CAPSTONE PROJECT: REAL ESTATE MACHINE LEARNING MODEL & APPLICATION

Western Governor University

Louis Wong

Table of Content

Letter of Transmittal

Summary of the Problem:

Since the recent surge in real estate prices and inflation, California real estate is becoming a national hot topic. Houses can be in the range of $500,000 to $50,000,000. But what actually determines this housing price tag? To find out what is going on, the answer can be revealed using machine learning. In creating a machine learning model, the information can be quantified, and find out what is the exact cost base on a few basic parameters.

Recommendation of the Solution (Data Product And Type):

This project will be relying on publicly available data sources only. Using what's publicly available, it is suggested to specify the model and analysis using the Santa Brabraba housing data set from the County of Santa Brabraba. The interpretation aspects of this project are important in this project. A simple machine learning model will be created that suits the need for interpretation and the capability to predict. The machine learning algorithm recommends to use is the multi-linear regression model. It is a supervised learning model. Most importantly, it suggests analyzing the different parameters at the end. Therefore, the question of which aspect truly affects the cost and at what quantified value can be understand.

Description of How the Proposed Solution Benefits the Client:

The interpretation of this model can act as a general guideline of what affects the price of real estate. Besides the understanding, the model itself can use as a prediction when unknown price data is given. Understanding the underlying prices of real estate can gain a huge advantage in spotting different, locate opportunities, and gives a competitive advantage in the market.

Objectives of the Project:

- The objective of this project is to create a machine learning model that fit the Santa Brabraba housing data set from the County of Santa Brabraba.
- Interpreting the model once it's trained.
- Create a simple application interface using this model.

Total Funding Requirement:

Estimated number of hours for the following:

Planning and Design: 8hrs

Development 24hrs

Documentation: 24hrs

Total: 56 hrs x $35 (Data Scientist avg. hourly wage)

The total cost price for this project is **$1,960** excluding hardware resources.

Letter of Transmittal

The Expertise of Developer Relevant to the Solution:

> The expertise required for this project is a data scientist or machine learning professional, who is proficient in python, Juypter lab, machine learning, and Sklearn.

Best Regard, Louis Wong.

lwong33@wgu.edu

Project Proposal

Problem Summary

Since the recent surge in real estate prices and inflation, California real estate is becoming a national hot topic. Houses can be in the range of $500,000 to $50,000,000. But what actually determines this housing price tag? To find out what is going on, the answer can be revealed using machine learning. In creating a machine learning model, the information can be quantified, and find out what is the exact cost base on a few basic parameters.

Application Benefits

The interpretation of this model can act as a general understanding of what affects the price of real estate. Besides the understanding, the model can use as a prediction when unknown price data is given. Understanding underlying prices of real estate can gain a huge advantage in spotting different, locate opportunities, and gives a competitive advantage in the market. The interpretation aspects of this project are important in this project. A simple machine learning model will be created that suits the need for interpretation and the capability to predict. Therefore, we can understand which aspect truly affects the cost and at what quantified value.

Application Description

The machine learning algorithm suggest to use is the multi-linear regression model, which is a supervised learning model. Most importantly, we are going to analyze the different parameters at the end and create a simple application interface using this model.

Data Description

This project will be relying on publicly available data sources only. Using what's publicly available, it is suggested to specify the model and analysis using the Santa Brabraba housing data set from the County of Santa Brabraba. The data may require some cleaning process.

Objective and Hypotheses

The objectives of this project are a functional machine learning model that can approximate Santa Barbara housing data, including a user interface application in the Jupyter lab, data visualization, and insightful interpretation results.

location seems to always contribute important factors to housing prices. A reasonable hypothesis can be drawn is that location is the number one most important factor that influences real estate prices.

Methodology

The Waterfall methodology will be used in this project since this project is a very straightforward machine learning project without much complication at the project level.

- **Requirements**

    - The requirement is to create multi parameters prediction model for the Santa Barbara real estate data set.
- **Design**
    - This project will use multi-linear regression as the foundation architecture of this model.
- **Implementation**
    - This project will machine learning model in Python using Sklearn.
- **Verification**
    - The prediction result will compare back to the test data set.
- **Maintenance**
    - Maintenance is not part of the scope of this project.

Funding Requirements

Estimated number of hours for the following:
Planning and Design: 8hrs
Development 24hrs
Documentation: 24hrs
Total: 56 hrs x $35 (Data Scientist avg. hourly wage)
The total cost price for this project is **$1,960** excluding hardware resources.

Stakeholders Impact

This project is for real estate analysis, real estate agencies, brokers, or anyone who has the interest to dissect real estate prices and understand the price of Santa Barbara County housing. This project gives the stakeholder, the ability to accurately approximate the underlying prices of real estate can gain a huge advantage in spotting different, opportunities, and gives a competitive advantage in the market.

Data Precautions

This housing data is publicly available on the government Santa Barbara County website. Although this data should ethically avoid being used as a direct mail campaign. However, it is not illegal.

Developer's Expertise

The expertise required for this project is a data scientist or machine learning professional, who is proficient in python, Juypter lab, machine learning, and Sklearn.

Executive Summary

Problem Statement

Since the recent surge in real estate prices and inflation, California real estate is becoming a national hot topic. Houses can be in the range of $500,000 to $50,000,000. But what actually determines this housing price tag? To find out what is going on, the answer can be revealed using machine learning. In creating a machine learning model, the information can be quantified, and find out what is the exact cost base on a few basic parameters, such as square footage, location, number of bedrooms, bathrooms, lot size, built year, etc.

Customer Summary

This application is for real estate analysis, real estate agencies, brokers, or anyone who has the interest to dissect real estate prices and understand the price of Santa Barba County housing. The user may need some basic knowledge of the Python environment to set up. However, little to no coding experience can run the analysis once it is properly set up.

Existing System Analysis

This project will require Windows 10/11 or other OS with Python and Jupyter Lab installed. Currently, most of the public does not have access to robust machine learning algorithms to understand real estate prices. Most real estate firms and brokers today are still relying on the “experts” and experience to give an estimation of real estate pricing instead of using a multi-parameter machine learning algorithm for accurate price approximation.

Data

Data is collected from Santa Baraba County, data will be downloaded as .xls file. Data will have to be clean and standardized before inputting into the machine learning model.

Project Methodology

This project will use the waterfall model.

- **Requirements**
  - The requirement is to create multi parameters prediction model for the Santa Barbara real estate data set.
- **Design**
  - This project will use multi-linear regression as the foundation architecture of this model.
- **Implementation**
  - This project will machine learning model in Python using Sklearn.

Executive Summary

- **Verification**
    - The prediction result will compare back to the test data set.
- **Maintenance**
    - Maintenance is not part of the scope of this project.

Project Outcomes

The deliverables of this project are a functional machine learning model that can accurately approximate Santa Barbara housing data, including a user interface application in the Jupyter lab, data visualization, and insightful interpretation results.

Implementation Plan

1. **Data collection,** load to Pandas data frame**.**
2. **Data Cleaning** will check if data have a null value.
3. **Data exploring** will create visualization and understanding of the data further.
4. **Data engineering** phase will standardize the data value and split the train and test the data set.
5. **Modeling** will finalize the fitting.
6. **Model Evaluation** will check the metric such as the R square value.
7. **Application integration** will create an interface for the users.

Evaluation Plan

- The project will be able to run without code-breaking.
- Prediction models are a good representation of the data by a minimum R square value of at least 0.5.
- The result provides accurate and insightful information about the data.
- The interface application is easy to use without advanced knowledge.

Resources and Costs

Programming Environment

- Hardware: PC/Mac
- OS Software: Window 10/11, Mac OS, Linus
- Other Software: Jupyter Lab and Python Environment

Environment Costs

Window 11 retail license costs anywhere between $139 and $309, Python and Jupyter Lab is open source (Free).

Executive Summary

Human Resource Requirements

Total: 56 hrs x $35 (Data Scientist avg. hourly wage)
The human resource cost for this project is $1,960.

Timeline and Milestones

Milestone	Time Duration	Start	End
Planning and Design	8hrs	3/28	3/29
Development	24hrs	3/29	3/31
Documentation	24hrs	3/31	3/4

Total: 56 hrs x $35 (Data Scientist avg. hourly wage)
The total cost price for this project is **$1,960** excluding hardware resources.

