In [31]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [32]:
```python
#Importing CSV File
#Since we are checking our model will be proprely able to detct the room will sold or not we have to use cla
df = pd.read_csv('House-Price.csv', header= 0)
```

In [33]:
```python
df.head()
```

Out[33]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterbod
0	24.0	32.31	0.538	6.575	65.2	4.35	3.81	4.18	4.01	24.7	4.98	YES	5.480	11.1920	Rive
1	21.6	37.07	0.469	6.421	78.9	4.99	4.70	5.12	5.06	22.2	9.14	NO	7.332	12.1728	Lak
2	34.7	37.07	0.469	7.185	61.1	5.03	4.86	5.01	4.97	22.2	4.03	NO	7.394	101.1200	Nor
3	33.4	32.18	0.458	6.998	45.8	6.21	5.93	6.16	5.96	21.3	2.94	YES	9.268	11.2672	Lak
4	36.2	32.18	0.458	7.147	54.2	6.16	5.86	6.37	5.86	21.3	5.33	NO	8.824	11.2896	Lak

DATA PRE-PROCESSING AND CLEANING

In [34]:
```python
# Checking Extended Dictonary(EDD)
#EDD will help to us check the data variation (i.e mean, std, count) and it will alsp help uis to find,
# Missing value and outliers
# we can aslo find outliers with help of (df.info)--> Gives information about dataframe
df.describe()
```

Out[34]:

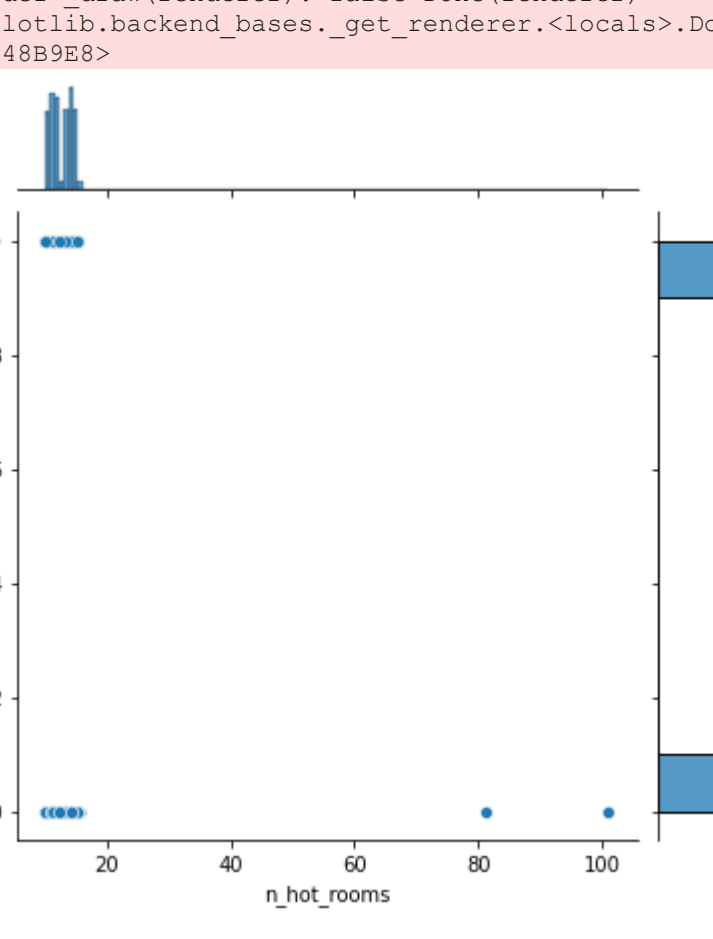
	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	22.528854	41.136779	0.554695	6.284634	68.574901	3.971996	3.628775	3.960672	3.618972	21.544466	12.653063
std	9.182176	6.860353	0.115878	0.702617	28.148861	2.108532	2.108580	2.119797	2.099203	2.164946	7.141062
min	5.000000	30.460000	0.385000	3.561000	2.900000	1.130000	0.920000	1.150000	0.730000	18.000000	1.730000
25%	17.025000	35.190000	0.449000	5.885500	45.025000	2.270000	1.940000	2.232500	1.940000	19.800000	6.950000
50%	21.200000	39.690000	0.538000	6.208500	77.500000	3.385000	3.010000	3.375000	3.070000	20.950000	11.360000
75%	25.000000	48.100000	0.624000	6.623500	94.075000	5.367500	4.992500	5.407500	4.985000	22.600000	16.955000
max	50.000000	57.740000	0.871000	8.780000	100.000000	12.320000	11.930000	12.320000	11.940000	27.400000	37.970000

In [35]:
```python
#Checking Values with help of graph
```

In [36]:
```python
#scatter plot
sns.jointplot(x='n_hot_rooms', y='Sold', data= df)
```

Out[36]: <seaborn.axisgrid.JointGrid at 0x16174040>

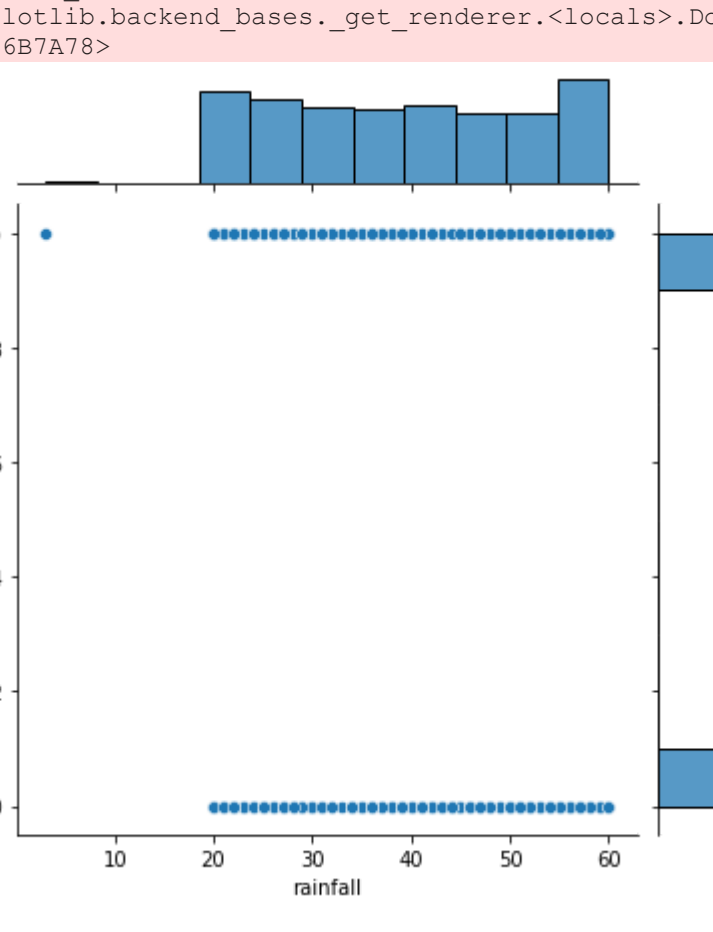
```
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x1648B9E8>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 242, in resize
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x1648B9E8>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x1648B9E8>
```



In [37]:
```python
#scatter plot to check missing value
sns.jointplot(x='rainfall', y='Sold', data = df)
```

Out[37]: <seaborn.axisgrid.JointGrid at 0x160d5bf8>

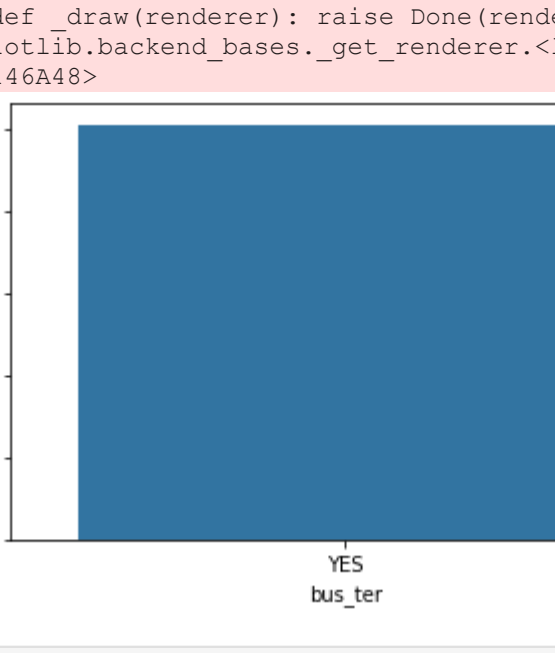
```
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x186B7A78>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 242, in resize
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x186B7A78>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x186B7A78>
```



In [38]:
```python
#count plot (To check catogircal variable)
sns.countplot(x='bus_ter', data = df)
```

Out[38]: <AxesSubplot:xlabel='bus_ter', ylabel='count'>

```
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x16146A48>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 242, in resize
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x16146A48>
Exception in Tkinter callback
Traceback (most recent call last):
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 1883, in __call__
    return self.func(*args)
  File "c:\python 3.8.5\lib\tkinter\__init__.py", line 804, in callit
    func(*args)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\_backend_tk.py", line 253, in idle_draw
    self.draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_tkagg.py", line 9, in draw
    super(FigureCanvasTkAgg, self).draw()
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backends\backend_agg.py", line 407, in draw
    self.figure.draw(self.renderer)
  File "c:\python 3.8.5\lib\site-packages\matplotlib\backend_bases.py", line 1544, in _draw
    def _draw(renderer): raise Done(renderer)
matplotlib.backend_bases._get_renderer.<locals>.Done: <matplotlib.backends.backend_agg.RendererAgg object at 0x16146A48>
```



In [39]:
```python
#Observation  (can be predicted by graph too)
#1) Missing value in n_hos_beds
#2) Outlier present in n_hot_rooms and rainfall
# 3) Bus ha only single catogical variable
```

In [40]:
```python
# Dealing with missing value---> Giving missing values a value which is mean value of that particular column
df.n_hos_beds = df.n_hos_beds.fillna(df.n_hos_beds.mean())
```

In [41]:
```python
df.info() ## after dealing with mising value we have same number of observation in every column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 19 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   price        506 non-null    float64
 1   resid_area   506 non-null    float64
 2   air_qual     506 non-null    float64
 3   room_num     506 non-null    float64
 4   age          506 non-null    float64
 5   dist1        506 non-null    float64
 6   dist2        506 non-null    float64
 7   dist3        506 non-null    float64
 8   dist4        506 non-null    float64
 9   teachers     506 non-null    float64
 10  poor_prop    506 non-null    float64
 11  airport      506 non-null    object
 12  n_hos_beds   506 non-null    float64
 13  n_hot_rooms  506 non-null    float64
 14  waterbody    506 non-null    object
 15  rainfall     506 non-null    int64
 16  bus_ter      506 non-null    object
 17  parks        506 non-null    float64
 18  Sold         506 non-null    int64
dtypes: float64(14), int64(2), object(3)
memory usage: 69.2+ KB
```

In [42]:
```python
# dealing with outliers
# from graph we came to know that in right most part of n_hot_room outlier is presnet

upper_limit = np.percentile(df.n_hot_rooms,[99])[0]
upper_limit
```

Out[42]: 15.399519999999999

In [43]:
```python
#Now Using Cappimg and florring technique to deal with outlier
#checking for upper limit values
df[df.n_hot_rooms > upper_limit]
```

Out[43]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterb
2	34.7	37.07	0.4690	7.185	61.1	5.03	4.86	5.01	4.97	22.2	4.03	NO	7.394	101.12	N
166	50.0	49.58	0.6050	7.929	96.2	2.11	1.91	2.31	1.86	25.3	3.70	YES	8.300	15.40	F
204	50.0	32.68	0.4161	8.034	31.9	5.41	4.80	5.28	4.99	25.3	2.88	YES	8.900	15.40	F
267	50.0	33.97	0.5750	8.297	67.0	2.60	2.13	2.43	2.52	27.0	7.44	YES	8.000	15.40	N
369	50.0	48.10	0.6310	6.683	96.8	1.55	1.28	1.65	0.94	19.8	3.73	YES	6.700	15.40	F
423	13.4	48.10	0.6140	6.103	85.1	2.08	1.80	2.34	1.87	19.8	23.29	NO	8.268	81.12	

In [44]:
```python
df.n_hot_rooms[(df.n_hot_rooms> 3*upper_limit)] = 3*upper_limit
```

```
<ipython-input-44-6cc413d964c9>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df.n_hot_rooms[(df.n_hot_rooms> 3*upper_limit)] = 3*upper_limit
```

In [45]:
```python
df[df.n_hot_rooms > upper_limit] #Now values has been reduced
```

Out[45]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterb
2	34.7	37.07	0.4690	7.185	61.1	5.03	4.86	5.01	4.97	22.2	4.03	NO	7.394	46.19856	N
166	50.0	49.58	0.6050	7.929	96.2	2.11	1.91	2.31	1.86	25.3	3.70	YES	8.300	15.40000	F
204	50.0	32.68	0.4161	8.034	31.9	5.41	4.80	5.28	4.99	25.3	2.88	YES	8.900	15.40000	F
267	50.0	33.97	0.5750	8.297	67.0	2.60	2.13	2.43	2.52	27.0	7.44	YES	8.000	15.40000	N
369	50.0	48.10	0.6310	6.683	96.8	1.55	1.28	1.65	0.94	19.8	3.73	YES	6.700	15.40000	F
423	13.4	48.10	0.6140	6.103	85.1	2.08	1.80	2.34	1.87	19.8	23.29	NO	8.268	46.19856	

In [47]:
```python
#similary dealing with rainfall coulumn.. but on lower side

lower_limit = np.percentile(df.rainfall, [1])[0]
lower_limit
```

Out[47]: 20.0

In [50]:
```python
#Checking values less than 20
df[df.rainfall < lower_limit]
```

Out[50]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterb
213	28.1	40.59	0.489	6.375	32.3	4.11	3.92	4.18	3.57	21.4	9.38	YES	7.562	10.2248	N

In [51]:
```python
#changing the value of outlier by florring and capping method
df.rainfall[(df.rainfall < 0.3*lower_limit)] = 0.3*lower_limit
```

```
<ipython-input-51-78c2414f55f1>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  df.rainfall[(df.rainfall < 0.3*lower_limit)] = 0.3*lower_limit
```

In [52]:
```python
df[df.rainfall < lower_limit]
```

Out[52]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterb
213	28.1	40.59	0.489	6.375	32.3	4.11	3.92	4.18	3.57	21.4	9.38	YES	7.562	10.2248	N

In [54]:
```python
#Deleting  Unwanted coloumns(i.e clumns(data) that will not effect model and which genrally conveys similar
df['dist_avg'] = (df.dist1 + df.dist2 + df.dist3 + df.dist4)/4
```

In [55]:
```python
df.describe()
```

Out[55]:

	price	resid_area	air_qual	room_num	age	dist1	dist2	dist3	dist4	teachers	poor_prop
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	22.528854	41.136779	0.554695	6.284634	68.574901	3.971996	3.628775	3.960672	3.618972	21.544466	12.653063
std	9.182176	6.860353	0.115878	0.702617	28.148861	2.108532	2.108580	2.119797	2.099203	2.164946	7.141062
min	5.000000	30.460000	0.385000	3.561000	2.900000	1.130000	0.920000	1.150000	0.730000	18.000000	1.730000
25%	17.025000	35.190000	0.449000	5.885500	45.025000	2.270000	1.940000	2.232500	1.940000	19.800000	6.950000
50%	21.200000	39.690000	0.538000	6.208500	77.500000	3.385000	3.010000	3.375000	3.070000	20.950000	11.360000
75%	25.000000	48.100000	0.624000	6.623500	94.075000	5.367500	4.992500	5.407500	4.985000	22.600000	16.955000
max	50.000000	57.740000	0.871000	8.780000	100.000000	12.320000	11.930000	12.320000	11.940000	27.400000	37.970000

In [56]:
```python
del df['dist1']
del df['dist2']
del df['dist3']
del df['dist4']
del df['bus_ter']
```

In [58]:
```python
df.head()
```

Out[58]:

	price	resid_area	air_qual	room_num	age	teachers	poor_prop	airport	n_hos_beds	n_hot_rooms	waterbody	rainfall	parks	Sold
0	24.0	32.31	0.538	6.575	65.2	24.7	4.98	YES	5.480	11.19200	River	23	0.049347	0
1	21.6	37.07	0.469	6.421	78.9	22.2	9.14	NO	7.332	12.17280	Lake	42	0.046146	1
2	34.7	37.07	0.469	7.185	61.1	22.2	4.03	NO	7.394	46.19856	None	38	0.045764	0
3	33.4	32.18	0.458	6.998	45.8	21.3	2.94	YES	9.268	11.26720	Lake	45	0.047151	0
4	36.2	32.18	0.458	7.147	54.2	21.3	5.33	NO	8.824	11.28960	Lake	55	0.039474	0

In [59]:
```python
#creating dummy variables
df = pd.get_dummies(df)  #adding dummy varibles to catogical column such that data wpuld be clean are vlear

#Not neccessary since we are only predicting wether the house_property will be sold or not these step will n
# on our data these step is must.
```

In [60]:
```python
df.head()
```

Out[60]:

	price	resid_area	air_qual	room_num	age	teachers	poor_prop	n_hos_beds	n_hot_rooms	rainfall	parks	Sold	dist_avg	airport_N
0	24.0	32.31	0.538	6.575	65.2	24.7	4.98	5.480	11.19200	23	0.049347	0	4.0875	
1	21.6	37.07	0.469	6.421	78.9	22.2	9.14	7.332	12.17280	42	0.046146	1	4.9675	
2	34.7	37.07	0.469	7.185	61.1	22.2	4.03	7.394	46.19856	38	0.045764	0	4.9675	
3	33.4	32.18	0.458	6.998	45.8	21.3	2.94	9.268	11.26720	45	0.047151	0	6.0650	
4	36.2	32.18	0.458	7.147	54.2	21.3	5.33	8.824	11.28960	55	0.039474	0	6.0625	

In [61]:
```python
del df['airport_NO']
del df['waterbody_None']
```

In [62]:
```python
df.head()
```

Out[62]:

	price	resid_area	air_qual	room_num	age	teachers	poor_prop	n_hos_beds	n_hot_rooms	rainfall	parks	Sold	dist_avg	airport_YE
0	24.0	32.31	0.538	6.575	65.2	24.7	4.98	5.480	11.19200	23	0.049347	0	4.0875	
1	21.6	37.07	0.469	6.421	78.9	22.2	9.14	7.332	12.17280	42	0.046146	1	4.9675	
2	34.7	37.07	0.469	7.185	61.1	22.2	4.03	7.394	46.19856	38	0.045764	0	4.9675	
3	33.4	32.18	0.458	6.998	45.8	21.3	2.94	9.268	11.26720	45	0.047151	0	6.0650	
4	36.2	32.18	0.458	7.147	54.2	21.3	5.33	8.824	11.28960	55	0.039474	0	6.0625	

Linear Discriminant Analsysis

In [63]:
```python
X = df.loc[:,df.columns != "Sold"]
```

In [64]:
```python
y = df['Sold']
```

In [65]:
```python
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
```

In [66]:
```python
#setting LDA model
clf_lda = LinearDiscriminantAnalysis()
```

In [67]:
```python
#data fitting
clf_lda.fit(X,y)
```

Out[67]: LinearDiscriminantAnalysis()

In [68]:
```python
y_pred_lda = clf_lda.predict(X)    # basically y_pred_lda ek array ho ga containg zeros and ones
```

In [69]:
```python
y_pred_lda
```

Out[69]:
```
array([1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
       0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,
       0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0,
       1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0,
       1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
       1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0,
       0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1],
      dtype=int64)
```

In [70]:
```python
#Checking Accuracy of model
from sklearn.metrics import accuracy_score, confusion_matrix
```

In [71]:
```python
confusion_matrix(y, y_pred_lda)
```

Out[71]:
```
array([[192,  84],
       [ 79, 151]], dtype=int64)
```

In [72]:
```python
accuracy_score(y , y_pred_lda)    #Model with great accuracy
```

Out[72]: 0.6778656126482213

In [73]:
```python
df['predicted_data'] = y_pred_lda  #You can check by comparing predicted_data column and Sold data
```

In [75]:
```python
df.head(20)
```

Out[75]:

	price	resid_area	air_qual	room_num	age	teachers	poor_prop	n_hos_beds	n_hot_rooms	rainfall	parks	Sold	dist_avg	airport_
0	24.0	32.31	0.538	6.575	65.2	24.7	4.98	5.480	11.19200	23	0.049347	0	4.0875	
1	21.6	37.07	0.469	6.421	78.9	22.2	9.14	7.332	12.17280	42	0.046146	1	4.9675	
2	34.7	37.07	0.469	7.185	61.1	22.2	4.03	7.394	46.19856	38	0.045764	0	4.9675	
3	33.4	32.18	0.458	6.998	45.8	21.3	2.94	9.268	11.26720	45	0.047151	0	6.0650	
4	36.2	32.18	0.458	7.147	54.2	21.3	5.33	8.824	11.28960	55	0.039474	0	6.0625	
5	28.7	32.18	0.458	6.430	58.7	21.3	5.21	7.174	14.22960	53	0.045910	1	6.0600	
6	22.9	37.87	0.524	6.012	66.6	24.8	12.43	6.958	12.18320	41	0.052170	1	5.5600	
7	22.1	37.87	0.524	6.172	96.1	24.8	19.15	5.842	12.17680	56	0.057075	1	5.9500	
8	16.5	37.87	0.524	5.631	100.0	24.8	29.93	5.930	12.13200	55	0.056302	0	6.0825	
9	18.9	37.87	0.524	6.004	85.9	24.8	17.10	9.478	14.15120	45	0.050727	0	6.5900	
10	15.0	37.87	0.524	6.377	94.3	24.8	20.45	6.000	11.12000	29	0.057775	1	6.3475	
11	18.9	37.87	0.524	6.009	82.9	24.8	13.27	9.278	13.15120	23	0.055237	1	6.2250	
12	21.7	37.87	0.524	5.889	39.0	24.8	15.71	5.534	10.17360	57	0.057423	1	5.4525	
13	20.4	38.14	0.538	5.949	61.8	19.0	8.26	5.908	14.16320	39	0.053464	0	4.7075	
14	18.2	38.14	0.538	6.096	84.5	19.0	10.26	6.964	13.14560	49	0.059882	0	4.4650	
15	19.9	38.14	0.538	5.834	56.5	19.0	8.47	8.498	14.15920	28	0.059751	0	4.4975	
16	23.1	38.14	0.538	5.935	29.3	19.0	6.58	5.462	10.18480	46	0.054699	0	4.5000	
17	17.5	38.14	0.538	5.990	81.7	19.0	14.67	5.450	11.14000	56	0.054785	1	4.2600	
18	20.2	38.14	0.538	5.456	36.6	19.0	11.69	8.504	12.16160	41	0.054251	0	3.7950	
19	18.2	38.14	0.538	5.727	69.5	19.0	11.28	8.564	12.14560	27	0.057770	1	3.8000	

In [ ]:
```python
#LDA is having higher predicting accuracy in these datset, so we have used data set
# we can aslo try KNN classsifier and Logistic Regression on these dataset
```