In [1]:
```
import pandas as pd
```

In [2]:
```
#import the dataset
df=pd.read_csv('houseprice.csv')
```

In [3]:
```
#show data info
df
```

Out[3]:

	Price	Bedroom	Space	Room	Lot	Tax	Bathroom	Garage	Condition
0	53.0	2.0	967.0	5.0	39.0	652.0	1.5	0.0	0.0
1	55.0	2.0	815.0	5.0	33.0	1000.0	1.0	2.0	1.0
2	56.0	3.0	900.0	5.0	35.0	897.0	1.5	1.0	0.0
3	58.0	3.0	1007.0	6.0	24.0	964.0	1.5	2.0	0.0
4	64.0	3.0	1100.0	7.0	50.0	1099.0	1.5	1.5	0.0
...	...	...	...	...	...	...	...	...	...
152	44.0	3.0	820.0	5.0	27.0	NaN	1.0	0.0	1.0
153	43.0	2.0	593.0	4.0	30.0	465.0	2.0	1.0	0.0
154	53.0	3.0	716.0	6.0	30.0	585.0	1.0	2.0	1.0
155	62.0	4.0	951.0	7.0	30.0	895.0	2.0	1.0	0.0
156	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

157 rows × 9 columns

In [8]:
```
#Handle missing values
df_filled = df.fillna(df.median())
```

In [9]:
```
#Select features and target variable
X = df_filled[['Space', 'Bedroom', 'Bathroom']]
y = df_filled['Price']
```

In [11]:
```
from sklearn.model_selection import train_test_split
```

In [12]:
```
#Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [13]:
```
from sklearn.linear_model import LinearRegression
```

In [14]:
```
#Initialize and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[14]:
```
▾ LinearRegression

LinearRegression()
```

In [15]:
```
#Predict house prices on the test set
y_pred = model.predict(X_test)
```

In [16]:
```
from sklearn.metrics import mean_squared_error
```

In [17]:
```
#Evaluate the model performance
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse}")
print(f"Model Coefficients: {model.coef_}")
print(f"Model Intercept: {model.intercept_}")
```

```
Mean Squared Error (MSE): 87.55499137071118
Model Coefficients: [ 0.01915244 -1.94098617  7.16845819]
Model Intercept: 31.22761796362404
```

In [18]:
```
#Predict on new data
new_data = pd.DataFrame({'Space': [1000], 'Bedroom': [3], 'Bathroom': [2]})
predicted_price = model.predict(new_data)
print(f"Predicted Price: {predicted_price[0]}")
```

```
Predicted Price: 58.89401451165316
```

In [ ]:
```

```