Project Summary

Project Purpose

California real estate is becoming the nation's hot topic, since the recent surge in prices and inflation. In Santa Baraba County, houses can be in the range of \$500,000 to $50,000,000, but what actually determines this housing price tag? Using machine learning we are going to find out what is going to quantify this information and find out what is the exact cost base on a few basic parameters, such as square footage, location, number of bedrooms, bathrooms, lot size, built year, etc. This project is a multi-linear regression machine learning model that can predict Santa Barbara housing data, including a user interface application in the Jupyter lab, data visualization, and insightful interpretation result.

Datasets

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
files = 'Santa Barabara.xls'
df = pd.read_excel(files)
dfselect = df[['Bedrooms','BathRooms', 'SqFootage', 'YearBuilt',
'YearEff','LotSize','PropUse','SitusCity','SalePrice']]
cleandf = dfselect.dropna()
cleandf['SalePrice'] = cleandf['SalePrice'].replace("[$,]", "",
regex=True).astype(int)
cleandf['SalePrice K'] = cleandf['SalePrice']/1000
cleandf = cleandf[cleandf['SalePrice K']<40000]
cleandf = cleandf.drop(columns=['SalePrice'])
cleandf = cleandf[cleandf['PropUse']!='Vacant Residential Land']
cleandf
```

```
WARNING *** OLE2 inconsistency: SSCS size is 0 but SSAT size is non-
zero
```

	Bedrooms	BathRooms	SqFootage	YearBuilt	YearEff	LotSize
0	3.0	2.00	1232.0	1962.0	1995.0	6098.40
1	4.0	2.25	1944.0	1961.0	1962.0	11761.20
2	4.0	2.00	1408.0	1961.0	1975.0	7405.20
3	3.0	3.00	2338.0	1963.0	2005.0	10890.00
5	4.0	1.75	1300.0	1963.0	1964.0	8712.00
...	...	...	...	...	...	...

```
2279       3.0       4.00     5241.0     2001.0   2001.0  1093791.60

2280       3.0       3.00     3904.0     1973.0   1990.0    63162.00

2281       4.0       4.00     3299.0     1996.0   1996.0    41251.32

2282       1.0       1.00      860.0     1965.0   1966.0    48787.20

2284       4.0       3.50     4094.0     1989.0   1990.0    40510.80


                      PropUse      SitusCity  SalePrice K
0     Single Family Residence    CARPINTERIA       1615.0
1     Single Family Residence    CARPINTERIA       1500.0
2     Single Family Residence    CARPINTERIA       1350.0
3     Single Family Residence    CARPINTERIA       1230.0
5     Single Family Residence    CARPINTERIA       1382.5
...                       ...            ...          ...
2279  Single Family Residence  SANTA BARBARA       6785.0
2280  Single Family Residence  SANTA BARBARA       4975.0
2281  Single Family Residence  SANTA BARBARA       4100.0
2282  Single Family Residence  SANTA BARBARA       2650.0
2284  Single Family Residence  SANTA BARBARA       5250.0

[1886 rows x 9 columns]
```

The data set was collected from Santa Baraba County, data set was downloaded as .xls file. The data has many null values and missing data. It was removed during the data cleaning process. There were also a few outliers above the 40 million dollars range, which was also removed since it can skew the majority of the population. The “SalesPrice” column was converted from string to integer. The 7 digit millions were too long to work with, it was reduced to K, Ex: 1M -> 1000K. After the cleaning, there were 1886 data left for this project.

Data Product Code

```
import sklearn
import math
import numpy
import matplotlib.pyplot as plt
import seaborn as sns

onehotdf = pd.get_dummies(cleandf, columns=['PropUse', 'SitusCity'],
drop_first=True)
x_data = onehotdf.drop(columns=['SalePrice K'])
y_data = onehotdf['SalePrice K']
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler().fit(x_data)
scaler.transform(x_data)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_data,y_data,
```

```
random_state=30)
from sklearn.linear_model import LinearRegression
LR = LinearRegression()
LR.fit(x_train,y_train)
```

```
LinearRegression()
```

This code provide data cleaning, feature engineerging, split train/test data set and fit the multi-linear regression model which is the core algorithinm of this project. The data was scaled and standardized before inputting into the machine learning model, it provided the necessary functionality of predictive outputs and weight the data equally for analysis purpose. The train data set is use as inputs of this model, the test set is hidden from the training, which were used as a verification method after.

Hypothesis verification

```
x_cols = x_data.columns
coef_df =
pd.DataFrame({"Factors":x_cols,"Coefficients":abs(LR.coef_)})
coef_summary_df = coef_df[0:6]
proptype_mean = coef_df[6:8]['Coefficients'].mean()
location_mean = coef_df[8:]['Coefficients'].mean()
coef_summary_df =
coef_summary_df.append({'Factors':'PropType','Coefficients':proptype_m
ean}, ignore_index=True)
coef_summary_df =
coef_summary_df.append({'Factors':'Location','Coefficients':location_m
ean}, ignore_index=True)
coef_summary_df.sort_values(by = 'Coefficients', ascending =
False).reset_index(drop=True).head()
```

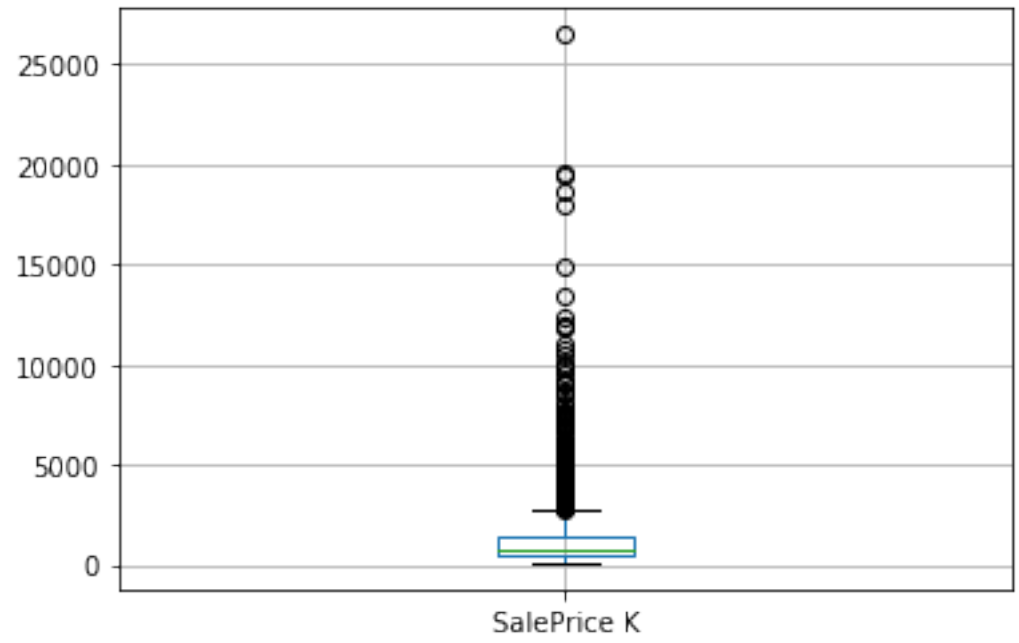
	Factors	Coefficients
0	Location	576.315367
1	Bedrooms	385.553787
2	PropType	142.494964
3	BathRooms	98.500932
4	YearBuilt	5.413614

location seems to always contribute important factors to housing prices. A reasonable initial hypothesis was that "location is the most important factor that influences real estate prices". From the finding, it was conclude that the top 5 most influential factors are location, numbers of bedrooms, residentual house or condo, number of bathrooms, and year built. Factors are in sequential order most to least. It was verfied that the original hypothesis was accpect that location is the main role of influences. Beside the hypothesis, from the summary pie chart below, location factor was conclude that it contribute close to half of the housing price.

Effective Visualizations and Reporting

```
cleandf.boxplot(column=['SalePrice K'])
```

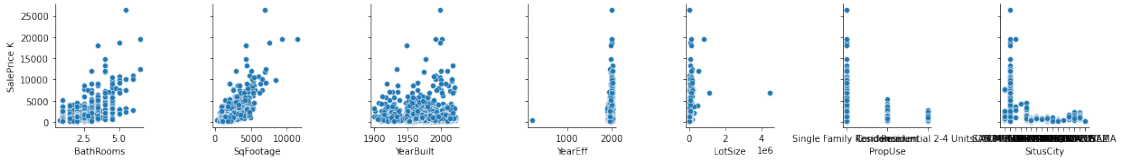
```
<AxesSubplot:>
```



Using the box plot, there were a few outliers that was spoted. Prices that were above 40 millions were outliers which was removed during cleaning process. Visualizationfrom help to analysis the data more effecitvely. This would not be as clear if the data was read one by one, there are close to 2000 rows in this dataset.

```
sns.pairplot(cleandf,x_vars=['BathRooms', 'SqFootage', 'YearBuilt',
'YearEff','LotSize','PropUse','SitusCity'], y_vars=['SalePrice K'],)
```

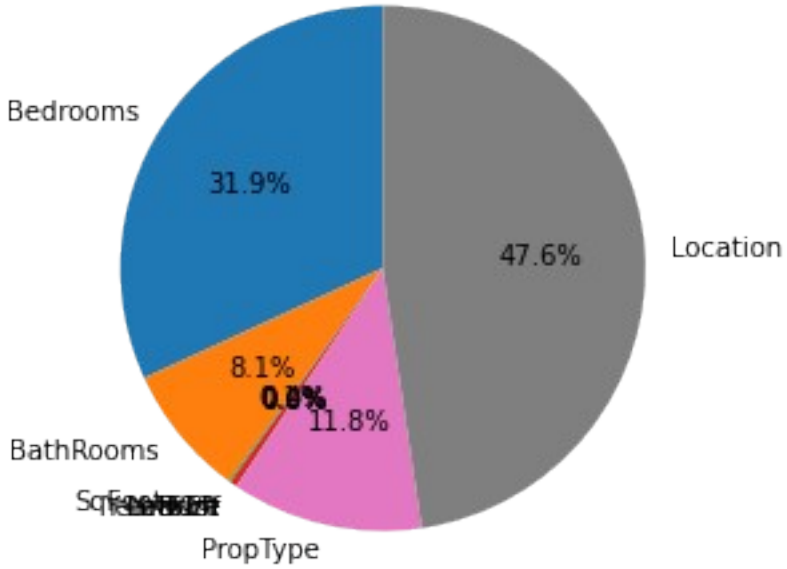
```
<seaborn.axisgrid.PairGrid at 0x1d2ec2408b0>
```



Using the pairplot, it visually indicate there was a trend in the data. All the parameters seems to affect the price some way, some is more, and some is less. Nonetheless, some kind of geometric pattern was observable. Which indicate the data was not random.

```
coef_df.sort_values(by = 'Coefficients', ascending =
False).reset_index(drop=True)
coef_summary_df = coef_df[0:6]
proptype_mean = coef_df[6:8]['Coefficients'].mean()
location_mean = coef_df[8:]['Coefficients'].mean()
coef_summary_df =
coef_summary_df.append({'Factors':'PropType','Coefficients':proptype_m
ean}, ignore_index=True)
coef_summary_df =
coef_summary_df.append({'Factors':'Location','Coefficients':location_m
ean}, ignore_index=True)
```

