real-estate

May 22, 2024

[2]:
```
import pandas as pd

# Load the dataset
data_path = './marketing.csv'
df = pd.read_csv(data_path)

# Display the first five rows of the dataframe
df.head()
```

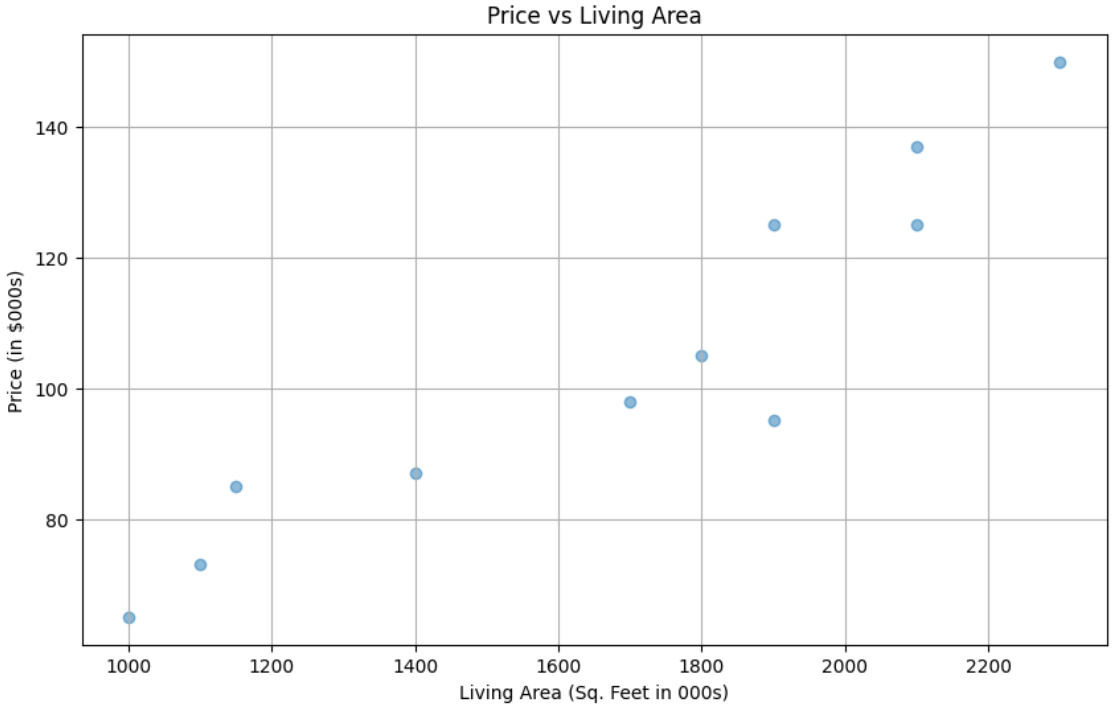
[2]:

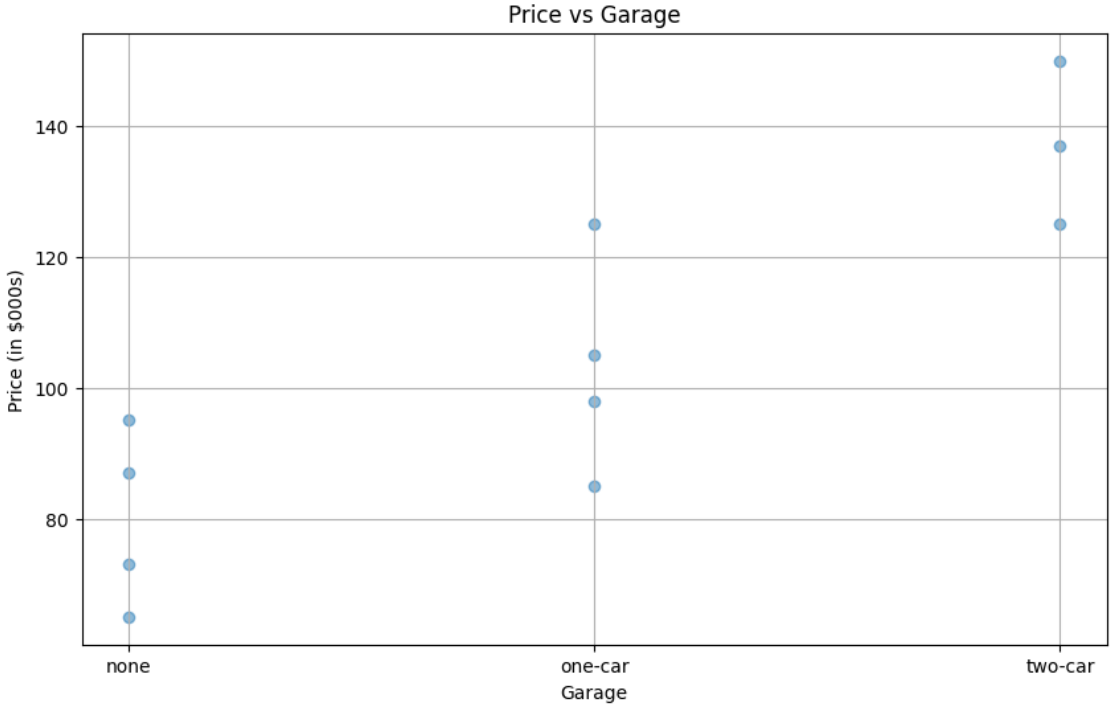
	Sq. Feet (in 000s)	Garage	Bedrooms	Price (in $000s)
0	1000	none	2	65
1	1100	none	2	73
2	1150	one-car	2	85
3	1400	none	3	87
4	1700	one-car	3	98

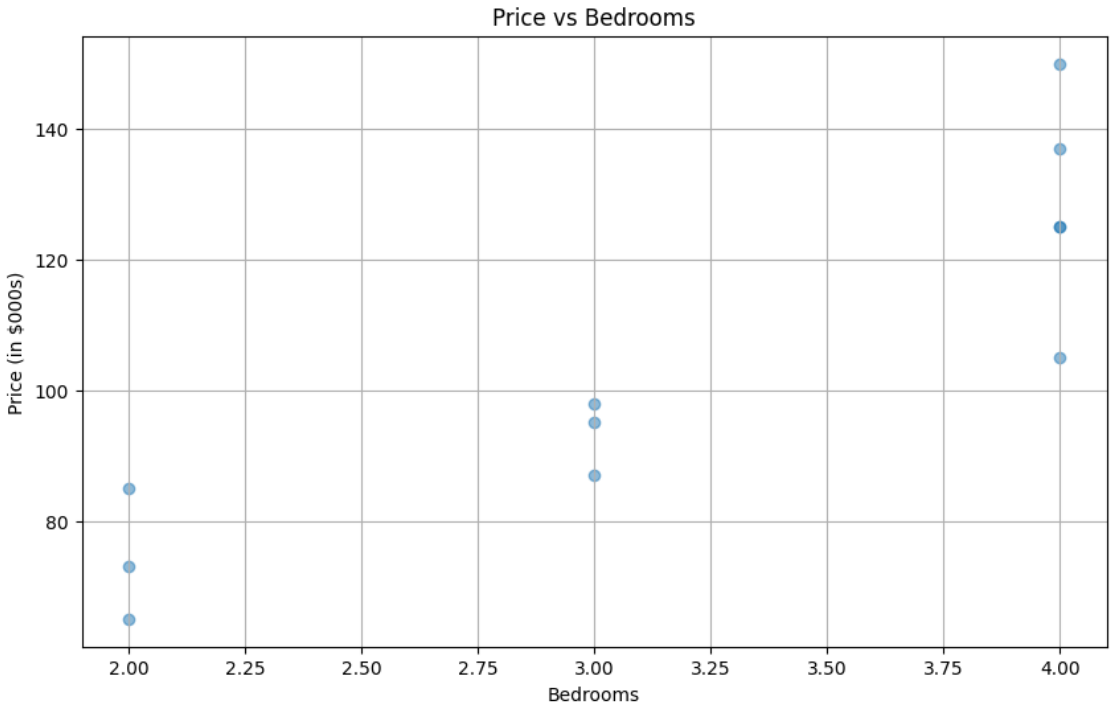
[3]:
```
import matplotlib.pyplot as plt

# Scatterplot for Price vs Living Area
plt.figure(figsize=(10, 6))
plt.scatter(df['Sq. Feet (in 000s)'], df['Price (in $000s)'], alpha=0.5)
plt.title('Price vs Living Area')
plt.xlabel('Living Area (Sq. Feet in 000s)')
plt.ylabel('Price (in $000s)')
plt.grid(True)
plt.show()

# Scatterplots for Price vs other attributes
for column in df.columns:
    if column != 'Price (in $000s)' and column != 'Sq. Feet (in 000s)':
        plt.figure(figsize=(10, 6))
        plt.scatter(df[column], df['Price (in $000s)'], alpha=0.5)
        plt.title(f'Price vs {column}')
        plt.xlabel(column)
        plt.ylabel('Price (in $000s)')
        plt.grid(True)
        plt.show()
```


Price vs Living Area
Price (in $000s)
140
120
100
80
1000
1200
1400
1600
1800
2000
2200
Living Area (Sq. Feet in 000s)


Price vs Garage
Price (in $000s)
140
120
100
80
none
one-car
two-car
Garage



[4]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Define the feature (Living Area) and target (Price)
X = df[['Sq. Feet (in 000s)']]
y = df['Price (in $000s)']

# Split the data 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)

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Display the model coefficients
print(f'Intercept: {model.intercept_}')
print(f'Coefficient: {model.coef_[0]}')
```

```
Intercept: 13.826900855095985
Coefficient: 0.05368153455049684
```

[5]:
```
# Calculate the R^2 score
r_squared = model.score(X_test, y_test)
```

```
print(f'Coefficient of Determination (R^2): {r_squared}')
```

```
Coefficient of Determination (R^2): 0.9442619843108755
```

[8]:
```
# Example prediction
living_area_example = pd.DataFrame({'Sq. Feet (in 000s)': [2.0]})  # Create a␣
 ↪DataFrame with the same column name
predicted_price = model.predict(living_area_example)
print(f'Predicted Price for {living_area_example["Sq. Feet (in 000s)"][0]}000␣
 ↪sq ft: ${predicted_price[0]:,.2f}000')
```

```
Predicted Price for 2.0000 sq ft: $13.93000
```

[7]:
```
import numpy as np
from sklearn.metrics import mean_squared_error

# Calculate predictions on the test set
y_pred = model.predict(X_test)

# Calculate the Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')

# Summary of the model quality
print(f'The R^2 value of {r_squared} indicates that the model explains␣
 ↪{r_squared*100:.2f}% of the variance in the selling price based on the␣
 ↪living area.')
print(f'The RMSE value of {rmse} provides an estimate of the average prediction␣
 ↪error in thousands of dollars.')
```

```
Mean Squared Error (MSE): 48.35582516674265
Root Mean Squared Error (RMSE): 6.953835284700282
The R^2 value of 0.9442619843108755 indicates that the model explains 94.43% of
the variance in the selling price based on the living area.
The RMSE value of 6.953835284700282 provides an estimate of the average
prediction error in thousands of dollars.
```