```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('house price data.csv')
```

```
data.head()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	wate
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	

```
# Data preprocessing
data = data.dropna()
data = data.drop_duplicates()
```

```
#Since there is date column therefore we directly cannot apply linear regressic
if 'date' in data.columns:
    data['date'] = pd.to_datetime(data['date'])
    data['year'] = data['date'].dt.year
    data['month'] = data['date'].dt.month
    data['day'] = data['date'].dt.day
    data['hour'] = data['date'].dt.hour
    data = data.drop(columns=['date'])
```

```
# Check if there are any other non-numeric columns and encode them if necessary
for column in data.select_dtypes(include=['object']).columns:
    data[column] = pd.factorize(data[column])[0]
```

```
# Split the dataset
X = data.iloc[:, :-1]
y = data.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random
```

```
# Implement Model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
# Evaluate on training data
y_train_pred = model.predict(X_train)
mse_train = mean_squared_error(y_train, y_train_pred)
r2_train = r2_score(y_train, y_train_pred)
print(f"Training MSE: {mse_train}")
print(f"Training R2: {r2_train}")
```

```
Training MSE: 0.0
Training R2: 1.0
```

```
# Model Evaluation
y_test_pred = model.predict(X_test)
mse_test = mean_squared_error(y_test, y_test_pred)
r2_test = r2_score(y_test, y_test_pred)
print(f"Test MSE: {mse_test}")
print(f"Test R2: {r2_test}")
```

```
Test MSE: 0.0
Test R2: 1.0
```

Start coding or generate with AI.