CONTENTS

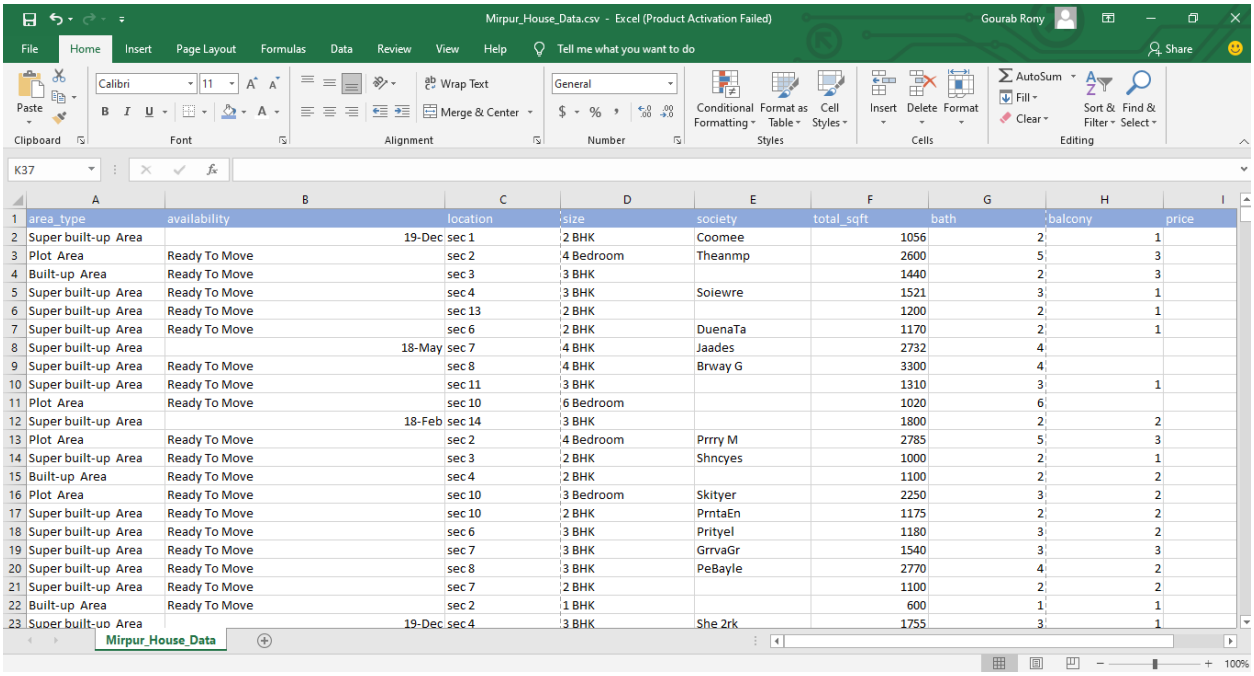
Chapter 1

1.1 Introduction

Home Rent Prediction System is an AI-based program that can predict the house price by analyzing the location, size of the room, and the total amount of bed-room. Here, we used a dataset that contains various information about houses including house prices. Our program predicts rent by analyzing the used dataset. We used Python programming language which is one of the best high level programming language for Artificial Intelligence.

1.2 Dataset

Our project Home Rent Prediction System contains a dataset named Mirpur_House_Data which is in CSV (comma separated values) format. because, CSV is a simple file format used to store tabular data, such as a spreadsheet or database. CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record.



	A	B	C	D	E	F	G	H	I
1	area_type	availability	location	size	society	total_sqft	bath	balcony	price
2	Super built-up Area	19-Dec	sec 1	2 BHK	Coomee	1056	2	1	
3	Plot Area	Ready To Move	sec 2	4 Bedroom	Theanmp	2600	5	3	
4	Built-up Area	Ready To Move	sec 3	3 BHK		1440	2	3	
5	Super built-up Area	Ready To Move	sec 4	3 BHK	Soiewre	1521	3	1	
6	Super built-up Area	Ready To Move	sec 13	2 BHK		1200	2	1	
7	Super built-up Area	Ready To Move	sec 6	2 BHK	DuenaTa	1170	2	1	
8	Super built-up Area	18-May	sec 7	4 BHK	Jaades	2732	4		
9	Super built-up Area	Ready To Move	sec 8	4 BHK	Brway G	3300	4		
10	Super built-up Area	Ready To Move	sec 11	3 BHK		1310	3	1	
11	Plot Area	Ready To Move	sec 10	6 Bedroom		1020	6		
12	Super built-up Area	18-Feb	sec 14	3 BHK		1800	2	2	
13	Plot Area	Ready To Move	sec 2	4 Bedroom	Prrry M	2785	5	3	
14	Super built-up Area	Ready To Move	sec 3	2 BHK	Shncyes	1000	2	1	
15	Built-up Area	Ready To Move	sec 4	2 BHK		1100	2	2	
16	Plot Area	Ready To Move	sec 10	3 Bedroom	Skityer	2250	3	2	
17	Super built-up Area	Ready To Move	sec 10	2 BHK	PrntaEn	1175	2	2	
18	Super built-up Area	Ready To Move	sec 6	3 BHK	Prityel	1180	3	2	
19	Super built-up Area	Ready To Move	sec 7	3 BHK	GrrvaGr	1540	3	3	
20	Super built-up Area	Ready To Move	sec 8	3 BHK	PeBayle	2770	4	2	
21	Super built-up Area	Ready To Move	sec 7	2 BHK		1100	2	2	
22	Built-up Area	Ready To Move	sec 2	1 BHK		600	1	1	
23	Super built-up Area	19-Dec	sec 4	3 BHK	She 2rk	1755	3	1	

Figure 1: Mirpur house data

1.2.1 Area_type

Area_type is the first column in our dataset. It shows us different types of areas of the location. We have four types of areas in our dataset, Super built-up Area, Plot Area, Built-up Area, and Carpet Area.

```
df1['area_type'].unique()
```

```
array(['Super built-up  Area', 'Plot  Area', 'Built-up  Area',
       'Carpet  Area'], dtype=object)
```

Figure 2: area_type

1.2.2 Availability

Availability is the second column in our dataset. It shows us the current status of houses which are available for rent or not.

