbance_Level','Gleason_Index','Tree_Height']
dependet_feature = ['Health_Status']
x = df[independet_feature]
y = df[dependet_feature]
```

[ ]:
```python
# Create and fit the scaler
scaler = StandardScaler()
scaler.fit(x)

# Transform the data
X_scaled = scaler.transform(x)
# X_test_scaled = scaler.transform(X_test)
```

[ ]:
```python
x
```

```
[ ]:      Disturbance_Level  Gleason_Index  Tree_Height
     0             0.004402       2.998681    20.835684
     1             0.000330       3.986987    24.307079
     2             0.472263       3.250667     9.013101
     3             0.029294       1.333210    19.496475
     4             0.974533       1.742321    18.008888
     ..                 …              …          …
     995           0.317082       1.030366    14.378997
     996           0.536056       3.290682    11.363233
     997           0.661191       3.868748    13.048025
     998           0.703485       3.980321    16.838336
     999           0.136755       4.619908     6.518286

     [1000 rows x 3 columns]
```

```
[ ]: y
```

```
[ ]:      Health_Status
     0                0
     1                3
     2                0
     3                2
     4                2
     ..             …
     995              2
     996              0
     997              0
     998              3
     999              0

     [1000 rows x 1 columns]
```

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
                                                    random_state=42)#to take ␣
      ↪the same 80% and 20% of data(same function of seed)
```

```
[ ]: print(X_train.shape)
     print(X_test.shape)
     print(y_train.shape)
     print(y_test.shape )
```

```
(800, 3)
(200, 3)
(800, 1)
(200, 1)
```

```
[ ]: # Correct the typo in the function name
from sklearn.linear_model import LogisticRegression

# Consider using a different variable name if you want to keep both models
model_logistic = LogisticRegression()
```

```
[ ]: model_logistic.fit(X_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1339:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

```
[ ]: LogisticRegression()
```

```
[ ]: y_pred = model_logistic.predict(X_test)
y_pred
```

```
[ ]: array([3, 3, 0, 2, 1, 3, 1, 3, 0, 0, 2, 0, 2, 2, 2, 2, 3, 2, 2, 0, 3, 1,
       2, 0, 1, 0, 0, 0, 0, 2, 0, 0, 1, 3, 2, 2, 2, 0, 0, 3, 1, 0, 0, 0,
       2, 0, 0, 3, 0, 2, 0, 0, 0, 3, 0, 0, 3, 0, 0, 0, 3, 2, 2, 0, 0, 2,
       2, 2, 3, 0, 2, 0, 3, 2, 2, 0, 3, 0, 0, 0, 0, 0, 0, 3, 3, 2, 0, 2,
       0, 0, 2, 0, 2, 2, 2, 0, 2, 0, 0, 2, 0, 2, 0, 0, 2, 0, 2, 2, 0, 0,
       3, 0, 0, 0, 0, 2, 0, 0, 0, 0, 3, 0, 2, 0, 2, 0, 2, 2, 3, 0, 2, 0,
       2, 0, 2, 3, 2, 1, 0, 2, 3, 2, 0, 1, 2, 3, 2, 2, 3, 1, 0, 0, 2, 2,
       0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 2, 1, 0, 1, 0, 3, 0, 2, 3, 2, 2, 0,
       0, 0, 2, 0, 0, 0, 0, 3, 0, 2, 1, 0, 0, 0, 0, 2, 0, 2, 0, 0, 2, 2,
       0, 2])
```

```
[ ]: X_train
```

```
[ ]: array([[ 1.2632559 , -0.42895896,  0.49557903],
        [ 0.51927969,  1.41503443, -1.25965087],
        [ 0.52703959,  0.10107796, -1.60739481],
        …,
        [-0.22955004, -1.39828461,  1.13253157],
        [ 0.5178118 , -1.48146136, -0.35322496],
        [ 0.10439668, -1.04785307,  1.4813321 ]])
```

```
[ ]: accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.665
```

```
[ ]: # mae = mean_squared_error(y_test,y_pred)
# r2 = r2_score(y_test,y_pred)
```

[ ]:
```
# print('Mean Squared Error:',mae)
# print('R2 Score:',r2)
```

[ ]:
```
# print(model.intercept_)
# print(model.coef_)
```

[ ]:
```
df
```

[ ]:
```
     Tree_Height  Gleason_Index  Disturbance_Level  Health_Status
0      20.835684       2.998681           0.004402              0
1      24.307079       3.986987           0.000330              3
2       9.013101       3.250667           0.472263              0
3      19.496475       1.333210           0.029294              2
4      18.008888       1.742321           0.974533              2
..           …              …                  …            …
995    14.378997       1.030366           0.317082              2
996    11.363233       3.290682           0.536056              0
997    13.048025       3.868748           0.661191              0
998    16.838336       3.980321           0.703485              3
999     6.518286       4.619908           0.136755              0

[1000 rows x 4 columns]
```

[ ]:
```

```

[ ]:
```
# Disturbance_Level = float(input('Disturbance_Level:'))
# Gleason_Index = float(input('Gleason_Index:'))
# Tree_Heigh = float(input('Tree_Height:'))
# user_input = [[Disturbance_Level,Gleason_Index,Tree_Heigh]]
# std = scaler.transform(user_input)
# health = model_logistic.predict(std)
# print('The Predicted Health:',health)
```

```
Disturbance_Level:6.518286
Gleason_Index:4.619908
Tree_Height:0.136755
The Predicted Health: [2]
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:493: UserWarning: X does
not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

[ ]:
```
# import seaborn as sns
# import matplotlib.pyplot as plt
# sns.regplot(x=y_pred,y=y_test)
# plt.xlabel('Predicted Sales')
# plt.ylabel('Actual Sales')
# plt.title('Actual vs Predicted Sales')
```

```
# plt.show()
```

[ ]:
```
# import pickle
# filename = 'advertising_model.pkl'
# pickle.dump(model,open(filename,'wb'))
```

[ ]:
```
# import pickle

# # Open the file in binary read mode ('rb')
# with open('advertising_model.pkl', 'rb') as f:
#     loaded_model = pickle.load(f)
```