```
labels = coef_summary_df['Factors']
values = coef_summary_df['Coefficients']
fig1, ax1 = plt.subplots()
ax1.pie(values, labels=labels,autopct='%1.1f%%', startangle=90 )
ax1.axis('equal')
plt.show()
```

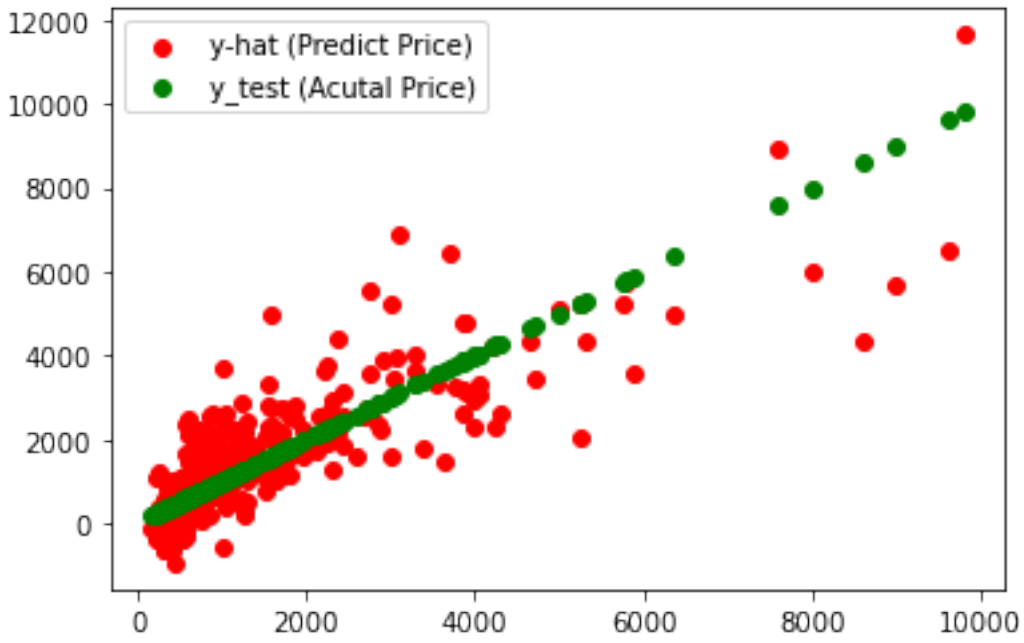


This is final summary pie chart after the model is complete. It is immediately obvious to spot the quantity relationship between different factors and the contribution to the price. Surprisingly, there were only 4 factors that influence 99% of the price, the rest affect less than 1%.

Accuracy analysis

```
import matplotlib.pyplot as plt
y_prediction = LR.predict(x_test)
y_comparedf = pd.DataFrame({'y-
hat':y_prediction.round(),'y_test':y_test})
plt.figure()
plt.scatter(x = y_comparedf['y_test'],y = y_comparedf['y-
hat'],label='y-hat (Predict Price)', c='red')
plt.scatter(x = y_comparedf['y_test'],y =
y_comparedf['y_test'],label='y_test (Acutal Price)', c='green')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x1d2ec5b0c70>
```



This chart plot against the prediction prices to the actual prices. It helps to understand how well the model does visually. The model is not perfect, there were residuals and deviation between prediction versus reailty. Nontheless, it does a good job capture the pattern, since the mean of the prediction seems to be align with the prices of the reailty. In this prespective, the model were successful.

```
LR.score(x_test, y_test)
```

```
0.6488986184301333
```

This number was the R-square score, it is 1 minus sum of residual square value. Generally speaking, any value above 50% indicate the model is effective. The closer to 1 to less error the model has. However, it is not too meaningful all by itself, this value will be most useful if compare to two or more different model with the same goal. The R-square score value can give an indication on which model perfrom the best.

Application Testing

For testing to confirm the functionality of the code, a white box approach of unit testing was done. It ensure each segment of the code block work as intended. Lastly, a blackbox approach of usability was done for the interface application. Ensure the application is smooth and usable.

```
# Application Interface

print('---------- APPLICATION INTERFACE ---------- ')
print('How many Bedrooms?(int)')
bedroom = int(input())
print('How many Bathrooms?(int or float)')
bathroom = float(input())
```

```
print('What is the Sqfootage?(int)')
sqft = int(input())
print('What is the lot size footage?(int)')
lotsize = int(input())
print('Enter 1 for Single Family Residence, 2 for Residential 2-4
Units:')
prop = int(input())
print('Enter the number of which city, 1:CARPINTERIA, 2:GOLETA,
3:GUADALUPE, 4:LOMPOC, 5:ALAMOS, 6:LOS OLIVOS, 7:MONTECITO, 8:NEW
CUYAMA, 9:ORCUTT, 10:SANTA BARBARA, 11:SANTA MARIA, 12:SANTA YNEZ,
13:SISQUOC, 14:SOLVANG, 15:SUMMERLAND')
city = int(input())
print('Enter Year Built:(int)')
yearbuilt = int(input())
print('Enter Effective Year:(int)')
yeareff = int(input())
print('---------- MODEL RESULTS ---------- ')

if city == 1:
    cityvalue = 'CARPINTERIA'
elif city == 2:
    cityvalue = 'GOLETA'
elif city == 3:
    cityvalue = 'GUADALUPE'
elif city == 4:
    cityvalue = 'LOMPOC'
elif city == 5:
    cityvalue = 'ALAMOS'
elif city == 6:
    cityvalue = 'LOS OLIVOS'
elif city == 7:
    cityvalue = 'MONTECITO'
elif city == 8:
    cityvalue = 'NEW CUYAMA'
elif city == 9:
    cityvalue = 'ORCUTT'
elif city == 10:
    cityvalue = 'SANTA BARBARA'
elif city == 11:
    cityvalue = 'SANTA MARIA'
elif city == 12:
    cityvalue = 'SANTA YNEZ'
elif city == 13:
    cityvalue = 'SISQUOC'
elif city == 14:
    cityvalue = 'SOLVANG'
elif city == 15:
    cityvalue = 'SUMMERLAND'

if prop == 1:
```

```
    propvalue = 'Single Family Residence'
elif prop == 2:
    propvalue = 'Residential 2-4 Units'

new_row = {'Bedrooms': bedroom, 'BathRooms': bathroom, 'SqFootage':
sqft, 'YearBuilt': yearbuilt, 'YearEff': yeareff,'LotSize':
lotsize,'PropUse': propvalue,'SitusCity': cityvalue,'SalePrice K': 0}
cleandf = cleandf.append(new_row, ignore_index=True)
onehotdf = pd.get_dummies(cleandf, columns=['PropUse', 'SitusCity'],
drop_first=True)
newx_data = onehotdf.drop(columns=['SalePrice K']).iloc[-1]
userxdata = scaler.transform([newx_data.to_numpy()])
yhatprice = LR.predict(userxdata)[0].round()
print('The Model Prediction Price is '+ "$
{:,.2f}".format(yhatprice*100))

---------- APPLICATION INTERFACE ----------
How many Bedrooms?(int)

 2

How many Bathrooms?(int or float)

 2

What is the Sqfootage?(int)

 2000

What is the lot size footage?(int)

 0

Enter 1 for Single Family Residence, 2 for Residential 2-4 Units:

 1

Enter the number of which city, 1:CARPINTERIA, 2:GOLETA, 3:GUADALUPE,
4:LOMPOC, 5:ALAMOS, 6:LOS OLIVOS, 7:MONTECITO, 8:NEW CUYAMA, 9:ORCUTT,
10:SANTA BARBARA, 11:SANTA MARIA, 12:SANTA YNEZ, 13:SISQUOC,
14:SOLVANG, 15:SUMMERLAND

 11

Enter Year Built:(int)

 1980

Enter Effective Year:(int)

 2030

---------- MODEL RESULTS ----------
The Model Prediction Price is $893,800.00
```

Application Files

Software Requirements:

- Any OS
- Python 3.9
- Jupyter Lab
- Python Libraries: pandas, sklearn, math, numpy, matplotlib, seaborn, xlrd
- Main Application Files: Real_Estate_ML/main_lab.ipynb running using Jupyter Lab, data is Real_Estate_ML/Santa Barabara.xls.
- Extra Content: Real_Estate_ML/Project_Summary.ipynb content same information in this current document.

User's Guide

Brief manual for the installation and use of the application and all steps necessary to establish an environment capable of running the application and producing the required results.

1. Install Python 3.9 https://www.python.org/downloads/ in your OS.
2. In CDM enter: pip install jupyterlab
3. Unzip and extract the "Real_estate_ML" file to a perfer location.
4. Open Jupyter Lab browser in CMD enter: jupyter lab
5. Open "main_lab.ipynb" in the Jupyter Lab browser.
6. Run the first cell to install all the Python libraries: pandas, sklearn, math, numpy, matplotlib, seaborn, xlrd
7. Run all cells.
8. Enter your housing information at the last cell.

If there is any trouble installing the jupyter lab, please check out https://jupyter.org/install. You may extract the "main_lab.ipynb" and "Santa Barabara.xls" to a new files, just make sure run the first cell and pip install all the libraries. Anaconda version of Jupyter Lab may provide a simpler install experience.

References and Sources

There are no citations or quotes used.