1.2.3 Location

Location is the third column in our dataset. It shows us the location of the houses. We have a total of fourteen unique locations in our dataset.

```
df1['location'].unique()
```

```
array(['sec 1', 'sec 2', 'sec 3', 'sec 4 ', 'sec 13', 'sec 6', 'sec 7',
       'sec 8', 'sec 11', 'sec 10', 'sec 14', 'sec6', 'sec13', 'sec 12'],
      dtype=object)
```

Figure 3: location

1.2.4 Size

Size is the fourth column in our dataset. It shows us the size of houses in BHK (bedroom, hall, and kitchen) format.

```
df1['size'].head(10)
```

```
0        2 BHK
1    4 Bedroom
2        3 BHK
3        3 BHK
4        2 BHK
5        2 BHK
6        4 BHK
7        4 BHK
8        3 BHK
9    6 Bedroom
Name: size, dtype: object
```

Figure 4: size

1.2.5 Society

Society is the fifth column in our dataset. It shows us different types of the society of houses.

1.2.6 Total_sqft

Total_sqft is the sixth column in our dataset. It shows us the total size of houses in the square feet format.

```
df1['total_sqft'].head(10)
```

```
0    1056
1    2600
2    1440
3    1521
4    1200
5    1170
6    2732
7    3300
8    1310
9    1020
Name: total_sqft, dtype: object
```

Figure 5: total_sqft

1.2.7 Bath

Bath is the seventh column of our dataset. It shows us the total number of baths that houses contain.

1.2.8 Balcony

Balcony is the eighth column of our dataset. It shows us the total number of balconies that houses contain.

1.2.9 Price

Price is the last column in our dataset. It shows us the rent of houses.

```
df1['price'].head(10)
```

```
0    20000
1    20000
2    21000
3    12000
4    12500
5    13000
6    13000
7    18000
8    10500
9    13000
Name: price, dtype: int64
```

Figure 6: Price

1.3 Problem Statement

In Bangladesh's capital city Dhaka, searching for houses that are available for rent is a very hard-working job for the tenant. There are many websites that help them to make their job easy, but it is not worth it. Because tenant looks for houses by their rent and number of the room. The existing system is not helping the tenant with their need.

1.4 Motivation

A better idea is to use the system which can improve the current system and help the tenants. This system is providing each entity the facility to approach the tenant by predicts the rent so that it will become much easier to find houses which is available for rent.

1.5 Summary

To find a house which is available for rent is a very hard-working job. Our system can make the job a little easy. And tenants can save a lot of time to find their beautiful house.

Chapter 2

2.1 Background

2.1.1 Introduction

The majority of the people in Dhaka have come to Dhaka for a variety of economic and social reasons. They are staying and working in Dhaka permanently

without a home of their own. The experiences and the house ownership process of the tenants generate knowledge and present a guideline for the policymakers and planners. According to research, about 68% of people who lived in Dhaka are tenants. So, it's very difficult for them to search for available houses on their own without any guidelines.

2.2 Literal Review

This work provides a literature review of the house finding process. We aim to help the tenant to save their time, make easy their job and improve the current system. With this system, tenants can find their house easier way and fast. In this system when tenants search by house square feet size, the number of rooms, and a number of bathrooms, it shows the suitable rent. So, our system can help the tenants.

2.3 Problem Analysis

In Dhaka, there are 60% to 65% of people are tenants. But the number of house owners is low who are willing to give rent their house. It is less than 20%. So, it's very hard to find a house for tenants. Tenants have to go door to door for searching the house or searching a house on different websites. On websites, tenants have only one option for finding the houses. it's the location option. So, it's a very stressful job to find a favorable house at an affordable price. Sometimes they cannot find any and waste their time.

2.4 Summary

Dhaka is the capital of Bangladesh with a population of 8,906,039. Every day thousands of people come here looking for hope. Our system is made for those hopeful tenants.

Chapter 3

Proposed Model, Milestone, Schedule, DFD, Use Case Diagram, ER-Diagram

3.1 Proposed Model

The scope of the project is that in a very short span provides the user with many facilities. It provides a very accurate prediction with an accuracy of more than 60%. So, users can trust it. It shows users the predicted rent when they give the location, total square feet of house, total bedroom, and total bath as input. The main purpose of this project is to reduce users' work to find their houses which are available for rent. This system predicts the rent by analyzing the given

dataset used in the system. We modified the dataset so that the system can give the user accurate prediction.

3.2 Milestone

- To reduces the users' work as much as possible to search houses which are available for rent.
- To use a statistical dataset which can give us all information about the house rent.
- To make the system as simple as possible
- To use an algorithm that can give us our rent prediction result

3.3 Schedule

	Task Name	Duration	Start	ETA
1	Complete project execution	35 days	16.02.21	21.03.20
2	Engineering	35 days	16.02.21	21.03.21
3	Project examination	1 day	16.02.21	16.02.21
4	Material specification	2 days	17.02.21	18.02.21
5	Material ordering	1 day	19.02.21	19.02.21
6	Equipment layouting	3 days	20.02.21	24.02.21
7	Supervision and meetings	3 days	25.02.12	28.02.21
8	Preparing distribution member	1 day	1.03.21	1.03.21
9	Project examination	1 day	3.03.21	3.03.21
10	Work distribution	2 days	6.03.21	8.03.21
11	Implementation	7 days	9.03.21	15.03.21
12	Testing	3 days	16.03.21	18.03.21
13	Final Testing	1 days	19.05.12	19.03.21
14	Documentation	2 days	18.03.21	20.03.21
15	Project Review	1 day	21.03.21	21.03.21

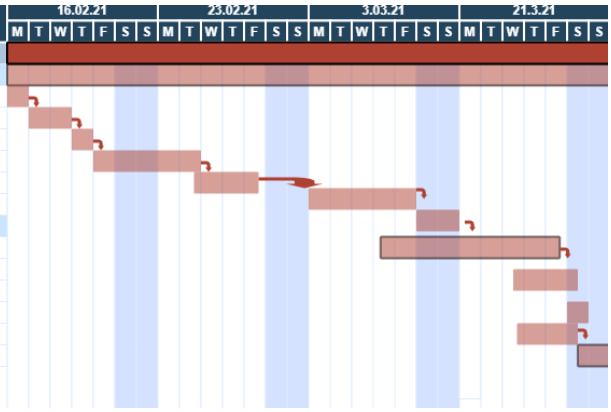


Figure 7: Schedule

3.4 DFD (Data Flow Diagram)

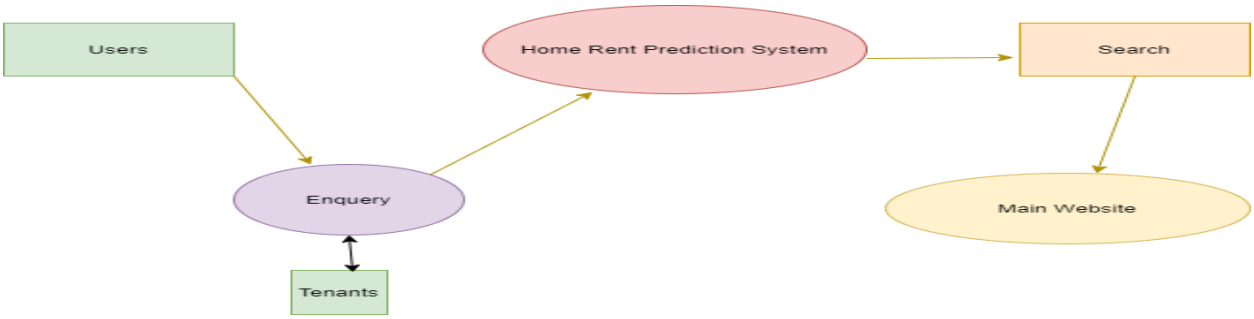


Figure 8: DFD

3.5 Use Case Diagram

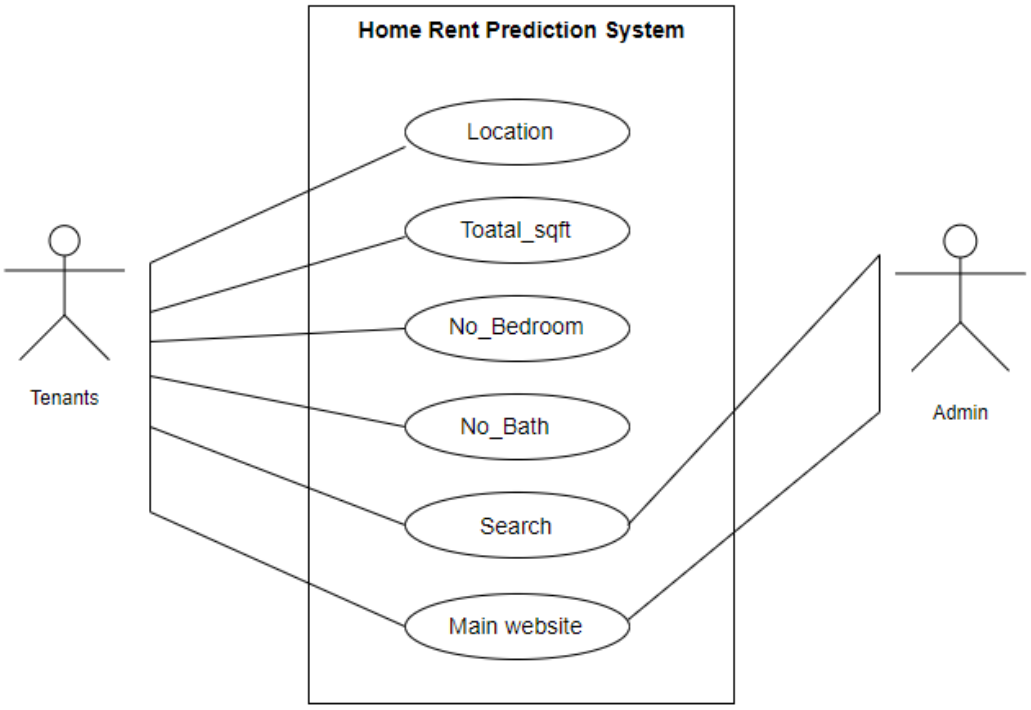


Figure 9: Use case diagram

3.6 ER-Diagram

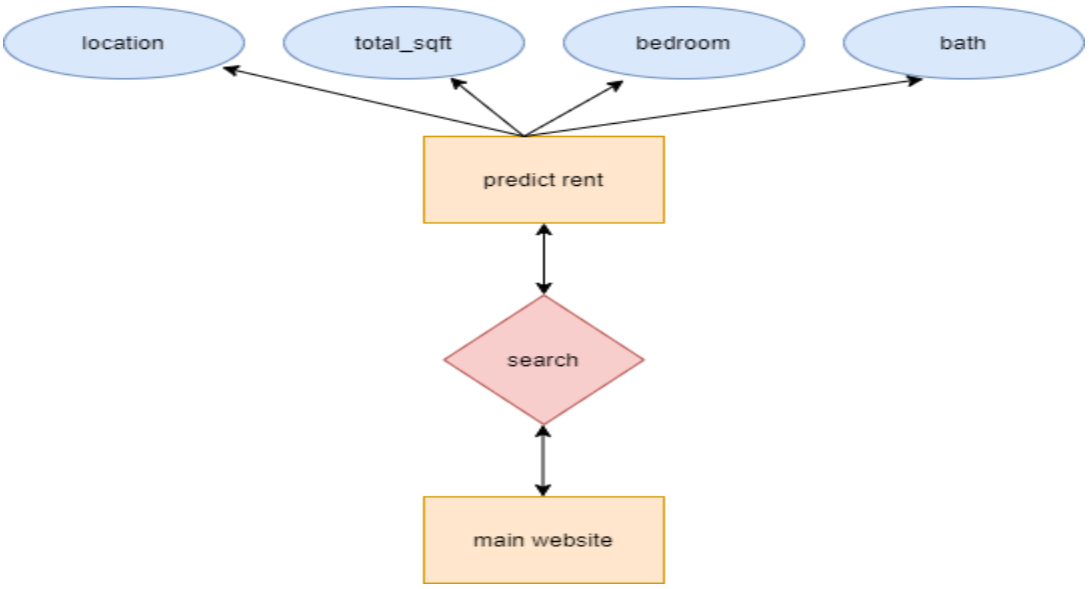


Figure 10: ER-diagram

Chapter 4

Implementation and Result

4.1 Implementation

Implementation is a basic expression of any quit project. To implement our project, we want a development model. We choosing the Linear Regression algorithm. Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Hypothesis function for Linear

Regression: $y = \theta_1 + \theta_2 * x$

Here,
x: input
y: labels
θ_1: intercept
θ_2: coefficient of x

4.1.1 Code

```
import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

%matplotlib inline

import matplotlib

matplotlib.rcParams["figure.figsize"] = (20,10)

df1 = pd.read_csv("Book2.csv")

df1.head()

df1.shape

df1.columns

df1['area_type'].head(10)

df1['area_type'].value_counts()

df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')
```

```
df2.shape
df2.isnull().sum()
df2.shape
df3 = df2.dropna()
df3.isnull().sum()
df3.shape
df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
df3.bhk.unique()
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
df3[~df3['total_sqft'].apply(is_float)].head(10)
def convert_sqft_to_num(x):
    tokens = x.split('-')
    if len(tokens) == 2:
        return (float(tokens[0])+float(tokens[1]))/2
    try:
        return float(x)
    except:
        return None
df4 = df3.copy()
df4.total_sqft = df4.total_sqft.apply(convert_sqft_to_num)
df4 = df4[df4.total_sqft.notnull()]
df4.head(2)
```

```
df4.loc[30]
df5 = df4.copy()
df5['price_per_sqft'] = df5['price']/df5['total_sqft']
df5.head()
df5_stats = df5['price_per_sqft'].describe()
df5_stats
df5.to_csv("bhp.csv",index=False)
df5.location = df5.location.apply(lambda x: x.strip())
location_stats = df5['location'].value_counts(ascending=False)
location_stats
location_stats.values.sum()
len(location_stats[location_stats>500])
len(location_stats)
len(location_stats[location_stats<=500])
location_stats_less_than_500 = location_stats[location_stats<=500]
location_stats_less_than_500
len(df5.location.unique())
df5.location = df5.location.apply(lambda x: 'other' if x in location_stats_less_than_500 else x)
len(df5.location.unique())
df5.head(10)
df5[df5.total_sqft/df5.bhk<300].head()
df5.shape
df6 = df5[~(df5.total_sqft/df5.bhk<300)]
df6.shape
df6.price_per_sqft.describe()
def remove_pps_outliers(df):
```

```
    df_out = pd.DataFrame()
    for key, subdf in df.groupby('location'):
        m = np.mean(subdf.price_per_sqft)
        st = np.std(subdf.price_per_sqft)
        reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=(m+st))]
        df_out = pd.concat([df_out,reduced_df],ignore_index=True)
    return df_out
df7 = remove_pps_outliers(df6)
df7.shape
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+', color='green',label='3 BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price (Hazar Bangladesh Taka)")
    plt.title(location)
    plt.legend()

plot_scatter_chart(df7,"sec 2")
plot_scatter_chart(df7,"sec 10")
def remove_bhk_outliers(df):
    exclude_indices = np.array([])
    for location, location_df in df.groupby('location'):
```

```
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indices, bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')
df8 = remove_bhk_outliers(df7)
# df8 = df7.copy()
df8.shape
plot_scatter_chart(df8,"sec 2")
plot_scatter_chart(df8,"sec 10")
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
plt.hist(df8.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
df8.bath.unique()
plt.hist(df8.bath,rwidth=0.8)
plt.xlabel("Number of bathrooms")
plt.ylabel("Count")
```

```
df8[df8.bath>10]
df8[df8.bath>df8.bhk+2]
df9 = df8[df8.bath<df8.bhk+2]
df9.shape
df9.head(2)
df10 = df9.drop(['size','price_per_sqft'],axis='columns')
df10.head(3)
dummies = pd.get_dummies(df10.location)
dummies.head(3)
df11 = pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns')
df11.head()
df12 = df11.drop('location',axis='columns')
df12.head(2)
df12.shape
X = df12.drop(['price'],axis='columns')
X.head(3)
X.shape
y = df12.price
y.head(3)
len(y)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=10)
from sklearn.linear_model import LinearRegression
lr_clf = LinearRegression()
lr_clf.fit(X_train,y_train)
lr_clf.score(X_test,y_test)
```

```
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score

cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

cross_val_score(LinearRegression(), X, y, cv=cv)
def predict_price(location,sqft,bath,bhk):
    loc_index = np.where(X.columns==location)[0][0]

    x = np.zeros(len(X.columns))
    x[0] = sqft
    x[1] = bath
    x[2] = bhk
    if loc_index >= 0:
        x[loc_index] = 1

    return lr_clf.predict([x])[0]

predict_price('sec 2',1000, 2, 2)
predict_price('sec 10',1000, 3, 3)
predict_price('sec 2',850, 3, 2)
```

4.1.2 Output

```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
```

```
(13320, 9)
```

```
Index(['area_type', 'availability', 'location', 'size', 'society',
       'total_sqft', 'bath', 'balcony', 'price'],
      dtype='object')
```

```
0    Super built-up  Area
1              Plot  Area
2          Built-up  Area
3    Super built-up  Area
4    Super built-up  Area
5    Super built-up  Area
6    Super built-up  Area
7    Super built-up  Area
8    Super built-up  Area
9              Plot  Area
Name: area_type, dtype: object
```

```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
```

```
(13320, 5)
```

```
location       0
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

```
location      0
size          0
total_sqft    0
bath          0
price         0
dtype: int64
```

```
(13247, 5)
```

```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

	location	size	total_sqft	bath	price	bhk
30	sec 1	4 BHK	2100 - 2850	4.0	13000	4
122	sec 11	4 BHK	3067 - 8156	4.0	17500	4
137	sec 7	2 BHK	1042 - 1105	2.0	20000	2
165	sec 6	2 BHK	1145 - 1340	2.0	24000	2
188	sec 2	2 BHK	1015 - 1540	2.0	13000	2
410	sec 6	1 BHK	34.46Sq. Meter	1.0	24000	1
549	sec 10	2 BHK	1195 - 1440	2.0	12000	2
648	sec 6	9 Bedroom	4125Perch	9.0	15000	9
661	sec 1	2 BHK	1120 - 1145	2.0	10000	2
672	sec 6	4 Bedroom	3090 - 5002	4.0	35000	4

	location	size	total_sqft	bath	price	bhk
0	sec 1	2 BHK	1056.0	2.0	20000	2
1	sec 2	4 Bedroom	2600.0	5.0	20000	4

```
location       sec 1
size           4 BHK
total_sqft      2475
bath              4
price         13000
bhk               4
Name: 30, dtype: object
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	sec 1	2 BHK	1056.0	2.0	20000	2	18.939394
1	sec 2	4 Bedroom	2600.0	5.0	20000	4	7.692308
2	sec 3	3 BHK	1440.0	2.0	21000	3	14.583333
3	sec 4	3 BHK	1521.0	3.0	12000	3	7.889546
4	sec 13	2 BHK	1200.0	2.0	12500	2	10.416667

```
count    13201.000000
mean        14.066712
std        120.441243
min          0.248699
25%          8.045052
50%         10.922993
75%         15.000000
max      13500.000000
Name: price_per_sqft, dtype: float64
```

```
sec 2     2313
sec 10    1766
sec 1     1594
sec 6     1592
sec 8      975
sec 7      973
sec 4      894
sec 13     884
sec 14     884
sec 11     708
sec 12     262
sec 3      180
sec6        89
sec13       87
Name: location, dtype: int64
```

```
13201
```

```
10
```

```
14
```

```
4
```

```
sec 12    262
sec 3     180
sec6       89
sec13      87
Name: location, dtype: int64
```

```
14
```

```
11
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	sec 1	2 BHK	1056.0	2.0	20000	2	18.939394
1	sec 2	4 Bedroom	2600.0	5.0	20000	4	7.692308
2	other	3 BHK	1440.0	2.0	21000	3	14.583333
3	sec 4	3 BHK	1521.0	3.0	12000	3	7.889546
4	sec 13	2 BHK	1200.0	2.0	12500	2	10.416667
5	sec 6	2 BHK	1170.0	2.0	13000	2	11.111111
6	sec 7	4 BHK	2732.0	4.0	13000	4	4.758419
7	sec 8	4 BHK	3300.0	4.0	18000	4	5.454545
8	sec 11	3 BHK	1310.0	3.0	10500	3	8.015267
9	sec 10	6 Bedroom	1020.0	6.0	13000	6	12.745098

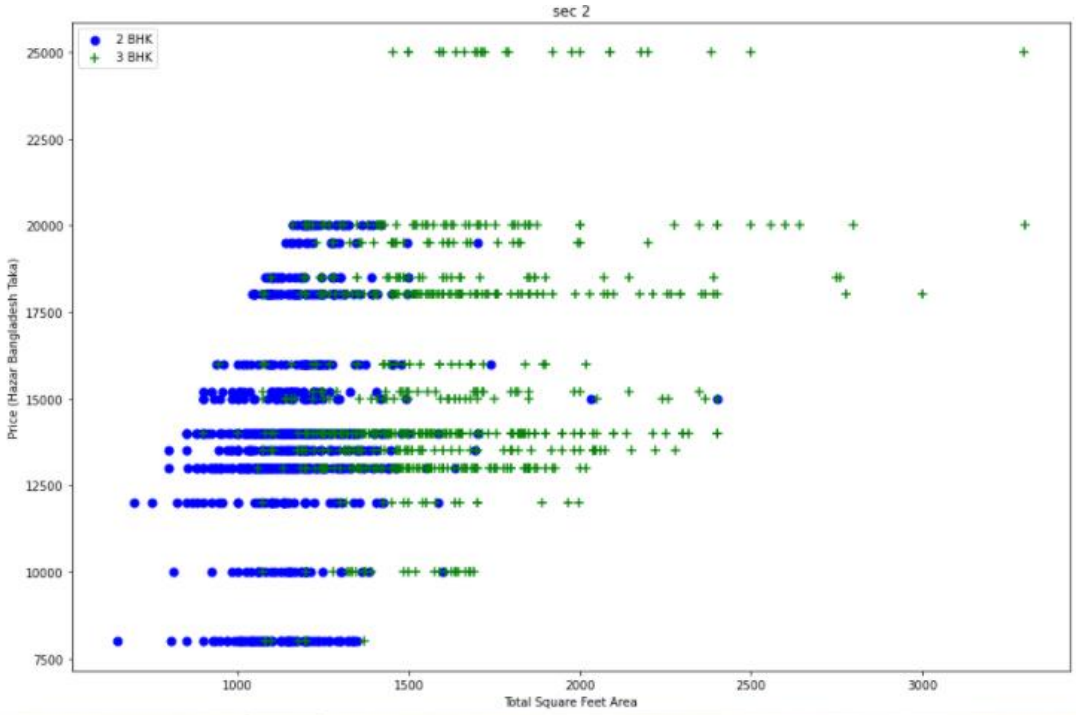
	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	sec 10	6 Bedroom	1020.0	6.0	13000	6	12.745098
45	sec 10	8 Bedroom	600.0	9.0	13000	8	21.666667
58	sec 13	6 Bedroom	1407.0	4.0	10000	6	7.107321
68	sec 1	8 Bedroom	1350.0	7.0	12000	8	8.888889
70	sec 7	3 Bedroom	500.0	3.0	13550	3	27.100000

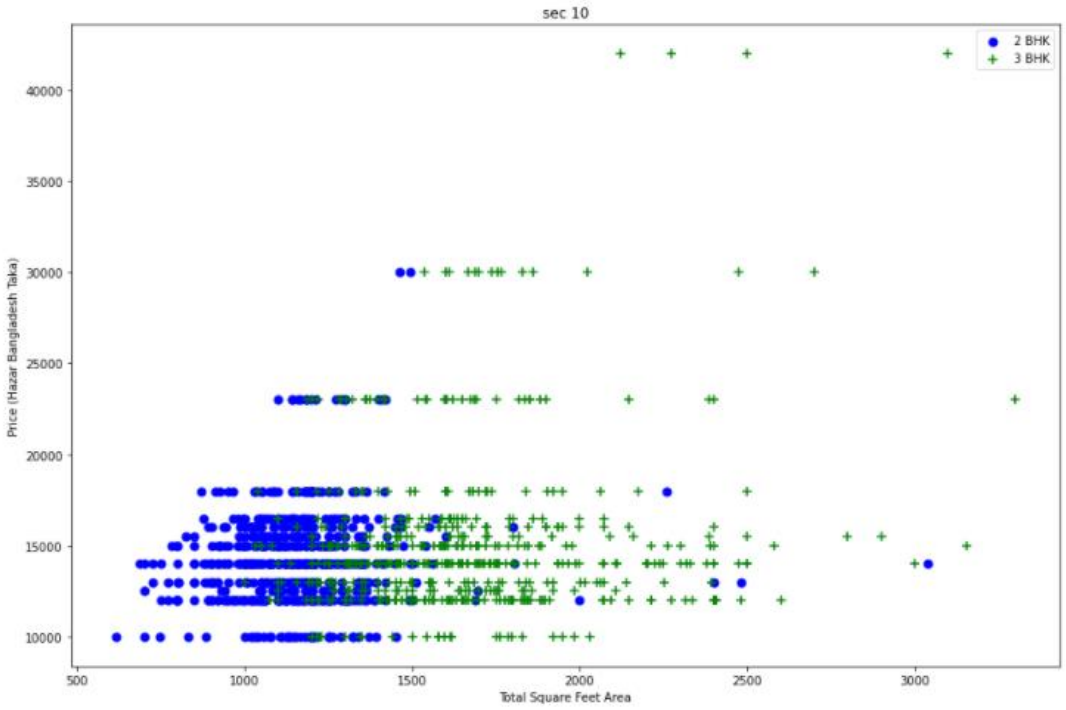
```
(13201, 7)
```

```
(12457, 7)
```

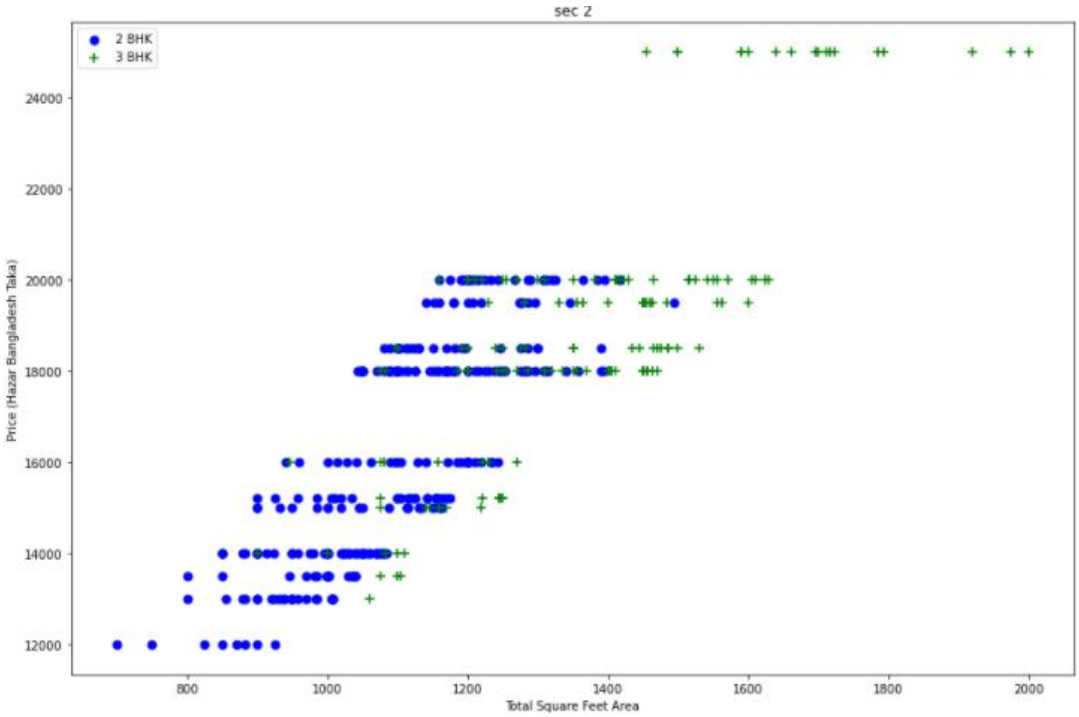
```
count    12457.000000
mean        12.262505
std          7.346777
min          0.248699
25%          7.936508
50%         10.800000
75%         14.492754
max        100.478469
Name: price_per_sqft, dtype: float64
```

