Linear regression

1. Import and visualize data

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

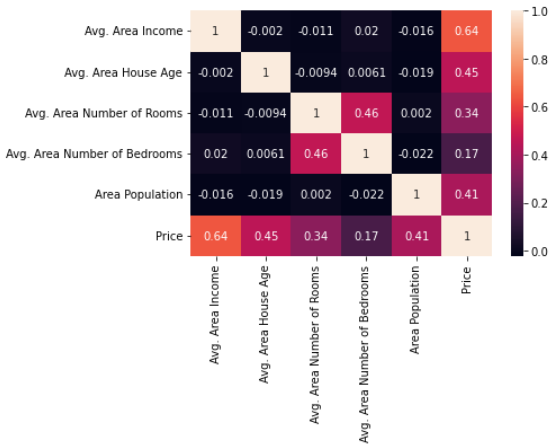
In [2]:
```
USAhousing = pd.read_csv('USA_Housing.csv')
```

In [3]:
```
USAhousing.columns
```

Out[3]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

In [4]:
```
sns.heatmap(USAhousing.corr(), annot=True)
```

Out[4]:
```
<AxesSubplot:>
```



2. Split data set to training data and testing data

In [5]:
```
from sklearn.model_selection import train_test_split
```

In [6]:
```
USAhousing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

In [7]:
```
x=USAhousing.iloc[:,0:4]
```

In [8]:
```
y=USAhousing['Price']
```

In [9]:
```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.4)
```

3. Build the linear regressing model

In [10]:
```
from sklearn.linear_model import LinearRegression
```

In [11]:
```
lm = LinearRegression()
```

In [12]:
```
lm.fit(X_train,y_train)
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
print(lm.intercept_)
```

```
-2053268.9801055775
```

In [14]:
```
coeff_df = pd.DataFrame(lm.coef_,x.columns,columns=['Coefficient'])
```

In [15]:
```
lm.intercept_
```

Out[15]:
```
-2053268.9801055775
```

In [16]:
```
lm.coef_
```

Out[16]:
```
array([ 2.16371590e+01,  1.62326458e+05,  1.19227576e+05, -7.15972990e+02])
```

In [17]:
```
coeff_df
```

Out[17]:

	Coefficient
Avg. Area Income	21.637159
Avg. Area House Age	162326.458416
Avg. Area Number of Rooms	119227.576301
Avg. Area Number of Bedrooms	-715.972990

4. Compare between original data and the result of prediction from the model

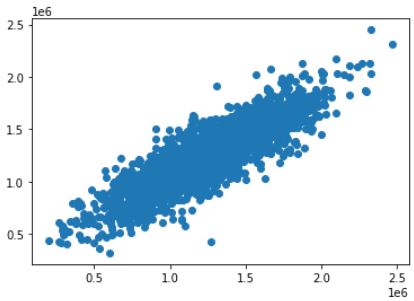
In [18]:
```
predictions = lm.predict(X_test)
```

In [19]:
```
predictions
```

Out[19]:
```
array([1648210.96766365,  897569.59255515, 1593447.07681869, ...,
       1513846.44789638,  980164.68214296, 1376068.44552167])
```

In [20]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,predictions)
```

Out[20]:
```
<matplotlib.collections.PathCollection at 0x7f988ebfd550>
```

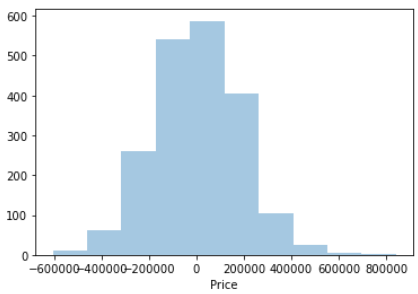


In [21]:
```
y_test == predictions
```

Out[21]:
```
3664    False
2476    False
3321    False
2601    False
2004    False
        ...
576     False
4495    False
3473    False
789     False
2610    False
Name: Price, Length: 2000, dtype: bool
```

In [22]:
```
sns.distplot((y_test-predictions),kde=False,bins=10);
```

```
/Users/jakapongtosunpul/opt/anaconda3/lib/python3.9/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



5. Calculate the error of the model

In [23]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
```

```
MAE: 144064.71162224805
```

In [24]:
```
print('MSE:', metrics.mean_squared_error(y_test, predictions))
```

```
MSE: 32647361244.609985
```

In [25]:
```
import numpy as np
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
RMSE: 180685.80808854353
```

In [ ]: