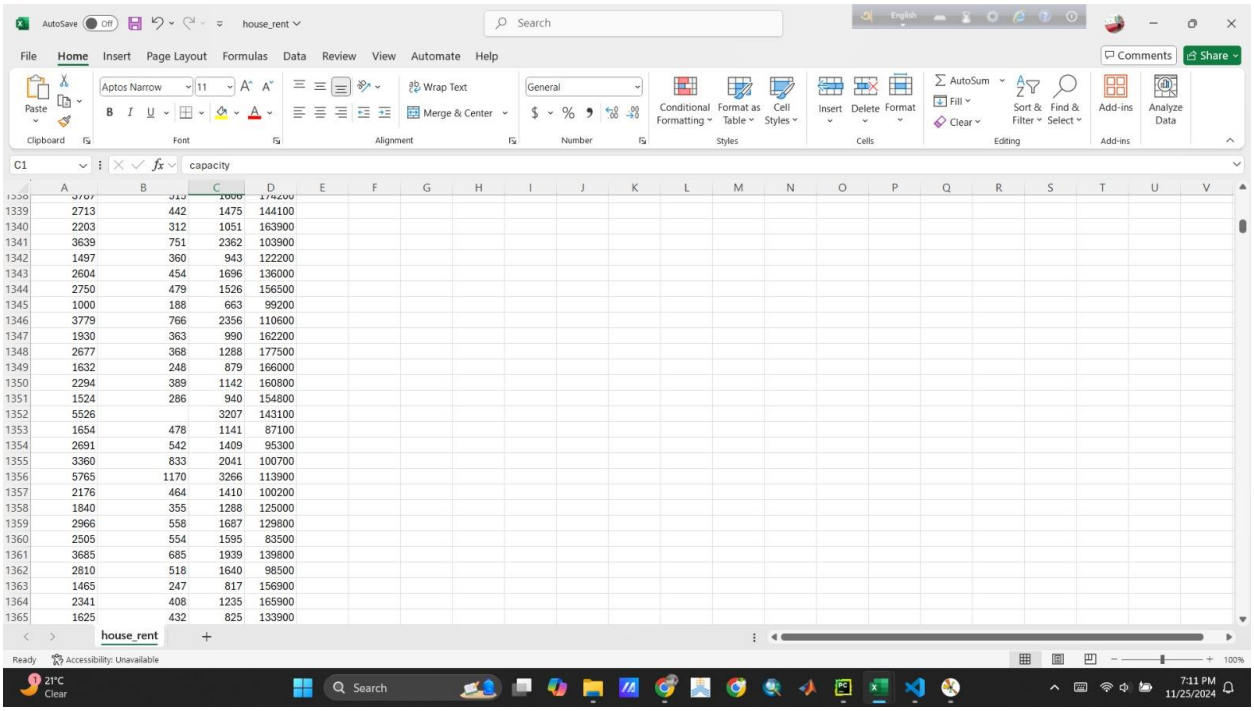
Peoblem_1_Missing data handling



	A	B	C	D
1338	3787	515	1866	174200
1339	2713	442	1475	144100
1340	2203	312	1051	163900
1341	3639	751	2362	103900
1342	1497	360	943	122200
1343	2604	454	1696	136000
1344	2750	479	1526	156500
1345	1000	188	663	99200
1346	3779	766	2356	110600
1347	1930	363	990	162200
1348	2677	368	1288	177500
1349	1632	248	879	166000
1350	2294	389	1142	160800
1351	1524	286	940	154800
1352	5526		3207	143100
1353	1654	478	1141	87100
1354	2691	542	1409	95300
1355	3360	833	2041	100700
1356	5765	1170	3266	113900
1357	2176	464	1410	100200
1358	1840	355	1288	125000
1359	2966	558	1687	129800
1360	2505	554	1595	83500
1361	3685	685	1939	139800
1362	2810	518	1640	98500
1363	1465	247	817	156900
1364	2341	408	1235	165900
1365	1625	432	825	133900

Code for missing data:

```
import pandas as pd

import numpy as np

path=r"E:\aquacrop\house_rent.csv"

df=pd.read_csv(path)

x=df.iloc[:,:].values


from sklearn.impute import SimpleImputer

imp = SimpleImputer(missing_values= np.nan, strategy="mean")

imp= imp.fit(x[:,:])

x[:,:]=imp.fit_transform(x[:,:])

#print(x)

#print(y)
```

```
from sklearn.preprocessing import LabelEncoder

lb=LabelEncoder()

x[:,0]=lb.fit_transform(x[:,0])

print(x)
```

Out Put:

```
[[7.330e+02 1.290e+02 3.220e+02 4.526e+05]
 [5.255e+03 1.106e+03 2.401e+03 3.585e+05]
 [1.316e+03 1.900e+02 4.960e+02 3.521e+05]
 ...
 [2.103e+03 4.850e+02 1.007e+03 9.230e+04]
 [1.709e+03 4.090e+02 7.410e+02 8.470e+04]
 [2.633e+03 6.160e+02 1.387e+03 8.940e+04]]
```

Problem_2_K-Means_Clustering

Code for Clustering:

```
import pandas as pd

import numpy as np

import  matplotlib.pyplot as plt

from sklearn.cluster import KMeans

path=r"E:\aquacrop\house_rent.csv"

df=pd.read_csv(path)

X=df.iloc[:,[1,2]].values


from sklearn.impute import SimpleImputer
```

```
imp = SimpleImputer(missing_values= np.nan, strategy="mean")

imp= imp.fit(X[:,:])

X[:,:]=imp.fit_transform(X[:,:])

from sklearn.preprocessing import LabelEncoder

lb=LabelEncoder()

X[:,0]=lb.fit_transform(X[:,0])

print(X)

#Applying K-Means

kmeans = KMeans(n_clusters= 4, init='k-means++', max_iter=300,
n_init=10, random_state=0)

y_kmeans = kmeans.fit_predict(X)

plt.scatter(X[y_kmeans == 0,0], X[y_kmeans == 0,1], s=100, c='red',
label='Çluster_1')

plt.scatter(X[y_kmeans == 1,0], X[y_kmeans == 1,1], s=100, c='green',
label='Çluster_2')

plt.scatter(X[y_kmeans == 2,0], X[y_kmeans == 2,1], s=100, c='black',
label='Çluster_3')

plt.scatter(X[y_kmeans == 3,0], X[y_kmeans == 3,1], s=100, c='cyan',
label='Çluster_4')

plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s = 300, c = 'Yellow', label='Centroids')

plt.title('The final clustered data')
```

```
plt.xlabel('The X-axis data')

plt.ylabel('The Y-axis data')

plt.legend()

plt.show()
```

Out put: