Curbio Inc. Research on New Market Opportunities

Devi Nadimpally
College of Business and Economics
California State University East Bay
Hayward, USA
dnadimpally@horizon.csueastbay.edu

Karan Malik
College of Business and Economics
California State University East Bay
Hayward, USA
kmalik6@horizon.csueastbay.edu

Manogna Reddy Peddyreddy
College of Business and Economics
California State University East Bay
Hayward, USA
mpeddyreddy@horizon.csueastbay.edu

Richa Arora
College of Business and Economics
California State University East Bay
Hayward, USA
rarora8@horizon.csueastbay.edu

Abstract— The real estate market in the Bay Area is hot, with the median price topping $1 million for the twelfth month in a row. The median sale price for a Bay Area home last month was $1.33 million. It is the price in the very middle of a data set, with exactly half of the houses priced for less and half-priced for more in the Bay Area real estate market. The rising prices show that the Bay Area housing market is distinguished by high demand, cheap mortgage rates, and a scarcity of available inventory. When a seller sells his home, he faces a lot of issues in terms of repair, but the traditional improvement process is time-consuming, unreliable, and can be risky when a home transaction is at stake. We believe there is a better way to complete pre-listing updates and demonstrate them. The project outcomes help Curbio Inc to make business decisions for introducing new plans for customers in various segments accordingly.

I. Introduction

Our project focuses on Curbio Inc which does fast fixups to full renovations and takes care of everything, so you don’t have to—with zero due until the home sells. It is a leading start-up in the home renovation space that provides pre-approved home renovation options to home sellers, based on datasets acquired from Zillow. The pre-approved options help Curbio to bring in a new source of revenue by adding a tiered based business model “a 6-8% share on increased price after the home sells + cost of renovation” (gold Tier), Renovate and sell the property for the client includes brokerage (Platinum Tier).

II. Problem statement

The aim of this project is to analyze and uncover insights from the dataset involved in the project and help curbio to build its tier-based business models. The objective of the project is to focus mainly on Curbio which is intending to build a user interface, through which a home seller can input the current state of the home. The analysis and modeling would help the company in taking key decisions about the outlook/growth opportunities for the coming years.

To assess the risk involved in the change of business model, the company wants to research/analyze the existing properties in Zillow and build its own model for assessment. Through our project, we are trying to help curbio in data-driven market research so that they can build the model for a better analysis. Solving the above problems is technically difficult and interesting because they are unique problems in a new industry with fierce competition from bigger players such as Home Light, Redfin, etc. The project briefly talks about various factors and research questions stated below.

III. Research questions

- What is the total number of homes available in each city?
- What is the most popular home type?
- What is the impact on price if new attributes are added?
- What is the impact on the price with the change in total number of bedrooms?
- What is the impact on the property price with the change in total number of bathrooms?
- What is the impact on the property price with the change in living area?
- How does the property price change with the change in lot area?
- What is the most expensive and cheapest home type taking into consideration the average price/sqft?
- What is the average price of a bedroom and bath in the bay area?
- How do prices in tier 1 and tier 2 cities differ according to different features?
- What are the average house price for 3 beds and 2 baths for different home types?
- What are the most popular cities with the highest elementary, middle, and high school ratings?
- How does the price fluctuate with having a good school that is rated well in the bay area?
- How does renovating change the dynamics of property in tier 1 and tier 2 cities based on crime and school data? A comparative study to show the change in $ value for the same work in different cities?

IV. models Used

A. *Linearr Regression*

Linear regression attempts to model the relationship between two variables. One variable is regarded as an explanatory variable, while the other is regarded as a dependent variable. A modeler, for example, might want to use a linear regression model to relate people's weights to their heights. Linear regression, like all forms of regression analysis, focuses on the conditional probability distribution of the response given the predictor values, rather than the joint probability distribution of all these variables, which is the domain of multivariate analysis. By fitting a linear equation to observed data, linear regression attempts to model the relationship between two variables.

B. *Ordinary Least Squares Regression*

Ordinary Least Squares regression (OLS) is a method for estimating the coefficients of linear regression equations that describe the relationship between one or more independent quantitative variables and a dependent variable (simple or multiple linear regression). Least squares is an abbreviation for the minimum squares error (SSE). Alternative approaches to OLS include maximum likelihood and the Generalized Method of Moments Estimator. OLS regression is used to get the coefficient values which describe how each attribute is contributing to predict the target attribute.

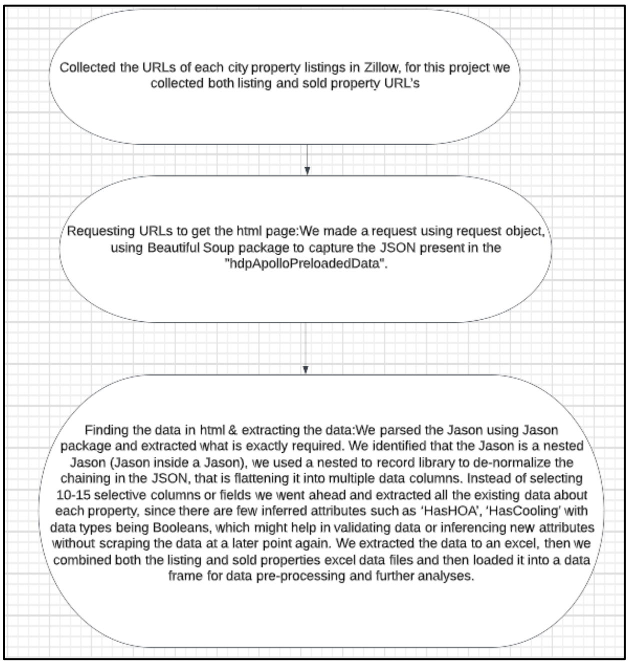
IV. IMPLEMENTATION AND EVALUATION

C. *Data Extraction*

In this modern world where data is one of the most valuable aids, any business can have. Data helps to generate insights, and to visualize relationships between different attributes, data collection/ extraction is a very important step in building any model. Data helps organizations end the guessing game and step up their strategies for development. There are so many ways to collect data, one of the most popular methods is web scraping. For housing-related study, We are relying on some datasets that are publicly available on Zillow. By making use of a variety of real estate metrics published by Zillow Research, will try to analyze and understand the performance of the housing segment. We are using different kinds of techniques to help answer a few questions for curbio to build its own model in the near future. We have followed the below web scraping process.

a) Web Scrapping Process

Identifying the target website:(Data source: https://www.zillow.com/



Collected the URLs of each city property listings in Zillow, for this project we collected both listing and sold property URL's

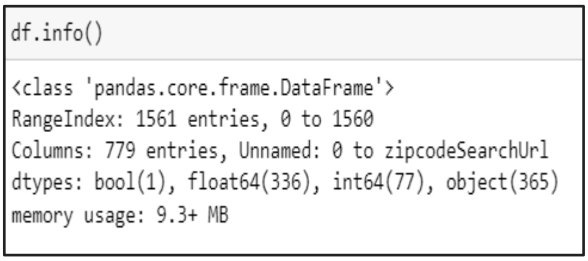
Requesting URLs to get the html page:We made a request using request object, using Beautiful Soup package to capture the JSON present in the "hdpApolloPreloadedData".

Finding the data in html & extracting the data:We parsed the Jason using Jason package and extracted what is exactly required. We identified that the Jason is a nested Jason (Jason inside a Jason), we used a nested to record library to de-normalize the chaining in the JSON, that is flattening it into multiple data columns. Instead of selecting 10-15 selective columns or fields we went ahead and extracted all the existing data about each property, since there are few inferred attributes such as 'HasHOA', 'HasCooling' with data types being Booleans, which might help in validating data or inferencing new attributes without scraping the data at a later point again. We extracted the data to an excel, then we combined both the listing and sold properties excel data files and then loaded it into a data frame for data pre-processing and further analyses.

Fig 1: Shows the data scraping process

b) Data Exploration

The data set that we extracted is a collection of data about 1561(rows) unique properties from 17 different cities in the bay area capturing 779(columns) unique data points about each property. These 779 unique data points contain data about various fields like property type, area of the property, number of bedrooms and bathrooms, etc.



```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1561 entries, 0 to 1560
Columns: 779 entries, Unnamed: 0 to zipcodeSearchUrl
dtypes: bool(1), float64(336), int64(77), object(365)
memory usage: 9.3+ MB
```

Fig 2: Shows information about the data

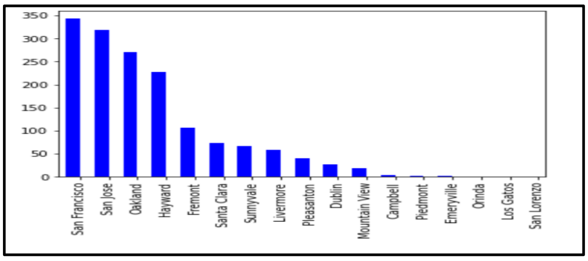


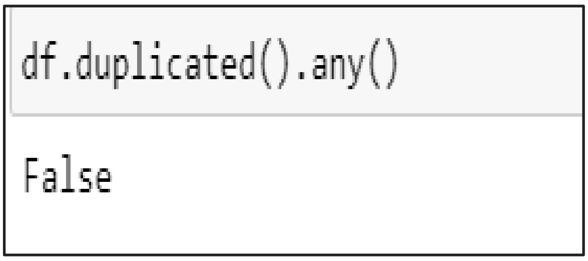
Fig 3: Shows different cities' data

B. *Data Preprocessing*

Data preprocessing refers to the various transformations that are applied to the data before feeding it into a model. Data preprocessing is a data mining technique of transferring the raw data into an understandable format, which is suitable for machine learning models. It is a very important step as we cannot work with raw data because raw data may contain inconsistent formatting, human errors, outliers and can also be incomplete.

b) Dropping unwanted columns: The 779 columns which we extracted from Zillow have so many Zillow derived and repeating data fields so we removed the unwanted data and shortlisted the columns to 253 for further analyses.

c) Data Deduplication: Data Deduplication is a technique to eliminate duplicate copies of repeating data. Data duplication may or may not be a big issue depending on the data sets we use and the purpose of the analyses, but duplicate data may be responsible for negative targeting implications, poor branding, unnecessary data storage, marketing budget waste and maybe a reason for many other effects, so it is best to check for duplicate values and remove them if there are any. We confirmed that our data do not have any duplicate values.



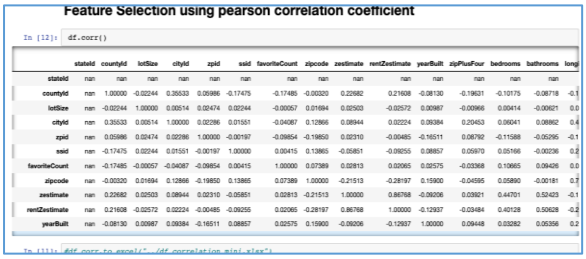
```
df.duplicated().any()

False
```

Fig 4: Shows if there is any duplicate data

d) Feature Selection: The input variables we feed to the model are called features, each column in our data set is a feature. To make a better model, we need to feed the model with important or necessary features instead of feeding the model with all the features in the dataset. The process where we automatically or manually select the features which contribute most to our prediction model is called feature selection.

Machine learning models follow a simple rule, whatever goes in comes out, if we feed the model with noise or unnecessary data we can expect noise in the output too, usually a good chunk of the data collected is noise or unimportant or waste data. Also, training a model with so much useless information will further decrease the accuracy or quality of the model and its time consuming, so selecting the good features is very important.

While there are so many feature selection techniques in machine learning, we are using the Pearson correlation coefficient technique in our project. Correlation coefficient is a measure that describes the degree to which the moment of two different variables is associated. If two variables are correlated then, we can predict one from the other. Since we will be predicting the 'Zestimate' attribute we are looking at the correlation coefficient of all the attributes with respect to 'Zestimate'. Correlation can be both positive and negative. If the score is high irrespective of the sign, they are highly correlated. We selected the features with correlation coefficient higher than 0.1and dropped the rest of the features, reducing the features to 51 from 253.



Feature Selection using pearson correlation coefficient

```
In [12]: df.corr()
```

	stateId	countyId	lotSize	cityId	zpid	ssid	favoriteCount	zipcode	zestimate	rentZestimate	yearBuilt	zipPlusFour	bedrooms	bathrooms	longi
stateId	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	
countyId	nan	1.00000	-0.02244	0.35533	0.05986	-0.17475	-0.17485	-0.00320	0.22682	0.21608	-0.08130	-0.19631	-0.10175	-0.08718	-0.1
lotSize	nan	-0.02244	1.00000	0.00514	0.02474	0.02244	-0.00057	0.01694	0.02503	-0.02572	0.00987	-0.00966	0.00414	-0.00621	0.0
cityId	nan	0.35533	0.00514	1.00000	0.02286	0.01551	-0.04087	0.12866	0.08944	0.02224	0.09384	0.20453	0.06041	0.08862	0.4
zpid	nan	0.05986	0.02474	0.02286	1.00000	-0.00197	-0.09854	-0.19850	0.02310	-0.00485	-0.16511	0.08792	-0.11588	-0.05295	-0.1
ssid	nan	-0.17475	0.02244	0.01551	-0.00197	1.00000	0.00415	0.13865	-0.05851	-0.09255	0.08857	0.05970	0.05166	-0.00236	0.2
favoriteCount	nan	-0.17485	-0.00057	-0.04087	-0.09854	0.00415	1.00000	0.07389	0.02813	0.02065	0.02575	-0.03368	0.10665	0.09426	0.0
zipcode	nan	-0.00320	0.01694	0.12866	-0.19850	0.13865	0.07389	1.00000	-0.21513	-0.28197	0.15900	-0.04595	0.05890	-0.00181	0.7
zestimate	nan	0.22682	0.02503	0.08944	0.02310	-0.05851	0.02813	-0.21513	1.00000	0.86768	-0.09206	0.03921	0.44701	0.52423	-0.1
rentZestimate	nan	0.21608	-0.02572	0.02224	-0.00485	-0.09255	0.02065	-0.28197	0.86768	1.00000	-0.12937	-0.03484	0.40128	0.50628	-0.2
yearBuilt	nan	-0.08130	0.00987	0.09384	-0.16511	0.08857	0.02575	0.15900	-0.09206	-0.12937	1.00000	0.09448	0.03282	0.05356	0.2

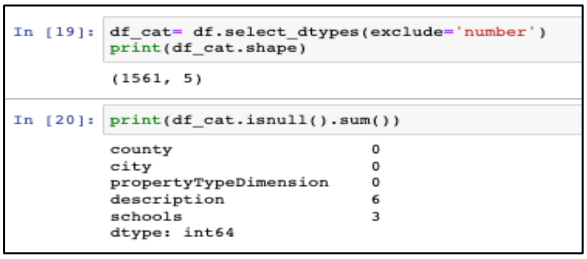
```
In [11]: #df.corr.to_excel("../df_correlation_mini.xlsx")
```

Fig 5: Shows feature selection

e) Data Classification

We have divided our data into two types:

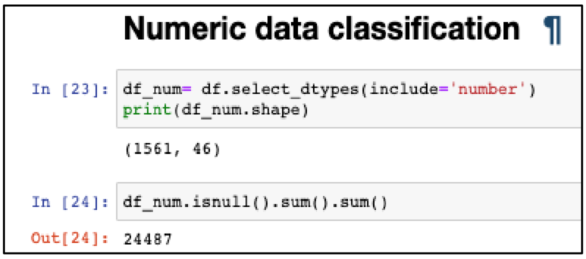
- *Categorical data:* The data which can be identified based on the names or labels given to them is called categorical data. In this project, we created a data frame 'df_cat' to store the categorical data. The below screenshot also shows the null values present in the data frame, which we deal with as we move forward in data preprocessing.



```
In [19]: df_cat= df.select_dtypes(exclude='number')
         print(df_cat.shape)

         (1561, 5)

In [20]: print(df_cat.isnull().sum())

         county                     0
         city                       0
         propertyTypeDimension      0
         description                6
         schools                    3
         dtype: int64
```

Fig 6: Creating data frame and checking null values and datatypes

Numerical data: The data which includes numbers and is not in the form of any descriptive form can be called as numerical data. In this project, we created a data frame 'df_num' to store the numerical data. The below screenshot also shows the null values present in the data frame, which we deal with as we move forward in data preprocessing.



Numeric data classification ¶

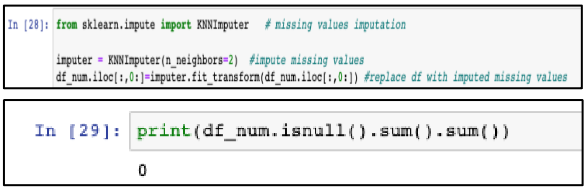
```
In [23]: df_num= df.select_dtypes(include='number')
         print(df_num.shape)

         (1561, 46)

In [24]: df_num.isnull().sum().sum()

Out[24]: 24487
```

Fig 7: Creating data frame and checking null values and datatypes

C. Missing Value imputation

It is quite common to see null values, also called as missing values in the data frame. Missing value imputation is a technique used to replace all these null values with some substitute value to retain the data/information from the data set. There are so many ways to impute missing values, in this project we used 3 different ways, depending on the type of the attribute we are dealing with.

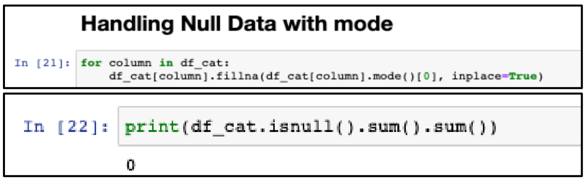
b) KNN Imputer: It is a very popular method from scikit-learn class, where we replace the missing values with the mean value of the k-neighbors from the dataset. It identifies the k-samples in the dataset that are near the missing value. In this project, we use knn imputer to replace all the missing values in the numeric data set.



```
In [28]: from sklearn.impute import KNNImputer   # missing values imputation

         imputer = KNNImputer(n_neighbors=2)  #impute missing values
         df_num.iloc[:,0:]=imputer.fit_transform(df_num.iloc[:,0:]) #replace df with imputed missing values

In [29]: print(df_num.isnull().sum().sum())

         0
```

Fig 8: Checking the null values

c) Mode Imputation: In this technique, we replace the missing value with the mode value or most frequently occurred value from the entire column. It is quite popular in dealing with categorical variables. In this project, we used this method to replace missing values in the categorical data frame.

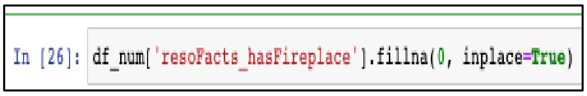


Handling Null Data with mode

```
In [21]: for column in df_cat:
             df_cat[column].fillna(df_cat[column].mode()[0], inplace=True)

In [22]: print(df_cat.isnull().sum().sum())

         0
```

Fig 9: Shows how we handle null data with mode

d) Other Imputation: Replacing null values in the fireplace by 0 and replacing nulls in Zestimate by price values



```
In [26]: df_num['resoFacts_hasFireplace'].fillna(0, inplace=True)
```

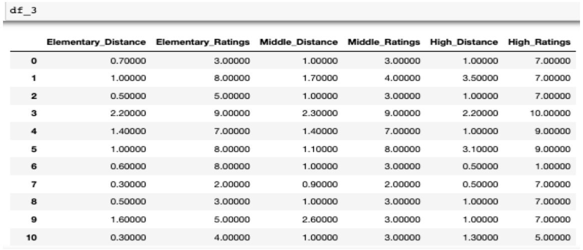
Fig 10: Shows how we have replaced the null values with 0

D. Feature Engineering

The process of creating a new feature from or using the existing features is called feature engineering. This can be done by using domain knowledge to select and transform the most relevant variable from the original dataset and transferring it into a desirable variable.

In this project, we are creating six different features from the 'school' attribute, which has useful information stored in it a very complicated way. The below table shows the six new features created from the 'schools' attribute which has data stored in it as below. (it is stored in a single cell) We have dropped the school column from the data sets and replaced it with 6 other columns.

Schools=[{'distance': 0.7, 'name': 'Parker Elementary', 'rating': 3, 'level': 'Elementary', 'studentsPerTeacher': 20, 'assigned': None, 'grades': 'K-8', 'link': 'https://www.greatschools.org/school?id=00249&state=CA', 'type': 'Public', 'size': 314, 'totalCount': 1, 'isAssigned': True}]



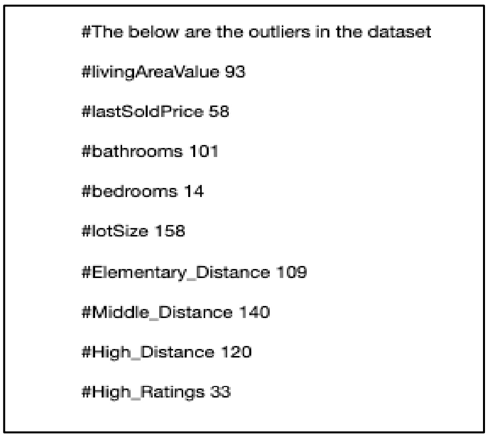
```
df_3
```

	Elementary_Distance	Elementary_Ratings	Middle_Distance	Middle_Ratings	High_Distance	High_Ratings
0	0.70000	3.00000	1.00000	3.00000	1.00000	7.00000
1	1.00000	8.00000	1.70000	4.00000	3.50000	7.00000
2	0.50000	5.00000	1.00000	3.00000	1.00000	7.00000
3	2.20000	9.00000	2.30000	9.00000	2.20000	10.00000
4	1.40000	7.00000	1.40000	7.00000	1.00000	9.00000
5	1.00000	8.00000	1.10000	8.00000	3.10000	9.00000
6	0.60000	8.00000	1.00000	3.00000	0.50000	1.00000
7	0.30000	2.00000	0.90000	2.00000	0.50000	7.00000
8	0.50000	3.00000	1.00000	3.00000	1.00000	7.00000
9	1.60000	5.00000	2.60000	3.00000	1.00000	7.00000
10	0.30000	4.00000	1.00000	3.00000	1.30000	5.00000

Table 1: Shows the six new features created from the schools

E. Detecting Outliers

Outliers are the data points that differ significantly away from other observations. In general, we remove any outliers present in a dataset before feeding it to the model, but simply removing outliers may not be the best practice to produce a better-fitted model or statistically significant results. What if the outlier value is a natural part of the dataset we are studying? So, in this project, we are leaving the outlier values in the dataset just the same. Below are the outliers present in the dataset.



```
#The below are the outliers in the dataset

#livingAreaValue 93

#lastSoldPrice 58

#bathrooms 101

#bedrooms 14

#lotSize 158

#Elementary_Distance 109

#Middle_Distance 140

#High_Distance 120

#High_Ratings 33
```

Fig 11: Shows outliers in the dataset

F. Removing Multi-Collinearity

Multicollinearity occurs when independent variables in a regression model are highly correlated to each other. To build a better model we should make sure that there is no relation between independent or predictor variables. Multicollinearity increases the standard errors of the coefficients. There are different ways to remove or to overcome multicollinearity. One way to measure multicollinearity is VIF (Variance Inflation Factor). The high VIF values indicate that there is high multicollinearity. There is no such standard value to compare or use for the VIF value, it all depends on the trial-and-error method and is completely the decision of the analyst. In this project, we are dropping all the columns whose VIF value is greater than 12.5.

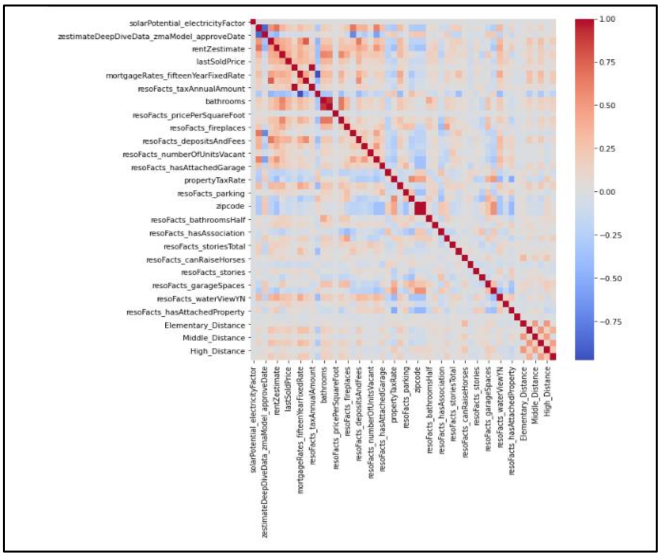
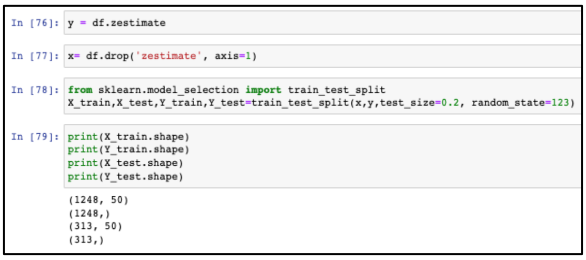


Fig 12: Heat Map showing correlation between the attributes

G. Train-Test data splitting

The data has been divided into Trian and Test as follows for further analyses. The best train-test ratio is 80:20, we are using 80 percent of the observations for training and the rest 20 percent is used for testing.

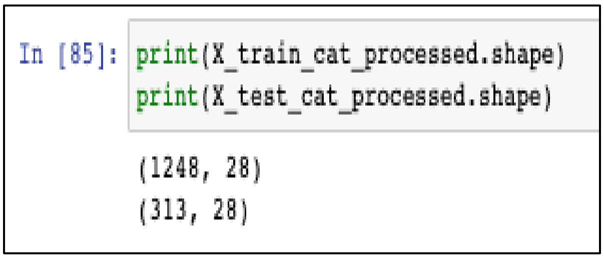


```
In [76]: y = df.zestimate

In [77]: x= df.drop('zestimate', axis=1)

In [78]: from sklearn.model_selection import train_test_split
         X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.2, random_state=123)

In [79]: print(X_train.shape)
         print(Y_train.shape)
         print(X_test.shape)
         print(Y_test.shape)

         (1248, 50)
         (1248,)
         (313, 50)
         (313,)
```

Fig 13: Splitting the data set

H. One-hot encoding & numeric data preparation

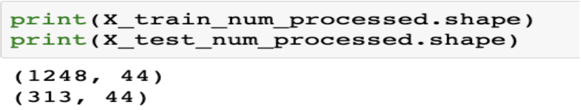
The process of converting categorical variables into a form that could be provided to ML algorithms to improve their prediction accuracy is called One-hot encoding. In this project, we converted the 'city', 'county' and 'propertyTypeDimension' attributes into numerical data by performing one-hot encoding. We were able to derive 28 new attributes from 3 attributes for categorical data.



```
In [85]: print(X_train_cat_processed.shape)
         print(X_test_cat_processed.shape)

         (1248, 28)
         (313, 28)
```

Fig 14: Train and test dataset using one-hot encoding

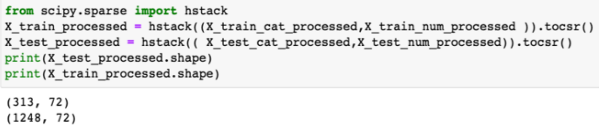
Now, that the one-hot encoding is done on the categorical data, we must convert the numerical data into the coo matrix to combine the data with categorical data.



```
print(X_train_num_processed.shape)
print(X_test_num_processed.shape)

(1248, 44)
(313, 44)
```

Fig 15: Train and test data on numerical data

We combined the categorical and numeric data frames, and we have our final data frames to feed the linear and OLS regression models as we can see in the below screenshot.



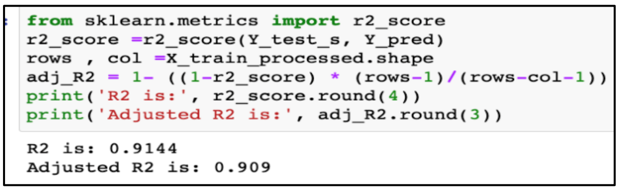
```
from scipy.sparse import hstack
X_train_processed = hstack((X_train_cat_processed,X_train_num_processed )).tocsr()
X_test_processed = hstack(( X_test_cat_processed,X_test_num_processed)).tocsr()
print(X_test_processed.shape)
print(X_train_processed.shape)

(313, 72)
(1248, 72)
```

Fig 16: Combined the categorical and numeric data frames

I. *Data Rescaling/Standardizing*

The process of bringing all the variables to the same scale is called standardizing. Standardizing variables is a simple process. To standardize the variables, we calculate the mean and standard deviation of the attribute, and then we subtract the mean and divide the value with the standard deviation for all the variables in the attribute.

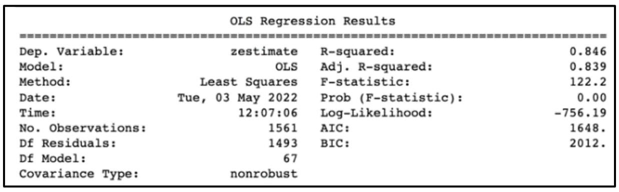
After standardizing the numerical values, we ran the Liner, model, on the rescaled data and got the R- square value of 0.9144 and an Adjusted R-square value of 0.909 for linear regression as shown in the below screenshot.



```
from sklearn.metrics import r2_score
r2_score =r2_score(Y_test_s, Y_pred)
rows , col =X_train_processed.shape
adj_R2 = 1- ((1-r2_score) * (rows-1)/(rows-col-1))
print('R2 is:', r2_score.round(4))
print('Adjusted R2 is:', adj_R2.round(3))

R2 is: 0.9144
Adjusted R2 is: 0.909
```

Fig 17: Combined the categorical and numeric data frames

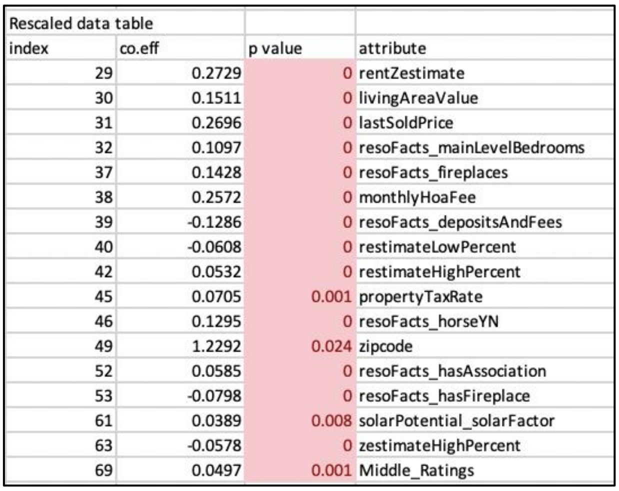
After standardizing the numerical values, we ran the OLS Regression model on the rescaled data and got the R- square value of 0.846 and an Adjusted R-square value of 0.839 for OLS regression as shown in the below screenshot.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:              zestimate   R-squared:                       0.846
Model:                            OLS   Adj. R-squared:                  0.839
Method:                 Least Squares   F-statistic:                     122.2
Date:                Tue, 03 May 2022   Prob (F-statistic):               0.00
Time:                        12:07:06   Log-Likelihood:                -756.19
No. Observations:                1561   AIC:                             1648.
Df Residuals:                    1493   BIC:                             2012.
Df Model:                          67
Covariance Type:            nonrobust
```

Table 2: Ols regression results

Our main goal in this project is to understand how each attribute is contributing to predicting the Zestimate, so we are using OLS regression. The below table shows how important is each attribute in predicting Zestimate, since the data is standardized, we can't exactly infer how much the coefficient value changes in dollars if we are increasing the significant attributes by one unit. But we can compare the attributes, like which attribute is more important than the other. The below table is showing the significant attributes and the coefficients in descending order.



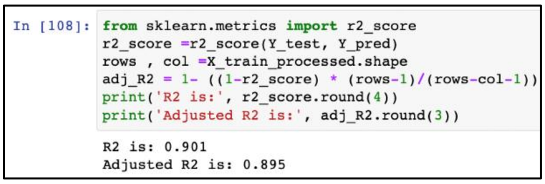
Rescaled data table			
index	co.eff	p value	attribute
29	0.2729	0	rentZestimate
30	0.1511	0	livingAreaValue
31	0.2696	0	lastSoldPrice
32	0.1097	0	resoFacts_mainLevelBedrooms
37	0.1428	0	resoFacts_fireplaces
38	0.2572	0	monthlyHoaFee
39	-0.1286	0	resoFacts_depositsAndFees
40	-0.0608	0	restimateLowPercent
42	0.0532	0	restimateHighPercent
45	0.0705	0.001	propertyTaxRate
46	0.1295	0	resoFacts_horseYN
49	1.2292	0.024	zipcode
52	0.0585	0	resoFacts_hasAssociation
53	-0.0798	0	resoFacts_hasFireplace
61	0.0389	0.008	solarPotential_solarFactor
63	-0.0578	0	zestimateHighPercent
69	0.0497	0.001	Middle_Ratings

Table 3: Rescaling the data table

Since we want to study how much each attribute is contributing to the Zestimate, we are running both the Linear and OLS regression models on the data without standardization in the next steps.

J. *Machine Learning Models*

b) *Linear Regression Model*

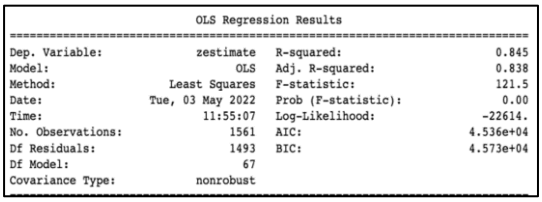


```
In [108]: from sklearn.metrics import r2_score
          r2_score =r2_score(Y_test, Y_pred)
          rows , col =X_train_processed.shape
          adj_R2 = 1- ((1-r2_score) * (rows-1)/(rows-col-1))
          print('R2 is:', r2_score.round(4))
          print('Adjusted R2 is:', adj_R2.round(3))

          R2 is: 0.901
          Adjusted R2 is: 0.895
```

Fig 18: Linear Regression model results

We have used linear regression to predict the value of Zestimate on the basis of other 31 independent attributes like school, lot size, last sold price, etc. we can see that the R-square value is 0.901 and the adjusted R-square value is 0.895, which means that it is a good model.

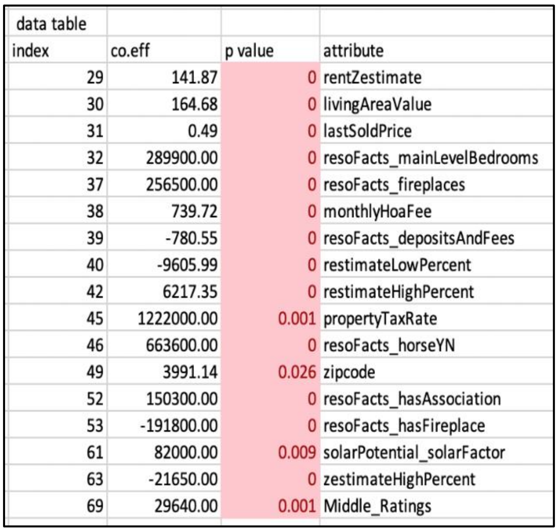
c) *OLS Regression Model*



```
                            OLS Regression Results
==============================================================================
Dep. Variable:              zestimate   R-squared:                       0.845
Model:                            OLS   Adj. R-squared:                  0.838
Method:                 Least Squares   F-statistic:                     121.5
Date:                Tue, 03 May 2022   Prob (F-statistic):               0.00
Time:                        11:55:07   Log-Likelihood:                -22614.
No. Observations:                1561   AIC:                         4.536e+04
Df Residuals:                    1493   BIC:                         4.573e+04
Df Model:                          67
Covariance Type:            nonrobust
```

Table 4: Ols regression results

We have used OLS regression to predict the value of Zestimate on the basis of other 31 independent attributes like school, lot size, last sold price, etc. we can see that the R-square is 0.845 and the adjusted R-square is 0.838, which means that it is a good model. Our main goal in this project is to understand how each attribute is contributing to predicting the Zestimate, so we are using OLS regression, The below table shows how much each attribute in contributing in predicting the Zestimate.



data table			
index	co.eff	p value	attribute
29	141.87	0	rentZestimate
30	164.68	0	livingAreaValue
31	0.49	0	lastSoldPrice
32	289900.00	0	resoFacts_mainLevelBedrooms
37	256500.00	0	resoFacts_fireplaces
38	739.72	0	monthlyHoaFee
39	-780.55	0	resoFacts_depositsAndFees
40	-9605.99	0	restimateLowPercent
42	6217.35	0	restimateHighPercent
45	1222000.00	0.001	propertyTaxRate
46	663600.00	0	resoFacts_horseYN
49	3991.14	0.026	zipcode
52	150300.00	0	resoFacts_hasAssociation
53	-191800.00	0	resoFacts_hasFireplace
61	82000.00	0.009	solarPotential_solarFactor
63	-21650.00	0	zestimateHighPercent
69	29640.00	0.001	Middle_Ratings

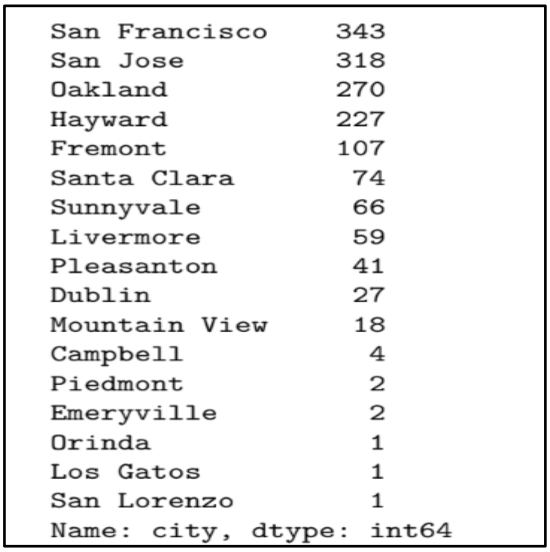
Table 5: Rescaling the data table

K. DATA ANALYSIS AND VISUALIZATION

Data visualization is a method for providing facts in the form of charts and graphs. It is often easier for us to take decisions using interactive visualization and to spot new trends. The simplicity with which enormous volumes of complex data may be displayed using charts or graphs instead of spreadsheets or reports is one of the advantages of displaying visual data. Here, “ matplotlib.pyplot, seaborn, plotly. express, plotly.graph_objs, make_subplots are some of the libraries we used to analyze and display information on our data.

b) Number of homes available in each city

we have created a series from the column 'city' and below is the table and bar plot showing the count of available homes in each city of the bay area.



```
San Francisco     343
San Jose          318
Oakland           270
Hayward           227
Fremont           107
Santa Clara        74
Sunnyvale          66
Livermore          59
Pleasanton         41
Dublin             27
Mountain View      18
Campbell            4
Piedmont            2
Emeryville          2
Orinda              1
Los Gatos           1
San Lorenzo         1
Name: city, dtype: int64
```

Table 5: Shows different cities with count of homes

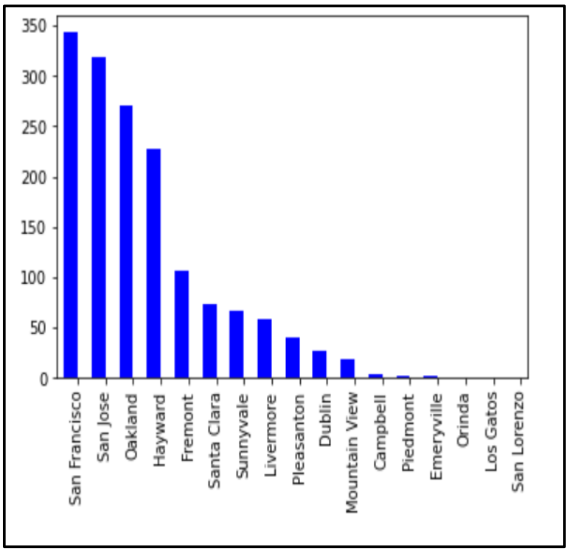
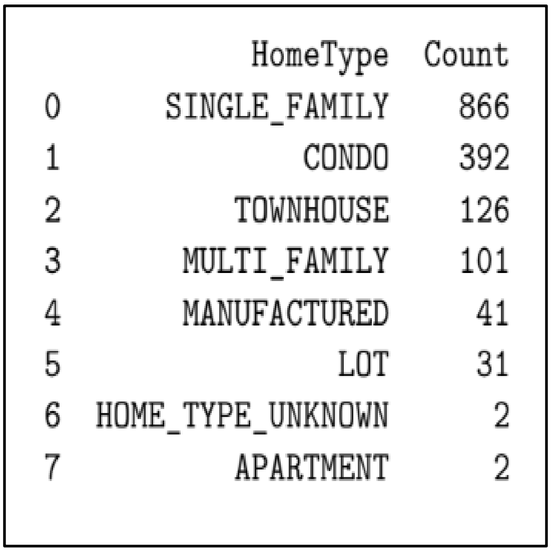


Fig 19: Graphic representation of city by home count

San Francisco, with 343 homes, has the most properties for sale, according to the graph above. There are 318 in San Jose and 270 in Oakland. The cities of Orinda, Los Gatos, and San Lorenzo have the fewest listings.

c) Most popular home type

To visualize the most popular home type, we used plotly. express and created a series of “home type” columns to show the frequency of each home type.



```
          HomeType  Count
0     SINGLE_FAMILY    866
1             CONDO    392
2         TOWNHOUSE    126
3      MULTI_FAMILY    101
4      MANUFACTURED     41
5               LOT     31
6  HOME_TYPE_UNKNOWN      2
7         APARTMENT      2
```

Table 6: Shows different cities with home type

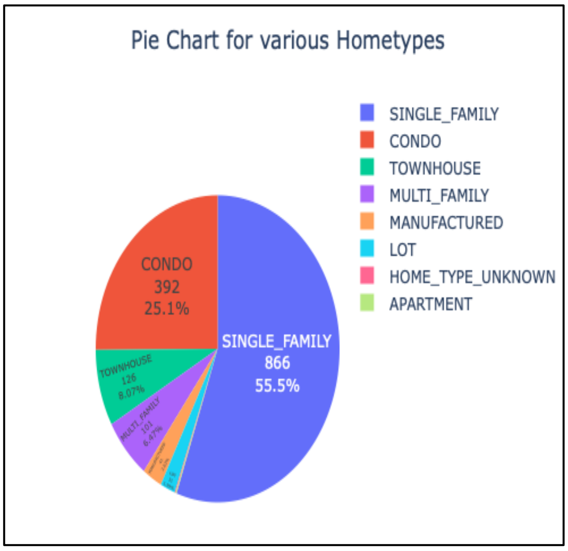


Table 6: Shows home count along with the type of home

Single-family houses are the most common home type, accounting for 55.5 percent of all available homes. Condominiums are the second most popular dwelling type, accounting for 25.1 percent of all homes. Townhouses, multi-family homes, and manufactured homes all contribute less than 10%, whereas lots account for roughly 1.99 percent of all available homes.

d) *Impact of having extra features*

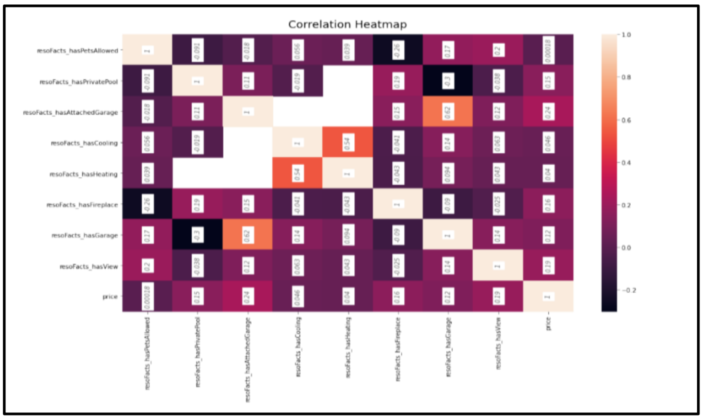


Fig 20: Heatmap showing the impact of different attributes on price

From the heatmap, we can deduce that resoFacts_hasAttachedGarage, resoFacts_hasView, resoFacts_hasFireplace, resoFacts_hasPrivatePool have a significant pricing influence. resofact_hasGarage, resoFacts_hasCooling, resoFacts_hasHeating, resoFacts_hasPetsAllowed on the other hand, have a less impact on price fluctuation.

e) *Variation of property price with bedrooms*

To analyze the Variation of Property prices with bedrooms. A scatter plot was mapped with parameters: Property Price, No. of Bedrooms, with Home Type taken as the marker.

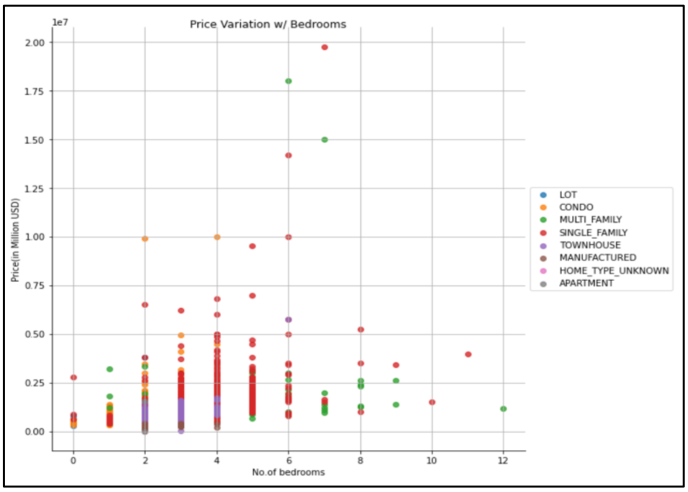


Fig 21: Scatter plot showing the property price along with the bedroom

The Scatter plot above illustrates the price of the property and the number of bedrooms, using the home type as the marker. We've noticed that as the number of bedrooms increases from 0 to 7, the price of single-family and condo homes increases. The price of a lot, townhouse, manufactured home, or apartment has increased slightly in relation to the number of bedrooms. we can conclude that price has a positive linear relationship with the number of bedrooms.

f) *Variation of property price with bathrooms*

To analyze the Variation of Property prices with Bathrooms. A scatter plot was mapped with parameters: Property Price, No. of Bathrooms, with Home Type taken as the marker.

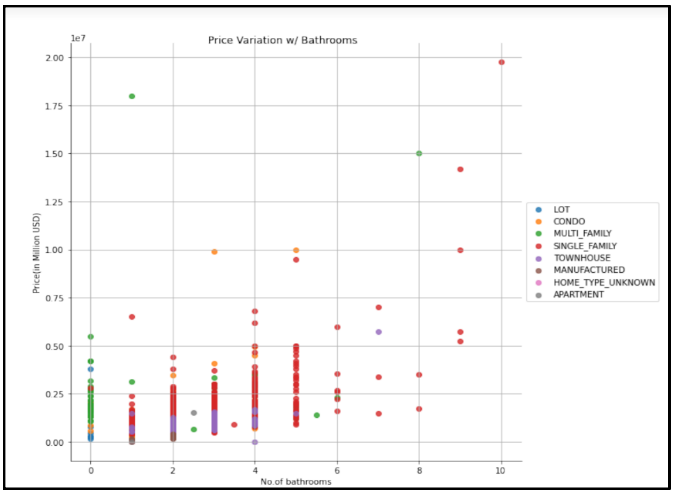


Fig 22: Scatter plot showing the change in the property price according to bathroom

In the graph above, we can see that the price of single-family homes rises as the number of bathrooms increases. Condo residences in the Bay Area with less than 5 bathrooms have a maximum price of $1 million.

g) Variation of property price with living area

To infer the Variation of Property Price with Living Area. A histogram was plotted with parameters: Property Price, Living Area; with Living Area partitioned into 4 categories, for a better understanding of the spread.

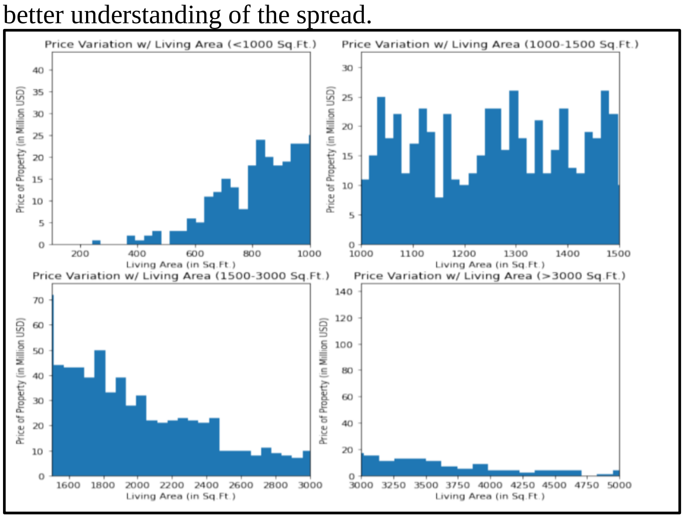


Fig 23: Bar plot showing the variation of property prices along with the living area

According to the plot, we can be see, the price rises when the living area is between 600 and 1500 square feet. When the living area is between 2400 and 5000 sq.ft., the price starts to drop rapidly.

h) Variation of property price with lot area

To analyze the Variation of Property Price with Lot Area. A scatter plot was mapped with parameters: Property Price, Lot Area, with County, Home Type taken as the marker.

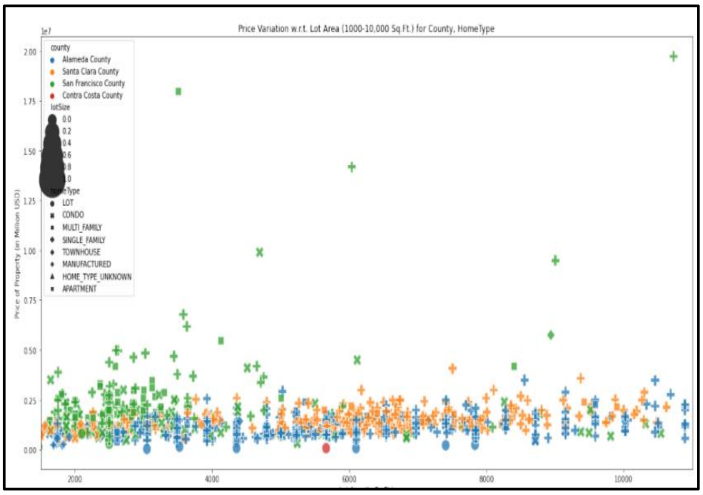


Fig 24: Scatter plot showing the change in the home price with lot area

According to the accompanying graph, residences in San Francisco County are mainly between 2000 and 5000 square feet. In San Francisco County, the median home price is under $1 million, with only a few residences over $1 million. Homes in Santa Clara and Alameda range in size from 2000 to 10,000 square feet. The property is valued at less than $500,000.

i) Most expensive and cheapest home in terms of average price per square feet

To draw visualization details about the most expensive and cheapest home types we used Home type and resoFacts_pricePerSquareFoot columns, grouped them by home type, and calculated the average price. Below is the table which shows the average price per square foot for each home type.

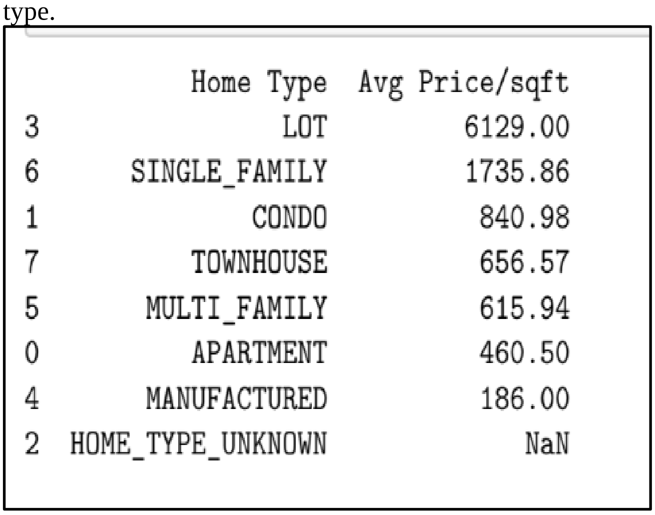
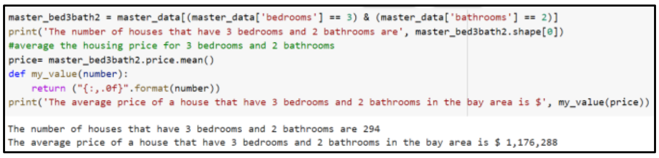

```
           Home Type  Avg Price/sqft
3                LOT         6129.00
6      SINGLE_FAMILY         1735.86
1              CONDO          840.98
7          TOWNHOUSE          656.57
5       MULTI_FAMILY          615.94
0          APARTMENT          460.50
4       MANUFACTURED          186.00
2  HOME_TYPE_UNKNOWN             NaN
```

Table 7: Average price per square feet according to the home type

According to the previous conclusions, lot homes are less popular than other home kinds, but they have the highest average price, $6129/sqft. The most common housing type, single family, costs around $1736/sqft. The average price per square foot for a condo and a townhouse is $840 and $815, respectively. The average price of an apartment is $460/sqft, whereas the average price of a manufactured home is $186/sqft.

j) Average housing price for 3 bed and 3 bath in bay area

We utilized properties with 3 bedrooms and 2 bathrooms to obtain the average dwelling price for 3 bed, 2 bath. The code below will return the number of homes with 3 bedrooms and 2 bathrooms, as well as their average price.



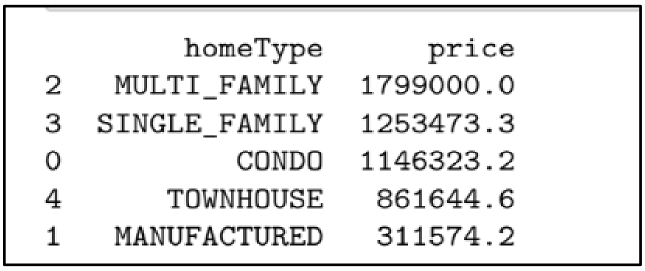
```
master_bed3bath2 = master_data[(master_data['bedrooms'] == 3) & (master_data['bathrooms'] == 2)]
print('The number of houses that have 3 bedrooms and 2 bathrooms are', master_bed3bath2.shape[0])
#average the housing price for 3 bedrooms and 2 bathrooms
price= master_bed3bath2.price.mean()
def my_value(number):
    return ("{:,.0f}".format(number))
print('The average price of a house that have 3 bedrooms and 2 bathrooms in the bay area is $', my_value(price))

The number of houses that have 3 bedrooms and 2 bathrooms are 294
The average price of a house that have 3 bedrooms and 2 bathrooms in the bay area is $ 1,176,288
```

Fig 24: Shows the average housing price for 3 bed and 2 bathrooms

The results came out to be, the number of houses that have 3 bedrooms and 2 bathrooms are 294. The average price of a house that have 3 bedrooms and 2 bathrooms in the bay area is $ 1,176,288.

k) Average housing price for 3 bed and 2 bath for different home types

We used the price and home type of all houses with 3bedrooms and 2bathrooms, grouped them by home type and average and calculated housing price.



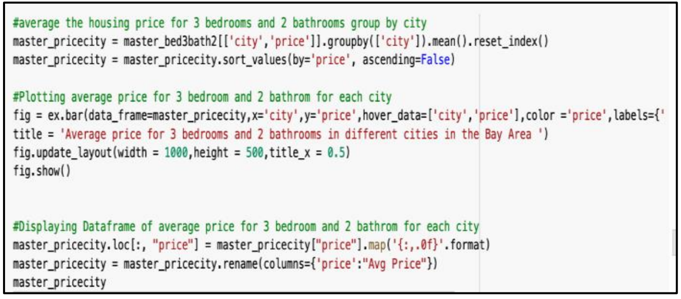
	homeType	price
2	MULTI_FAMILY	1799000.0
3	SINGLE_FAMILY	1253473.3
0	CONDO	1146323.2
4	TOWNHOUSE	861644.6
1	MANUFACTURED	311574.2

Table 8: Average housing price for 3 bedrooms and 2 bathrooms

We see that the average housing price for a three-bedroom, two-bathroom multifamily home is around 1.8 million dollars. The price difference between a three-bedroom, two-bathroom single family home and a three-bedroom, two-bathroom condo is 107k dollars. In other words, the average price of a single-family home is around 1.25 million dollars, while the average price of a condo is around 1.15 million dollars. The average cost of a townhouse is $862,000. Manufactured homes are the least expensive, costing less than $400,000.

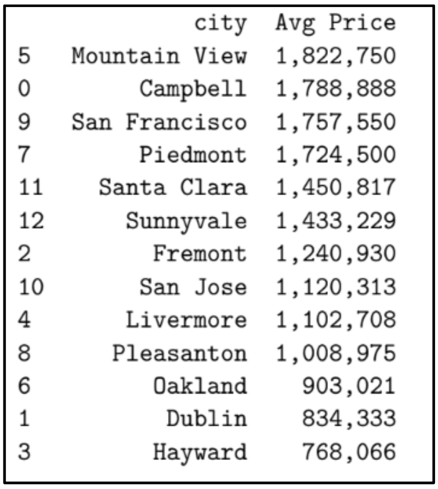
l) Average housing price for 3bedroom and 2 bathrooms in bay area by cities

We used plotly.express to plot the average housing price. Parameters we used are price and city, which are grouped by city. Below is the code to determine Average housing price of each city.



```
#average the housing price for 3 bedrooms and 2 bathrooms group by city
master_pricecity = master_bed3bath2[['city','price']].groupby(['city']).mean().reset_index()
master_pricecity = master_pricecity.sort_values(by='price', ascending=False)

#Plotting average price for 3 bedroom and 2 bathrom for each city
fig = ex.bar(data_frame=master_pricecity,x='city',y='price',hover_data=['city','price'],color ='price',labels={'
title = 'Average price for 3 bedrooms and 2 bathrooms in different cities in the Bay Area ')
fig.update_layout(width = 1000,height = 500,title_x = 0.5)
fig.show()


#Displaying Dataframe of average price for 3 bedroom and 2 bathrom for each city
master_pricecity.loc[:, "price"] = master_pricecity["price"].map('{:,.0f}'.format)
master_pricecity = master_pricecity.rename(columns={'price':"Avg Price"})
master_pricecity
```

Fig 25: Shows the average housing price for 3 bed and 2 bathrooms



	city	Avg Price
5	Mountain View	1,822,750
0	Campbell	1,788,888
9	San Francisco	1,757,550
7	Piedmont	1,724,500
11	Santa Clara	1,450,817
12	Sunnyvale	1,433,229
2	Fremont	1,240,930
10	San Jose	1,120,313
4	Livermore	1,102,708
8	Pleasanton	1,008,975
6	Oakland	903,021
1	Dublin	834,333
3	Hayward	768,066

Table 9: Average housing price for 3 bedroom and 2 bathroom

In terms of three bedrooms and two bathrooms, we can see that Mountain View is the most costly area, followed by Campbell and San Francisco. A home in Campbell, San Francisco, or Piedmont can be purchased for 1.7 million to 1.8 million dollars. Pleasanton, Oakland, Dublin, and Hayward are the cheapest cities to buy a property for under a million dollars when compared to other cities.

m) Cities with the highest elementary, middle, and high school ratings

The parameters used to plot the histograms are Elementary_ratings, Middle_ratings, High_ratings, and city which are grouped by city.

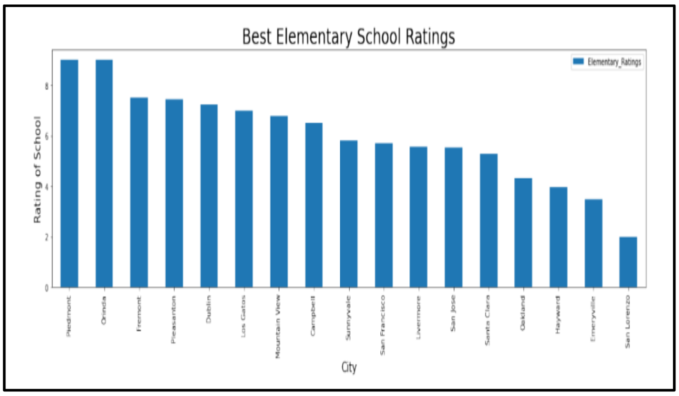


Fig 26: Bar plot showing the city with the highest elementary, middle and high school ratings

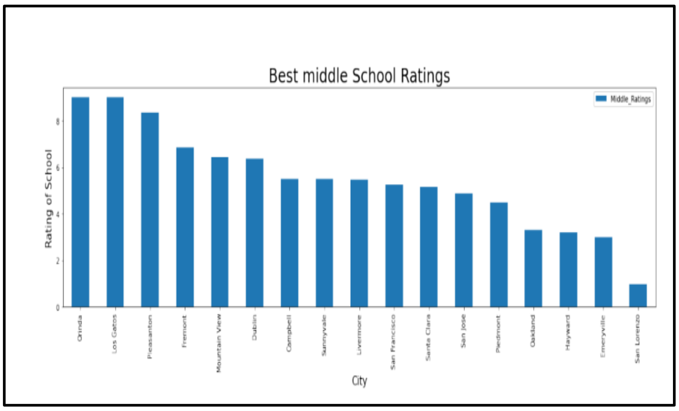


Fig 27: Bar plot showing the best middle school rating

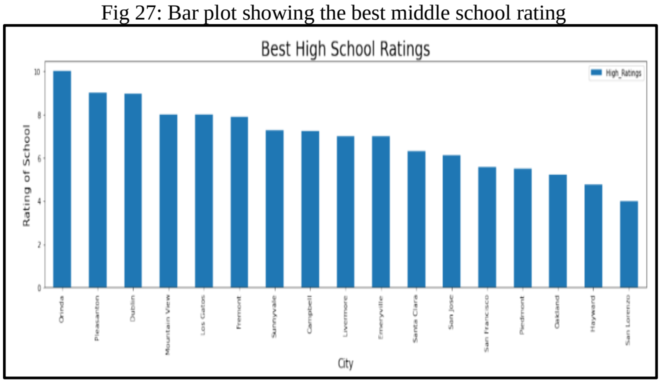
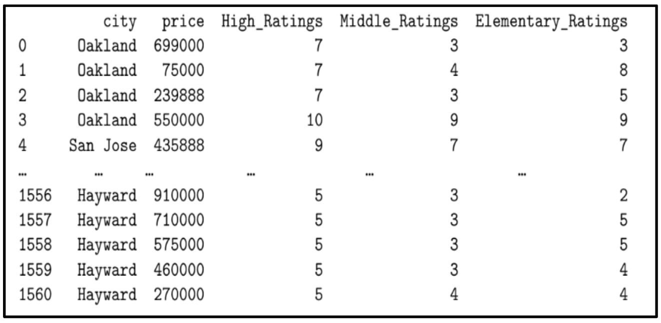


Fig 28: Best high school ratings

From the above figures, we observe that Piedmont has the highest elementary school rating, while middle and high school ratings are not great. Orinda has the best elementary, middle and high school ratings. San Lorenzo school ratings are very low compared to other cities. Pleasanton, Dublin, Fremont, Mountain View, and Los Gatos are the safest places to buy a home considering the school ratings.

n) Impact of school ratings on housing prices in the bay area

Created a data frame with attributes city, price, elementary rating, middle ratings, and high school ratings. The data is plotted using subplots having the same x-axis to help us to understand the impact on price with respect to the school rating.



```
          city   price  High_Ratings  Middle_Ratings  Elementary_Ratings
0      Oakland  699000             7               3                   3
1      Oakland   75000             7               4                   8
2      Oakland  239888             7               3                   5
3      Oakland  550000            10               9                   9
4     San Jose  435888             9               7                   7
...        ...     ...           ...             ...                 ...
1556   Hayward  910000             5               3                   2
1557   Hayward  710000             5               3                   5
1558   Hayward  575000             5               3                   5
1559   Hayward  460000             5               3                   4
1560   Hayward  270000             5               4                   4
```

Table 9: Average housing price for 3 bedrooms and 2 bathrooms

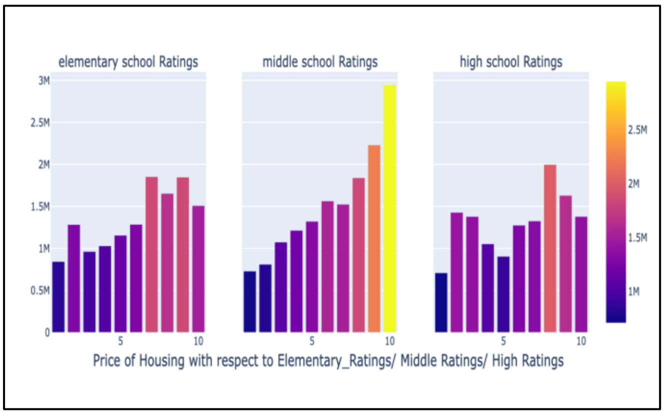


Figure 29: Impact of school rating on housing prices in bay area

The following graphs show that Piedmont has the top elementary school rating, but the middle and high school ratings aren't quite as outstanding. In elementary, middle, and high school, Orinda has the greatest ratings. San Lorenzo's school ratings are below average when compared to other cities.

o) Variation of property price with location

To infer the variation of property price with location price and city parameters were used

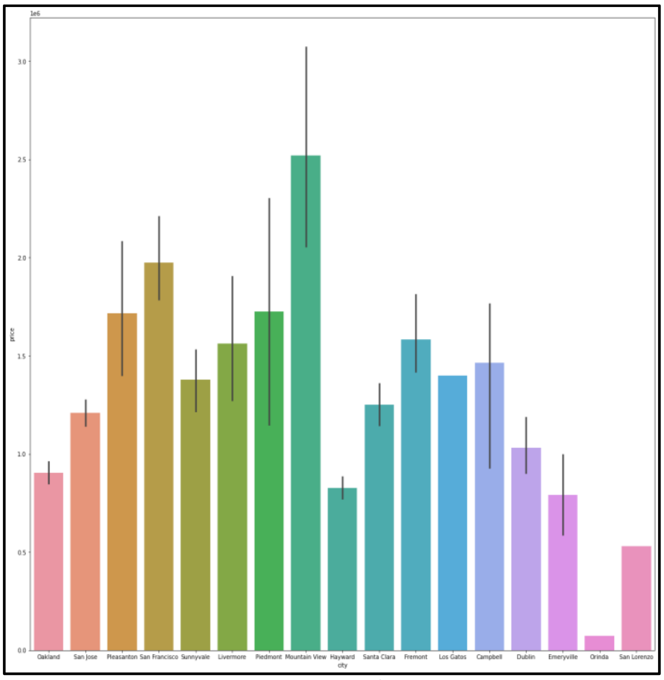


Fig 30: Bar plot showing the variation of property price along with the location

We can see from the graph above that the pricing is lower, despite the fact that Orinda has the greatest school ratings. Mountain View is the most expensive, followed by San Francisco, Piedmont, and Pleasanton.

V. METHODOLOGY

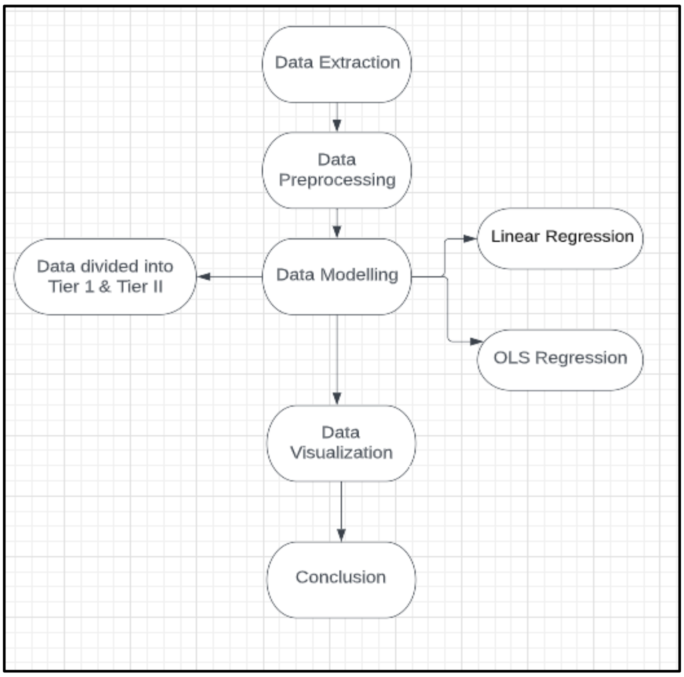


Fig 31: Shows the flow of the report

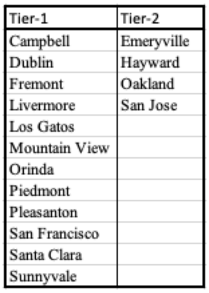
p) Strategy Building for CURBIO inc

In the above data pre-processing steps, we used feature selection to select the features which we are feeding the model with, and we can see how the R-square values are in both linear and OLS regression. But our main goal is to develop strategies for Curbio for profit maximization.

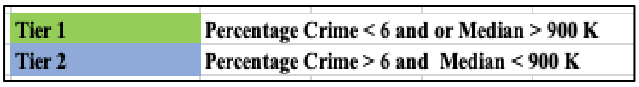
To develop these strategies, we are dividing the dataset into two tiers Tier 1 and Tier2.

q) Tier-1 and Tier-2 Classification

We classified the dataset into two tiers, Tier-1 and Tier-2 based on the percentage of crime and the median value of owner-occupied housing units from the FBI crime reports excel found in the google. The cities whose crime percentage is greater than six and or whose median value of owner-occupied housing units is greater than 900k are considered as Tier-1 cities, and the cities whose crime percentage is less than six and median value of owner-occupied housing units is greater than 900k are considered as Tier-2 cities. The below screenshots show the same.



Tier-1	Tier-2
Campbell	Emeryville
Dublin	Hayward
Fremont	Oakland
Livermore	San Jose
Los Gatos	
Mountain View	
Orinda	
Piedmont	
Pleasanton	
San Francisco	
Santa Clara	
Sunnyvale	



Tier 1	**Percentage Crime < 6 and or Median > 900 K**
Tier 2	**Percentage Crime > 6 and Median < 900 K**

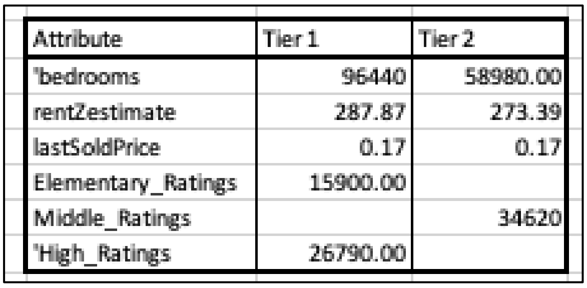
Table 10: Tier 1 and tier 2 cities

After feeding the above Tier-1 and Tier-2 data respectively to Linear and OLS regression models where features are selected manually, the R-square values are as shown in the following figures, the features selected for these machine learning models are as follows:

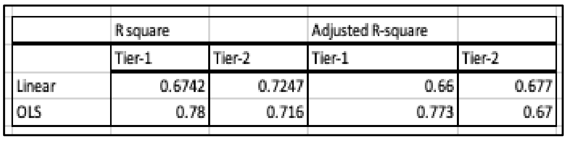
Numerical attributes = ['bedrooms', 'rentZestimate', 'lastSoldPrice','resoFacts_hasFireplace', Elementary_Ratings','Elementary_Distance', 'Middle_Ratings', 'High_Ratings', 'High_Distance']

Categorical attributes = ['city', 'county']

The above features are finalized after removing multicollinearity and trying many different combinations to generate a meaningful outcome from the datasets.



Attribute	Tier 1	Tier 2
'bedrooms	96440	58980.00
rentZestimate	287.87	273.39
lastSoldPrice	0.17	0.17
Elementary_Ratings	15900.00	
Middle_Ratings		34620
'High_Ratings	26790.00	



	R square		Adjusted R-square	
	Tier-1	Tier-2	Tier-1	Tier-2
Linear	0.6742	0.7247	0.66	0.677
OLS	0.78	0.716	0.773	0.67

Table 11: Tier 1 and tier 2 cities

VI.CONCLUSION

- Based on the above tier 1 and tier 2 table it is pretty evident that tier 1 cities are more bang for the buck for the company. On average, if we compare the two sister cities of San Jose and Santa Clara, on average tier 1 city had an increase of 97k in property value compared to 56k in tier 2 cities. The other variables which are significant via probability also infer the same. Having said that, since the cost of materials and labor is constant across the cities in the bay area the company gains more profit on the properties in tier 1 compared to tier 2. For a robust and sustainable business model, Curbio inc needs to launch multiple plans with equal weightage specific to a certain audience.

VII.SUGGESTIONS

- Some suggestions like a "6-8% share on increased price after the home sells + cost of renovation (gold Tier).
- We can limit the risk by rolling out plan A (gold tier) to only Tier 1 cities. Faster time to market, quick turnaround time, less risk of home not selling because no brokerage and listing is required.
- Renovate and sell the property for the client including brokerage (Platinum Tier).
- Since the profit is not much in Tier 2 cities compared to Tier 1 in similar renovation conditions, we should bundle brokerage services to maximize profit based on the high significance conditions from the above table.
- For tier 2 cities curbio inc should promote/limit its product to only the Platinum tier to gauge similar profits from the same cities.
- Statistically, since we can predict the increase in price based on 3rd party data, it would be good to use public data and build its own models rather than doing it from scratch.
- More investments can go into building in house validation models so as to adjust to market conditions based on certain factors like inflation, interest rate, socio-economic, geopolitical and pandemic conditions.

VIII. FUTURE ENHANCEMENTS AND LIMITATIONS

A data analyst's job is never completely done, there will always be room for further improvements. There are a couple of enhancements that we can do in the near future, they are as follows*:*

- **Dataset Size, source, and variety of data:** Zillow only shows first 20 pages of the data per URL, and it frequently throws captcha error, to avoid caught Ing up in the captcha we had to throw a wait time in the code, also Zillow keeps on changing their websites, so we can’t use the same code we used the before day, we had to keep on updating the code, which made the web scraping part time-consuming.
- **Limited Data**: The data we collected for this project is only 1561 rows, In the future, if we can collect large amounts of data, then we can discover many trends in the real estate market. Even though Zillow is the only source we used to collect the data for this project, we can collect the data from multiple other sources like costar, fly homes, redfin, and many other sources available in the market.
- **Collecting a variety of data** includes data on different property types like single-family, multi-family, condo, townhouses, apartments, duals or so. This can lead to uncovering many more inferences from the real estate market. In this project we collected data from only 16 cities in the bay area, we can widen the scope and collect the data from all the cities in the bay area.
- **Domain knowledge:** Even though all of us have a basic knowledge of the real estate market, we don’t know the real estate market in depth. In this project we used Pearson correlation coefficient for feature selection, but a proper domain knowledge would help us to make this process easier and more meaningful.

IX. REFERENCES

[1] **https://www.xlstat.com/en/solutions/features/ordinary-least-squares-regression-ols**

[2] **https://www.zyte.com/learn/what-is-web-scraping/**

[3] **https://www.forbes.com/sites/forbesagencycouncil/2019/10/01/the-age-of-analytics-and-the-importance-of-data-quality/?sh=2105971a5c3c**

[4] **https://www.jotform.com/data-collection-methods/**

[5] **https://www.promptcloud.com/blog/web-scraping-advantages-and-dis-advantages/**

[6] **https://www.dataaxlegenie.com/blog/this-is-why-duplicate-data-is-bad-for-you/**

[7] **https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/**

[8] **https://www.simplilearn.com/tutorials/machine-learning-tutorial/feature-selection-in-machine-learning**

[9] **https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/**

[10] **https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/**

[11] **https://www.investopedia.com/ask/answers/032515/what-does-it-mean-if-correlation-coefficient-positive-negative-or-zero.asp**

[12] **https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/**

[13] **https://www.voxco.com/blog/categorical-data-vs-numerical-data/#:~:text=Definition-,Categorical%20data%20refers%20to%20a%20data%20type%20that%20can%20be,any%20language%20or%20descriptive%20form.&text=Also%20known%20as%20qualitative%20data%20as%20it%20qualifies%20data%20before%20classifying%20it.**

[14] **https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10**

[15] **https://statisticsbyjim.com/regression/standardize-variables-regression**