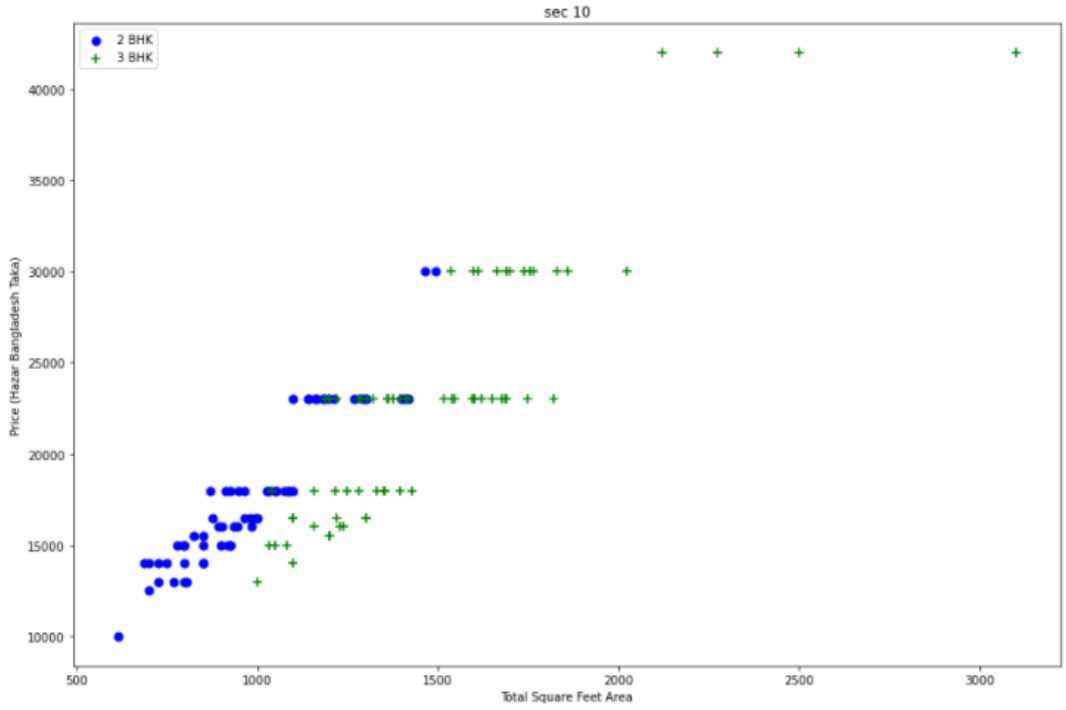
```
(9812, 7)
```


sec 2
2 BHK
3 BHK
Price (Hazar Bangladesh Taka)
Total Square Feet Area

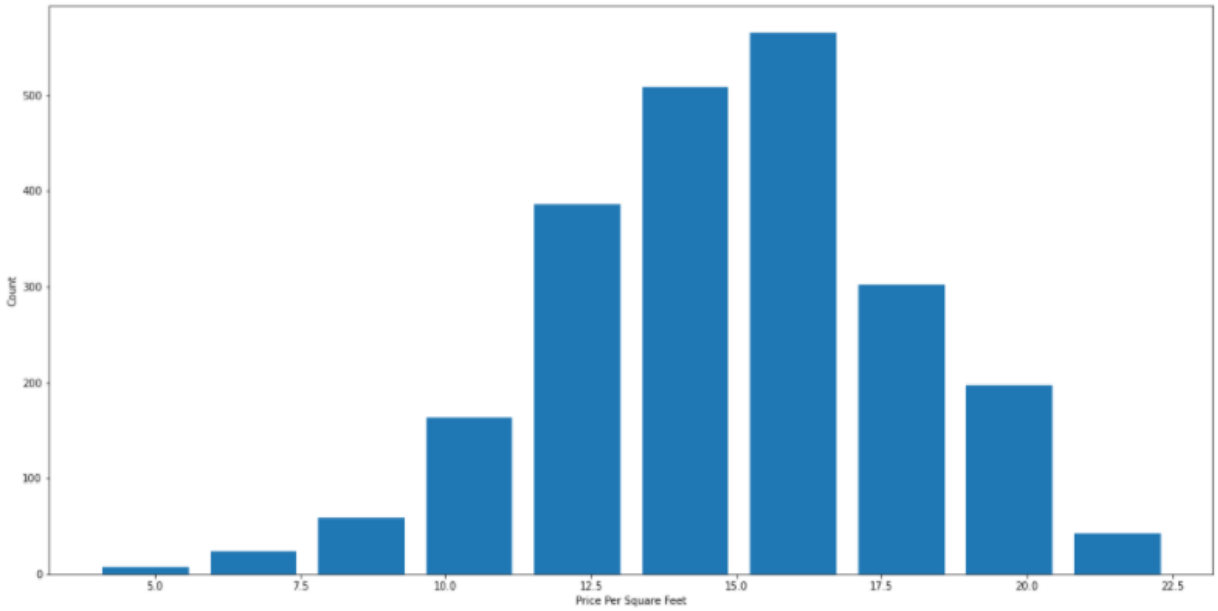

sec 10
2 BHK
3 BHK
Price (Hazar Bangladesh Taka)
Total Square Feet Area

```
(2253, 7)
```


sec 2
2 BHK
3 BHK
Price (Hazar Bangladesh Taka)
Total Square Feet Area
24000
22000
20000
18000
16000
14000
12000
800
1000
1200
1400
1600
1800
2000

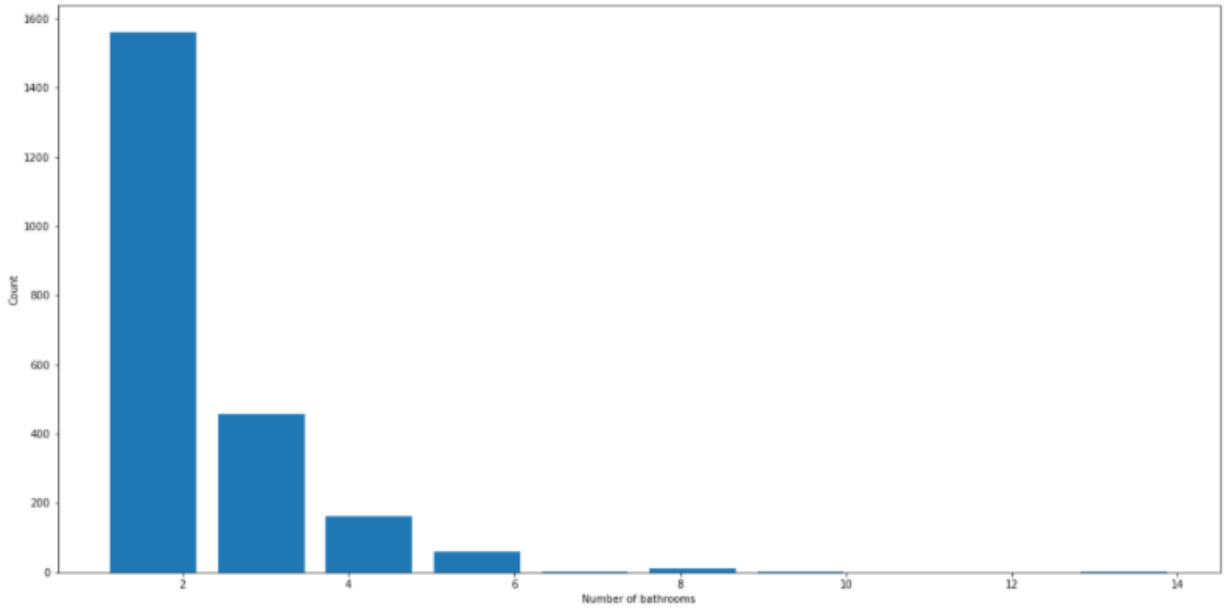

sec 10
2 BHK
3 BHK
Price (Hazar Bangladesh Taka)
Total Square Feet Area
40000
35000
30000
25000
20000
15000
10000
500
1000
1500
2000
2500
3000

```
Text(0, 0.5, 'Count')
```




```
array([ 2.,  3.,  4.,  7.,  1.,  5.,  8.,  6.,  9., 14.])
```

```
Text(0, 0.5, 'Count')
```



	location	size	total_sqft	bath	price	bhk	price_per_sqft
3669	sec 13	9 Bedroom	3300.0	14.0	14500	9	4.393939

	location	size	total_sqft	bath	price	bhk	price_per_sqft
1342	sec 1	3 Bedroom	2400.0	6.0	40000	3	16.666667
3669	sec 13	9 Bedroom	3300.0	14.0	14500	9	4.393939

```
(2246, 7)
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	other	3 BHK	1440.0	2.0	21000	3	14.583333
1	other	3 BHK	1075.0	2.0	13000	3	12.093023

	location	total_sqft	bath	price	bhk
0	other	1440.0	2.0	21000	3
1	other	1075.0	2.0	13000	3
2	other	945.0	2.0	13000	2

	other	sec 1	sec 10	sec 11	sec 13	sec 14	sec 2	sec 4	sec 6	sec 7	sec 8
0	1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0

	location	total_sqft	bath	price	bhk	sec 1	sec 10	sec 11	sec 13	sec 14	sec 2	sec 4	sec 6	sec 7	sec 8
0	other	1440.0	2.0	21000	3	0	0	0	0	0	0	0	0	0	0
1	other	1075.0	2.0	13000	3	0	0	0	0	0	0	0	0	0	0
2	other	945.0	2.0	13000	2	0	0	0	0	0	0	0	0	0	0
4	other	1242.0	2.0	13000	2	0	0	0	0	0	0	0	0	0	0
5	other	883.0	2.0	10000	2	0	0	0	0	0	0	0	0	0	0

	total_sqft	bath	price	bhk	sec 1	sec 10	sec 11	sec 13	sec 14	sec 2	sec 4	sec 6	sec 7	sec 8
0	1440.0	2.0	21000	3	0	0	0	0	0	0	0	0	0	0
1	1075.0	2.0	13000	3	0	0	0	0	0	0	0	0	0	0

```
(2246, 14)
```

	total_sqft	bath	bhk	sec 1	sec 10	sec 11	sec 13	sec 14	sec 2	sec 4	sec 6	sec 7	sec 8
0	1440.0	2.0	3	0	0	0	0	0	0	0	0	0	0
1	1075.0	2.0	3	0	0	0	0	0	0	0	0	0	0
2	945.0	2.0	2	0	0	0	0	0	0	0	0	0	0

