In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error,mean_squared_error
```

In [2]:

```
# data from magicbricks.com
main = pd.read_csv("House_Rent_Dataset.csv")
main
```

Out[2]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	
0	2022-05-18	2	10000	1100	Ground out of 2	Super Area	Bandel	Kolkata	Unfurnished	Bache
1	2022-05-13	2	20000	800	1 out of 3	Super Area	Phool Bagan, Kankurgachi	Kolkata	Semi-Furnished	Bache
2	2022-05-16	2	17000	1000	1 out of 3	Super Area	Salt Lake City Sector 2	Kolkata	Semi-Furnished	Bache
3	2022-07-04	2	10000	800	1 out of 2	Super Area	Dumdum Park	Kolkata	Unfurnished	Bache
4	2022-05-09	2	7500	850	1 out of 2	Carpet Area	South Dum Dum	Kolkata	Unfurnished	
...	...	...	...	...	...	...	...	...	...	
4741	2022-05-18	2	15000	1000	3 out of 5	Carpet Area	Bandam Kommu	Hyderabad	Semi-Furnished	Bache
4742	2022-05-15	3	29000	2000	1 out of 4	Super Area	Manikonda, Hyderabad	Hyderabad	Semi-Furnished	Bache
4743	2022-07-10	3	35000	1750	3 out of 5	Carpet Area	Himayath Nagar, NH 7	Hyderabad	Semi-Furnished	Bache
4744	2022-07-06	3	45000	1500	23 out of 34	Carpet Area	Gachibowli	Hyderabad	Semi-Furnished	
4745	2022-05-04	2	15000	1000	4 out of 5	Carpet Area	Suchitra Circle	Hyderabad	Unfurnished	

4746 rows × 12 columns

In [3]:

```
main_shuffled = main.sample(frac=1,random_state=42)
main_shuffled
```

Out[3]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	
1566	2022-06-23	2	16000	1100	2 out of 4	Super Area	Jaya Nagar Block 7, Jayanagar	Bangalore	Unfurnished	Bache
3159	2022-05-12	2	12000	800	2 out of 3	Super Area	Manikandan Nagar	Chennai	Furnished	Bache
538	2022-05-10	2	28000	518	5 out of 12	Carpet Area	Bhandup West	Mumbai	Semi-Furnished	Bache
2630	2022-06-08	3	8000	1500	1 out of 1	Carpet Area	Cherlopalli TIRUPATI	Delhi	Unfurnished	Bache
4418	2022-07-06	3	46000	2235	Ground out of 10	Carpet Area	Gachibowli	Hyderabad	Unfurnished	
...	...	...	...	...	...	...	...	...	...	
4426	2022-06-29	3	25000	1500	1 out of 2	Carpet Area	Ayodhya Nagar, Quthbullapur	Hyderabad	Semi-Furnished	
466	2022-06-13	3	20000	1200	3 out of 3	Super Area	Bansdroni	Kolkata	Unfurnished	Bache
3092	2022-07-06	2	20000	800	13 out of 17	Carpet Area	Vadapalani	Chennai	Semi-Furnished	
3772	2022-05-17	3	85000	3500	Ground out of 1	Carpet Area	T Nagar	Chennai	Semi-Furnished	Bache
860	2022-06-20	1	25000	450	5 out of 7	Carpet Area	Goregaon West	Mumbai	Semi-Furnished	

4746 rows × 12 columns

In [4]:

```
# test data
test = main_shuffled.tail(1000)
test
```

Out[4]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	
2467	2022-05-06	3	40000	1650	1 out of 3	Carpet Area	Dakshini Pitampura	Delhi	Semi-Furnished	Bac
1645	2022-05-27	2	12000	950	1 out of 2	Super Area	Nri Layout	Bangalore	Semi-Furnished	Bac
3417	2022-05-20	2	5000	800	Ground out of 2	Super Area	Ayappakkam	Chennai	Unfurnished	Bac
3476	2022-05-09	3	10000	1000	1 out of 8	Carpet Area	Kelambakkam, Old Mahabalipuram Road	Chennai	Semi-Furnished	Bac
3941	2022-06-10	4	25000	1815	2 out of 5	Super Area	Old Bowenpally	Hyderabad	Semi-Furnished	Bac
...	...	...	...	...	...	...	...	...	...	
4426	2022-06-29	3	25000	1500	1 out of 2	Carpet Area	Ayodhya Nagar, Quthbullapur	Hyderabad	Semi-Furnished	
466	2022-06-13	3	20000	1200	3 out of 3	Super Area	Bansdroni	Kolkata	Unfurnished	Bac
3092	2022-07-06	2	20000	800	13 out of 17	Carpet Area	Vadapalani	Chennai	Semi-Furnished	
3772	2022-05-17	3	85000	3500	Ground out of 1	Carpet Area	T Nagar	Chennai	Semi-Furnished	Bac
860	2022-06-20	1	25000	450	5 out of 7	Carpet Area	Goregaon West	Mumbai	Semi-Furnished	

1000 rows × 12 columns

In [5]:

```
training = main_shuffled.drop(test.index)
training
```

Out[5]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	
1566	2022-06-23	2	16000	1100	2 out of 4	Super Area	Jaya Nagar Block 7, Jayanagar	Bangalore	Unfurnished	Bac
3159	2022-05-12	2	12000	800	2 out of 3	Super Area	Manikandan Nagar	Chennai	Furnished	Bac
538	2022-05-10	2	28000	518	5 out of 12	Carpet Area	Bhandup West	Mumbai	Semi-Furnished	Bac
2630	2022-06-08	3	8000	1500	1 out of 1	Carpet Area	Cherlopalli TIRUPATI	Delhi	Unfurnished	Bac
4418	2022-07-06	3	46000	2235	Ground out of 10	Carpet Area	Gachibowli	Hyderabad	Unfurnished	
...	...	...	...	...	...	...	...	...	...	
1854	2022-05-11	2	8900	1100	1 out of 2	Super Area	Sarjapur Road	Bangalore	Semi-Furnished	Bac
4248	2022-06-22	2	12500	1200	2 out of 5	Super Area	Kompally	Hyderabad	Semi-Furnished	Bac
2872	2022-05-30	2	20000	950	2 out of 3	Carpet Area	Geeta Colony PRWS Block 3 And 7, Rajgarh Colony	Delhi	Semi-Furnished	Bac
3030	2022-05-12	2	14000	1000	1 out of 3	Super Area	Kanathur Reddikuppam	Chennai	Unfurnished	Bac
1205	2022-07-09	3	120000	1130	3 out of 7	Carpet Area	Bandra West	Mumbai	Semi-Furnished	Bac

3746 rows × 12 columns

In [6]:

```
train = training.sample(frac=0.8,random_state=42)
train
```

Out[6]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status
46	2022-05-20	2	4600	400	3 out of 3	Carpet Area	Behala	Kolkata	Semi-Furnished
518	2022-06-08	2	45000	1320	Lower Basement out of 2	Super Area	Ballygunge	Kolkata	Furnished
3930	2022-07-06	3	75000	2395	14 out of 17	Carpet Area	Vittal Rao Nagar, Hitech City	Hyderabad	Semi-Furnished
201	2022-06-24	1	10000	700	1 out of 2	Carpet Area	Garia	Kolkata	Unfurnished
3520	2022-07-10	3	65000	1444	11 out of 14	Super Area	Nungambakkam	Chennai	Semi-Furnished
...	...	...	...	...	...	...	...	...	...
2224	2022-06-27	1	6500	450	Ground out of 2	Carpet Area	Garden City University	Bangalore	Semi-Furnished
2293	2022-05-23	1	4000	250	Ground out of 2	Super Area	Vijayanagar	Bangalore	Unfurnished
1814	2022-06-22	2	9750	575	Ground out of 2	Carpet Area	Varanasi	Bangalore	Semi-Furnished
1950	2022-05-23	2	12000	650	2 out of 4	Super Area	Bommanahalli	Bangalore	Semi-Furnished
4037	2022-05-06	2	10000	1090	2 out of 4	Super Area	Hyder Shah Kote, Chevella Road	Hyderabad	Unfurnished

2997 rows × 12 columns

In [7]:

```
validation = training.drop(train.index)
validation
```

Out[7]:

	Posted On	BHK	Rent	Size	Floor	Area Type	Area Locality	City	Furnishing Status	
3159	2022-05-12	2	12000	800	2 out of 3	Super Area	Manikandan Nagar	Chennai	Furnished	Bac
4418	2022-07-06	3	46000	2235	Ground out of 10	Carpet Area	Gachibowli	Hyderabad	Unfurnished	
4080	2022-06-18	2	12000	500	1 out of 2	Super Area	Saidabad Colony, Sayeedabad	Hyderabad	Furnished	Bac
3792	2022-07-06	4	140000	2600	2 out of 2	Carpet Area	Ranjith Road	Chennai	Semi-Furnished	Bac
2323	2022-07-05	3	60000	1850	3 out of 3	Carpet Area	Halasuru	Bangalore	Furnished	Bac
...	...	...	...	...	...	...	...	...	...	
4727	2022-06-28	3	30000	214	2 out of 2	Super Area	Jubilee Hills	Hyderabad	Furnished	
3878	2022-06-28	4	75000	3800	3 out of 10	Carpet Area	Financial District, Nanakram Guda	Hyderabad	Semi-Furnished	Bac
2360	2022-05-05	3	24000	1400	2 out of 4	Super Area	Jalahalli, Jalahalli, Outer Ring Road	Bangalore	Semi-Furnished	Bac
3502	2022-06-29	3	60000	1800	9 out of 9	Super Area	in Saligramam	Chennai	Semi-Furnished	
3030	2022-05-12	2	14000	1000	1 out of 3	Super Area	Kanathur Reddikuppam	Chennai	Unfurnished	Bac

749 rows × 12 columns

Number of rows in training, validation and test sets, along with the structure, datatypes and value counts of the dataframes

In [8]:

```
print(train.dtypes)
print("The number of rows and columns are :",len(train),",",len(train.columns))
```

```
Posted On            object
BHK                   int64
Rent                  int64
Size                  int64
Floor                object
Area Type            object
Area Locality        object
City                 object
Furnishing Status    object
Tenant Preferred     object
Bathroom              int64
Point of Contact     object
dtype: object
The number of rows and columns are : 2997 , 12
```

In [9]:

```
print(test.dtypes)
print("The number of rows and columns are :",len(test),",",len(test.columns))
```

```
Posted On            object
BHK                   int64
Rent                  int64
Size                  int64
Floor                object
Area Type            object
Area Locality        object
City                 object
Furnishing Status    object
Tenant Preferred     object
Bathroom              int64
Point of Contact     object
dtype: object
The number of rows and columns are : 1000 , 12
```

In [10]:

```
print(validation.dtypes)
print("The number of rows and columns are :",len(validation),",",len(validation.columns))
```

```
Posted On            object
BHK                   int64
Rent                  int64
Size                  int64
Floor                object
Area Type            object
Area Locality        object
City                 object
Furnishing Status    object
Tenant Preferred     object
Bathroom              int64
Point of Contact     object
dtype: object
The number of rows and columns are : 749 , 12
```

Data Cleaning:

- Analyse the data and identify which columns are not relevant for house rent prediction task. Drop those columns from the dataframes.*

In [11]:

```
train.drop(['Posted On','Area Type','Area Locality','Tenant Preferred','Point of Contact'
```

Out[11]:

	BHK	Rent	Size	Floor	City	Furnishing Status	Bathroom
46	2	4600	400	3 out of 3	Kolkata	Semi-Furnished	1
518	2	45000	1320	Lower Basement out of 2	Kolkata	Furnished	2
3930	3	75000	2395	14 out of 17	Hyderabad	Semi-Furnished	3
201	1	10000	700	1 out of 2	Kolkata	Unfurnished	1
3520	3	65000	1444	11 out of 14	Chennai	Semi-Furnished	3
...	...	...	...	...	...	...	...
2224	1	6500	450	Ground out of 2	Bangalore	Semi-Furnished	1
2293	1	4000	250	Ground out of 2	Bangalore	Unfurnished	1
1814	2	9750	575	Ground out of 2	Bangalore	Semi-Furnished	2
1950	2	12000	650	2 out of 4	Bangalore	Semi-Furnished	2
4037	2	10000	1090	2 out of 4	Hyderabad	Unfurnished	2

2997 rows × 7 columns

Check for missing values and logically impute the dataset.

In [12]:

```
print("FOR NA VALUE:\n",train.isna().sum(),'\n')
print("FOR NULL VALUE:\n",train.isnull().sum(),'\n')
print("From above we can derive that there is no null value and no Na Values, therefore t
```

```
FOR NA VALUE:
 Posted On            0
BHK                   0
Rent                  0
Size                  0
Floor                 0
Area Type             0
Area Locality         0
City                  0
Furnishing Status     0
Tenant Preferred      0
Bathroom              0
Point of Contact      0
dtype: int64

FOR NULL VALUE:
 Posted On            0
BHK                   0
Rent                  0
Size                  0
Floor                 0
Area Type             0
Area Locality         0
City                  0
Furnishing Status     0
Tenant Preferred      0
Bathroom              0
Point of Contact      0
dtype: int64

From above we can derive that there is no null value and no Na Values, ther
efore there is no need of logical imputation
```

Identify any categorical valued columns (non-numeric) and convert them to numeric.

In [13]:

```
print("Not Found")
```

```
Not Found
```

Exploratory Analysis (On training set):

Plot the house rents against the dependent variable of “size”. See if there is a uniform linear trend between the dependent and independent variables. Make accurate axis and legend. Save the plot in a png file.

In [14]:

```
df = train[['Size','Rent']]
df=df.tail(1000)
df.plot(kind='scatter',x='Size',y='Rent',color='cyan')
plt.title('ScatterPlot of Rent vs Size')
plt.legend(["Rent for the x Size Room"])
plt.show()
```

Find average rent prices in different cities and report which city has the highest average rent.

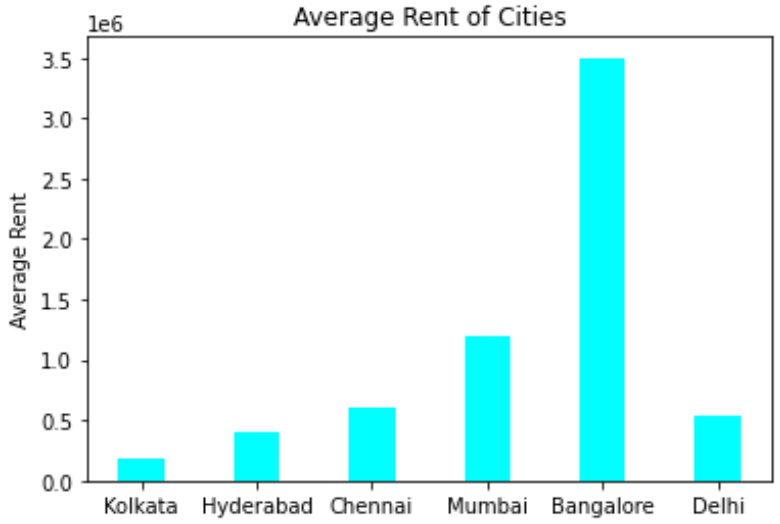
In [15]:

```
df1 = train[['Rent','City']]
m=df1.groupby(["City"],sort=True)['Rent'].mean()
print(m)
print(m.max())
x = train['Rent']
y = train['City']
plt.bar(y,x,color='cyan',width=0.4)
plt.title("Average Rent of Cities")
plt.ylabel("Average Rent")
```

```
City
Bangalore    27585.763060
Chennai      22145.766372
Delhi        30994.748031
Hyderabad    20448.101266
Kolkata      11909.943953
Mumbai       85821.845907
Name: Rent, dtype: float64
85821.84590690209
```

Out[15]:

```
Text(0, 0.5, 'Average Rent')
```



Regression :

In [16]:

```
train_part = train[['BHK','Rent','Bathroom']]
train_part
```

Out[16]:

	BHK	Rent	Bathroom
46	2	4600	1
518	2	45000	2
3930	3	75000	3
201	1	10000	1
3520	3	65000	3
...	...	...	...
2224	1	6500	1
2293	1	4000	1
1814	2	9750	2
1950	2	12000	2
4037	2	10000	2

2997 rows × 3 columns

In [17]:

```
model = LinearRegression()
```

Train a linear regression model on the training set partition by taking only one dependent variable of "size". Calculate the error on the validation set

In [18]:

```
model.fit(train_part,train['Size'])
```

Out[18]:

```
LinearRegression()
```

Plot the model predictions of rent values alongside the actual rent values taken for the validation set.Show the legend, axes and color-coded predictions and ground truth for differentiating

In [19]:

```
test_prediction = model.predict(test[['BHK','Rent','Bathroom']])
# print(test_prediction)
df2 = pd.DataFrame(data=test_prediction,columns=["Predicted Rent"])
df2
```

Out[19]:

	Predicted Rent
0	1243.710957
1	933.428892
2	928.085563
3	1495.771253
4	1796.129994
...	...
995	1507.221243
996	1228.444304
997	939.535553
998	1553.021204
999	654.443466

1000 rows × 1 columns

In [20]:

```
df3=train[['Rent']].tail(1000)
df3
```

Out[20]:

	Rent
4518	35000
1472	20000
265	8000
1461	42000
2340	120000
...	...
2224	6500
2293	4000
1814	9750
1950	12000
4037	10000

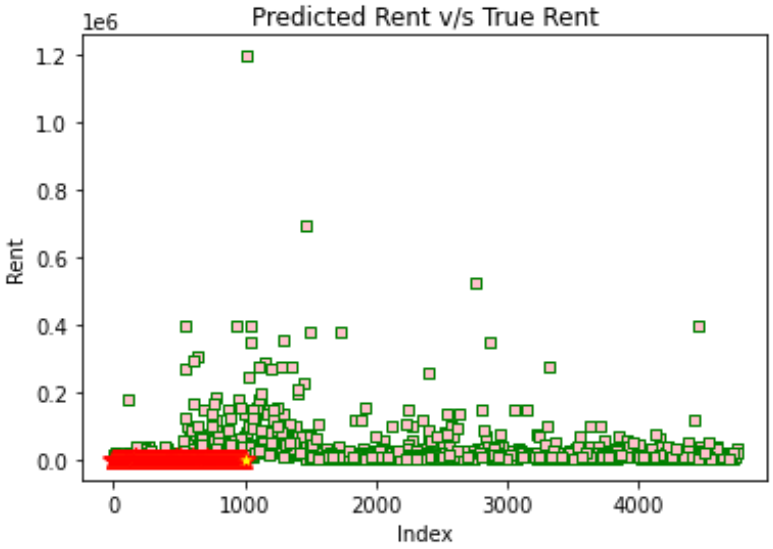
1000 rows × 1 columns

In [21]:

```
df3Idx=df3.index.values
# print(df3Idx)
df2Idx=df2.index.values
# print(df2Idx)
```

In [22]:

```
plt.scatter(y=df3['Rent'],x=df3Idx,c="pink",linewidths=1,marker="s",edgecolor="green",s=2
plt.scatter(y=df2['Predicted Rent'],x=df2Idx,c="yellow",linewidths=1,marker="*",edgecolor
plt.title("Predicted Rent v/s True Rent")
plt.xlabel("Index")
plt.ylabel("Rent")
plt.show()
```



In [23]:

```
mae = mean_absolute_error(test['Size'],test_prediction)
mae
```

Out[23]:

```
272.23283544227377
```

In [24]:

```
mse = mean_squared_error(test['Size'],test_prediction)
mse
```

Out[24]:

```
184862.6749401498
```

Create a function for calculating the RMSE values for the predictions Vs the actual ground truth rent values. RMSE = SQRT(Σ ((F(xi) - yi)2)/N), Here F(x) are the prediction values, N are the number of rows

In [25]:

```
rmse = np.sqrt(mse)
rmse
```

Out[25]:

```
429.9565965770845
```