

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

```
In [2]: from sklearn.model_selection import train_test_split
```

```
In [3]: from sklearn.linear_model import LinearRegression
```

```
In [4]: from sklearn.metrics import mean_squared_error, r2_score
```

```
In [5]: import matplotlib.pyplot as plt
```

```
In [12]: ds=pd.read_csv("train.csv")
```

```
In [13]: ds
```

Out[13]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	..
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	..
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	..
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	..
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	..
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	..
...	...	...	...	...	...	...	...	...	...	...	..
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	..
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	..
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	..
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	..
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	..

1460 rows × 81 columns



```
In [14]: features = ["GrLivArea", "BedroomAbvGr", "FullBath"]
```

```
In [16]: X = ds[features]
```

```
In [17]: y = ds["SalePrice"]
```

```
In [18]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [19]: model = LinearRegression()
```

```
In [20]: model.fit(X_train, y_train)
```

```
Out[20]: 

LinearRegression



LinearRegression()


```

```
In [21]: y_pred = model.predict(X_test)
```

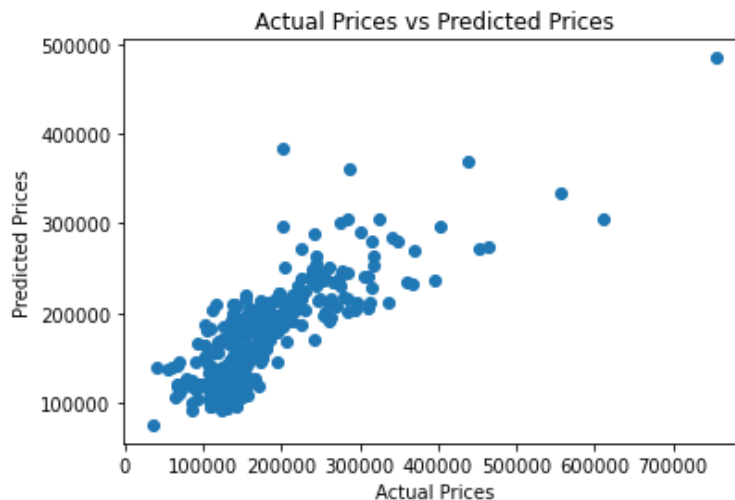
```
In [22]: mse = mean_squared_error(y_test, y_pred)
```

```
In [23]: r2 = r2_score(y_test, y_pred)
```

```
In [24]: print(f'Mean Squared Error: {mse}')  
print(f'R-squared: {r2}')
```

Mean Squared Error: 2806426667.247853  
R-squared: 0.6341189942328371

```
In [25]: # Plotting predictions against actual values  
plt.scatter(y_test, y_pred)  
plt.xlabel("Actual Prices")  
plt.ylabel("Predicted Prices")  
plt.title("Actual Prices vs Predicted Prices")  
plt.show()
```



```
In [27]: new_data = pd.DataFrame({'GrLivArea': [2000], 'BedroomAbvGr': [3], 'FullBath': [2]})  
predicted_price = model.predict(new_data)  
print(f'Predicted Price: {predicted_price[0]}')
```

Predicted Price: 240377.51479736285

```
In [ ]:
```

