

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```

```
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/3rd-programming-competition-bahrain-ai/
house_data_train4.csv
/kaggle/input/3rd-programming-competition-bahrain-ai/house_data_test4.
csv
```

```
# Load the training and testing datasets
df_train = pd.read_csv('/kaggle/input/3rd-programming-competition-
bahrain-ai/house_data_train4.csv')
df_test = pd.read_csv('/kaggle/input/3rd-programming-competition-
bahrain-ai/house_data_test4.csv')
```

```
df_train.head()
```

	id	sale_price	bedrooms	bathrooms	metersq_living	metersq_lot
0	2	442573.0	3	1.75	1780	7567
1	3	800000.0	4	2.25	2510	9963
2	4	205000.0	3	0.75	1080	5025
3	5	353000.0	4	1.50	1100	9600
4	6	230000.0	3	1.00	1380	10112

	waterfront	view	condition	grade	metersq_above	metersq_basement
0	0	0	3	7	1290	490
1	0	0	4	9	2200	310
2	0	0	3	5	1080	0
3	0	0	4	6	1100	0
4	0	0	4	7	940	440

	yr_built	block_no	location1	location2	metersq_built
0	1980	34	0.7314	-0.225	1910

```

8645
1      1967      5      0.5973      -0.177      3110
9963
2      1948     146      0.4936      -0.335      1370
6000
3      1960      33      0.7000      -0.175      1100
9630
4      1963      23      0.3196      -0.351      1240
10112

```

```
df_train.tail()
```

```

      id  sale_price  bedrooms  bathrooms  metersq_living
metersq_lot \
28980  33305    230000.0         6         3.00         2180
7220
28981  33306    230000.0         3         1.00         1020
12289
28982  33307    225000.0         3         1.75         1460
8372
28983  33308    225000.0         2         1.00          830
5720
28984  33309    220000.0         3         1.75         1230
7200

```

```

      floors  waterfront  view  condition  grade  metersq_above \
28980      2.0           0      0          3       7         2180
28981      1.0           0      0          4       7         1020
28982      1.0           0      0          4       7         1460
28983      1.0           0      0          4       6          830
28984      1.0           0      0          3       7         1230

```

```

      metersq_basement  yr_built  block_no  location1  location2 \
28980                0      1980      2040      1.5346      -1.495
28981                0      1967      2046      1.6535      -1.345
28982                0      1981      2035      1.5983      -1.277
28983                0      1950      2038      1.9639      -1.529
28984                0      1986      2018      1.5181      -1.553

```

```

      metersq_built  metersq_land
28980            2260           7344
28981            1300           9894
28982            1220           7803
28983            1150           5250
28984            1540           7210

```

```
# Identifying the unique number of values in the dataset
```

```
df_train.nunique()
```

```

id      28985
sale_price  3427

```

```
bedrooms          13
bathrooms         30
metersq_living    1000
metersq_lot       9185
floors            5
waterfront        2
view              5
condition         5
grade            12
metersq_above     910
metersq_basement  293
yr_built         116
block_no         140
location1        9209
location2        1427
metersq_built     755
metersq_land      8214
dtype: int64
```

```
# Check for missing values
df_train.isnull().sum()
```

```
id                0
sale_price        0
bedrooms          0
bathrooms         0
metersq_living    0
metersq_lot       0
floors            0
waterfront        0
view              0
condition         0
grade            0
metersq_above     0
metersq_basement  0
yr_built         0
block_no         0
location1        0
location2        0
metersq_built     0
metersq_land      0
dtype: int64
```

```
df_test.isnull().sum()
```

```
id                0
bedrooms          0
bathrooms         0
metersq_living    0
metersq_lot       0
floors            0
```

```
waterfront      0
view            0
condition       0
grade          0
metersq_above   0
metersq_basement 0
yr_built       0
block_no       0
location1       0
location2       0
metersq_built   0
metersq_land    0
dtype: int64
```

```
# Separate the features and target variable for both datasets
train_features = df_train.drop(['id', 'sale_price', 'waterfront'],
axis=1)
train_target = df_train['sale_price']
test_features = df_test.drop(['id', 'waterfront'], axis=1)
test_target = df_test['id']
```

```
df2 = pd.DataFrame(test_target, columns = ['id'])
print(type(df2))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
# Import Random Forest Regressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
```

```
# Create a Random Forest Regressor
reg = RandomForestRegressor()
```

```
# Train the model using the training sets
reg.fit(train_features, train_target)
```

```
RandomForestRegressor()
```

```
# Model prediction on train data
y_pred = reg.predict(test_features)
```

```
#now converting the results to a dataframe
print(type(y_pred))
df = pd.DataFrame(y_pred, columns = ['price'])
print(type(df))
df['price'].round()
```

```
<class 'numpy.ndarray'>
<class 'pandas.core.frame.DataFrame'>
```

```
0      1076898.0
1      446054.0
```

```
2      518480.0
3      599360.0
4      587198.0
```

```
...
7243    291608.0
7244    216432.0
7245    194419.0
7246    215848.0
7247    130404.0
```

Name: price, Length: 7248, dtype: float64

#adding the Id column to the dataframe

```
submission = pd.concat([df2,df.abs()],axis=1)
submission.to_csv('/kaggle/working/submission.csv',index=False) # save
to notebook output
print(submission)
```

```
      id      price
0    17292  1076897.50
1    17293   446053.99
2    17294   518479.72
3    17295   599360.30
4    17296   587198.15
...
7243  36230   291607.98
7244  36231   216432.46
7245  36232   194418.99
7246  36233   215848.30
7247  36234   130403.50
```

[7248 rows x 2 columns]