In [2]:
```
!pip install opendatasets
import opendatasets as od
od.download("https://www.kaggle.com/datasets/muhammadbinimran/housing-price-prediction-data")
```

```
Requirement already satisfied: opendatasets in /usr/local/lib/python3.10/dist-packages (0.1.22)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from opendataset
s) (4.67.1)
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (from opendatase
ts) (1.6.17)
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from opendataset
s) (8.1.7)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle-
>opendatasets) (1.17.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.10/dist-packages (fro
m kaggle->opendatasets) (2024.12.14)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from k
aggle->opendatasets) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle->
opendatasets) (2.32.3)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from ka
ggle->opendatasets) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle->o
pendatasets) (2.2.3)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle->op
endatasets) (6.2.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from blea
ch->kaggle->opendatasets) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (fr
om python-slugify->kaggle->opendatasets) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-package
s (from requests->kaggle->opendatasets) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requ
ests->kaggle->opendatasets) (3.10)
Skipping, found downloaded files in "./housing-price-prediction-data" (use force=True to force do
wnload)
```

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


data = pd.read_csv("/content/housing-price-prediction-data/housing_price_dataset.csv")
```

In [4]:
```
data.head()
```

Out[4]:

	SquareFeet	Bedrooms	Bathrooms	Neighborhood	YearBuilt	Price
0	2126	4	1	Rural	1969	215355.283618
1	2459	3	2	Rural	1980	195014.221626
2	1860	2	1	Suburb	1970	306891.012076
3	2294	2	1	Urban	1996	206786.787153
4	2130	5	2	Suburb	2001	272436.239065

In [5]:
```
data.isna().sum()
```

Out[5]:

	0
SquareFeet	0
Bedrooms	0
Bathrooms	0
Neighborhood	0
YearBuilt	0
Price	0

dtype: int64

In [6]:

```
from sklearn import linear_model
from sklearn import preprocessing

le = preprocessing.LabelEncoder()
data["Neighborhood"] = le.fit_transform(data["Neighborhood"])

data.head()
```

Out[6]:

	SquareFeet	Bedrooms	Bathrooms	Neighborhood	YearBuilt	Price
0	2126	4	1	0	1969	215355.283618
1	2459	3	2	0	1980	195014.221626
2	1860	2	1	1	1970	306891.012076
3	2294	2	1	2	1996	206786.787153
4	2130	5	2	1	2001	272436.239065

In [7]:

```
reg = linear_model.LinearRegression()
reg.fit(data[["SquareFeet","Bedrooms","Bathrooms","Neighborhood","YearBuilt"]], data.Price)
```

Out[7]:

```
▾ LinearRegression
LinearRegression()
```

In [8]:

```
reg.coef_
```

Out[8]:

```
array([  99.3491404 , 5076.32859775, 2830.03311391,  773.9244851 ,
        -11.01201598])
```

In [9]:

```
reg.intercept_
```

Out[9]:

```
23178.580465551786
```

Predict price for following features: SquareFeet 2500

Bedrooms 3

Bathrooms 2

Neighborhood Urban (2)- for, Rural 0, Suburban 1, Urban 2

YearBuilt 2007

In [15]:

```
reg.predict([[2500,3,2,2,2007]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[15]:

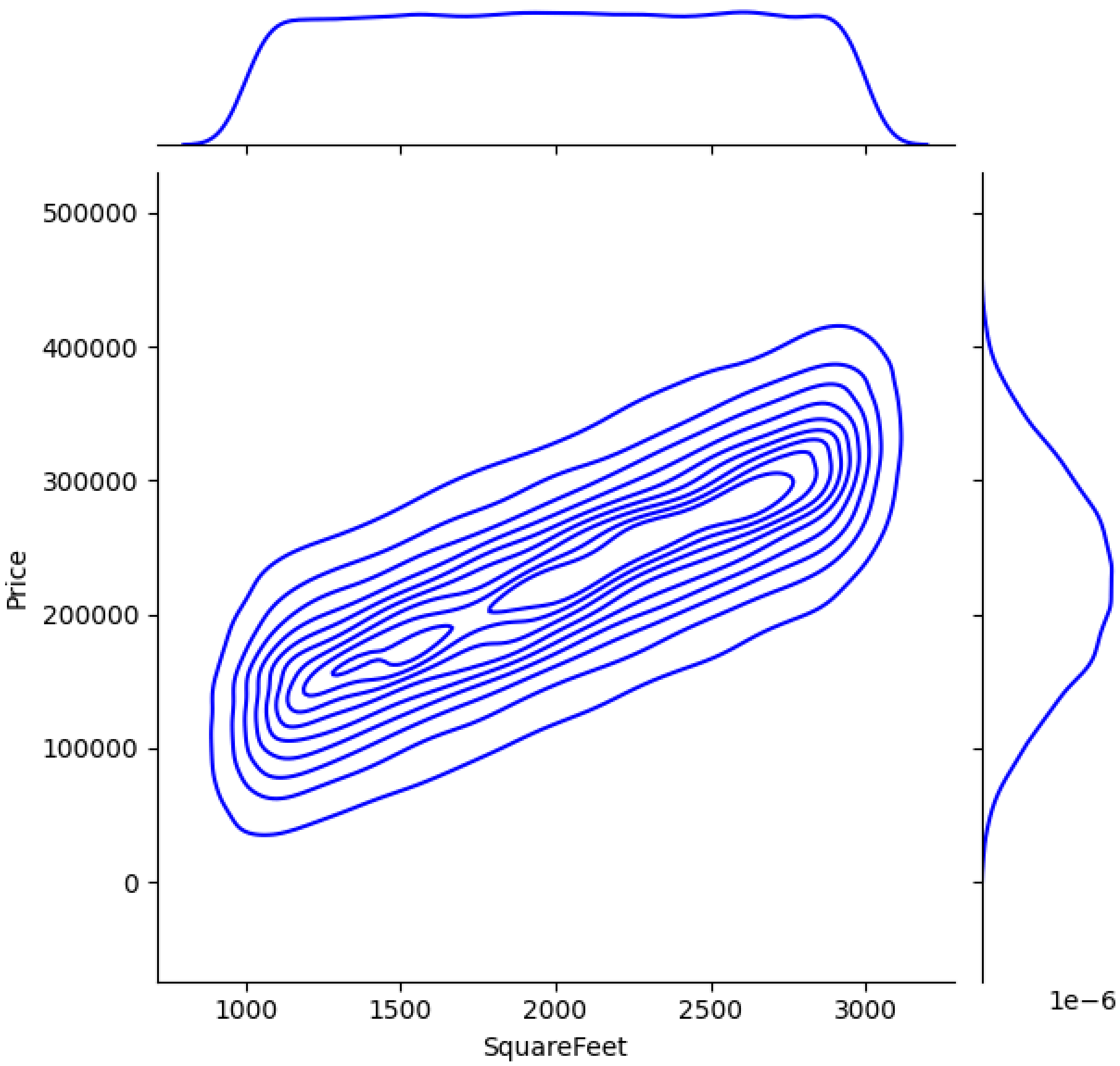
```
array([271887.21639664])
```

In [16]:

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.jointplot(x="SquareFeet", y="Price", data=data, kind = "kde", color = "b")
```

Out[16]:

```
<seaborn.axisgrid.JointGrid at 0x7bb97a2ddff0>
```



In [17]:

```
data.corr()
```

Out[17]:

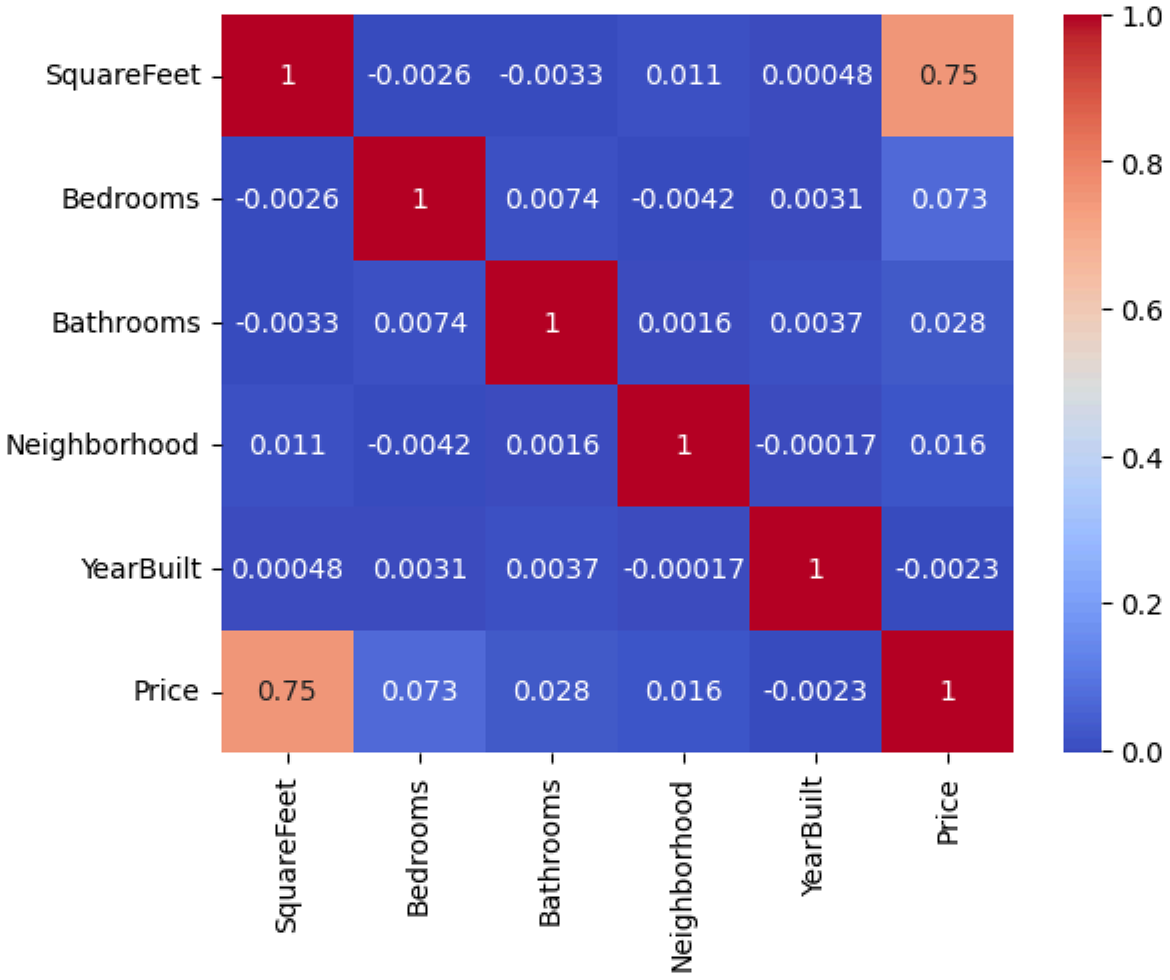
	SquareFeet	Bedrooms	Bathrooms	Neighborhood	YearBuilt	Price
SquareFeet	1.000000	-0.002638	-0.003275	0.011186	0.000482	0.750720
Bedrooms	-0.002638	1.000000	0.007405	-0.004208	0.003147	0.072624
Bathrooms	-0.003275	0.007405	1.000000	0.001613	0.003748	0.028418
Neighborhood	0.011186	-0.004208	0.001613	1.000000	-0.000174	0.016429
YearBuilt	0.000482	0.003147	0.003748	-0.000174	1.000000	-0.002288
Price	0.750720	0.072624	0.028418	0.016429	-0.002288	1.000000

In [18]:

```
sns.heatmap(data.corr(), annot = True, cmap = "coolwarm")
```

Out[18]:

```
<Axes: >
```


SquareFeet
Bedrooms
Bathrooms
Neighborhood
YearBuilt
Price
SquareFeet
Bedrooms
Bathrooms
Neighborhood
YearBuilt
Price