

```
In [1]: import pandas as pd
import numpy as np
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: df=pd.read_excel('DS - Assignment Part 1 data set.xlsx')
df
```

```
Out[2]:
```

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area
0	2012.916667	32.0	84.87882	10	24.98298	121.54024	1	575	37.9
1	2012.916667	19.5	306.59470	9	24.98034	121.53951	2	1240	42.2
2	2013.583333	13.3	561.98450	5	24.98746	121.54391	3	1060	47.3
3	2013.500000	13.3	561.98450	5	24.98746	121.54391	2	875	54.8
4	2012.833333	5.0	390.56840	5	24.97937	121.54245	1	491	43.1
...
409	2013.000000	13.7	4082.01500	0	24.94155	121.50381	3	803	15.4
410	2012.666667	5.6	90.45606	9	24.97433	121.54310	2	1278	50.0
411	2013.250000	18.8	390.96960	7	24.97923	121.53986	1	503	40.6
412	2013.000000	8.1	104.81010	5	24.96674	121.54067	1	597	52.5
413	2013.500000	6.5	90.45606	9	24.97433	121.54310	2	1097	63.9

414 rows × 9 columns

```
In [3]: df.isnull().sum()
```

```
Out[3]: Transaction date      0
House Age      0
Distance from nearest Metro station (km)  0
Number of convenience stores  0
latitude      0
longitude     0
Number of bedrooms  0
House size (sqft)  0
House price of unit area  0
dtype: int64
```

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 9 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Transaction date                         414 non-null    float64
1   House Age                               414 non-null    float64
2   Distance from nearest Metro station (km) 414 non-null    float64
3   Number of convenience stores             414 non-null    int64
4   latitude                                 414 non-null    float64
5   longitude                                414 non-null    float64
6   Number of bedrooms                      414 non-null    int64
7   House size (sqft)                       414 non-null    int64
8   House price of unit area                 414 non-null    float64
dtypes: float64(6), int64(3)
memory usage: 29.2 KB
```

```
In [5]: df['Transaction date']=df['Transaction date'].astype(str)
```

```
In [6]: df['Transaction date']=df['Transaction date'].apply(lambda x: x.split('.')[0])
```

```
In [7]: df
```

Out [7]:

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area
0	2012	32.0	84.87882	10	24.98298	121.54024	1	575	37.9
1	2012	19.5	306.59470	9	24.98034	121.53951	2	1240	42.2
2	2013	13.3	561.98450	5	24.98746	121.54391	3	1060	47.3
3	2013	13.3	561.98450	5	24.98746	121.54391	2	875	54.8
4	2012	5.0	390.56840	5	24.97937	121.54245	1	491	43.1
...
409	2013	13.7	4082.01500	0	24.94155	121.50381	3	803	15.4
410	2012	5.6	90.45606	9	24.97433	121.54310	2	1278	50.0
411	2013	18.8	390.96960	7	24.97923	121.53986	1	503	40.6
412	2013	8.1	104.81010	5	24.96674	121.54067	1	597	52.5
413	2013	6.5	90.45606	9	24.97433	121.54310	2	1097	63.9

414 rows × 9 columns

In [8]:

```
df['Transaction date']=df['Transaction date'].astype(int)
```

In [9]:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 9 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Transaction date                      414 non-null    int32
1   House Age                            414 non-null    float64
2   Distance from nearest Metro station (km)  414 non-null    float64
3   Number of convenience stores          414 non-null    int64
4   latitude                             414 non-null    float64
5   longitude                             414 non-null    float64
6   Number of bedrooms                   414 non-null    int64
7   House size (sqft)                    414 non-null    int64
8   House price of unit area              414 non-null    float64
dtypes: float64(5), int32(1), int64(3)
memory usage: 27.6 KB
```

In [10]:

```
df['House Age']=np.ceil(df['House Age']).astype(int)
```

In [11]:

```
df['Distance from nearest Metro station (km)']=df['Distance from nearest Metro station (km)'].astype(int)
```

In [12]:

```
df
```

Out [12]:

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area
0	2012	32	84	10	24.98298	121.54024	1	575	37.9
1	2012	20	306	9	24.98034	121.53951	2	1240	42.2
2	2013	14	561	5	24.98746	121.54391	3	1060	47.3
3	2013	14	561	5	24.98746	121.54391	2	875	54.8
4	2012	5	390	5	24.97937	121.54245	1	491	43.1
...
409	2013	14	4082	0	24.94155	121.50381	3	803	15.4
410	2012	6	90	9	24.97433	121.54310	2	1278	50.0
411	2013	19	390	7	24.97923	121.53986	1	503	40.6
412	2013	9	104	5	24.96674	121.54067	1	597	52.5
413	2013	7	90	9	24.97433	121.54310	2	1097	63.9

414 rows × 9 columns

Merger the two coloumnns

```
In [13]: df['Total Price of House']= df['House size (sqft)'] * df['House price of unit area']
```

```
In [14]: df
```

Out[14]:

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area	Total Price of House
0	2012	32	84	10	24.98298	121.54024	1	575	37.9	21792.5
1	2012	20	306	9	24.98034	121.53951	2	1240	42.2	52328.0
2	2013	14	561	5	24.98746	121.54391	3	1060	47.3	50138.0
3	2013	14	561	5	24.98746	121.54391	2	875	54.8	47950.0
4	2012	5	390	5	24.97937	121.54245	1	491	43.1	21162.1
...
409	2013	14	4082	0	24.94155	121.50381	3	803	15.4	12366.2
410	2012	6	90	9	24.97433	121.54310	2	1278	50.0	63900.0
411	2013	19	390	7	24.97923	121.53986	1	503	40.6	20421.8
412	2013	9	104	5	24.96674	121.54067	1	597	52.5	31342.5
413	2013	7	90	9	24.97433	121.54310	2	1097	63.9	70098.3

414 rows × 10 columns

```
In [15]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 414 entries, 0 to 413
Data columns (total 10 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Transaction date                      414 non-null    int32
1   House Age                            414 non-null    int32
2   Distance from nearest Metro station (km)  414 non-null    int32
3   Number of convenience stores          414 non-null    int64
4   latitude                             414 non-null    float64
5   longitude                            414 non-null    float64
6   Number of bedrooms                   414 non-null    int64
7   House size (sqft)                    414 non-null    int64
8   House price of unit area              414 non-null    float64
9   Total Price of House                  414 non-null    float64
dtypes: float64(4), int32(3), int64(3)
memory usage: 27.6 KB
```

```
In [16]: df.describe()
```

Out[16]:

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area	Total Price of House
count	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000	414.000000
mean	2012.695652	18.144928	1083.376812	4.094203	24.969030	121.533361	1.987923	931.475845	37.980193	35597.804106
std	0.460687	11.414839	1262.159441	2.945562	0.012410	0.015347	0.818875	348.910269	13.606488	19890.892075
min	2012.000000	0.000000	23.000000	0.000000	24.932070	121.473530	1.000000	402.000000	7.600000	5440.400000
25%	2012.000000	9.250000	289.000000	1.000000	24.963000	121.528085	1.000000	548.000000	27.700000	20482.700000
50%	2013.000000	17.000000	492.000000	4.000000	24.971100	121.538630	2.000000	975.000000	38.450000	31224.200000
75%	2013.000000	28.750000	1453.500000	6.000000	24.977455	121.543305	3.000000	1234.750000	46.600000	48439.375000
max	2013.000000	44.000000	6488.000000	10.000000	25.014590	121.566270	3.000000	1500.000000	117.500000	161327.500000

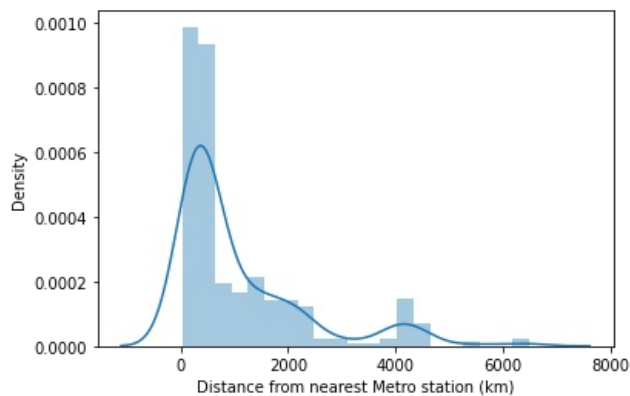
```
In [17]: sns.heatmap(df.corr(),annot=True)
```





```
In [18]: sns.distplot(df['Distance from nearest Metro station (km)'])
```

```
Out[18]: <AxesSubplot:xlabel='Distance from nearest Metro station (km)', ylabel='Density'>
```



```
In [19]: x=df.iloc[:, :-1]
y=df['Total Price of House']
```

```
In [20]: x
```

```
Out[20]:
```

	Transaction date	House Age	Distance from nearest Metro station (km)	Number of convenience stores	latitude	longitude	Number of bedrooms	House size (sqft)	House price of unit area
0	2012	32	84	10	24.98298	121.54024	1	575	37.9
1	2012	20	306	9	24.98034	121.53951	2	1240	42.2
2	2013	14	561	5	24.98746	121.54391	3	1060	47.3
3	2013	14	561	5	24.98746	121.54391	2	875	54.8
4	2012	5	390	5	24.97937	121.54245	1	491	43.1
...
409	2013	14	4082	0	24.94155	121.50381	3	803	15.4
410	2012	6	90	9	24.97433	121.54310	2	1278	50.0
411	2013	19	390	7	24.97923	121.53986	1	503	40.6
412	2013	9	104	5	24.96674	121.54067	1	597	52.5
413	2013	7	90	9	24.97433	121.54310	2	1097	63.9

414 rows × 9 columns

```
In [21]: y
```

```
Out[21]: 0      21792.5
          1      52328.0
          2      50138.0
          3      47950.0
          4      21162.1
          ...
         409     12366.2
         410     63900.0
         411     20421.8
         412     31342.5
         413      70098.3
Name: Total Price of House, Length: 414, dtype: float64
```

```
In [22]: 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=1)
```

```
In [23]: from sklearn.linear_model import LinearRegression
         Lr=LinearRegression()
         Lr.fit(x_train,y_train)
         y_pred=Lr.predict(x_test)
```

```
In [24]: from sklearn.metrics import r2_score
```

```
In [25]: r2_score(y_test,y_pred)
```

```
Out[25]: 0.9290204411341823
```

```
In [26]: Lr.score(x_train,y_train)
```

```
Out[26]: 0.940891075498062
```

```
In [27]: from sklearn.neighbors import KNeighborsRegressor
         from sklearn.tree import DecisionTreeRegressor
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.ensemble import BaggingRegressor
         KNN=KNeighborsRegressor()
         DTR=DecisionTreeRegressor()
         RTR=RandomForestRegressor()
         BR=BaggingRegressor()
```

```
In [28]: from sklearn.metrics import accuracy_score
```

```
In [29]: def House(model):
         model.fit(x_train,y_train)
         ypred=model.predict(x_test)
         print(r2_score(y_test,y_pred))
```

```
In [30]: House(KNN)
```

```
0.9290204411341823
```

```
In [31]: House(DTR)
```

```
0.9290204411341823
```

```
In [32]: House(RTR)
```

```
0.9290204411341823
```

```
In [33]: House(BR)
```

In [34]: `df.columns`

Out[34]: Index(['Transaction date', 'House Age',
'Distance from nearest Metro station (km)',
'Number of convenience stores', 'latitude', 'longitude',
'Number of bedrooms', 'House size (sqft)', 'House price of unit area',
'Total Price of House'],
dtype='object')

```
In [ ]: Transaction_date=int(input('Transaction date :'))
House_Age=int(input('enrter House Age :'))
distance_from_nearest_Metro_station=int(input('enter Distance from nearest Metro station (km) :'))
Number_of_convenience_stores=int(input('enter Number of convenience stores :'))
latitude=float(input('enter latitude :'))
longitude=float(input('enter longitude :'))
Number_of_bedrooms=int(input('enter Number of bedrooms :'))
House_size=int(input('enter House size (sqft) :'))
House_price_of_unit_area=eval(input('enter House price of unit area :'))

res=Lr.predict([[Transaction_date, House_Age,
distance_from_nearest_Metro_station,
Number_of_convenience_stores, latitude, longitude,
Number_of_bedrooms, House_size, House_price_of_unit_area]])
print('PRICE OF HOUSE IS:',res,'RS ONLY')
```

In []: