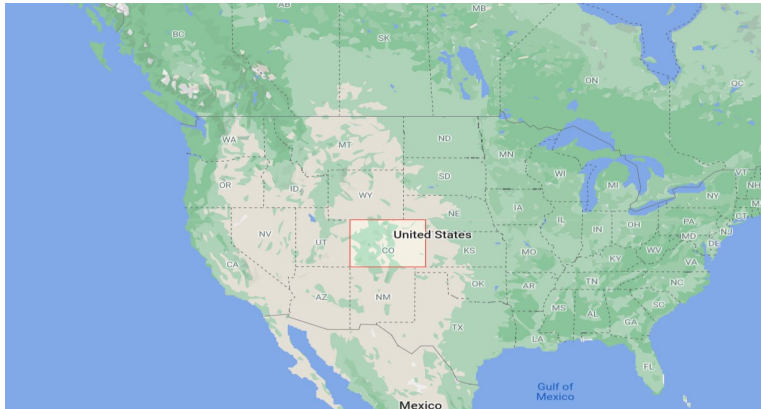
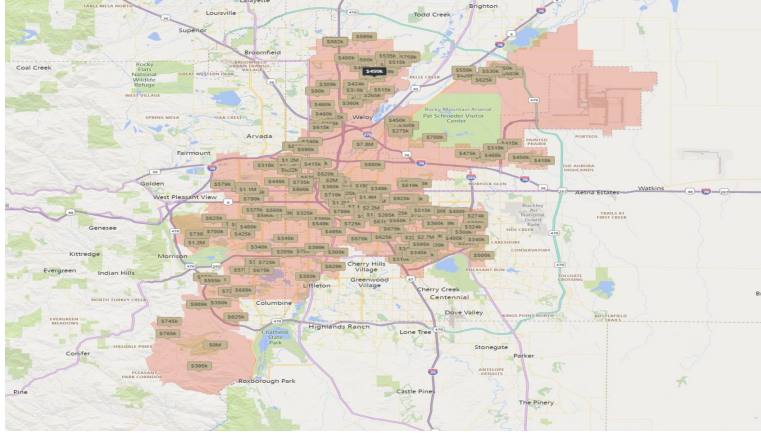


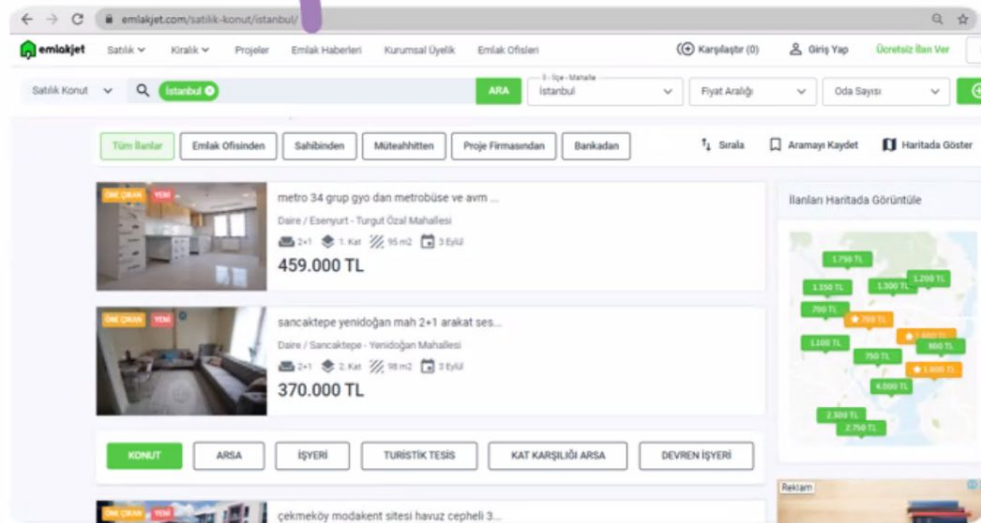
Denver, Colorado, USA House Price Prediction With Machine Learning in Python



Serdar Celebi

INTRODUCTION

- **FOCUSING:** Make a Regression model that shows a good estimation on prices.
- **OBJECTIVE :** What is shown in emlakjet dataset or what features are crucial in dataset?
- **GOALS:** Finding the fit model on the prices with the help of distinct columns.





