BAN 620

Alameda and Santa Clara Housing Analysis

By Soha Mohamed and Srivalli Chadalavada

Date May 2022

Summary

The Purpose of this project is to **conduct fresh analysis on the Housing situation in Alameda and Santa Clara through predicting the best model to be used for future price predictions based on web scrapped data in May 2022.**

A brief analysis is conducted on the factors that affect the housing prices in the United States. From there, we will zoom in into the housing situation in **Alameda and Santa Clara.**
The housing prices are soaring in general, using the **Multiple Regression model**, we will predict the best model for predicting to be used in the housing pricing situation in Alameda and Santa Clara. We will use all the variables then backward elimination to eliminate unnecessary variables and assure the efficiency of our model as well.

Then we will use The **Neural Network model** as a benchmark to the multiple regression model used. We will apply Neutral Networks with different iterations, hidden layers and will use Grid Search to create an improved model.

Introduction:

Housing and Factors Determining prices:

According to Pew Research Center, "A rising share of Americans say the availability of affordable housing is a major problem in their local community. **In October 2021**, about half of Americans **(49%) said this was a major problem where they live, up 10 percentage points from early 2018**. In the same survey, **70% of Americans said young adults today have a harder time buying a home than their parents' generation did."**

According to the White House, The Pandemic has seen extraordinary growth in home values. The Pandemic caused significant market disruptions, like increased demand and rising building costs as well as other long-term supply constraints. The **Case-Shiller U.S. The National Home Price index has risen by 18.6 percent during 2020 t**hen again, a new record **of 45 percent during 2022** and still going up creating the strongest growth in the history of the series **(Figure 1).**

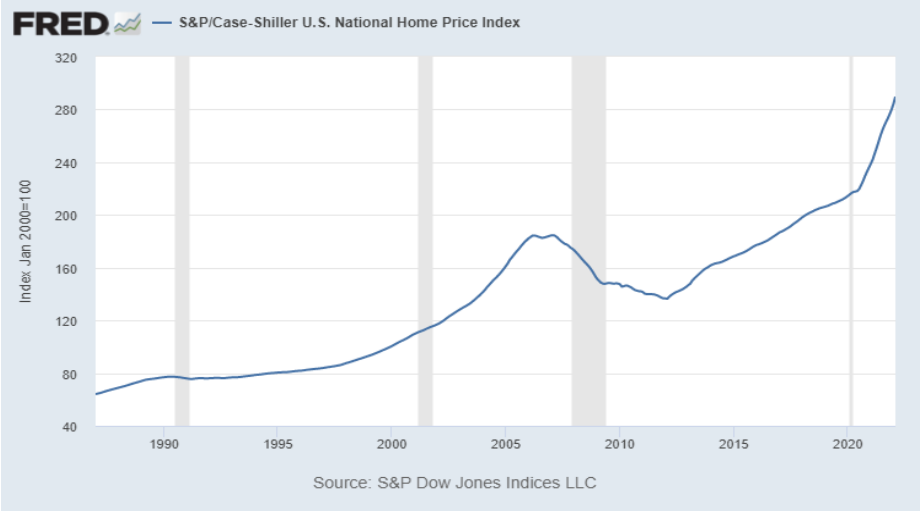


Figure 1: Case-Shiller US National Home Price Index

According to the **Gardner** Report on the housing in Northern California, the **most affordable** county relative to average prices **in the first Q of 2022 is Shasta**. **Santa Clara** is again the **most expensive market**. **Santa Clara and Alameda** counties rose by **19.4% and 18.3%** respectively. In addition, the **number of days of homes listed** has **dropped by 7 days** comparing the first quarter of 2021 to the first quarter of 2022. **Economists attribute this to low inventory and a high demand for homes coupled with rise in inflation.**

Looking at the S&P Case-Shiller U.S. National Home Price Index, we can see a **5% increase between Jan 2022 and Feb 2022,** which is almost the **same increase between March,2010 till March 2013**. This can be clearing displayed on (Figure 2)

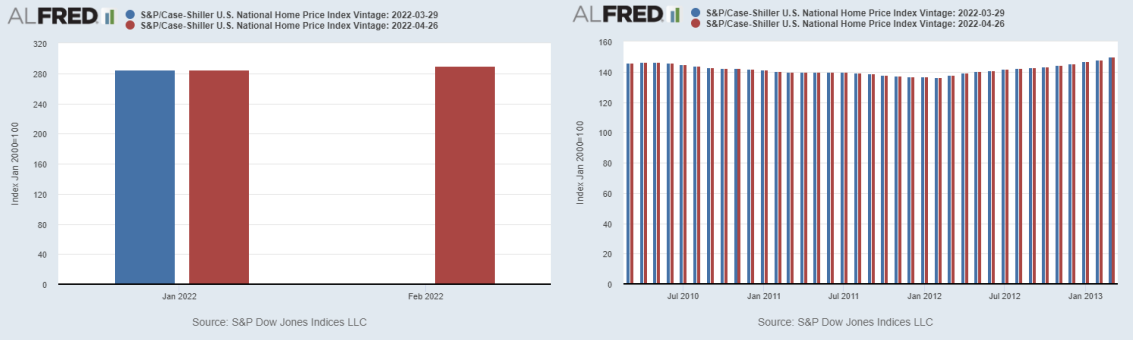


Figure 2: Case-Shiller national Home Price Index between Jan 2022 and Feb 2022 VS March,2010 till March 2013.

The trend is remarkably high. It would be helpful to predict the situation in Almeda and Santa Clara. Based on the Analysis that will be conducted we will be able to get more insights. We will have clarity on the developments unless radical variable came into the equation. By radical variable here we mean something that is imposed by the government: maybe a regulation that would ease the supply process to assure smooth demand.

In this section we will check more on the **annual changes in prices for housing** and the **average days taken in each county for a home to be sold.** As per **Gardner analysis** conducted on the **first quarter of 2022**, a county like **SAN LUIS OBISPO** going up by **26.3%** and a county like **NAPA** going up by **4.5%.** Such a difference in the range between counties is expected given other individual variables related to each county.

Figure 3 goes more into the annual increases by county in Almeda and Santa Clara counties.

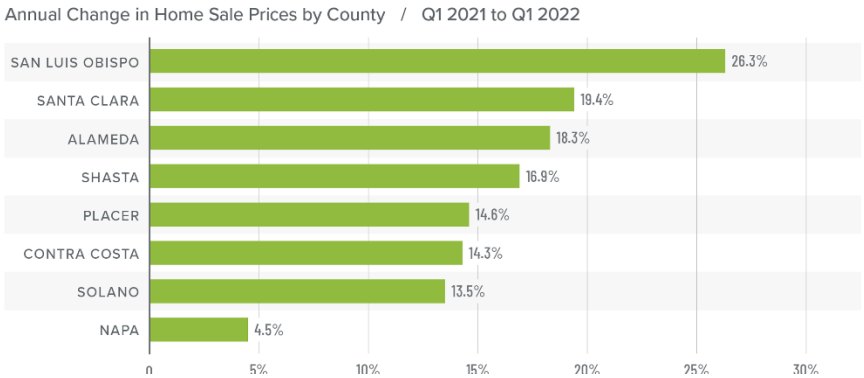


Figure 3: Annual change in housing sales prices by county—2021-2022

Looking at the average days needed to sell a home in Almeda and Santa Clara; The least number of days goes to **Santa Clara (13) days,** while the highest number of days goes **Shasta (85) days**. For this section we can conclude that **Santa Clara is an attractive county for residency** even though the housing sales **prices went up by 19.4%**. *Figure 4* illustrates more on the average number of days for sales per county.

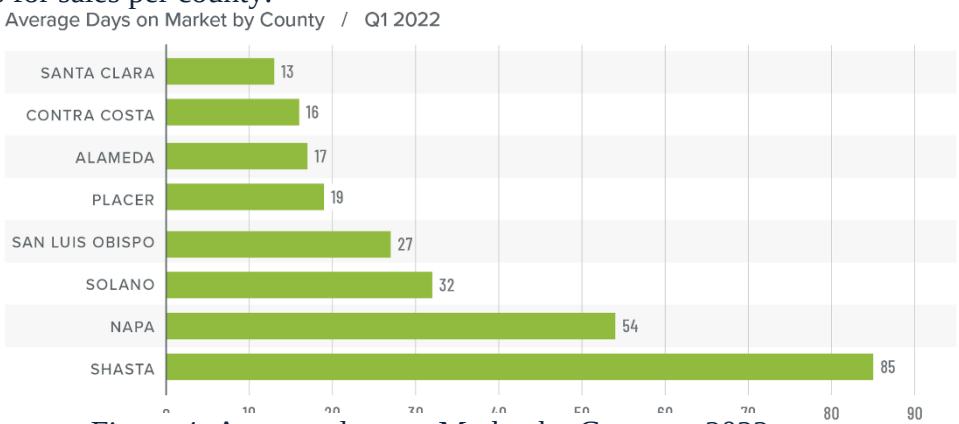


Figure 4: Average days on Market by County – 2022

In the next sections, we will assess what key factors affects home prices and build different models for future price predictions.

Main Chapter:

Conducting data analysis in real life, we usually follow 6 main steps:

a- **Define the analysis objective:** Our project goal is to assess the factors that affect the selling price across Alameda and Santa Clara counties then predict the housing prices through building a model/s for the new records.

b- **Data Collection (data sourcing and collection):** Scraping in python using BeautifulSoup library from REDFIN website then export the outcomes into a CSV formatted file to be used in the analysis. If the Web scraped data are used without exporting data scraped at a certain time, a continuous update will occur, and the codes might crash as well. For efficiency and better resources utilization, export will be conducted in our analysis.

c- **Data Wrangling (Data Cleaning, Exploratory Data exploration including visualization):** In this section, we will remove null values, check correlation between variables in relation to the price. Initial visualizations will be conducted to visualize the data distributions. We will mainly develop the data to be ready for the models' developments.

d- **Model/s Development** (Select, Build and Test models): At this stage we will develop models based on all variables involved and based on reduced variables as well. Reduced variables help in the accuracy and efficiency as well. Multiple Linear regression and Neural networks models to be used in our analysis.

e- **Model Deployment:** Deploying the best model for future price predictions. In our case it would be the best model to predict prices.

f- **Insights and meeting the stated objective/s:** Insights will be developed along each model and stated research objective to be met.

B- Data Collection (data sourcing and collection):

We chose web scrapping for the data collection. During the web scrapping, lots of challenges were faced. In this section we will briefly state the challenges and how did we overcome them. Figure 5 explains briefly the Web Scraping process.

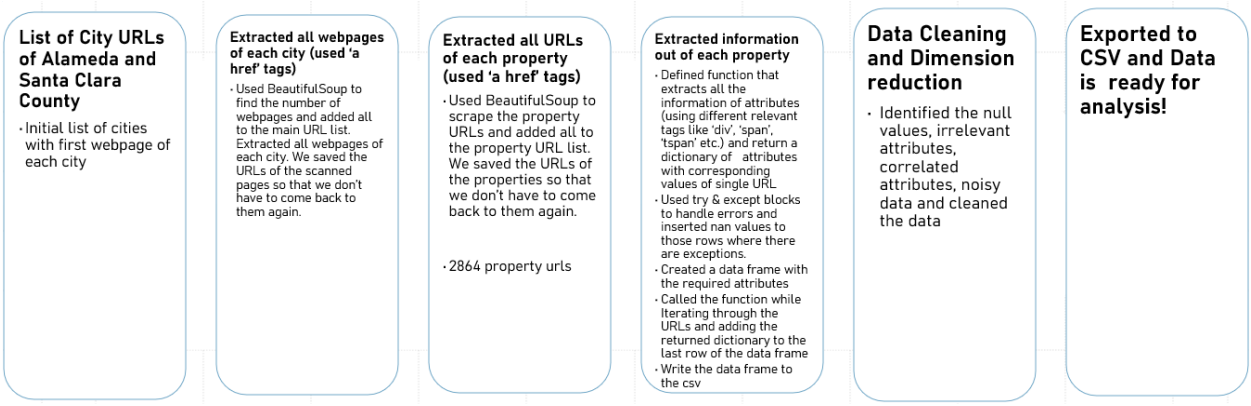


Figure 5: Web scrapping process

- **List of URLs**
- **Extract all webpages for each city**
- **Extract all URLs of each property**
- **Extract information out of each property**
- **Data Cleaning and dimension reduction(initial)**
- **Export data to csv file format**

Challenges faced:

- **IP blocking**: Blocking is triggered when the server detects many requests from the same IP address or when a search robot makes several parallel requests.
- **Captcha:** CAPTCHA allows distinguishing a person from a robot.
- **Slow or unstable load speed**: Website slow-down in content loading. Matter fact sometimes it did not load at all specially when receiving many access requests. With refreshing, the parser will not know how to handle such a situation and data collection may be interrupted.
- **Over written data disruptions and time constraints:** During the analysis process since there are momentum updates, lots of errors and warning kept showing up.

Procedures took to overcome this:

- **Set a real User Agent**: The User Agent (UA) is a string in the request header that specifies the browser and operating system for the web server. If the user agent used does not belong to one of the major browsers, some sites will block requests. Hence the user agent is set as Mozilla/5.0 for the headers parameter request method to avoid IP blocking and Captcha.
- **Set random intervals in between your requests:** Instead of overloading the site with many requests, few seconds time delays between requests were done. URLs of the scanned pages to avoid coming back for them again. Using time.sleep (5) a time delay of 5 seconds is set between requests.
- **Avoid scraping during peak hours:** Better to collect data during off-peak periods, so as not to interfere with the site work. It also plays a big role for the parser itself, as it will

significantly increase the speed of data collection. This is what was done and worked well during the web scraping stage

- **Synchronize the web scraped data to csv file:** doing that helped in minimizing the continuous coding crashes faced.

After conducting the web scraping process, **35 initial variables were chosen** to be the initially used. The types of variables were initially changed to be meet the analysis requirements. Variables names used as mentioned on REDFINE, for easier references to the readers of this analysis. These variables were reduced during data wrangling then in later stage during modelling.

1. ***List Price****:* A list price is the price of a home for sale set by a seller and listing agent. The unit of list price is '$'.
2. ***Address, City, State, Zip code****:* These variables give the exact location of advertised property. Zip code is the postal code used by state postal.
3. ***Property URL:*** The links or URLs of all the properties in cities of 2 Counties.
4. ***Beds:*** the total number of Bedrooms available in the advertised property.
5. ***Baths*****:** the total number of Bathrooms available in the advertised property.
6. ***Living sqft:*** *R*efers to the "living area" of the home. It is the area that will be heated or cooled.
7. ***Status*****:** It signifies whether the property is active, pending or sold out.
8. ***Property Type*****:** It marks the type of advertised property such as Condo, Single, double etc.
9. ***Year Built*****:** It implies the year in which the advertised property was built.

10. ***Est. Monthly Payment*:** It is the amount you pay back in your home loan every month. It consists of Principal and interest, property taxes and Homeowner's insurance.
11. ***Price per Square Feet:*** This is the list or sale price of a home divided by the number of total square feet. It determines how much a buyer will pay for a square foot of space.
12. ***Drought Score*:** This score is made up of three parts on a scale of 100: sensitivity, exposure, and ability to adapt. Provides information on climate risk to location of property.
13. ***Walk Score*:** It measures the walkability of any address using a patented system. Amenities within 5 minutes of distance are given maximum points.
14. ***Neighbourhood Homes*:** It is the number of total houses sold near the advertised property.
15. ***Transit Score*:** Transit score implies about the facility of public transportation option and how far it is from the advertised property.
16. ***Groceries Stores*:** It shows the total number of grocery stores nearby the advertised property.
17. ***Services*:** It shows the number of services available nearby the advertised property. It includes pre-school, fuel station, hair salon, nail salon, etc.
18. ***Emergency*:** It shows the number of Emergency services available nearby the advertised property. It includes Fire station, Police department and Hospital.

19. ***Shopping*:** It shows the total number of shopping mall, drug store and convenience store are available nearby the advertised property. It includes Safeway, 7-Eleven.
20. ***Food and Drink*:** It shows the number of refreshment shops/ lounges are available nearby the advertised property. It includes Hotel Bar, Coffee shop, Restaurants, etc.
21. ***County*:** It shows in which county the address of advertised property is located. We are analysing two Counties, i.e., Santa Clara and Alameda.
22. ***Competitive Score*:** Market Competition calculated over past 6 months on a scale of 100.
23. **has_supercenter:** Whether the advertised property have at least one supermarket in the supercenter list {Walmart, Target, …} located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
24. **has_starbucks:** Whether the advertised property have at least one starbucks located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
25. **has_boba:** Whether the house have at least boba shop located in the neighbourhood. If it has at least one takes the values of 1, otherwise. 0
26. **has_mall:** Whether the house have at least mall located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
27. **has_indian_restaurant:** Whether the house have at least an Indian restaurant located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.

28. **has_major_indian_grocery:** Whether the house have at least an Indian grocery located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
29. **has_chinese_restaurant:** Whether the house have at least a Chinese restaurant located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
30. **has_mexican_restaurant:** Whether the house have at least a Mexican restaurant located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.
31. **page_fav_count_30:** The number of times this home has been favorited on Redfin for the current MLS listing from past 30 days.
32. **page_fav_all_time_count:** The number of times this home has been favorited on Redfin for the current MLS listing overall.
33. **page_view_count:** The number of views the current MLS listing has had on the Redfin website since it was imported from the MLS.
34. **HOA Dues:** Homeowners association fees are monthly dues collected by homeowners' associations from property owners. These fees are standard for most purchased condominiums, apartments, and planned communities.
35. **has_major_entertainment:** Whether the house have at least a theatre located in the neighbourhood. If it has at least one takes the values of 1, otherwise 0.

The results of web scraping after initial cleaning were 2213 rows and 35 columns. Property types are 'Condo', 'Single', and 'Townhouse' to be the focus in this analysis for conciseness.

C- Data Wrangling (Data Cleaning, Exploratory Data exploration including visualization):

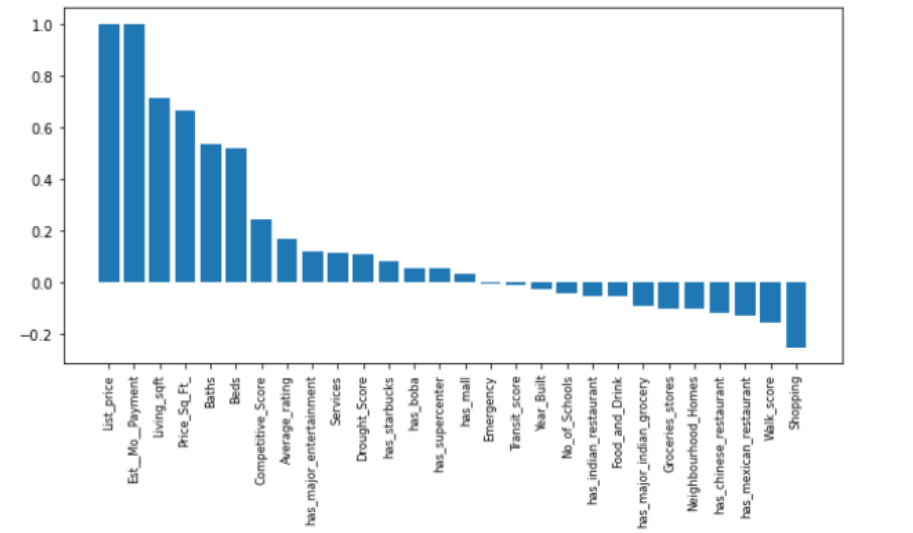
We will start this stage by conducting correlation analysis between price (outcome) and 34 predictors (Inputs).

```
c = df1.corr()
correlation=c["List_price"].sort_values(ascending=False)
correlation=pd.DataFrame(correlation)
correlation
```

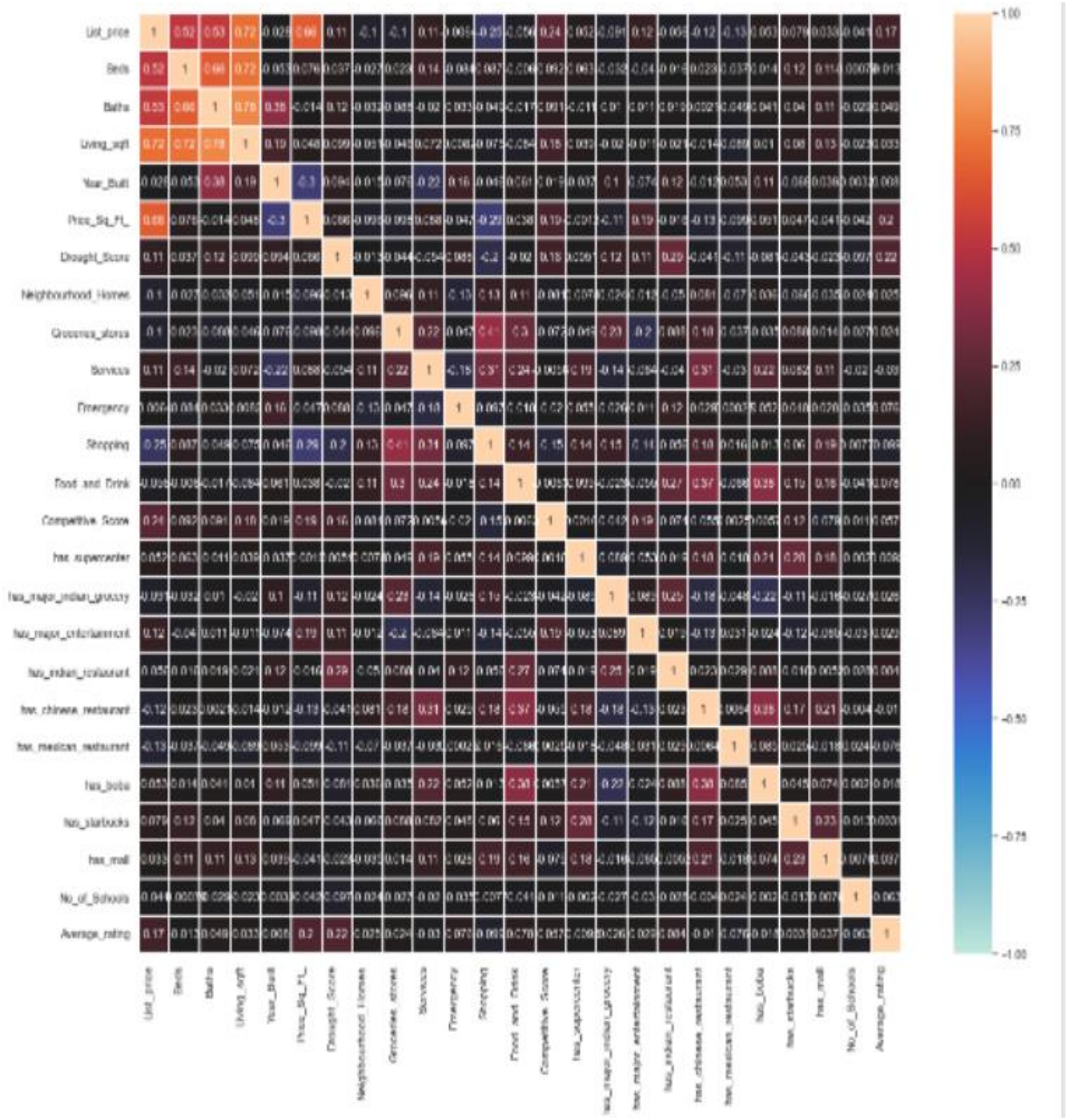
	List_price
List_price	1.000000
Est__Mo__Payment	0.999362
Living_sqft	0.715915
Price_Sq_Ft_	0.663444
Baths	0.532644
Beds	0.521508
Competitive_Score	0.242802
Average_rating	0.166675
has_major_entertainment	0.121357
Services	0.111544
Drought_Score	0.107535
has_starbucks	0.079471
has_boba	0.052952
has_supercenter	0.052286
has_mall	0.033374
Emergency	-0.006417
Transit_score	-0.008599
Year_Built	-0.028232
No_of_Schools	-0.041412
has_indian_restaurant	-0.055580
Food_and_Drink	-0.055773
has_major_indian_grocery	-0.090890
Groceries_stores	-0.101625
Neighbourhood_Homes	-0.103177
has_chinese_restaurant	-0.118088
Walk_score	-0.158007
Shopping	-0.251336

As we can see Estimated monthly payment, living square feet, and price per square feet are highly positively correlated with List_price (outcome). Variables like Emergency and has_mall will be least considered since their values are less than 10%. Values like shopping will be considered negatively correlated to price.

Plotting correlation for better understanding:



Using heat map for better understanding to the multicollinearity using figures and colours. The lighter the colours the more corelated the variable to price in a positive or a negative form.

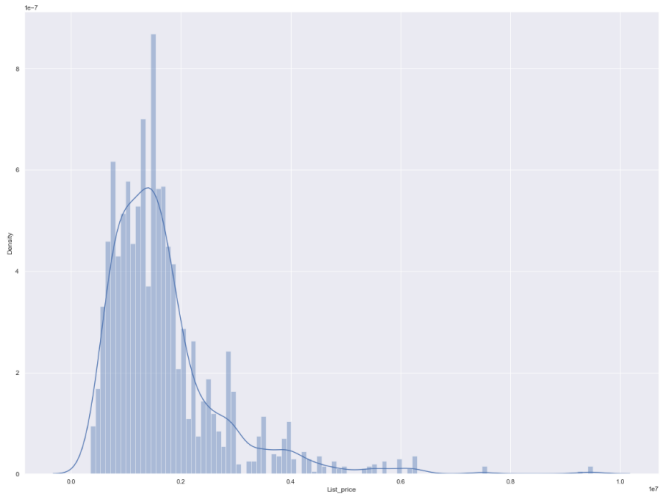


More visualizations were conducted to construct a better understanding to different involved relations:

- Checking distribution of list price

```
#checking the distribution of listprice
sns.distplot(df1['List_price'], color='b', bins=100, hist_kws={'alpha': 0.4})
sns.set(rc = {'figure.figsize':(6,6)})
```

```
C:\Users\stsc.LENOVO-127992\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated 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)
```

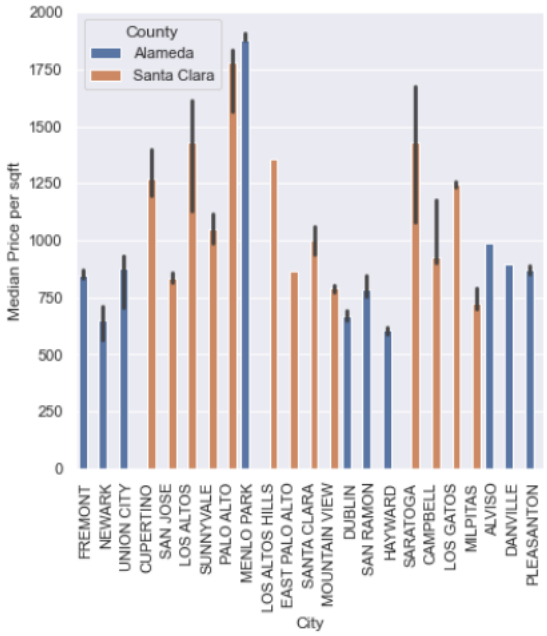


- Visualize the variation of average school rating related to price taking into consideration the county.

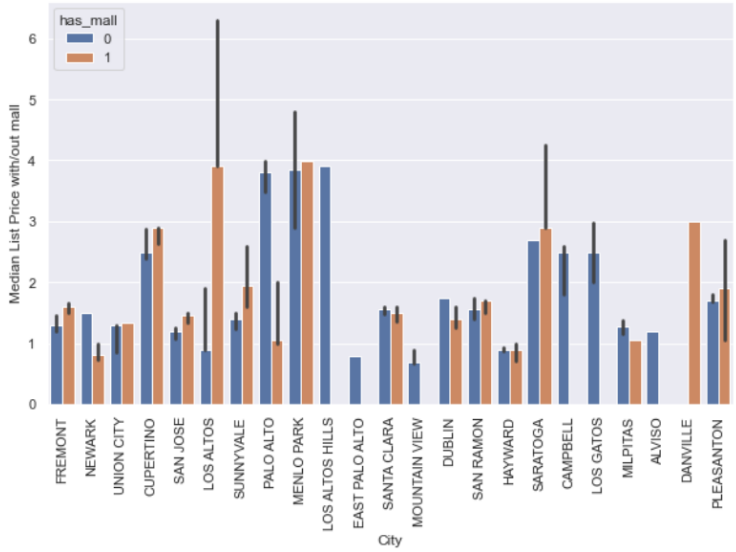
```
#analysing the varaition of average rating over the median price per sqft.
sns.lineplot(x = "Average_rating", y = 'Price_Sq_Ft_', data=df1, estimator = median, hue = 'County')
sns.set(rc = {'figure.figsize':(6,6)})
# plt.title('Opening Prices')
plt.xlabel('Avg Schools rating')
plt.ylabel('Median Price_Sq_Ft')
plt.show()
```

- Visualizing the price median for Almeda and Santa Clara

```
#analysing the median price per sqft over the cities and counties
from numpy import median
sns.barplot(x = "City", y = 'Price_Sq_Ft_', data=df1, hue = "County", estimator = median)
sns.set(rc = {'figure.figsize':(8,6)})
# plt.title('Opening Prices')
# plt.xlabel('Date')
plt.ylabel('Median Price per sqft')
plt.xticks(rotation=90)
plt.show()
```



- Visualizing the presence of shopping areas to price:



Removing variables that are correlated to each other.

- Shopping is correlated with Grocery stores
- has_Boba with Food& drink
- has boba with has_chinese_restaurant

The shape of the cleaned data frame after removing null values for certain columns and filling the null values using various methods such as forward filling, filling with median value for certain columns and removing correlated and unnecessary columns are (2213, 29).

Listing the data types after the above cleaning steps:

```
List_price                 float64
Address                     object
City                        object
State                       object
Beds                       float64
Baths                      float64
Living_sqft                float64
Property_Type               object
Year_Built                 float64
Price_Sq_Ft_               float64
Drought_Score              float64
Neighbo  'hood_Homes       float64
Emergency                    int64
Competitive_Score          float64
has_supercenter              int64
has_major_indian_grocery     int64
has_major_entertainment      int64
has_indian_restaurant        int64
has_chinese_restaurant       int64
has_mexican_restaurant       int64
has_boba                     int64
has_starbucks                int64
property_url                object
has_mall                     int64
Zipcode                     object
No_of_Schools              float64
Average_rating             float64
County                      object
Walk_score_cat              object
dtype: object
```

Listing data types that will be used in Analysis:

```
df_new.dtypes
```

```
List_price                          float64
Address                              object
City                                 object
State                                object
Beds                                float64
Baths                               float64
Living_sqft                         float64
Year_Built                          float64
Price_Sq_Ft_                        float64
Drought_Score                       float64
Neighbourhood_Homes                 float64
Emergency                             int64
Competitive_Score                   float64
has_supercenter                       int64
has_major_indian_grocery              int64
has_major_entertainment               int64
has_indian_restaurant                 int64
has_chinese_restaurant                int64
has_mexican_restaurant                int64
has_boba                              int64
has_starbucks                         int64
property_url                         object
has_mall                              int64
Zipcode                              object
No_of_Schools                       float64
Average_rating                      float64
Walk_score_cat_Somewhat Walkable      uint8
Walk_score_cat_Very Walkable          uint8
Walk_score_cat_Walkers Paradise       uint8
Property_Type_Single                  uint8
Property_Type_Townhouse               uint8
County_Santa Clara                    uint8
dtype: object
```

Listing variables after replacing spaces with underscore sign:

```
print('Modified column titles with no space and one word for titles:')
price_df.columns = [s.strip().replace(' ', '_') for s in price_df.columns]
price_df.columns
```

```
Modified column titles with no space and one word for titles:

Index(['List_price', 'Address', 'City', 'State', 'Beds', 'Baths',
       'Living_sqft', 'Year_Built', 'Price_Sq_Ft_', 'Drought_Score',
       'Neighbourhood_Homes', 'Emergency', 'Competitive_Score',
       'has_supercenter', 'has_major_indian_grocery',
       'has_major_entertainment', 'has_indian_restaurant',
       'has_chinese_restaurant', 'has_mexican_restaurant', 'has_boba',
       'has_starbucks', 'property_url', 'has_mall', 'Zipcode', 'No_of_Schools',
       'Average_rating', 'Walk_score_cat_Somewhat_Walkable',
       'Walk_score_cat_Very_Walkable', 'Walk_score_cat_Walkers_Paradise',
       'Property_Type_Single', 'Property_Type_Townhouse',
       'County_Santa_Clara'],
      dtype='object')
```

D- Model/s Development (Select, Build and Test models):

Dropping Address, City, State, Zipcode and property_url columns because they do not affect the List_price.

The potential outcome variable and the predictor variables (26) are listed below:

```
predictors = ['Beds', 'Baths', 'Living_sqft', 'Year_Built',
       'Price_Sq_Ft_', 'Drought_Score', 'Neighbourhood_Homes', 'Emergency',
       'Competitive_Score', 'has_supercenter', 'has_major_indian_grocery',
       'has_major_entertainment', 'has_indian_restaurant',
       'has_chinese_restaurant', 'has_mexican_restaurant', 'has_boba',
       'has_starbucks', 'has_mall', 'No_of_Schools', 'Average_rating',
       'Walk_score_cat_Somewhat_Walkable', 'Walk_score_cat_Very_Walkable',
       'Walk_score_cat_Walkers_Paradise', 'Property_Type_Single',
       'Property_Type_Townhouse', 'County_Santa_Clara']
outcome = 'List_price'
```

Conducting Partitioning (Training and validation):

- Training partition: 1327
- Validation partition: 886

In order to avoid an overfitting situation, we use partitions to develop our data by using train_test_split() with test-size at 0.40. It means the training partition contains 60% of the data to develop the model and Validation contains 40% of data to evaluate on new records' performance.

Regression Model for prices:

Regression Model for Prices Training Set

Intercept: -1707772.56

	Predictor	Coefficient
0	Beds	-17082.81
1	Baths	29381.11
2	Living_sqft	1075.38
3	Year_Built	34.10
4	Price_Sq_Ft_	1693.12
5	Drought_Score	2030.47
6	Neighbourhood_Homes	-314.52
7	Emergency	5699.00
8	Competitive_Score	-4393.03
9	has_supercenter	128698.68
10	has_major_indian_grocery	-9390.14
11	has_major_entertainment	15291.38
12	has_indian_restaurant	-117036.73
13	has_chinese_restaurant	-116929.28
14	has_mexican_restaurant	16684.02
15	has_boba	38717.67
16	has_starbucks	40038.25
17	has_mall	-95229.71
18	No_of_Schools	5847.67
19	Average_rating	18961.08
20	Walk_score_cat_Somewhat_Walkable	52772.85
21	Walk_score_cat_Very_Walkable	66688.21
22	Walk_score_cat_Walkers_Paradise	-58190.13
23	Property_Type_Single	-134266.65
24	Property_Type_Townhouse	-32387.25
25	County_Santa_Clara	22564.41

Since the outcome variable is numerical, so we will conduct multiple linear regression and neural network models to suggest the best model to be used for future prices predictions.

1- Conducting Multiple Linear Regression model with 26 variables:

```
Accuracy Measures for Training Set - All Variables

Regression statistics

                     Mean Error (ME) : -0.0000
      Root Mean Squared Error (RMSE) : 322849.3457
           Mean Absolute Error (MAE) : 194073.9406
         Mean Percentage Error (MPE) : 2.1051
Mean Absolute Percentage Error (MAPE) : 13.9432

Accuracy Measures for Validation Set - All Variables

Regression statistics

                     Mean Error (ME) : -12055.8340
      Root Mean Squared Error (RMSE) : 267077.8858
           Mean Absolute Error (MAE) : 178908.7909
         Mean Percentage Error (MPE) : 1.7451
Mean Absolute Percentage Error (MAPE) : 12.9972
```

RMSE in the training partition is 322849.35 and, in the validation, partition is 267077.89. The difference between RMSE in the training and validation partitions is approximately 17.2% which is acceptable. The RMSE is very high in general.
MAPE in the training partition is 13.94 % and, in the validation, partition is 12.997%. The difference between the training partition and the validation partition is minimal. Both rates are accepted.

Based on the above, we can conclude that there is no overfitting. The model is good for prediction. The errors using RMSE is high, Will develop a multiple regression model using Backward elimination algorithm to remove unnecessary variables and improve this error rate.

2- Conducting Multiple Linear Regression with Backward Elimination Algorithm:

This algorithm is used to find the best variables that are influencing the list price. This method is adopted because it is computationally cheaper and reasonable with the number of records used in this analysis.

```
Best Variables from Backward Elimination Algorithm
['Living_sqft', 'Price_Sq_Ft_', 'Emergency', 'has_supercenter', 'has_indian_res
taurant', 'has_chinese_restaurant', 'has_boba', 'has_mall', 'Average_rating',
'Walk_score_cat_Somewhat_Walkable', 'Walk_score_cat_Very_Walkable', 'Property_T
ype_Single']
```

With the 12 best variables obtained from this algorithm the Multiple Linear regression model is built and the regression coefficients and accuracy measures are given below.

```
Accuracy Measures for Training Set based on Backward elimination algorithm

Regression statistics

                    Mean Error (ME) : -0.0000
       Root Mean Squared Error (RMSE) : 323866.5611
            Mean Absolute Error (MAE) : 194572.1751
          Mean Percentage Error (MPE) : 2.0949
Mean Absolute Percentage Error (MAPE) : 13.9218

Accuracy Measures for Validation Set based on Backward elimination algorithm

Regression statistics

                    Mean Error (ME) : -13714.1137
       Root Mean Squared Error (RMSE) : 268675.0587
            Mean Absolute Error (MAE) : 180650.3527
          Mean Percentage Error (MPE) : 1.6380
Mean Absolute Percentage Error (MAPE) : 13.0804
```

RMSE in the training partition is 323866.56 and, in the validation, partition is 268675.06. The difference between RMSE in the training and validation partitions is approximately 17.04% which is acceptable. The RMSE is very high in general.
MAPE in the training partition is 13.92 % and, in the validation, partition is 13.08%. The difference between the training partition and the validation partition is minimal. Both rates are accepted.
Based on the above, we can conclude that there is no overfitting. The model is good for prediction.

3- Conducting Neural Network model with26 variables, 1000 iterations and 6 hidden layers:

We conduct normalization or data scaling when we have large variations between predictors or input variables. We generally do predictors scaling for better prediction results as suggested. Scaling is done through normalized weights using zscore. Basically, we calculate the difference between each point and the mean then divide the result by the sample standard deviation.
The standardization of predictor variables is done in this model to avoid any dominance of variables with higher values over the variable with lower values.

Calculating Intercepts and Network weights:

```
Intercepts for Housing Neural Network Model based on 1000 iterations, 6 hidden layers
[array([ -426.90398922,  1123.10071085,  -678.89624444, -1200.63507127,
       -1491.51467721,   857.58729829]), array([1388.47630119])]

Network Weights for Housing Neural Network Model based on 1000 iterations, 6 hidden layers
[array([[ 1.05649374e+02, -7.10846414e+01,  6.03581841e+01,
          1.76097670e+02, -1.86108208e+02,  3.87228545e+02],
       [ 6.29366546e+01,  1.07020702e+02, -1.38611018e+02,
         -2.47936806e+02,  1.47627140e+02, -3.36864641e+02],
       [ 3.71399206e+02,  4.19058263e+02,  6.97548204e+01,
          2.80386381e+02,  4.17749431e+02, -7.35486128e+01],
       [-1.44062938e+01,  3.74162612e+01,  4.97661688e+01,
         -1.91009877e+01,  1.23080496e+02,  2.50610649e-01],
       [ 3.35013834e+02,  3.29513609e+02,  2.70546216e+02,
          5.30520620e+01,  3.16028884e+02,  2.31405007e+02],
       [ 9.11614778e+01,  5.92719433e+01, -1.24570799e+02,
         -3.30341495e+01, -8.89356987e+00, -2.18870856e+02],
       [ 5.58090389e+01,  2.53605659e+01, -4.93195694e+01,
         -2.88055127e+02, -3.21120078e+02, -8.39345058e+01],
       [ 5.66043845e+00, -2.86766677e+01, -4.04638384e+01,
         -6.41082757e+01,  1.43926514e+02,  1.31468778e+02],
       [-8.49822469e+01, -2.11413091e+01, -7.31833465e+01,
          1.45886631e+02,  5.43846903e+00,  7.49115507e+01],
       [ 5.75210849e+02,  3.61075233e+01,  5.57027643e+01,
          7.61470903e+01,  3.27959847e+01, -1.05529862e+02],
       [-1.33192738e+01, -3.26374710e+01, -1.42328434e+01,
         -3.19031166e+02,  1.49968728e+02,  1.19169415e+02],
       [ 2.59436608e+01,  1.40658309e+01, -3.64276889e+02,
         -1.30375087e+02,  4.00542964e+02, -8.18251609e+00],
       [-3.74262915e+01,  7.46427392e+01, -2.48407198e+01,
          1.70307528e+01, -3.27153127e+01, -2.56010370e+02],
       [ 1.76528567e+01, -9.94285588e+01,  7.32256825e+01,
         -1.66992296e+02,  1.30335907e+01,  6.51009463e+02],
       [ 1.38508038e+02,  7.84797057e+01,  2.00034047e+02,
         -2.56304405e+02,  1.61927777e+02, -2.89738144e+02],
       [ 1.10024965e+02, -2.35782735e+01,  4.96478344e+01,
         -5.09402215e+01, -2.17945927e+01,  1.54334643e+02],
       [ 7.12575558e+01,  2.83913942e+00,  1.77130442e+01,
         -1.13629952e+02, -6.14873627e+01, -2.17662817e+01],
       [ 3.16612435e+00, -4.47764357e+01, -4.93669181e+01,
          5.57983048e+01, -3.00045477e+02,  1.75025450e+02],
       [ 5.29310273e+00,  1.26513281e+00,  8.02125634e+00,
          1.20375737e+01,  3.39448526e+01, -1.19695582e+01],
       [ 9.10839441e+01, -1.66979069e+01, -3.77430784e+01,
         -3.31135143e+00,  6.88596840e+01,  7.10083131e+01],
       [ 1.26039397e+02, -1.23607756e+01, -3.78055039e+01,
         -2.24506535e+02, -5.21268340e+01,  1.20370585e+01],
       [-8.13942493e+01,  5.97375968e+01, -5.77424642e+01,
          4.51717329e+01,  6.60526653e+01, -2.27503085e+02],
       [-2.49645822e+02,  4.86087968e+01, -6.93690903e+01,
          6.88609602e+01, -2.11502016e+01, -2.31589329e+02],
       [ 5.81141335e+01,  7.94298953e+01,  4.98594062e+01,
          3.59340752e+01,  1.99899524e+02, -2.03859882e+02],
       [ 4.05600491e+01,  3.70263194e+00, -3.94485725e+01,
         -3.13611193e+01,  2.71838269e+01, -3.10650608e-01],
       [ 2.07551675e+02,  5.12454026e+01, -2.41163950e+02,
          2.60935022e+02,  1.49424553e+02, -1.73713427e+02]]), array([[ 536.5404385 ],
       [1052.31105543],
       [ 893.85526665],
       [ 998.92100862],
       [1267.20227368],
       [ 284.40669445]])]
```

```
Accuracy Measures for Training Partition for Neural Network based on 1000 iterations

Regression statistics

                     Mean Error (ME) : -644.1920
      Root Mean Squared Error (RMSE) : 143620.6449
           Mean Absolute Error (MAE) : 92913.0213
         Mean Percentage Error (MPE) : -0.6655
Mean Absolute Percentage Error (MAPE) : 5.7893

Accuracy Measures for Validation Partition for Neural Network based on 1000 iterations

Regression statistics

                     Mean Error (ME) : 7181.6976
      Root Mean Squared Error (RMSE) : 186188.3034
           Mean Absolute Error (MAE) : 119183.1186
         Mean Percentage Error (MPE) : -0.1468
Mean Absolute Percentage Error (MAPE) : 6.8942
```

RMSE in the training partition is 143620.6449 and, in the validation, partition is 186188.3034. The difference betw een RMSE in the training and validation partitions is approximately 42568 which is acceptable. The RMSE is very hi gh in general.
MAPE in the training partition is 5.7893 % and, in the validation, partition is 6.8942% The difference between the t raining partition and the validation partition is minimal. Both rates are accepted.
Based on the above, we can conclude that there is no overfitting. The model is good for prediction.

4- Conducting Neural Network model with 26 variables, 500 iterations and 6 hidden layers:

```
Accuracy Measures for Training Partition for Neural Network based on 500 iterat
ions

Regression statistics

                     Mean Error (ME) : -533.4747
      Root Mean Squared Error (RMSE) : 145398.9278
           Mean Absolute Error (MAE) : 93892.8191
         Mean Percentage Error (MPE) : -0.6544
Mean Absolute Percentage Error (MAPE) : 5.8916

Accuracy Measures for Validation Partition for Neural Network based on 500 iter
ations

Regression statistics

                     Mean Error (ME) : 7702.0849
      Root Mean Squared Error (RMSE) : 185386.6290
           Mean Absolute Error (MAE) : 118198.2962
         Mean Percentage Error (MPE) : -0.1264
Mean Absolute Percentage Error (MAPE) : 6.8861
```

RMSE in the training partition is 145398,9278 and, in the validation, partition is 185386,6290. The difference betw een RMSE in the training and validation partitions is approximately 39987.7 which is acceptable. The RMSE is very high in general.
MAPE in the training partition is 5.891 % and, in the validation, partition is 6.886% The difference between the training partition and the validation partition is almost 1%. Both rates are accepted.
Based on the above, we can conclude that there is no overfitting. The model is good for prediction.

5- Conducting Neural Network using Grid Search CV, 1000 iterations and 13 hidden layers:

Using Grid Search CV, the best parameters are identified and applied those in the neural network model. The accuracy scores with the best parameter i.e. hidden_layer_sizes = 13 are presented below.

```
Accuracy Measures for Training Partition for Neural Network based on grid search

Regression statistics

                     Mean Error (ME) : -1355.2949
      Root Mean Squared Error (RMSE) : 102690.9313
           Mean Absolute Error (MAE) : 72557.2777
         Mean Percentage Error (MPE) : -0.4903
Mean Absolute Percentage Error (MAPE) : 5.1004

Accuracy Measures for Validation Partition for Neural Network based on grid search

Regression statistics

                     Mean Error (ME) : -7234.1836
      Root Mean Squared Error (RMSE) : 148069.9658
           Mean Absolute Error (MAE) : 104994.6890
         Mean Percentage Error (MPE) : -0.8196
Mean Absolute Percentage Error (MAPE) : 7.3130
```

RMSE in the training partition is 102690.9313 and, in the validation, partition is 148069.9658
. The difference between RMSE in the training and validation partitions is approximately 45379.03 which is accepta ble. The RMSE is very high in general.
MAPE in the training partition is 5.1004 % and, in the validation, partition is 7.313% The difference between the training partition and the validation partition is almost 2%. Both rates are accepted.
Based on the above, we can conclude that there is no overfitting. The model is good for prediction.

Conclusion:

```
Accuracy Measures for Training Set - All Variables

Regression statistics

                      Mean Error (ME) : -0.0000
       Root Mean Squared Error (RMSE) : 322849.3457
            Mean Absolute Error (MAE) : 194073.9406
          Mean Percentage Error (MPE) : 2.1051
Mean Absolute Percentage Error (MAPE) : 13.9432

Accuracy Measures for Validation Set - All Variables

Regression statistics

                      Mean Error (ME) : -12055.8340
       Root Mean Squared Error (RMSE) : 267077.8858
            Mean Absolute Error (MAE) : 178908.7909
          Mean Percentage Error (MPE) : 1.7451
Mean Absolute Percentage Error (MAPE) : 12.9972
```

```
Accuracy Measures for Training Set based on Backward elimination algorithm

Regression statistics

                      Mean Error (ME) : -0.0000
       Root Mean Squared Error (RMSE) : 323866.5611
            Mean Absolute Error (MAE) : 194572.1751
          Mean Percentage Error (MPE) : 2.0949
Mean Absolute Percentage Error (MAPE) : 13.9218

Accuracy Measures for Validation Set based on Backward elimination algorithm

Regression statistics

                      Mean Error (ME) : -13714.1137
       Root Mean Squared Error (RMSE) : 268675.0587
            Mean Absolute Error (MAE) : 180650.3527
          Mean Percentage Error (MPE) : 1.6380
Mean Absolute Percentage Error (MAPE) : 13.0804
```

```
Accuracy Measures for Training Partition for Neural Network based on 1000 iterations

Regression statistics

                      Mean Error (ME) : -644.1920
       Root Mean Squared Error (RMSE) : 143620.6449
            Mean Absolute Error (MAE) : 92913.0213
          Mean Percentage Error (MPE) : -0.6655
Mean Absolute Percentage Error (MAPE) : 5.7893

Accuracy Measures for Validation Partition for Neural Network based on 1000 iterations

Regression statistics

                      Mean Error (ME) : 7181.6976
       Root Mean Squared Error (RMSE) : 186188.3034
            Mean Absolute Error (MAE) : 119183.1186
          Mean Percentage Error (MPE) : -0.1468
Mean Absolute Percentage Error (MAPE) : 6.8942
```

```
Accuracy Measures for Training Partition for Neural Network based on 500 iterat
ions

Regression statistics

                      Mean Error (ME) : -533.4747
       Root Mean Squared Error (RMSE) : 145398.9278
            Mean Absolute Error (MAE) : 93892.8191
          Mean Percentage Error (MPE) : -0.6544
Mean Absolute Percentage Error (MAPE) : 5.8916

Accuracy Measures for Validation Partition for Neural Network based on 500 iter
ations

Regression statistics

                      Mean Error (ME) : 7702.0849
       Root Mean Squared Error (RMSE) : 185386.6290
            Mean Absolute Error (MAE) : 118198.2962
          Mean Percentage Error (MPE) : -0.1264
Mean Absolute Percentage Error (MAPE) : 6.8861
```

```
Accuracy Measures for Training Partition for Neural Network based on grid search

Regression statistics

                      Mean Error (ME) : -1355.2949
       Root Mean Squared Error (RMSE) : 102690.9313
            Mean Absolute Error (MAE) : 72557.2777
          Mean Percentage Error (MPE) : -0.4903
Mean Absolute Percentage Error (MAPE) : 5.1004

Accuracy Measures for Validation Partition for Neural Network based on grid search

Regression statistics

                      Mean Error (ME) : -7234.1836
       Root Mean Squared Error (RMSE) : 148069.9658
            Mean Absolute Error (MAE) : 104994.6890
          Mean Percentage Error (MPE) : -0.8196
Mean Absolute Percentage Error (MAPE) : 7.3130
```

1-

Looking at the RMSE in the 5 models, we can conclude that:

- RMSE values in using neural networks are away lower than using multiple regression model.
- MAPE value in the neural network work using Grid Search CV is not the lowest, it is in the average of the 5 models and the third in the neural network models.
- No overfitting noticed in any of the models, all of them can be used for predictions.

The minimum the error the better the model to be used in predictions.

Though the MAPE value in the neural network work using Grid Search CV is not the lowest, it is in the average of the 5 models and the third in the neural network models. We will still select this model given the RMSE value that never reached such a low level in any of the models conducted.

The best model for price predictions would be with the one developed using neural networks (Grid search.CV) with 13 hidden resulted in the lower RMSE (148069).

Bibliography:

Work cited in MLA format

- "Housing Prices and Inflation." *The White House*, The United States Government, 30 Nov. 2021, https://www.whitehouse.gov/cea/written-materials/2021/09/09/housing-prices-and-inflation/.

- Schaeffer, Katherine. "Key Facts about Housing Affordability in the U.S." *Pew Research Center*, Pew Research Center, 23 Mar. 2022, https://www.pewresearch.org/fact-tank/2022/03/23/key-facts-about-housing-affordability-in-the-u-s/.

- Schaeffer, Katherine. "Key Facts about Housing Affordability in the U.S." Pew Research Center, Pew Research Center, 23 Mar. 2022, https://www.pewresearch.org/fact-tank/2022/03/23/key-facts-about-housing-affordability-in-the-u-s/

- "Northern California." Windermere Real Estate, https://www.windermere.com/market-update/northern-california.

- "S&P/Case-Shiller U.S. National Home Price Index." FRED, 26 Apr. 2022, https://fred.stlouisfed.org/series/CSUSHPISA.

"S&P/Case-Shiller U.S. National Home Price Index." FRED, 26 Apr. 2022, https://fred.stlouisfed.org/series/CSUSHPISA.

- "Northern California." Windermere Real Estate, https://www.windermere.com/market-update/northern-california.

"S&P/Case-Shiller U.S. National Home Price Index." FRED, 26 Apr. 2022, https://fred.stlouisfed.org/series/CSUSHPISA.