

```
In [1]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestRegressor
```

```
In [2]: data = pd.read_excel("C:\\\\Users\\\\Joshitha\\\\OneDrive\\\\Hình ảnh\\\\HousePricePrediction.xlsx")
```

```
In [3]: data
```

```
Out[3]:
```

	Id	MSSubClass	MSZoning	LotArea	LotConfig	BldgType	OverallCond	YearBuilt	YearRemodAdd	Exterior1st	BsmtFinSF2	TotalBsmtSF	SalePrice
0	0	60	RL	8450	Inside	1Fam	5	2003	2003	VinylSd	0.0	856.0	208500.0
1	1	20	RL	9600	FR2	1Fam	8	1976	1976	MetalSd	0.0	1262.0	181500.0
2	2	60	RL	11250	Inside	1Fam	5	2001	2002	VinylSd	0.0	920.0	223500.0
3	3	70	RL	9550	Corner	1Fam	5	1915	1970	Wd Sdng	0.0	756.0	140000.0
4	4	60	RL	14260	FR2	1Fam	5	2000	2000	VinylSd	0.0	1145.0	250000.0
...
2914	2914	160	RM	1936	Inside	Twnhs	7	1970	1970	CemntBd	0.0	546.0	NaN
2915	2915	160	RM	1894	Inside	TwnhsE	5	1970	1970	CemntBd	0.0	546.0	NaN
2916	2916	20	RL	20000	Inside	1Fam	7	1960	1996	VinylSd	0.0	1224.0	NaN
2917	2917	85	RL	10441	Inside	1Fam	5	1992	1992	HdBoard	0.0	912.0	NaN
2918	2918	60	RL	9627	Inside	1Fam	5	1993	1994	HdBoard	0.0	996.0	NaN

2919 rows × 13 columns

```
In [4]: data['SalePrice'] = data['SalePrice'].fillna(data['SalePrice'].mean())
```

```
In [5]: df = pd.DataFrame(data)
print(df.isnull().sum())
```

```
Id          0
MSSubClass  0
MSZoning    4
LotArea      0
LotConfig     0
BldgType     0
OverallCond   0
YearBuilt     0
YearRemodAdd  0
Exterior1st   1
BsmtFinSF2   1
TotalBsmtSF   1
SalePrice     0
dtype: int64
```

```
In [6]: data = data.dropna()
```

```
In [8]: from sklearn.preprocessing import LabelEncoder
obj_cols = ['MSZoning', 'LotConfig', 'BldgType', 'Exterior1st']
le = LabelEncoder()
for col in obj_cols:
    data[col] = le.fit_transform(data[col])
```

```
C:\Users\Joshitha\AppData\Local\Temp\ipykernel_27232\3884859511.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data[col] = le.fit_transform(data[col])
C:\Users\Joshitha\AppData\Local\Temp\ipykernel_27232\3884859511.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data[col] = le.fit_transform(data[col])
C:\Users\Joshitha\AppData\Local\Temp\ipykernel_27232\3884859511.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data[col] = le.fit_transform(data[col])
C:\Users\Joshitha\AppData\Local\Temp\ipykernel_27232\3884859511.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
data[col] = le.fit_transform(data[col])
C:\Users\Joshitha\AppData\Local\Temp\ipykernel_27232\3884859511.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
In [9]: df = pd.DataFrame(data)
print(df.isnull().sum())
```

```
Id          0
MSSubClass  0
MSZoning    0
LotArea      0
LotConfig     0
BldgType     0
OverallCond   0
YearBuilt     0
YearRemodAdd  0
Exterior1st   0
BsmtFinSF2   0
TotalBsmtSF   0
SalePrice     0
dtype: int64
```

```
In [10]: X = data.drop(['SalePrice'], axis=1)
Y = data['SalePrice']
```

```
In [11]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42, shuffle=True)
```

```
In [12]: scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
In [13]: model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, Y_train)
```

```
Out[13]:
```

▼ RandomForestRegressor
● ●

► Parameters

```
In [14]: Y_pred = model.predict(X_test_scaled)
```

```
In [15]: from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
print(f"MAE : {mean_absolute_error(Y_test, Y_pred):.2f}")
print(f"R² : {r2_score(Y_test, Y_pred):.3f}")
print(mean_squared_error(Y_test, Y_pred))
```

```
MAE : 12,043.46
R² : 0.810
694256483.9704959
```

```
In [16]: input_data = (0, 60, 3, 8450, 4, 0, 5, 2003, 2003, 12, 0.0, 856.0)
input_data_array = np.asarray(input_data).reshape(1, -1)
input_data_scaled = scaler.transform(input_data_array)
predicted_price = model.predict(input_data_scaled)
print(f"Predicted House Price: {predicted_price[0]:.2f}")
```

Predicted House Price: 201463.40

```
C:\Users\Joshitha\AppData\Roaming\Python\Python313\site-packages\sklearn\utils\validation.py:2749: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
warnings.warn(
```