```
(2246, 13)
```

```
0    21000
1    13000
2    13000
Name: price, dtype: int64
```

```
2246
```

```
0.701811631158008
```

```
array([0.53472331, 0.62433502, 0.67509161, 0.61406028, 0.60296465])
```

```
15276.218974954492
```

```
16519.18286028543
```

```
13824.604476963583
```

Figure 11: Ouput

4.2 System Setup

i. Package: Operating System (Any)
ii. Software: Browser (Any), Python (new update version), Anaconda, Jupyter Notebook, Microsoft Excel
iii. Hard Disk: 120 GB (minimum)
iv. Ram: 2 GB (minimum)
v. Processor: Dual core or above

4.3 Results and Discussion

Home Rent Prediction System is an AI-based program that can predict the house price by analyzing the location, size of the room, and the total amount of bed-room. Here, we used a dataset that contains various information about houses including house prices. Our program predicts rent by analyzing the used dataset. We used Python programming language which is one of the best high level programming language for Artificial Intelligence.

The majority of the people in Dhaka have come to Dhaka for a variety of economic and social reasons. They are staying and working in Dhaka permanently without a home of their own. The experiences and the house ownership process of the tenants generate knowledge and present a guideline for the policymakers and planners. According to research, about 68% of people who lived in Dhaka are tenants. So, it's very difficult for them to search for available houses on their own without any guidelines.

To find a house which is available for rent is a very hard-working job. Our system can make the job a little easy. And tenants can save a lot of time to find their beautiful house.

4.4 Existing System Analysis

In the currently existing system, tenants search for houses on the website by searching the only location. So, it is a boring and frustrating job to do. They are wasting too much time with this process.

4.5 Summary

This system is providing high-quality prediction with maximum accuracy about House rent.

Chapter 5

Standard, Ethics, Challenge

5.1 Standard

Ensure an adequate, timely, and easily accessible dataset that gives us an accurate result. Ensure adequate system for the operation of a sustainable prediction system. Implement a top-quality management system within the service. Our vision is to possess a well-organized, coordinated, standardized, and quality insertion service that ensures adequate, safe, and timely shows the rent

to the tenants. The tenants should understand the risks of view the predicted rent by the system.

5.2 Ethics

Home rent must be ethical. The how house owner and the tenant must follow the ethical rule. No fraud is acceptable in the home rent system. So always verify the information before renting a house. Tenants and the house owner can Develop straightforward contracts. House owner must be Reasonable with rent increases. Tenants should do proper maintenance of the house. Both tenants and house owners must respect privacy. A good tenant pays rent on time. But even the best tenant is going to occasionally make a mistake or have an issue where they miss a payment by a day or two. So, the house owner should show some leniency. The move-out process is often one of the biggest points of contention between landlords and tenants. Tenants want to get their security deposits back, while landlords typically look for any excuse to keep them. So, tenants shouldn't nitpick on move-out.

5.3 Challenges

The main challenge of building our project is to make a straightforward system for the tenants. To make our system we face many problems. But, due to hard teamwork and proper communication with our teammates, we come out through problems. To build a perfect dataset we are doing many things. Like, data load, handle NA values, Feature Engineering, Dimensionality Reduction, Outlier Removal Using Business Logic, Outlier Removal Using Standard Deviation and Mean, Outlier Removal Using Bathrooms Feature, One Hot Encoding for Location, build a Model, Test the model for few properties, etc. We use K Fold cross-validation to measure the accuracy of our Linear-Regression model.

Thus, we build our system and completed all the challenges.

5.4 Summary

To build a straightforward page with standards, ethics, and challenge it's not easy for us to end our project. But we tried our greatest to assist those tenants who seek houses each day. We believed that our system will help those tenants.

Chapter 6

Conclusion

6.1 Introduction

This system is easy to use for all. Home Rent Prediction System is an AI-based program that can predict the house price by analyzing the location, size of the room, and the total amount of bed-room. Here, we used a dataset that contains various information about houses including house prices. Our program predicts rent by analyzing the used dataset. We used Python programming language which is one of the best high level programming language for Artificial Intelligence.

The majority of the people in Dhaka have come to Dhaka for a variety of economic and social reasons. They are staying and working in Dhaka permanently without a home of their own. The experiences and the house ownership process of the tenants generate knowledge and present a guideline for the policymakers and planners. According to research, about 68% of people who lived in Dhaka are tenants. So, it's very difficult for them to search for available houses on their own without any guidelines.

The scope of the project is that in a very short span provides the user with many facilities. It provides a very accurate prediction with an accuracy of more than 60%. So, users can trust it. It shows users the predicted rent when they give the location, total square feet of house, total bedroom, and total bath as input.

To find a house which is available for rent is a very hard-working job. Our system can make the job a little easy. And tenants can save a lot of time to find their beautiful house.

6.2 Future Work

The Main future scope of our proposed model about the Home Rent Prediction system is to try it in the real world means we have to implement it with better results. We can improve this system with more accurate results. This system can be used in many websites by special features. It helps the websites to perform better and help the user to enjoy the sites.

Reference

[1]. Codebasics. “Machine Learning & Data Science Project” Jan-1, 2020. Available: https://www.youtube.com/watch?v=rdfbcdP75KI&list=PLeo1K3hjS3uu7clOTtwsp94PcHbzqpAdg&index=1

[2]. Bproperty. Available: https://www.bproperty.com/

[3]. Rentalhomebd. Available: https://rentalhomebd.com/

[4]. Samsuddin Ahmed, Md. Mahbubur Rahman, Sabirah Islam, “House Rent Estimation in Dhaka City by Multi-Layer Perceptions Neural Network” Aug-2014 DOI: 10.14257/ijunesst.2014.7.4.26 .

Available: https://www.researchgate.net/publication/291197777_House_Rent_Estimation_in_Dhaka_City_by_Multi_Layer_Perceptions_Neural_Network

[5]. Python Programmer. “What is Pandas? Why and How to Use Pandas in Python” May-24, 2018.

Available: https://www.youtube.com/watch?v=dcqPhpY7tWk

[6]. Python Programmer. “Watch this if you want to LEARN MATPLOTLIB for PYTHON!” Jun-28, 2018.

Available: https://www.youtube.com/watch?v=6rKe2IEIu8c

[7]. Corey Schafer. “Jupyter Notebook Tutorial: Introduction, Setup, and Walkthrough” Sep-22, 2016.

Available: https://www.youtube.com/watch?v=HW29067qVWk

[8]. Kaggle. “Bengaluru House Price Data” 2018.

Available: https://www.kaggle.com/amitabhajoy/bengaluru-house-price-data

Figures