King County House Sales

A report by Jacinta Fiona

King County House Sales

Title:
Microsoft Movie Studio

Author:
Jacinta Fiona

Copyright Date:
09/07/2023

Introduction

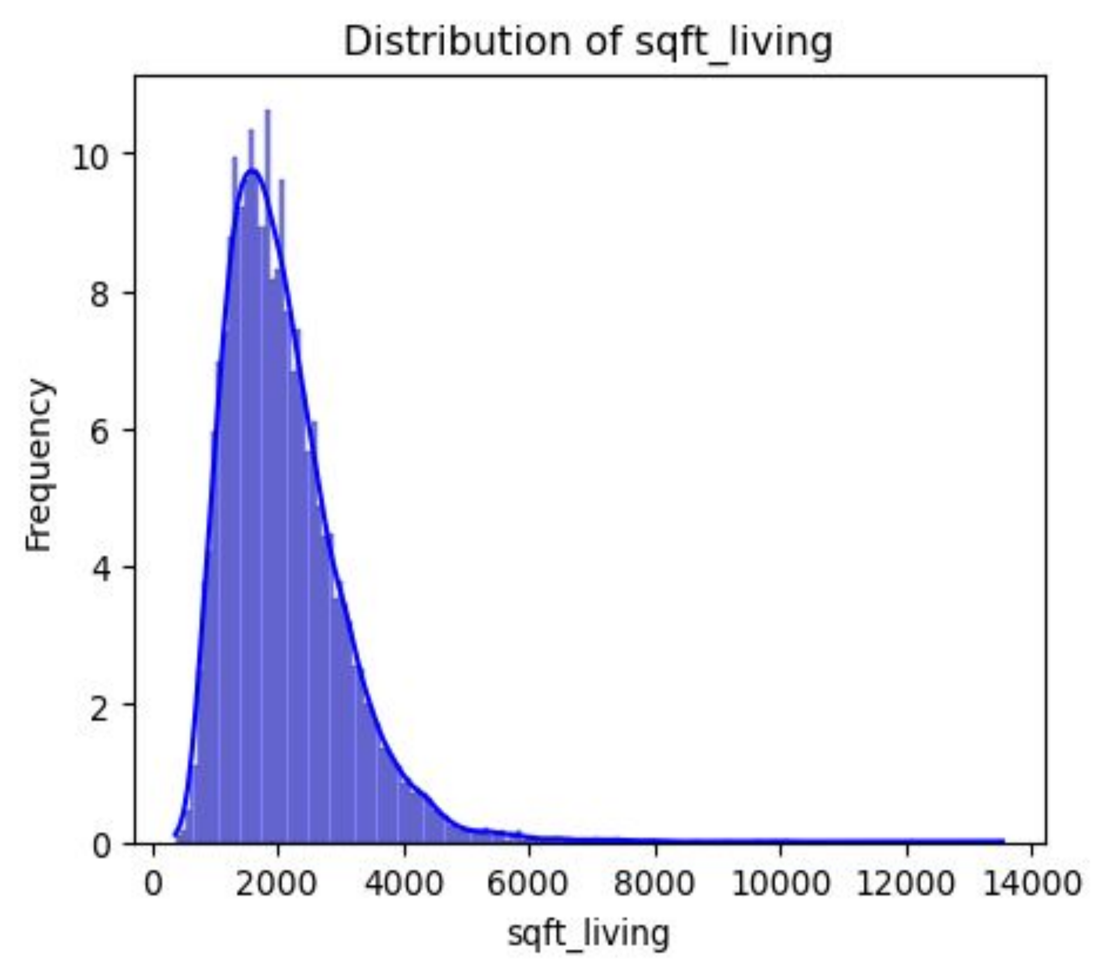
When done strategically, remodeling improves your home’s value and marketability. You might notice that a home that is similar to your home in age, size and layout has been appraised at a much higher value than your home. The most likely reason is that the home has been upgraded. Homes that have been upgraded with modern features or layouts attract more homebuyers and higher offers.

Problem statement

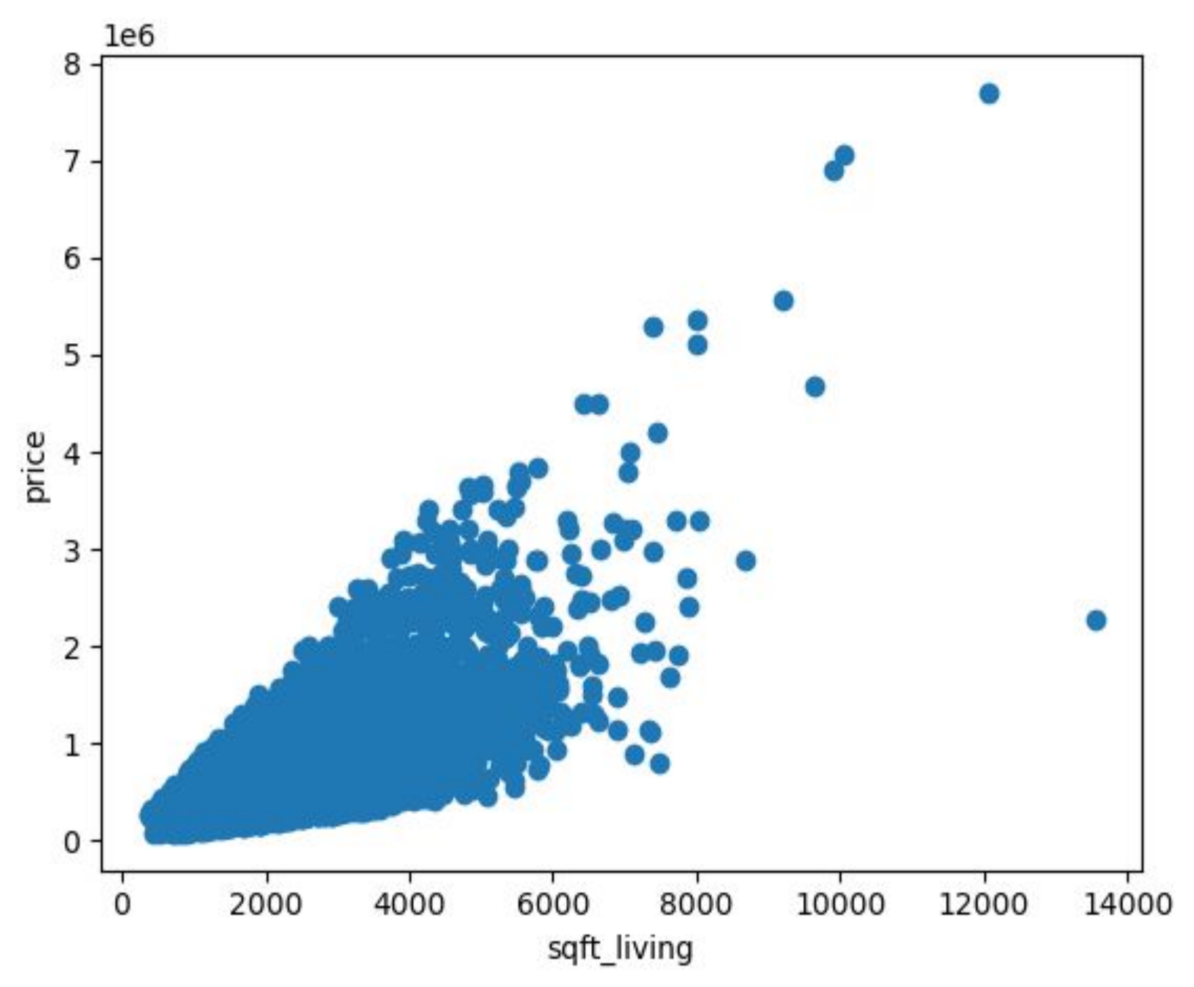
The business problem is to provide advice to homeowners about how home renovations might increase the estimated value of their homes using multiple linear regression to predict the estimated value of homes in King County.

Main Objective

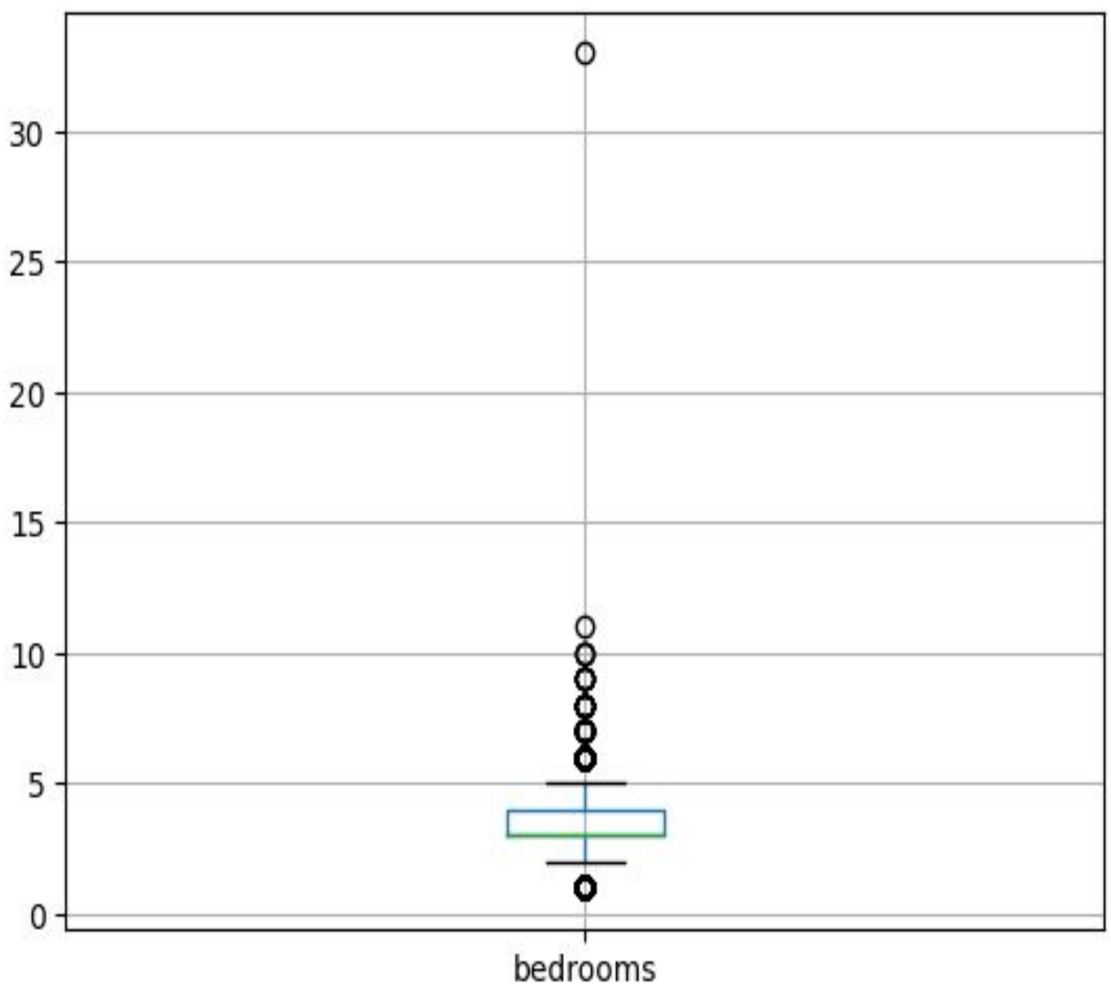
The main objective of this project is to develop a multiple linear regression model that predicts the estimated value of homes in King County.



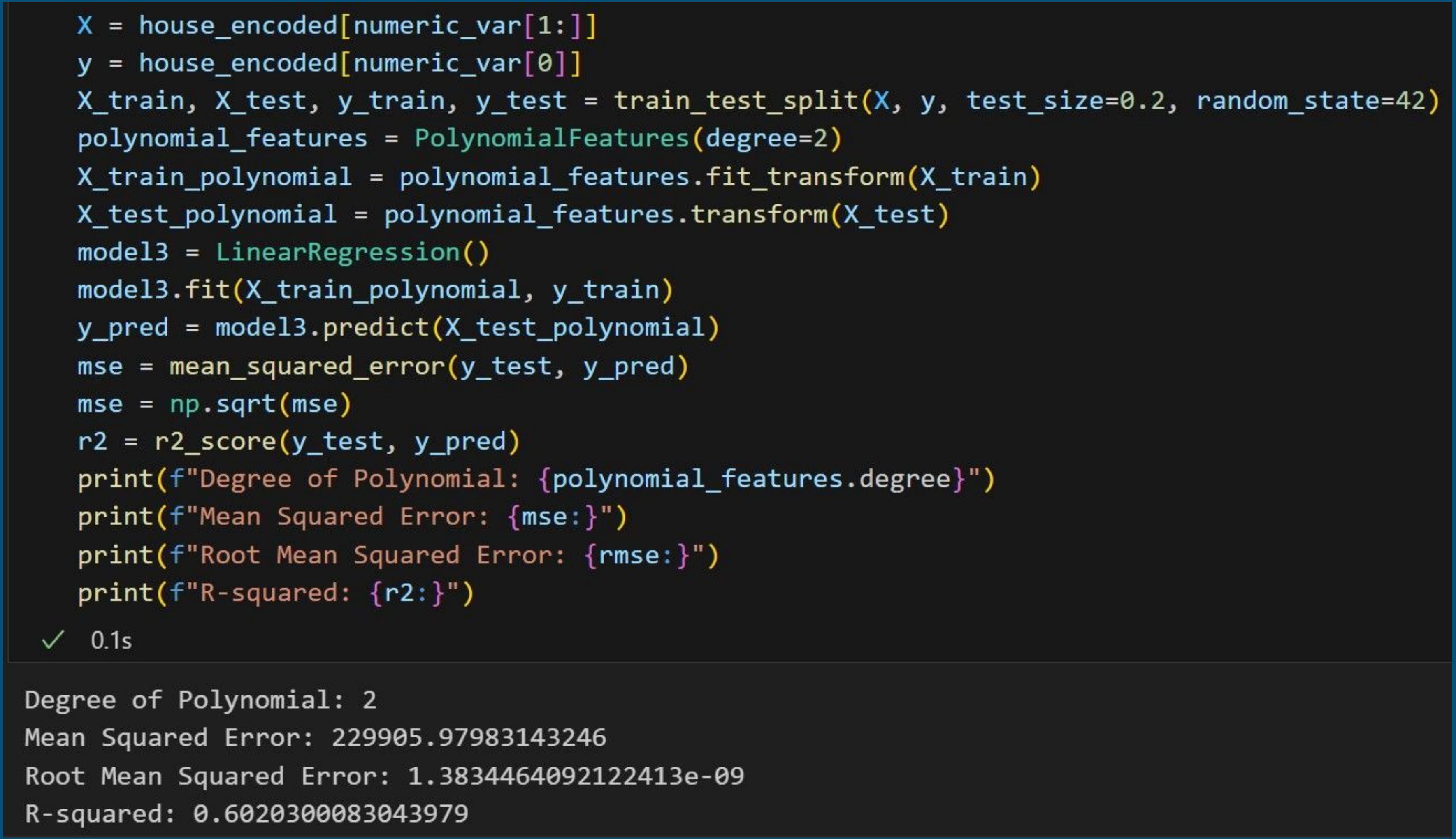
A plot showing the descriptive statistics for square footage of living.



A linear relationship between square footage of living space in the home and price.

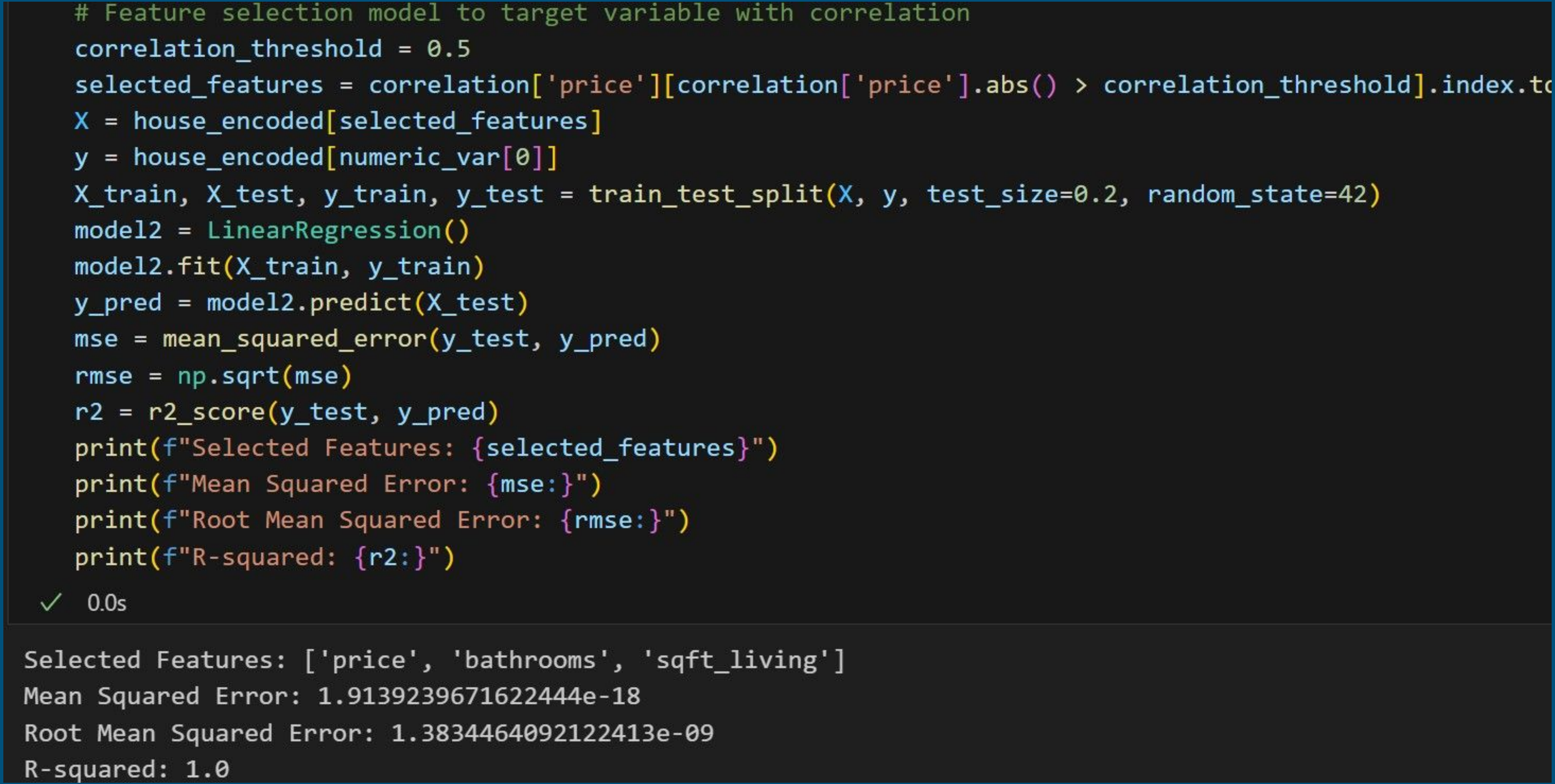


A box plot showing the distribution in the number of bedrooms.



```python
X = house_encoded[numeric_var[1:]]
y = house_encoded[numeric_var[0]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
polynomial_features = PolynomialFeatures(degree=2)
X_train_polynomial = polynomial_features.fit_transform(X_train)
X_test_polynomial = polynomial_features.transform(X_test)
model3 = LinearRegression()
model3.fit(X_train_polynomial, y_train)
y_pred = model3.predict(X_test_polynomial)
mse = mean_squared_error(y_test, y_pred)
mse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"Degree of Polynomial: {polynomial_features.degree}")
print(f"Mean Squared Error: {mse:}")
print(f"Root Mean Squared Error: {rmse:}")
print(f"R-squared: {r2:}")
```

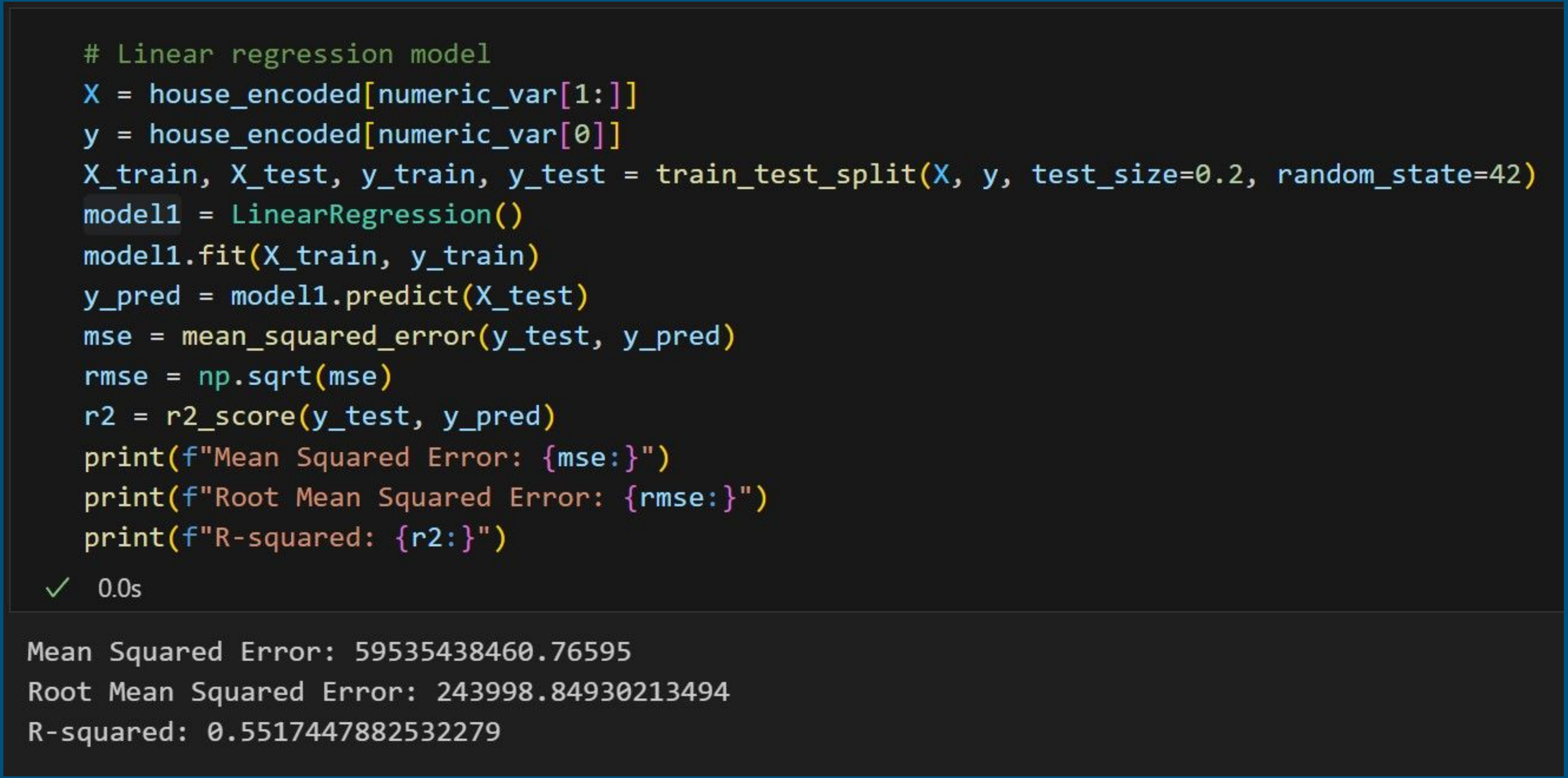
```
Degree of Polynomial: 2
Mean Squared Error: 229905.97983143246
Root Mean Squared Error: 1.3834464092122413e-09
R-squared: 0.6020300083043979
```



```python
# Feature selection model to target variable with correlation
correlation_threshold = 0.5
selected_features = correlation['price'][correlation['price'].abs() > correlation_threshold].index.to
X = house_encoded[selected_features]
y = house_encoded[numeric_var[0]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model2 = LinearRegression()
model2.fit(X_train, y_train)
y_pred = model2.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"Selected Features: {selected_features}")
print(f"Mean Squared Error: {mse:}")
print(f"Root Mean Squared Error: {rmse:}")
print(f"R-squared: {r2:}")
```

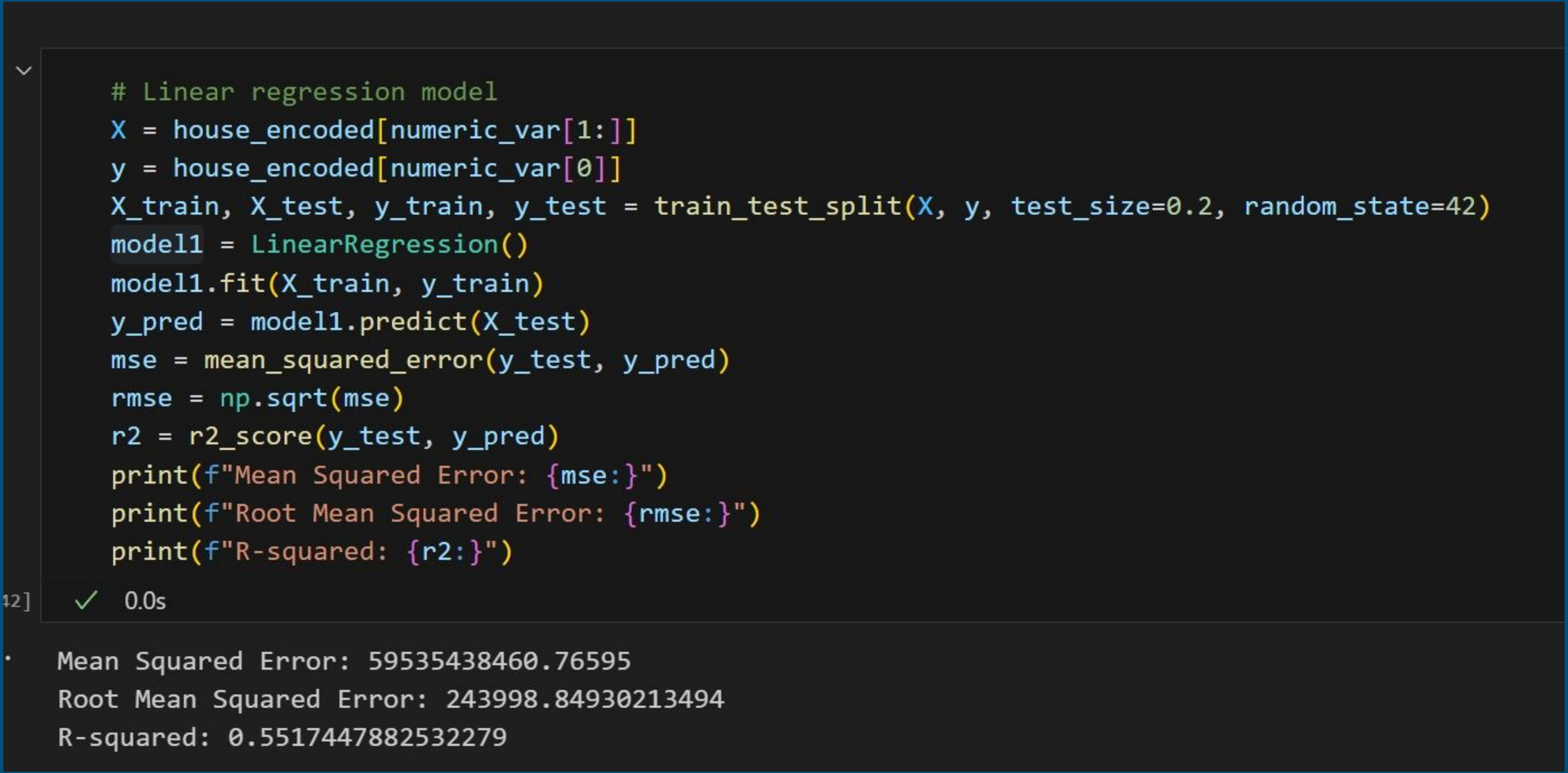
✓ 0.0s

```
Selected Features: ['price', 'bathrooms', 'sqft_living']
Mean Squared Error: 1.913923967162244e-18
Root Mean Squared Error: 1.3834464092122413e-09
R-squared: 1.0
```



```python
# Linear regression model
X = house_encoded[numeric_var[1:]]
y = house_encoded[numeric_var[0]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model1 = LinearRegression()
model1.fit(X_train, y_train)
y_pred = model1.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:}")
print(f"Root Mean Squared Error: {rmse:}")
print(f"R-squared: {r2:}")
```

```
Mean Squared Error: 59535438460.76595
Root Mean Squared Error: 243998.84930213494
R-squared: 0.5517447882532279
```



```python
# Linear regression model
X = house_encoded[numeric_var[1:]]
y = house_encoded[numeric_var[0]]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model1 = LinearRegression()
model1.fit(X_train, y_train)
y_pred = model1.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:}")
print(f"Root Mean Squared Error: {rmse:}")
print(f"R-squared: {r2:}")
```

```
Mean Squared Error: 59535438460.76595
Root Mean Squared Error: 243998.84930213494
R-squared: 0.5517447882532279
```

Conclusions

- The baseline model has a Root Mean Squared Error of 1.39e-9 and explains 100% of the variance.
- The feature selection model has a Root Mean Squared Error of 1.38e-9 and explains 100% of the variance.
- The Linear regression model has a Root Mean Squared Error of 243998.85 and explains 55% of the variance.
- The Polynomial regression model has a Root Mean Squared Error of 1.38e-9 and explains 60% of the variance.
- The feature selection model has the best balance of coefficients.

Reccomendations

- The model we should use the feature selection model as it has the lowest Root Mean Squared Error which indicates a better model performance.
- It would also be recommended that we use the feature selection model since it explain 100% of the variance showing that the whole proportion of the variance is accounted for by the model.
- There is a linear relationship between the price and square foot of living for homeowners who want to develop pricier units for commercial purposes.