

```

In [8]: 1 import pandas as pd
        2 import numpy as np
        3 from sklearn.model_selection import train_test_split
        4 from sklearn.ensemble import RandomForestRegressor
        5 from sklearn import metrics
        6
        7 df = pd.read_csv("kc_house_data.csv")
        8
        9 nf = df
       10 date_string = nf['date'].tolist()
       11 date_int = [int(str[:4]) for str in date_string]
       12 df['year'] = date_int
       13
       14
       15 xf = df
       16 xf = xf.drop(['id', 'date', 'price', 'zipcode'], 1)
       17
       18 yf = df
       19 yf = yf.iloc[:, 2]
       20
       21
       22 xs=(xf-xf.mean())/(xf.std())
       23
       24
       25 rfreg = RandomForestRegressor(n_estimators=10)
       26
       27 X_train, X_test = train_test_split(xs, train_size=0.80, random_stat
       28 Y_train, Y_test = train_test_split(yf, train_size=0.80, random_stat
       29
       30
       31 rfreg.fit(X_train, Y_train)
       32 y_pred = rfreg.predict(X_test)
       33
       34 importance = rfreg.feature_importances_
       35 score = rfreg.score(X_test, Y_test)
       36 absolute_error = metrics.mean_absolute_error(Y_test, y_pred)
       37 mse = metrics.mean_squared_error(Y_test, y_pred)
       38 rmse = np.sqrt(mse)
       39
       40 print("\nThe importance of all attributes is given by ",importance)
       41 print("\nNumber of estimators : 10")
       42 print("\nThe score given by Random forest regressor is ", round( sc
       43 print("\nMean Absolute Error is ", round(absolute_error,2))
       44 print("\nRoot Mean Squared Error is ", round(rmse,2))
       45

```

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_split.py:2026: FutureWarning: From version 0.21, test_size will always complement train_size unless both are specified.
FutureWarning)

The importance of all attributes is given by [0.00332677 0.00725831
0.27839696 0.01424092 0.00220892 0.02943334
0.01068537 0.00289646 0.31839314 0.01816735 0.00619043 0.0258785
0.00243877 0.15714028 0.07256439 0.03316242 0.01567519 0.00194249]

Number of estimators : 10

The score given by Random forest regressor is 0.88

Mean Absolute Error is 69657.35

Root Mean Squared Error is 121011.88

```
In [6]: 1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestRegressor
5 from sklearn import metrics
6
7
8
9 df = pd.read_csv("kc_house_data.csv")
10
11
12 nf = df
13 date_string = nf['date'].tolist()
14 date_int = [int(str[:4]) for str in date_string]
15 df['year'] = date_int
16
17
18 xf = df
19 xf = xf.drop(['id', 'date', 'price', 'zipcode'], 1)
20
21 yf = df
22 yf = yf.iloc[:, 2]
23
24 xs=(xf-xf.mean())/(xf.std())
25
26
27 rfreg = RandomForestRegressor(n_estimators=50)
28
29 X_train, X_test = train_test_split(xs, train_size=0.80, random_stat
30 Y_train, Y_test = train_test_split(yf, train_size=0.80, random_stat
31
32 rfreg.fit(X_train, Y_train)
33 y_pred = rfreg.predict(X_test)
34
35 importance = rfreg.feature_importances_
36 score = rfreg.score(X_test, Y_test)
37 absolute_error = metrics.mean_absolute_error(Y_test, y_pred)
38 mse = metrics.mean_squared_error(Y_test, y_pred)
39 rmse = np.sqrt(mse)
40
41 print("\nthe importance of all attributes is given by " importance)
```

```

41 print( \n\nThe importance of all attributes is given by ", importance)
42 print("\nNumber of estimators : 50")
43 print("\nThe score given by Random forest regressor is ", round( score, 2))
44 print("\nMean Absolute Error is ", round(absolute_error,2))
45 print("\nRoot Mean Squared Error is ", round(rmse,2))
46
47

```

```

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_split.py:2026: FutureWarning: From version 0.21, test_size will always complement train_size unless both are specified.
FutureWarning)

```

```

The importance of all attributes is given by [0.00326747 0.00872524
0.32843356 0.0150819 0.00238291 0.03616433
0.01021542 0.00312404 0.25460029 0.01793267 0.00502582 0.02742723
0.00252228 0.16227697 0.07248574 0.03440844 0.01355986 0.00236583]

```

```

Number of estimators : 50

```

```

The score given by Random forest regressor is 0.89

```

```

Mean Absolute Error is 65453.95

```

```

Root Mean Squared Error is 114396.42

```

```

In [5]: 1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestRegressor
5 from sklearn import metrics
6
7
8
9 df = pd.read_csv("kc_house_data.csv")
10
11 nf = df
12 date_string = nf['date'].tolist()
13 date_int = [int(str[:4]) for str in date_string]
14 df['year'] = date_int
15
16 xf = df
17 xf = xf.drop(['id', 'date', 'price', 'zipcode'], 1)
18
19 yf = df
20 yf = yf.iloc[:, 2]
21
22 xs=(xf-xf.mean())/(xf.std())
23
24
25 rfreg = RandomForestRegressor(n_estimators=100)
26
27 X_train, X_test = train_test_split(xs, train_size=0.90, random_state=42)

```

```

28 Y_train, Y_test = train_test_split(yt, train_size=0.90, random_stat
29
30 rfreg.fit(X_train, Y_train)
31 y_pred = rfreg.predict(X_test)
32
33 importance = rfreg.feature_importances_
34 score = rfreg.score(X_test, Y_test)
35 absolute_error = metrics.mean_absolute_error(Y_test, y_pred)
36 mse = metrics.mean_squared_error(Y_test, y_pred)
37 rmse = np.sqrt(mse)
38
39
40 print("\nThe importance of all attributes is given by ",importance)
41 print("\nNumber of estimators : 100")
42 print("\nThe score given by Random forest regressor is ", round( sc
43 print("\nMean Absolute Error is ", round(absolute_error,2))
44 print("\nRoot Mean Squared Error is ", round(rmse,2))
45
46

```

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_split.py:2026: FutureWarning: From version 0.21, test_size will always complement train_size unless both are specified.
FutureWarning)

The importance of all attributes is given by [0.00347032 0.00738769 0.29830547 0.01570872 0.00204956 0.03192644 0.01012144 0.00334655 0.28942194 0.02186616 0.00590878 0.02930165 0.00204034 0.16187868 0.06980053 0.03163767 0.01377082 0.00205725]

Number of estimators : 100

The score given by Random forest regressor is 0.88

Mean Absolute Error is 64076.29

Root Mean Squared Error is 117422.13

```

In [3]: 1 import pandas as pd
2 import numpy as np
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestRegressor
5 from sklearn import metrics
6
7
8 df = pd.read_csv("kc_house_data.csv")
9
10 nf = df
11 date_string = nf['date'].tolist()
12 date_int = [int(str[:4]) for str in date_string]
13 df['year'] = date_int
14
15 xf = df

```

```

16 xf = xf.drop(['id', 'date', 'price', 'zipcode'], 1)
17
18 yf = df
19 yf = yf.iloc[:, 2]
20
21 xs=(xf-xf.mean())/(xf.std())
22
23 rfreg = RandomForestRegressor(n_estimators=200)
24
25 X_train, X_test = train_test_split(xs, train_size=0.90, random_state=42)
26 Y_train, Y_test = train_test_split(yf, train_size=0.90, random_state=42)
27
28 rfreg.fit(X_train, Y_train)
29 y_pred = rfreg.predict(X_test)
30
31 importance = rfreg.feature_importances_
32 score = rfreg.score(X_test, Y_test)
33 absolute_error = metrics.mean_absolute_error(Y_test, y_pred)
34 mse = metrics.mean_squared_error(Y_test, y_pred)
35 rmse = np.sqrt(mse)
36
37
38 print("\nThe importance of all attributes is given by ", importance)
39 print("\nNumber of estimators : 200")
40 print("\nThe score given by Random forest regressor is ", round(score, 2))
41 print("\nMean Absolute Error is ", round(absolute_error, 2))
42 print("\nRoot Mean Squared Error is ", round(rmse, 2))
43
44

```

/anaconda3/lib/python3.7/site-packages/sklearn/model_selection/_split.py:2026: FutureWarning: From version 0.21, test_size will always complement train_size unless both are specified.
FutureWarning)

The importance of all attributes is given by [0.00311684 0.0076099 0.29586088 0.01495023 0.00201219 0.03413939 0.01071733 0.00320667 0.29097303 0.02062599 0.00540868 0.02785056 0.00225962 0.16197825 0.0701258 0.03267487 0.01432985 0.00215993]

Number of estimators : 200

The score given by Random forest regressor is 0.88

Mean Absolute Error is 63821.76

Root Mean Squared Error is 116520.68

In []:

1

