Importing Python libraries

In [1]:

```
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt
import pandas
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
```

--

--

Loading dataset

In [2]:

```
data = pd.read_csv('Occupancy_Estimation.csv')
```

--

Exploratory data analysis

Five top records of data

In [3]:

```
data.head()
```

Out[3]:

	Date	Time	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4_Ligh
0	2017/12/22	10:49:41	24.94	24.75	24.56	25.38	121	34	53	4
1	2017/12/22	10:50:12	24.94	24.75	24.56	25.44	121	33	53	4
2	2017/12/22	10:50:42	25.00	24.75	24.50	25.44	121	34	53	4
3	2017/12/22	10:51:13	25.00	24.75	24.56	25.44	121	34	53	4
4	2017/12/22	10:51:44	25.00	24.75	24.56	25.44	121	34	54	4

Five last records of data

In [4]:

```
data.tail()
```

Out[4]:

	Date	Time	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4
10124	2018/01/11	08:58:07	25.06	25.13	24.69	25.31	6	7	33	
10125	2018/01/11	08:58:37	25.06	25.06	24.69	25.25	6	7	34	
10126	2018/01/11	08:59:08	25.13	25.06	24.69	25.25	6	7	34	
10127	2018/01/11	08:59:39	25.13	25.06	24.69	25.25	6	7	34	
10128	2018/01/11	09:00:09	25.13	25.06	24.69	25.25	6	7	34	

Coloumns/features in data

In [5]:

```
data.columns
```

Out[5]:

```
Index(['Date', 'Time', 'S1_Temp', 'S2_Temp', 'S3_Temp', 'S4_Temp', 'S1_Light',
       'S2_Light', 'S3_Light', 'S4_Light', 'S1_Sound', 'S2_Sound', 'S3_Sound',
       'S4_Sound', 'S5_CO2', 'S5_CO2_Slope', 'S6_PIR', 'S7_PIR',
       'Room_Occupancy_Count'],
      dtype='object')
```

Length of data

In [6]:

```
print('lenght of data is', len(data))
```

```
lenght of data is 10129
```

Shape of data

In [7]:

```
data.shape
```

Out[7]:

```
(10129, 19)
```

Data information

In [8]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10129 entries, 0 to 10128
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Date                  10129 non-null  object 
 1   Time                  10129 non-null  object 
 2   S1_Temp               10129 non-null  float64
 3   S2_Temp               10129 non-null  float64
 4   S3_Temp               10129 non-null  float64
 5   S4_Temp               10129 non-null  float64
 6   S1_Light              10129 non-null  int64  
 7   S2_Light              10129 non-null  int64  
 8   S3_Light              10129 non-null  int64  
 9   S4_Light              10129 non-null  int64  
 10  S1_Sound              10129 non-null  float64
 11  S2_Sound              10129 non-null  float64
 12  S3_Sound              10129 non-null  float64
 13  S4_Sound              10129 non-null  float64
 14  S5_CO2                10129 non-null  int64  
 15  S5_CO2_Slope          10129 non-null  float64
 16  S6_PIR                10129 non-null  int64  
 17  S7_PIR                10129 non-null  int64  
 18  Room_Occupancy_Count  10129 non-null  int64  
dtypes: float64(9), int64(8), object(2)
memory usage: 1.5+ MB
```

Data types of all coloumns

In [9]:

```
data.dtypes
```

Out[9]:

```
Date                        object
Time                        object
S1_Temp                    float64
S2_Temp                    float64
S3_Temp                    float64
S4_Temp                    float64
S1_Light                     int64
S2_Light                     int64
S3_Light                     int64
S4_Light                     int64
S1_Sound                   float64
S2_Sound                   float64
S3_Sound                   float64
S4_Sound                   float64
S5_CO2                       int64
S5_CO2_Slope               float64
S6_PIR                       int64
S7_PIR                       int64
Room_Occupancy_Count         int64
dtype: object
```

Checking Null Values

In [10]:

```
data[data.isnull().any(axis=1)].head()
```

Out[10]:

Date	Time	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Light	S4_Light	S1_Sou

Data Description

In [11]:

```
data.describe()
```

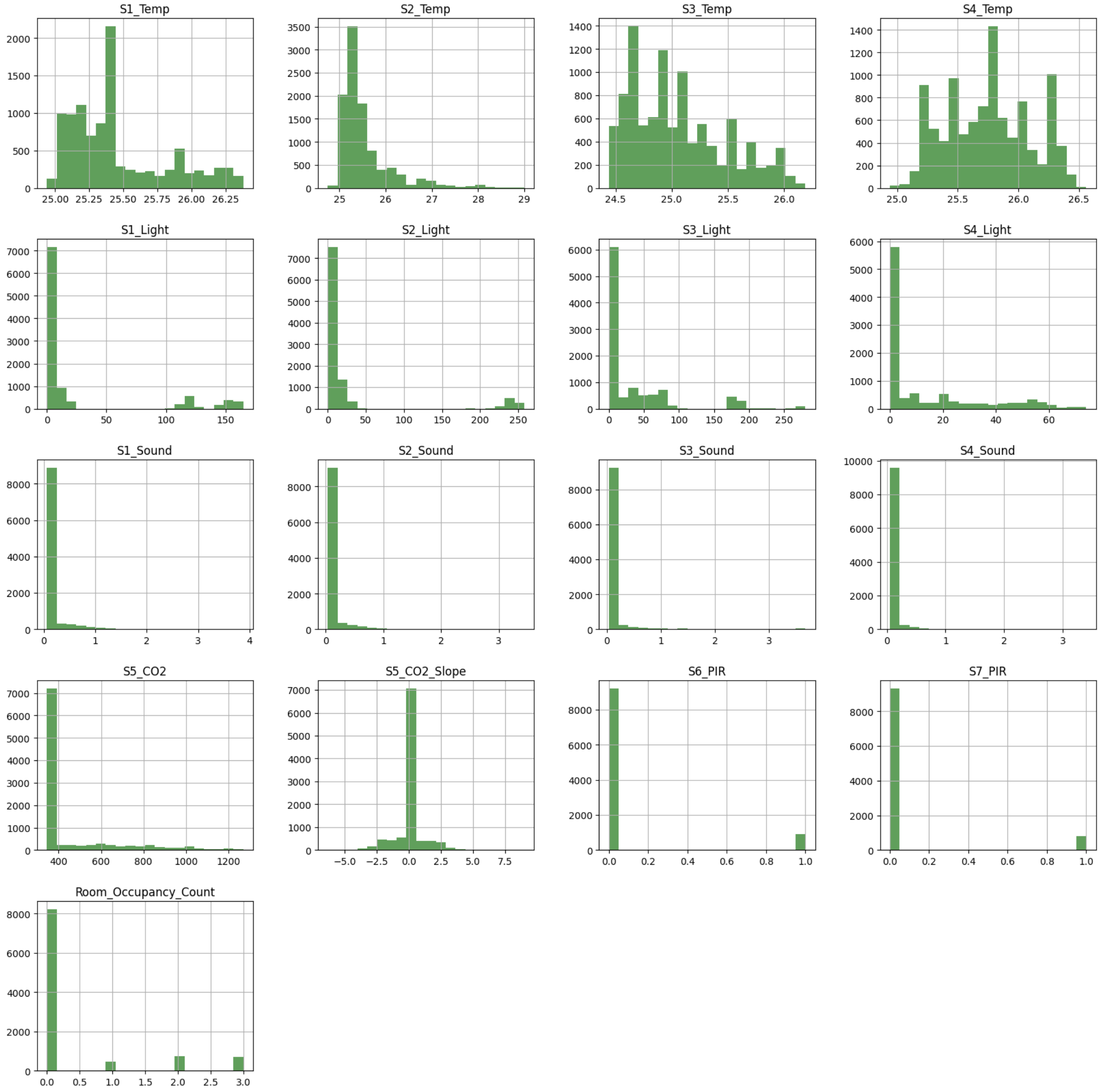
Out[11]:

	S1_Temp	S2_Temp	S3_Temp	S4_Temp	S1_Light	S2_Light	S3_Lig
count	10129.000000	10129.000000	10129.000000	10129.000000	10129.000000	10129.00000	10129.0000
mean	25.454012	25.546059	25.056621	25.754125	25.445059	26.01629	34.2484
std	0.351351	0.586325	0.427283	0.356434	51.011264	67.30417	58.4007
min	24.940000	24.750000	24.440000	24.940000	0.000000	0.00000	0.0000
25%	25.190000	25.190000	24.690000	25.440000	0.000000	0.00000	0.0000
50%	25.380000	25.380000	24.940000	25.750000	0.000000	0.00000	0.0000
75%	25.630000	25.630000	25.380000	26.000000	12.000000	14.00000	50.0000
max	26.380000	29.000000	26.190000	26.560000	165.000000	258.00000	280.0000

Numeric features distrubution

In [12]:

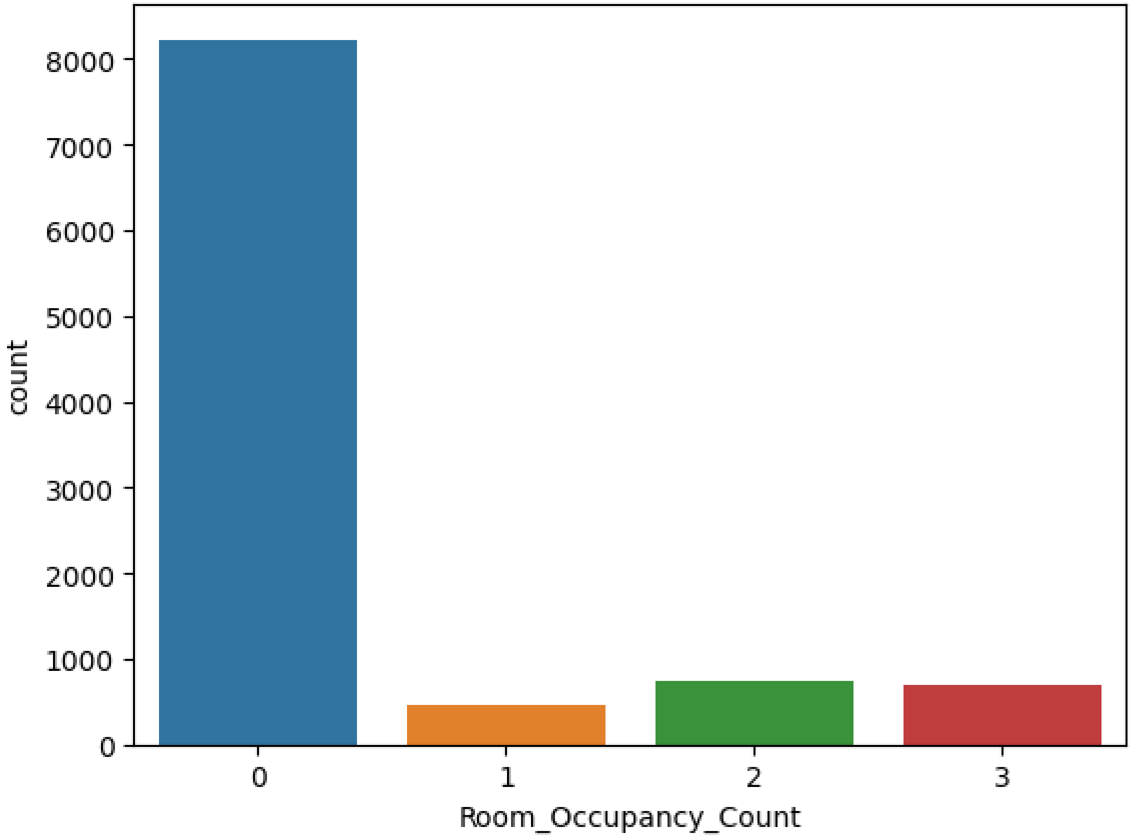
```
data.hist(figsize=(20,20),bins = 20, color="#107009AA")
plt.title("Numeric Features Distribution")
plt.show()
```



Room Occupancy Count

In [13]:

```
sns.countplot(data= data, x = "Room_Occupancy_Count")
plt.show()
```

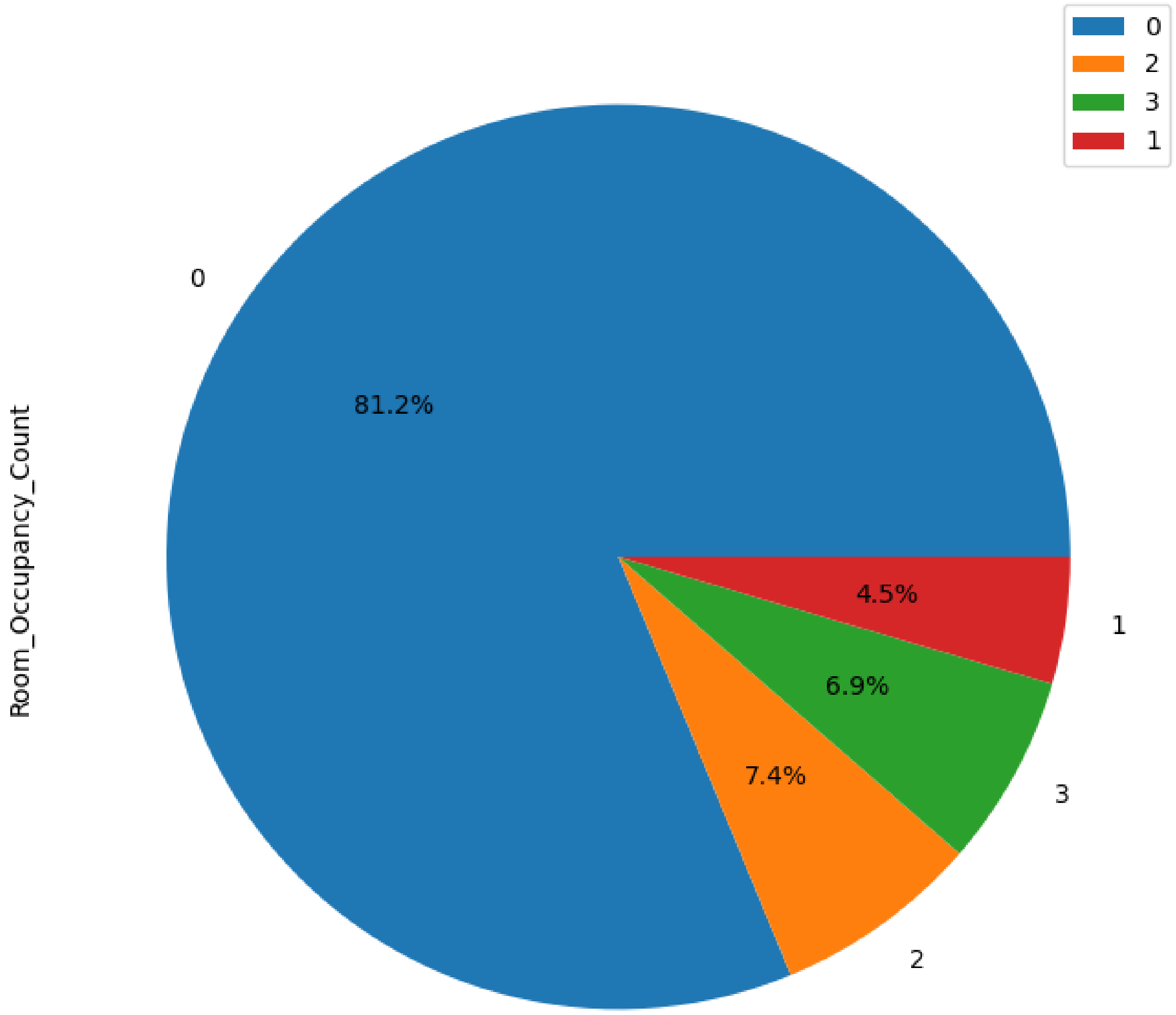


In [14]:

```
data["Room_Occupancy_Count"].value_counts().head(7).plot(kind = 'pie', autopct='%1.1f
```

Out[14]:

```
<matplotlib.legend.Legend at 0x1ec72446b20>
```



By getting features and Class

In [15]:

```
y=data['Room_Occupancy_Count']
X=data.drop(columns=['Room_Occupancy_Count','Date','Time'])
```

Spliting Dataset into 70% Training and 30% Testing

In [16]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state
```

--

----------------------------Machine Learning Algorithms--------------------------------

--

MLP

In [17]:

```
mlp=MLPClassifier()
mlp= mlp.fit(X_train , y_train)
mlp
```

Out[17]:

```
▾ MLPClassifier
MLPClassifier()
```

Accuracy

In [18]:

```
y_pred1 = mlp.predict(X_test)
print('Accuracy score= {:.2f}'.format(mlp.score(X_test, y_test)))
```

```
Accuracy score= 0.99
```

Confusion Matrix

In [19]:

```
print('\n')
print("Confusion Matrix")
print('\n')
CR=confusion_matrix(y_test, y_pred1)
print(CR)
print('\n')
```

```


Confusion Matrix


[[2462    0    0    1]
 [   0  144    1    0]
 [   0    0  214    4]
 [  12    0   17  184]]
```

In progress . . .