In [161]:

```
# import pyhton libraries
import numpy as np # It will tske care of numerical data
import pandas as pd # It will import excel file
# import data visualization library
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

(https://getlin

In [162]:

```
data=pd.read_csv(r"C:\Users\SUMIT SHARMA\Desktop\Housing.csv")
```

In [163]:

```
# Check rows and columns in the data set using .shape
data.shape
```

Out[163]:

```
(545, 13)
```

Out[163]:

```
(545, 13)
```

In [164]:

```
# Checking information about the dataset using .info()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   price             545 non-null    int64 
 1   area              545 non-null    int64 
 2   bedrooms          545 non-null    int64 
 3   bathrooms         545 non-null    int64 
 4   stories           545 non-null    int64 
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64 
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB
```

In [165]:

```
data.isnull().sum()
```

Out[165]:

```
price               0
area                0
bedrooms            0
bathrooms           0
stories             0
mainroad            0
guestroom           0
basement            0
hotwaterheating     0
airconditioning     0
parking             0
prefarea            0
furnishingstatus    0
dtype: int64
```

(https://getlin

In [166]:

```
data['mainroad'].unique()
```

Out[166]:

```
array(['yes', 'no'], dtype=object)
```

In [167]:

```
data['parking'].unique()
```

Out[167]:

```
array([2, 3, 0, 1], dtype=int64)
```

In [168]:

```
data['parking'].value_counts()
```

Out[168]:

```
0    299
1    126
2    108
3     12
Name: parking, dtype: int64
```

In [169]:

```
features = data[['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
'parking', 'prefarea', 'furnishingstatus']]
```

In [170]:

```
for i in features:
 print(i)
print(features[i].value_counts())
```

```
price
area
bedrooms
bathrooms
stories
mainroad
guestroom
basement
hotwaterheating
airconditioning
parking
prefarea
furnishingstatus
semi-furnished    227
unfurnished       178
furnished         140
Name: furnishingstatus, dtype: int64
```

(https://getlin

In [171]:

```
df = data
```

In [172]:

```
df['mainroad'] = df['mainroad'].map({'yes': 1, 'no': 0})
df['guestroom'] = df['guestroom'].map({'yes': 1, 'no': 0})
df['basement'] = df['basement'].map({'yes': 1, 'no': 0})
df['hotwaterheating'] = df['hotwaterheating'].map({'yes': 1, 'no': 0})
df['airconditioning'] = df['airconditioning'].map({'yes': 1, 'no': 0})
df['prefarea'] = df['prefarea'].map({'yes': 1, 'no': 0})
```

In [173]:

```
df
```

Out[173]:

```
        price  area  bedrooms  bathrooms  stories  mainroad  guestroom  basement  hotwa
  0  13300000  7420         4          2        3         1          0         0
  1  12250000  8960         4          4        4         1          0         0
  2  12250000  9960         3          2        2         1          0         1
  3  12215000  7500         4          2        2         1          0         1
  4  11410000  7420         4          1        2         1          1         1
...       ...   ...       ...        ...      ...       ...        ...       ...
540   1820000  3000         2          1        1         1          0         1
541   1767150  2400         3          1        1         0          0         0
542   1750000  3620         2          1        1         1          0         0
543   1750000  2910         3          1        1         0          0         0
544   1750000  3850         3          1        2         1          0         0

545 rows × 13 columns
```

(https://getlin

LINEAR REGRESSION

In [174]:

```
furnish = pd.get_dummies(df['furnishingstatus'],drop_first=True)
furnish
```

Out[174]:

```
     semi-furnished  unfurnished
  0               0            0
  1               0            0
  2               1            0
  3               0            0
  4               0            0
...             ...          ...
540               0            1
541               1            0
542               0            1
543               0            0
544               0            1

545 rows × 2 columns
```

In [175]:

```
data = pd.concat([df,furnish],axis = 1) # For merging data
data
```

Out[175]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	1	0	1	
541	1767150	2400	3	1	1	0	0	0	
542	1750000	3620	2	1	1	1	0	0	
543	1750000	2910	3	1	1	0	0	0	
544	1750000	3850	3	1	2	1	0	0	

545 rows × 15 columns

(https://getlin

In [176]:

```
data.drop(['furnishingstatus'],axis = 1,inplace=True)
data
```

Out[176]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	
...	...	...	...	...	...	...	...	...	
540	1820000	3000	2	1	1	1	0	1	
541	1767150	2400	3	1	1	0	0	0	
542	1750000	3620	2	1	1	1	0	0	
543	1750000	2910	3	1	1	0	0	0	
544	1750000	3850	3	1	2	1	0	0	

545 rows × 14 columns

(https://getlin

In [177]:

```
data.describe().T
```

Out[177]:

	count	mean	std	min	25%	50%	7
price	545.0	4.766729e+06	1.870440e+06	1750000.0	3430000.0	4340000.0	57400
area	545.0	5.150541e+03	2.170141e+03	1650.0	3600.0	4600.0	63
bedrooms	545.0	2.965138e+00	7.380639e-01	1.0	2.0	3.0	
bathrooms	545.0	1.286239e+00	5.024696e-01	1.0	1.0	1.0	
stories	545.0	1.805505e+00	8.674925e-01	1.0	1.0	2.0	
mainroad	545.0	8.587156e-01	3.486347e-01	0.0	1.0	1.0	
guestroom	545.0	1.779817e-01	3.828487e-01	0.0	0.0	0.0	
basement	545.0	3.504587e-01	4.775519e-01	0.0	0.0	0.0	
hotwaterheating	545.0	4.587156e-02	2.093987e-01	0.0	0.0	0.0	
airconditioning	545.0	3.155963e-01	4.651799e-01	0.0	0.0	0.0	
parking	545.0	6.935780e-01	8.615858e-01	0.0	0.0	0.0	
prefarea	545.0	2.348624e-01	4.243022e-01	0.0	0.0	0.0	
semi-furnished	545.0	4.165138e-01	4.934337e-01	0.0	0.0	0.0	
unfurnished	545.0	3.266055e-01	4.694024e-01	0.0	0.0	0.0	

(https://getlin

In [178]:

```
data.tail()
```

Out[178]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwat
540	1820000	3000	2	1	1	1	0	1	
541	1767150	2400	3	1	1	0	0	0	
542	1750000	3620	2	1	1	1	0	0	
543	1750000	2910	3	1	1	0	0	0	
544	1750000	3850	3	1	2	1	0	0	

In [179]:

```
data.head()
```

Out[179]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwate
0	13300000	7420	4	2	3	1	0	0	
1	12250000	8960	4	4	4	1	0	0	
2	12250000	9960	3	2	2	1	0	1	
3	12215000	7500	4	2	2	1	0	1	
4	11410000	7420	4	1	2	1	1	1	

(https://getlin

LINEAR REGRESSION ANALYSIS

In [180]:

```
from sklearn.linear_model import LinearRegression
```

In [181]:

```
data.columns
```

Out[181]:

```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'semi-furnished', 'unfurnished'],
      dtype='object')
```

In [182]:

```
# X is always independent variable or feature and y is always dependent feature
X = data[['area', 'bathrooms', 'stories', 'mainroad',
'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
'parking', 'prefarea', 'semi-furnished', 'unfurnished']]
```

In [ ]:

```
y = data[['price']]
```

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=4
```

In [ ]:

```
 from sklearn.linear_model import LinearRegression
```

In [ ]:

```
reg = LinearRegression().fit(X_train, y_train)
reg.score(X_test,y_test)
```

MODEL ACCURACY IS 64.40%

(https://getlin

MODEL PREDICTED GRAPH

In [ ]:

```
y_pred = reg.predict(X_test)
y_pred
```

In [ ]:

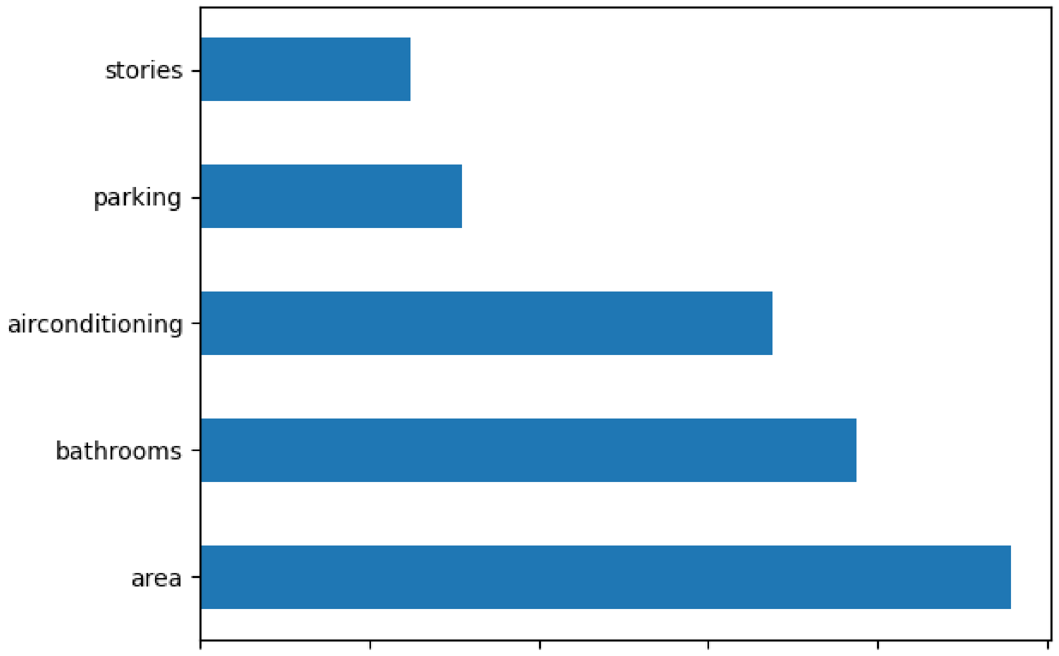
```
from sklearn.ensemble import ExtraTreesRegressor
model = ExtraTreesRegressor()
model.fit(X,y)
ExtraTreesRegressor()
```

In [ ]:

```
print(model.feature_importances_)
```

In [183]:

```
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(5).plot(kind='barh')
plt.show()
```



(https://getlin

In [ ]: