In [4]:

```
import pandas as pd
from sklearn.metrics import mean_squared_error
```

In [38]:

```
df = pd.read_csv('Real estate.csv')
```

In [40]:

```
df = df.drop(df.columns[0] ,axis = 1)
```

In [42]:

```
df
x = df.iloc[:,:6]
y = df.iloc[:,-1]
```

In [43]:

```
x
```

Out[43]:

	X1 transaction date	X2 house age	X3 distance to the nearest MRT station	X4 number of convenience stores	X5 latitude	X6 longitude
0	2012.917	32.0	84.87882	10	24.98298	121.54024
1	2012.917	19.5	306.59470	9	24.98034	121.53951
2	2013.583	13.3	561.98450	5	24.98746	121.54391
3	2013.500	13.3	561.98450	5	24.98746	121.54391
4	2012.833	5.0	390.56840	5	24.97937	121.54245
...	...	...	...	...	...	...
409	2013.000	13.7	4082.01500	0	24.94155	121.50381
410	2012.667	5.6	90.45606	9	24.97433	121.54310
411	2013.250	18.8	390.96960	7	24.97923	121.53986
412	2013.000	8.1	104.81010	5	24.96674	121.54067
413	2013.500	6.5	90.45606	9	24.97433	121.54310

414 rows × 6 columns

In [44]:

```
y
```

Out[44]:

```
0      37.9
1      42.2
2      47.3
3      54.8
4      43.1
       ... 
409    15.4
410    50.0
411    40.6
412    52.5
413    63.9
Name: Y house price of unit area, Length: 414, dtype: float64
```

In [33]:

```
x
```

Out[33]:

	506	13	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Un
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.
...	...	...	...	...	...	...	...	...	...	...	...
502	0.06263	0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21
503	0.04527	0	11.93	0	0.573	6.12	76.7	2.2875	1	273	21
504	0.06076	0	11.93	0	0.573	6.976	91	2.1675	1	273	21
505	0.10959	0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21

	506	13	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Un
506	0.04741	0	11.93	0	0.573	6.03	80.8	2.505	1	273	21

506 rows × 13 columns

In [36]:

```
y
```

Out[36]:

```
1        24
2      21.6
3      34.7
4      33.4
5      36.2
       ...
502    22.4
503    20.6
504    23.9
505      22
506    11.9
Name: Unnamed: 13, Length: 506, dtype: object
```

In [45]:

```
import math
n =  len(df)
maxk = int(math.sqrt(len(df)))
```

In [47]:

```
mse_values = []
```

In [48]:

```
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 = 102)
from sklearn.neighbors import KNeighborsRegressor
for i in range(1,maxk):
    model = KNeighborsRegressor(n_neighbors=i)
    model.fit(x_train,y_train)
    predicted = model.predict(x_test)
    mse_error = mean_squared_error(y_test,predicted)
    mse_values.append(mse_error)
```

In [49]:

```
print(mse_values)
```

```
[62.46855999999999, 49.58268, 52.730755555555554, 58.64525999999999, 63.3356864,
68.67109333333332, 70.36192489795918, 70.31558500000001, 69.93693728395064,
69.58304159999999, 68.62950280991735, 68.13115833333335, 68.3538901775148,
69.01559877551021, 69.6200792888889, 69.24840687499999, 68.6647626297578,
68.6024103703704, 67.9558129639889]
```

In [53]:

```
import numpy as np
elbow = 0;
inds = np.argsort(mse_values)
for i in inds:
    diff1 = mse_values[i-1] - mse_values[i]
    diff2 = mse_values[i] - mse_values[i+1]
    elbow = i
    if(diff1>0 and diff2<0):
        break
elbow = elbow + 1;
```

In [54]:

```
elbow
```

Out[54]:

```
2
```

In [56]:

```
rknn = KNeighborsRegressor(n_neighbors = elbow)
rknn.fit(x_train,y_train)
```

Out[56]:

```
KNeighborsRegressor(n_neighbors=2)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

KNeighborsRegressor

```
KNeighborsRegressor(n_neighbors=2)
```

In [58]:

```
k = np.arange(1,maxk)
```

In [60]:

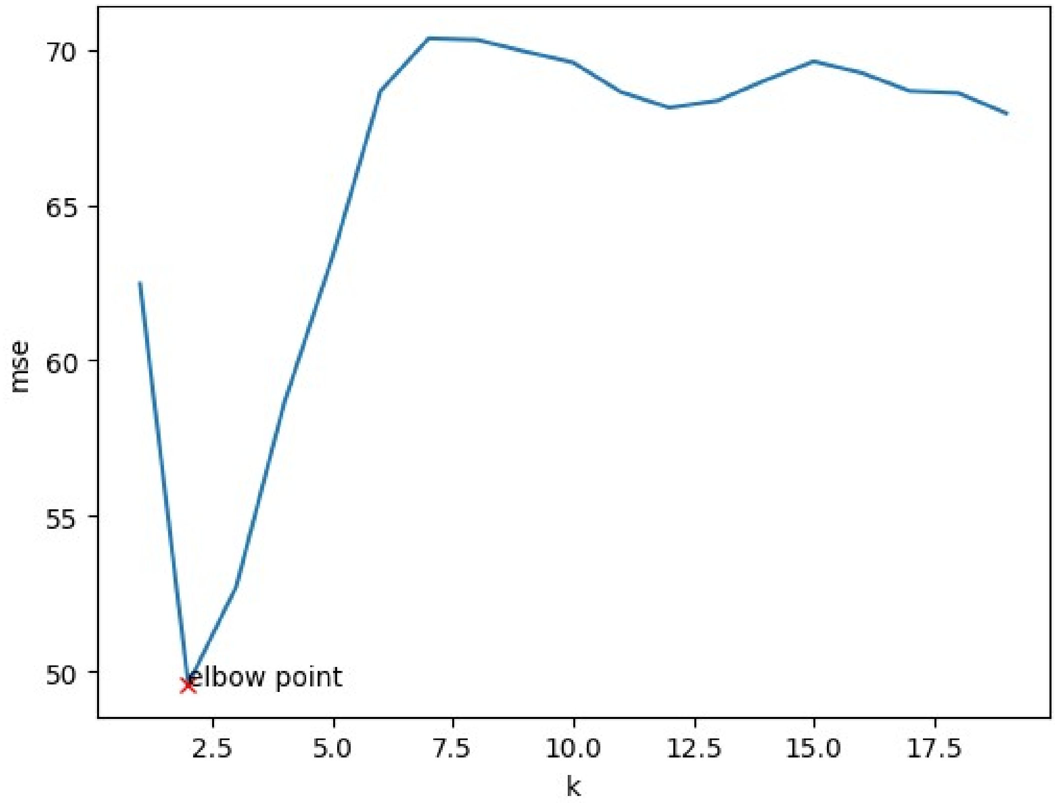
```
import matplotlib.pyplot as plt
```

In [63]:

```
plt.xlabel("k")
plt.ylabel("mse")
plt.plot(k,mse_values)
plt.plot(elbow,mse_values[elbow-1],'rx')
plt.annotate("elbow point" , (elbow,mse_values[elbow-1]))
```

Out[63]:

```
Text(2, 49.58268, 'elbow point')
```



In [ ]: