Mini Project Linear Regression Model

December 9, 2020

1 Linear Regression Machine Learning Project for House Price Prediction

1.0.1 Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

%matplotlib inline
```

1.0.2 Importing Data and Checking out.

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

[3]:
```
HouseDF.head()
```

[3]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05

```
                                                 Address
0  208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1  188 Johnson Views Suite 079\nLake Kathleen, CA...
2  9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3                          USS Barnett\nFPO AP 44820
4                         USNS Raymond\nFPO AE 09386
```

[4]:
```
HouseDF.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
```

[5]:
```
HouseDF.describe()
```

[5]:
```
       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
count       5000.000000          5000.000000                5000.000000   
mean       68583.108984             5.977222                   6.987792   
std        10657.991214             0.991456                   1.005833   
min        17796.631190             2.644304                   3.236194   
25%        61480.562388             5.322283                   6.299250   
50%        68804.286404             5.970429                   7.002902   
75%        75783.338666             6.650808                   7.665871   
max       107701.748378             9.519088                  10.759588   

       Avg. Area Number of Bedrooms  Area Population         Price  
count                   5000.000000      5000.000000  5.000000e+03  
mean                       3.981330     36163.516039  1.232073e+06  
std                        1.234137      9925.650114  3.531176e+05  
min                        2.000000       172.610686  1.593866e+04  
25%                        3.140000     29403.928702  9.975771e+05  
50%                        4.050000     36199.406689  1.232669e+06  
75%                        4.490000     42861.290769  1.471210e+06  
max                        6.500000     69621.713378  2.469066e+06  
```

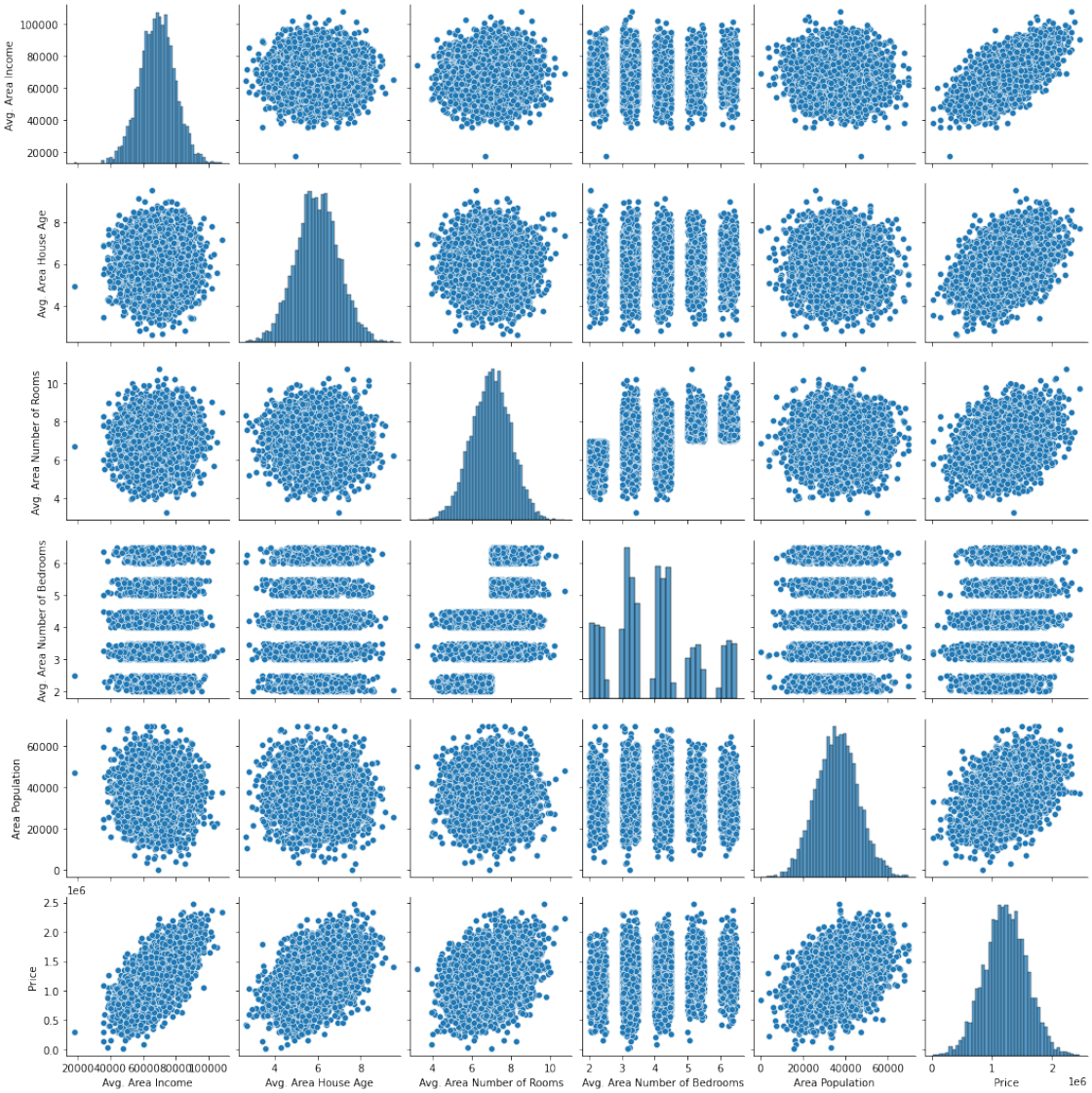
[6]:
```
HouseDF.columns
```

[6]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

1.1 Exploratory Data Analysis for House Price Prediction

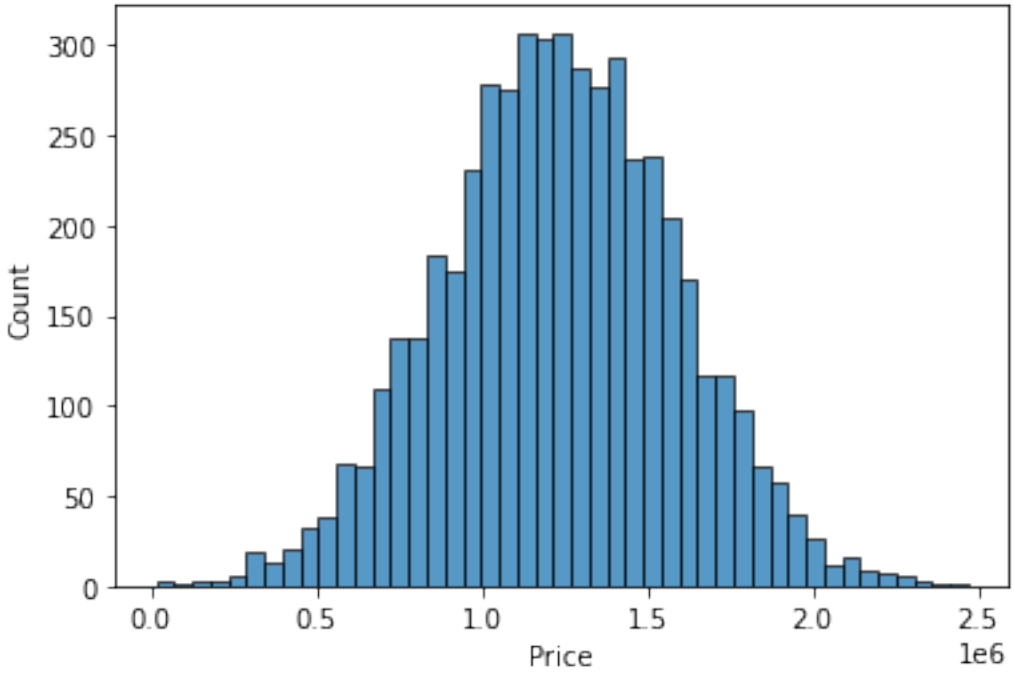
[7]:
```
sns.pairplot(HouseDF)
```

[7]:
```
<seaborn.axisgrid.PairGrid at 0x7efcd86a5880>
```



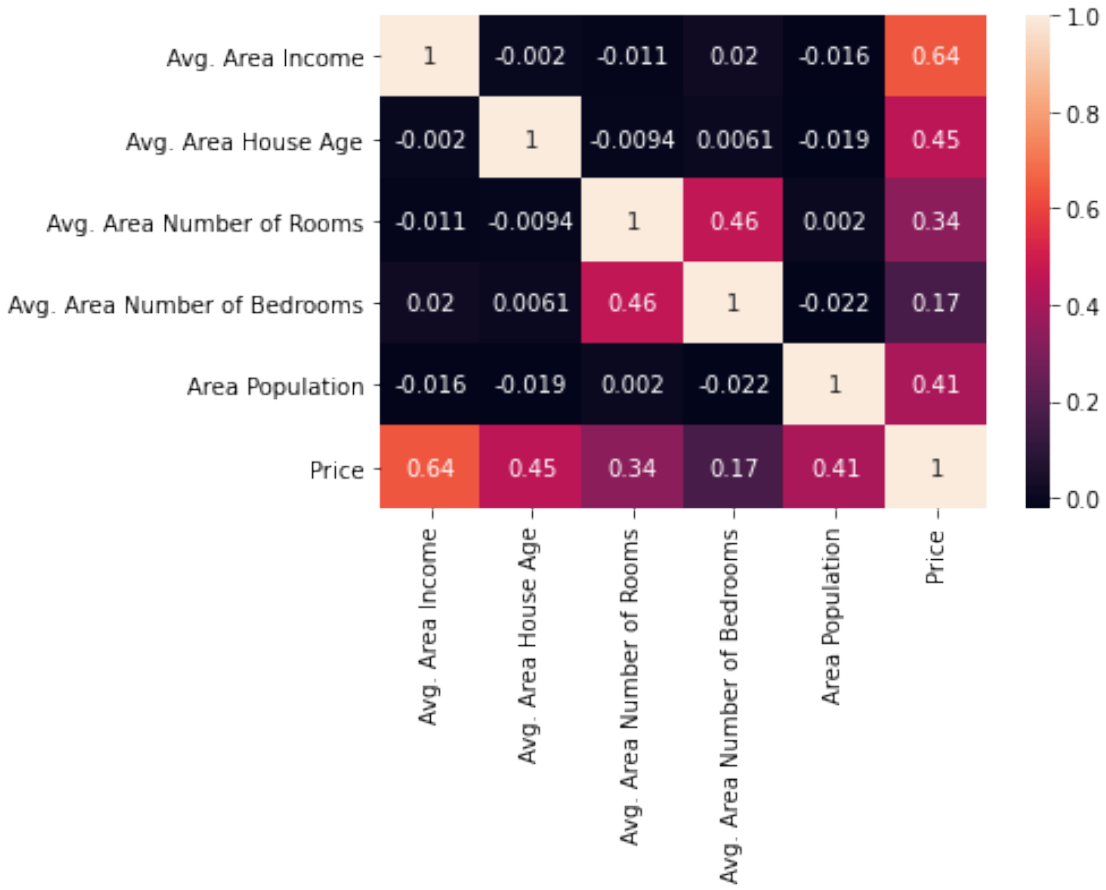
[8]:
```
sns.histplot(HouseDF['Price'])
```

[8]: <AxesSubplot:xlabel='Price', ylabel='Count'>



[12]:
```
sns.heatmap(HouseDF.corr() , annot=True)
```

[12]: <AxesSubplot:>



1.2 Training a Linear Regression Model

1.2.1 X and y List

[13]:
```
X = HouseDF[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of␣
↪Rooms',
               'Avg. Area Number of Bedrooms', 'Area Population']]

y = HouseDF['Price']
```

1.2.2 Split Data into Train, Test

[14]:
```
from sklearn.model_selection import train_test_split
```

[15]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.40,)#␣
↪random_state=101)
```

1.3 Creating and Training the LinearRegression Model

[16]:
```
from sklearn.linear_model import LinearRegression
```

[17]:
```
lm =  linear_model.LinearRegression()
```

[18]:
```
lm.fit(X_train,y_train)
```

[18]:
```
LinearRegression()
```

1.4 LinearRegression Model Evaluation

[19]:
```
print(lm.intercept_)
```

```
-2637599.1421310618
```

[20]:
```
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```

[20]:

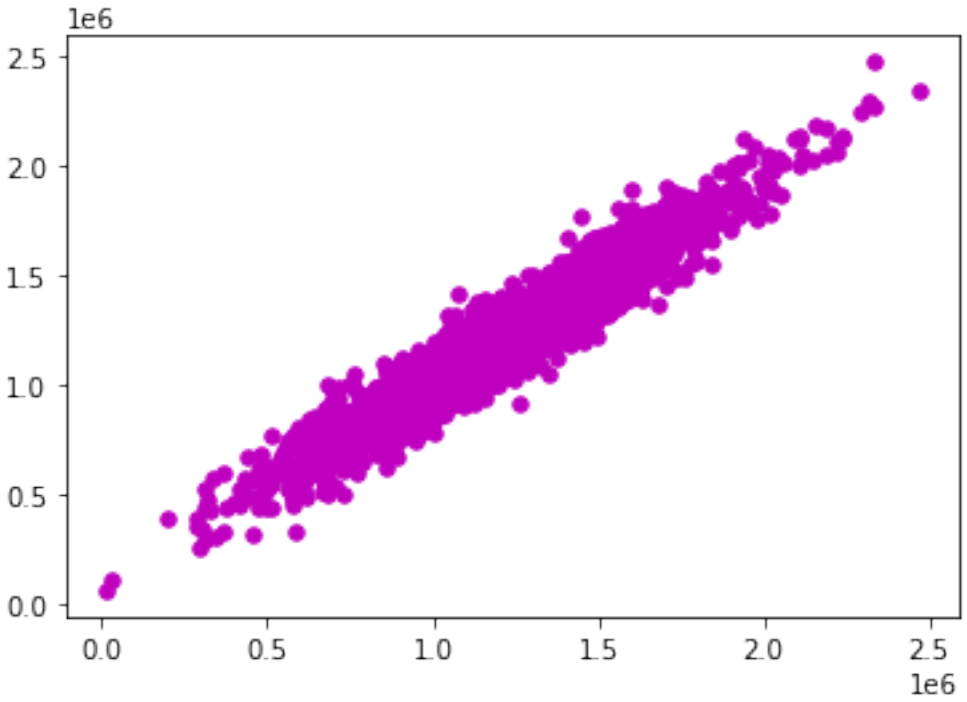
	Coefficient
Avg. Area Income	21.543966
Avg. Area House Age	165690.511404
Avg. Area Number of Rooms	120212.512681
Avg. Area Number of Bedrooms	1737.769687
Area Population	15.318771

1.5 Predictions from our Linear Regression Model

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

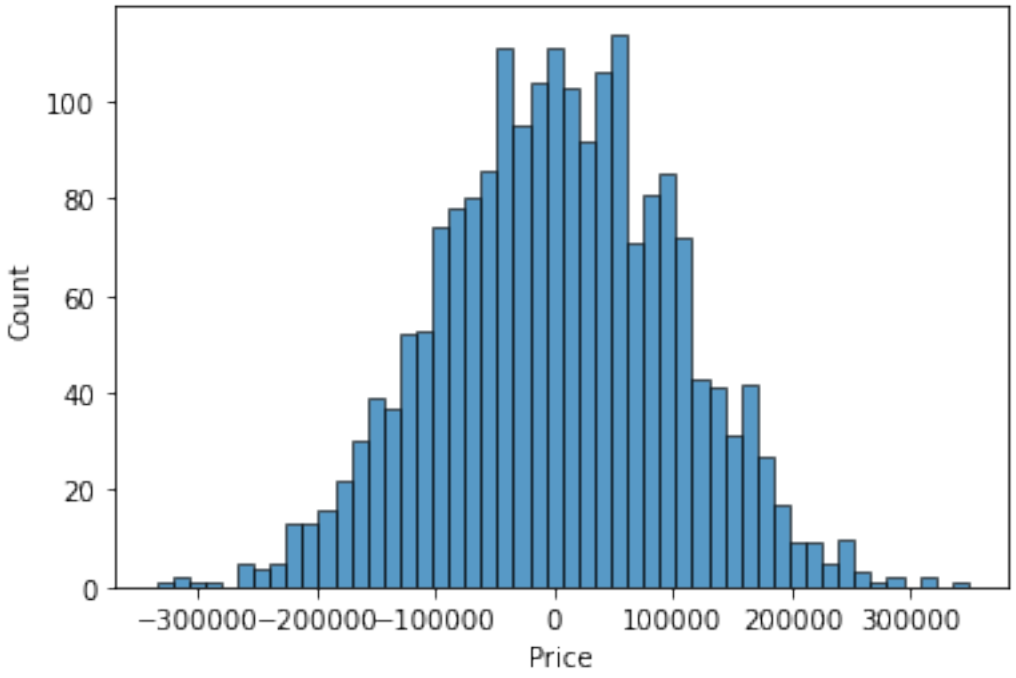
[22]:
```
#plt.scatter(y_test,predictions) plt.plot(X_test, y_pred, color='blue')
plt.scatter(y_test, y_pred, color = "m", marker = "o", s = 30)
```

[22]:
```
<matplotlib.collections.PathCollection at 0x7efca8db9310>
```



In the above scatter plot, we see data is in line shape, which means our model has done good predictions.

[23]:
```
sns.histplot((y_test-y_pred),bins=50);
```



In the above histogram plot, we see data is in bell shape (Normally Distributed), which means our model has done good predictions.

1.6 Regression Evaluation Metrics

[24]:
```
from sklearn import metrics
```

[25]:
```
print('MAE (Mean Absolute Error is ):', metrics.mean_absolute_error(y_test,␣
 ↪y_pred))
print('MSE (Mean Square Error is ):', mean_squared_error(y_test, y_pred))
print('RMSE( Root Mean Square Error is ):', np.sqrt(metrics.
 ↪mean_squared_error(y_test, y_pred)))
print('Coefficient of determination (Regression score ): %.2f' %␣
 ↪r2_score(y_test, y_pred))
```

```
MAE (Mean Absolute Error is ): 80301.66530852049
MSE (Mean Square Error is ): 9991562818.77865
RMSE( Root Mean Square Error is ): 99957.8051918841
Coefficient of determination (Regression score ): 0.92
```

[ ]:
```

```