METHODOLOGIES AND TOOLS

- To gather dataset, we made some web scraping from emlakjet by applying BeautifulSoup model .
- After that approach, we used libraries and their coding tools such as;
 - matplotlib, seaborn (to visualize and make users closer)
 - Linear Regression tools such as OLS,Ridge, ElasticNet Regression, Lasso etc.



Gathering the data using BeautifulSoup



Introduction

Business need: Predicting movie scores on IMDB

Solution: In order to predict the Movie ratings we will develop a linear regression model.





CENTURY 21

FIND A HOME

SELL A HOME

MORTGAGE

JOIN C21

MORE +

888-732-6139

Denver

Price

Beds

Baths

Filter

Keyword

MAP

GRID

Firestone
Dacono
Fort Lupton
Hudson
Aristocrat Ranchettes
Brighton
Lochbuie
Erie
Gunbarrel
Lafayette
Louisville
Superior
Broomfield
Rocky Flats National Wildlife Refuge
Golden
Centennial
Highlands Ranch
Parker
The Pinery
Aetna Estates

\$480k \$595k
\$400k \$535k
\$499k
\$559k 30k
\$90k \$424k 15k
\$360k
\$460k
\$615k
\$450k \$275k
\$825k \$7.8M
\$318k \$415k \$800k
\$5 \$1.1M \$73k \$2M \$340k \$19k
\$424k \$1.4M
\$220k \$590k \$325k \$786k \$265k \$360k \$434k
\$700k \$40k \$45k \$49k \$360k \$324k
\$1.2M \$34k \$27k \$399k \$6 \$639k \$340k
\$725k \$629k \$500k
\$55k \$350k
\$689k
\$809k
\$745k \$765k \$8M
\$395k

Search Entire Map

Denver Homes for Sale (274)

Sort: Coming Soon ▾

SAVE SEARCH

NEWLY LISTED

FOR SALE

\$499,950

or rent

\$2,940/mo est.

more info

3 beds

1 bath

1 half bath

2,112 sq. ft

10765 Madison St Thornton CO 80233

Courtesy Of HomeSmart Realty Partners

NEWLY LISTED

FOR SALE

\$2,000,000

3 beds

3 baths

1 half bath

3,592 sq. ft

1020 15th Street #42-D P. Denver CO 80202

Listed By CENTURY 21 Tenika Real Estate

NEWLY LISTED

FOR SALE

\$620,000

2 beds

Hi there! 🏡 Welcome to Century 21.

What brings you in today? 🏠

This site uses cookies and related technologies, as described in our privacy policy, for purposes that may enhance your experience, or advertising. You may choose to consent to our use of these technologies, or manage your preferences.

Buy a home

Sell a home

Rent a home

Manage Settings

```
<!DOCTYPE html>
<html lang="en" data-usagetrack-config="{"clickDefault": "ga-event", "viewDefault": "ga-event", "viewOnReady": true}" class="wf-bodoniuw-i4-active wf-jaffacitweb-n4-active wf-jaffacitweb-n3-active wf-jaffacitweb-n6-active wf-jaffacitweb-n7-active wf-active" style>
  <link type="text/css" rel="stylesheet" id="dark-mode-custom-link">
  <link type="text/css" rel="stylesheet" id="dark-mode-general-link">
  <style lang="en" type="text/css" id="dark-mode-custom-style"></style>
  <style lang="en" type="text/css" id="dark-mode-native-style"></style>
  <head></head>
  <!--[if lte IE 9] <body id="srp" class="res prp ie9"><![endif-->
  <body id="srp" class="res prp map-view" data-new-gr-c-s-check-loaded="14.1041.0" data-gr-ext-installed>
    <!-- Google Tag Manager (noscript) -->
    <noscript></noscript>
    <!-- End Google Tag Manager (noscript) -->
    <div id="bodyMain">
      <div id="ModallWindow"></div>
      <div id="content_blackbar"></div>
      <header class="site-header" data-usagetrack-group="Header"></header>
      <div id="prop-search-container" data-usagetrack-group="Hybrid Mapping">
        <div id="filters-container"></div>
        <div id="full-width-container">
          <div id="results-container">
            <div id="results-parent">
              <div class="card-wrap" data-usagetrack-group="Property Results">
                <input type="hidden" id="numResults" value="274">
                <input type="hidden" id="cacheServerSearchURL" name="cacheServerSearchURL" value="&amp;lid=CCOOEiW&amp;xc=ENX&amp;rp=20&amp;ics=1&amp;so=s&amp;start=0&amp;f=comingsoon">
                <div class="filter-wrap"></div>
                <div class="infinite-container">
                  <div class="Infinite-item property-card clearfix property-card-RE EN018940838 initialized" data-id="REN018940838" data-source-id="3y d-IRESCO-956364" data-brand-cd="REN" data-listing-id data-mls="956364" data-zip="80233" data-latitude="39.89155" data-longitude="-104.9447" data-link="/property/10765-madison-st-thornton-co-80233-RE N018940838">
                    <before>
                  <div class="nproperty-card-clin">
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

Get the data- Web scraping

```
✓ 1 properties = soup.find_all("div", {"class": "property-card-primary-info"})
   properties
```

```
✓ [12] len(properties)
```

20

```
✓ 13 properties[0]
```

```
✓ [14] all_info = []
```

```
for i in properties:
    info = {}
    info['title'] = i.find("div", {"class": "pdp-listing-type sale"}).text
    info['price'] = i.find("a", {"class": "listing-price"}).text.strip()
    info['beds'] = i.find("div", {"class": "property-beds"}).text.strip()
    info['bath'] = i.find("div", {"class": "property-baths"}).text.strip()
    info['sqft'] = i.find("div", {"class": "property-sqft"}).text.strip()
    info['city_info'] = i.find("div", {"class": "property-city"}).text.split()[-1]

    all_info.append(info)
```

```
✓ [15] print(all_info)
```

```
[{'title': 'FOR SALE', 'price': '$595,000', 'beds': '3 beds', 'bath': '2 baths', 'sqft': '2,366 sq. ft', 'city_info': '80231'}, {'title': 'FOR SALE', 'price': '$360,000', 'beds': '2 beds', 'bath': '2 baths', 'sqft': '936 sq. ft', 'city_info': '80204'}, {'title': 'FOR SALE', 'price': '$499,950', 'beds': '3 beds', 'bath': '1 bath', 'sqft': '2,1
```

```
✓ [16] len(all_info)
```

20

```
✓ 17 all_properties = []
```

```
for i in tqdm(range(1,34)):
    r = requests.get("https://www.century21.com/real-estate/denver-co/CCODEHVR/2/as=CC0MVA4A3DCC0ALUNH4S3CC0BRIQHTCHD2CC0BHOOP1E1D3D2CC0COPHRECE1TY9DCC0ENGL1W00R7DCC0HEDHRS0M0DCC0AF4FET1E3DCC0LITL1E1T0H2DCC0LOUTSV1LLE3DCC0COPOR1S0M3DCC0THORNT0H2CC0M1STH1STE1H3DCC0M1EATR1DGE&searchKey=f505881d-a16a-4836-a430-9ea664844d8b&com
    headers={'User-agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:61.0) Gecko/20100101 Firefox/61.0'})

    properties = soup.find_all("div", {"class": "property-card-primary-info"})
    for item in properties:
        info = {}
        info['title'] = item.find("div", {"class": "pdp-listing-type sale"}).text
        info['price'] = item.find("a", {"class": "listing-price"}).text.strip()
        info['beds'] = item.find("div", {"class": "property-beds"}).text.strip()
        info['bath'] = item.find("div", {"class": "property-baths"}).text.strip()
        info['sqft'] = item.find("div", {"class": "property-sqft"}).text.strip()
        info['city_info'] = item.find("div", {"class": "property-city"}).text.split()[-1]

        all_properties.append(info)
    print(all_properties)
```


Data output

✓ [22] df
0s

	Title	Price	Beds	Bath	SqFt	city_info
0	FOR SALE	\$595,000	3 beds	2 baths	2,366 sq. ft	80231
1	FOR SALE	\$360,000	2 beds	2 baths	936 sq. ft	80204
2	FOR SALE	\$499,950	3 beds	1 bath	2,112 sq. ft	80233
3	FOR SALE	\$350,000	3 beds	2 baths	1,144 sq. ft	80003
4	FOR SALE	\$843,030	4 beds	4 baths	4,571 sq. ft	80016
...
655	FOR SALE	\$646,200	3 beds	3 baths	2,507 sq. ft	80027
656	FOR SALE	\$620,000	2 beds	1 bath	1,084 sq. ft	80202
657	FOR SALE	\$1,400,000	4 beds	2 baths	3,088 sq. ft	80206
658	FOR SALE	\$775,000	6 beds	4 baths	4,968 sq. ft	80016
659	FOR SALE	\$735,900	3 beds	3 baths	1,675 sq. ft	80204

660 rows × 6 columns

Data Processing

- ✓ [27] `df['Bath']= df['Bath'].apply(lambda x : x.replace('s',''))`
- ✓ [28] `df['Bath'] = df['Bath'].apply(lambda x : x.replace('bath',''))`
- ✓ [29] `df['Beds'] = df['Beds'].apply(lambda x : x.replace('beds',''))`
`df['Beds']`
- ✓  [30] `df['Beds'] = df['Beds'].apply(lambda x : x.replace('bed',''))`
- ✓ [31] `df['Price'] = df['Price'].apply(lambda x : x.replace('$',''))`
- ✓ [32] `df['Price'] = df['Price'].apply(lambda x : x.replace(',',''))`
- ✓ [33] `df['SqFt'] = df['SqFt'].apply(lambda x : x.replace('sq. ft',''))`
- ✓ [34] `df['SqFt'] = df['SqFt'].apply(lambda x : x.replace(',',''))`
- ✓ [35] `# type conversion`
`df[['Price','SqFt','Beds','Bath']]= df[['Price','SqFt','Beds','Bath']].astype(int)`
- ✓ [36] `df.info()`

OS

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 660 entries, 0 to 659
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Title       660 non-null   object
 1   Price       660 non-null   int64
 2   Beds       660 non-null   int64
 3   Bath       660 non-null   int64
 4   SqFt       660 non-null   int64
 5   city_info  660 non-null   object
dtypes: int64(4), object(2)
memory usage: 31.1+ KB
```

Data Processing

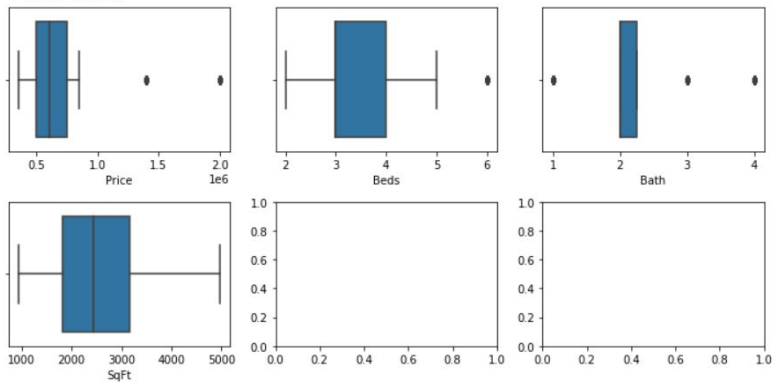
✓ 1s

Outlier Analysis

```
fig, axs = plt.subplots(2,3, figsize = (10,5))
plt1 = sns.boxplot(df['Price'], ax = axs[0,0])
plt2 = sns.boxplot(df['Beds'], ax = axs[0,1])
plt3 = sns.boxplot(df['Bath'], ax = axs[0,2])
plt1 = sns.boxplot(df['SqFt'], ax = axs[1,0])
```

```
plt.tight_layout();
```

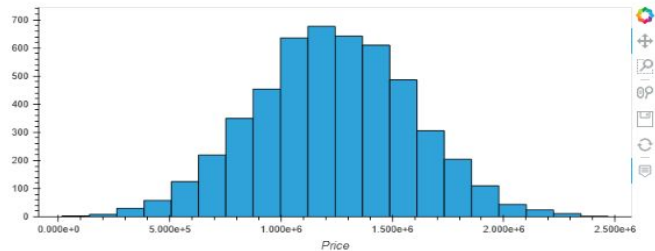
⚠ /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable
FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable
FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable
FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable
FutureWarning



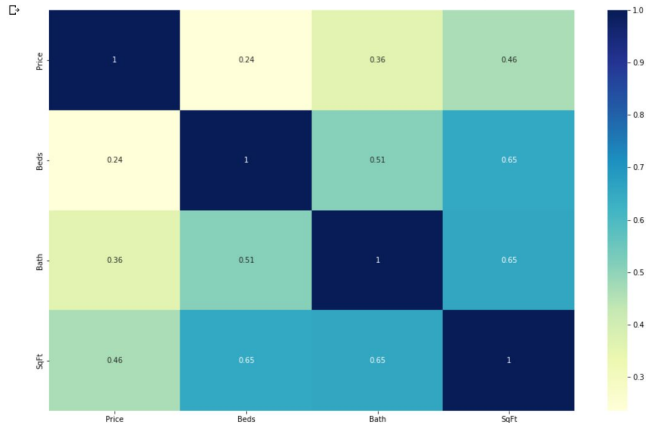
Exploratory Data Analysis

```
In [15]: USAHousing.hvplot.hist("Price")
```

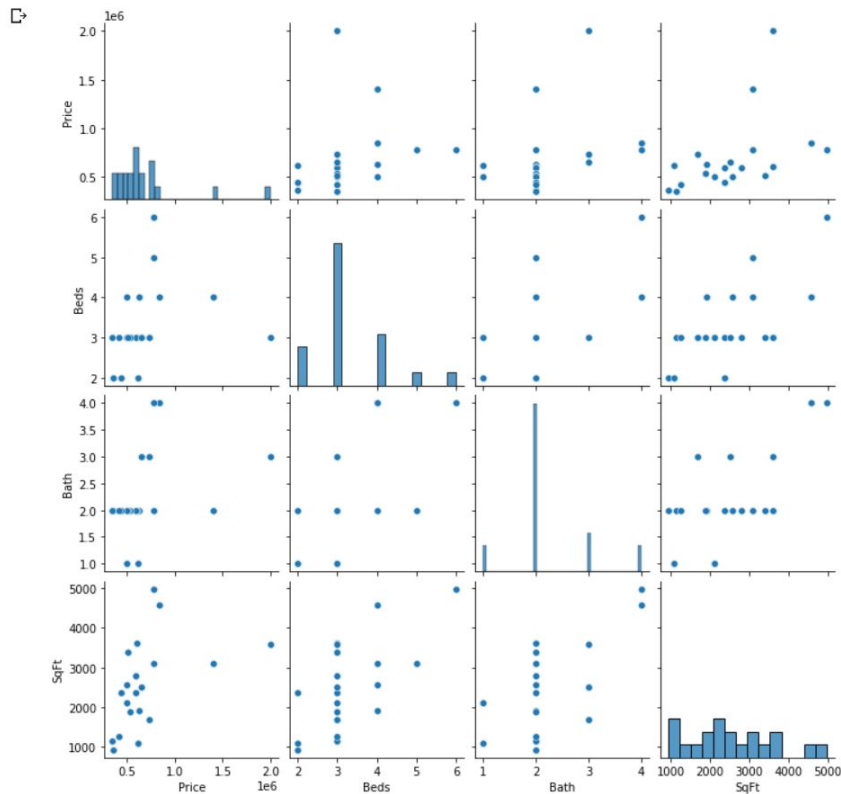
```
Out[15]:
```



```
# Let's check the correlation coefficients to see which variables are highly correlated  
plt.figure(figsize = (16, 10))  
sns.heatmap(df.corr(), annot = True, cmap="YlGnBu")  
plt.show()
```



```
sns.pairplot(df)  
plt.show()
```



✓ Regression Evaluation Metrics

Here are three common evaluation metrics for regression problems:

- **Mean Absolute Error (MAE)** is the mean of the absolute value of the errors:

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- **Mean Squared Error (MSE)** is the mean of the squared errors:

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

- **Root Mean Squared Error (RMSE)** is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

🔗 Comparing these metrics:

- **MAE** is the easiest to understand, because it's the average error.
- **MSE** is more popular than MAE, because MSE "punishes" larger errors, which tends to be useful in the real world.
- **RMSE** is even more popular than MSE, because RMSE is interpretable in the "y" units.

All of these are **loss functions**, because we want to minimize them.

Model Results

Linear Regression

R^2 : 0.8975574856866375

Root Mean Squared Error: 115272.59421

CV R^2 : 0.910746

Ridge

R^2 : 0.8975574856866375

Root Mean Squared Error: 114728.35240476725

Lasso

R^2 : 0.8935089334837565

Root Mean Squared Error: 113692.11382114442

Q & A