CS 171 Sec 4 Group Project

Tanya Gupta, Nhat Le, Tyler Nguyen

Table of Contents

1. Business Background (What has happened before, why need to develop this model...)

Licensed appraisers need to know the price of houses in San Jose, California, USA based on the trends of house rates. Appraisers usually look at the location, education, and quality of the house to determine the price, but the inflation and how much the house prices have been going up over the years is important too. San Jose's housing market has extremely high rates, and that is due to its location. It is important for the appraiser to consider not just location, but also trends in how the house market works. The price of the house matters for its buyers, real estate agents, and builders. First, the house must be affordable for a certain population of buyers. Real estate agents also get a portion of the price, which has to be accounted for, and builders also have to make profit after building the house. Overall, the cost of living overtime changes, and so do house prices. For this reason, the appraiser needs to value the house at the right price. Our model will help the appraiser value house prices in the future according to the current market.

2. Objective (Develop a model to answer what business question)

Our objective is to develop a Time-Series Model that allows us to see how mid-tier house prices in San Jose have changed over time. This allows appraisers to answer the question: "What will the average house price be in San Jose in the future? How will the house prices change?"

3. Data

a. Data Source

https://www.zillow.com/research/data/

From here, we chose Home Values, All-Homes Time-Series Raw, Metro & U.S.

Home values data table we used.

We also experimented with unemployment rates as the exogenous variable.

https://fred.stlouisfed.org/series/SANJ906URN

Unemployment data table we used.

We tried using interest rate data.

b. Features Included

The data source of home value includes Region Name (City) and the average house prices of each city each month, starting from January 1996, till October 2022. Since we only want to predict house prices for San Jose, we used all the data from Row 37, or “Size Rank” 35. For the unemployment data, we used the given features, since only the monthly date is given and unemployment rates by percentage.

c. Data Cleaning (how to handle missing values, is there an outlier,...)

We used Forward Fill, or pandas.DataFrame.fillna(method='ffill',axis=1, inplace=True) to handle missing values.

d. Feature Engineering (any transformation, binning to category data...)

We extracted just one city’s data, San Jose. We also only used the Region

Name column and the prices as we did not need Region ID and Region Type. This is a form of feature extraction. To make it a time series project, we also had to convert the dataframes into series. Lastly, the unemployment data had its dates at the end of the month instead of the beginning like the house value data. In order to fix that, we converted the data to DateTime type and back to series so that the dates were consistent.

We use the housing data in the San Jose as dependent variable and we need to choose an independent variable for SARIMAX model among interest rates, unemployment rates, average housing prices. These variables does not pass the cointegration test so we need to transform data using moving averages. We performed 3-month smoothing averages and 6-month smoothing averages on all 4 datasets, but the smoothing values still did not pass the test.

e. **Dimension Reduction(should we apply dimension reduction in this data or not?)**

We did not need to apply dimension reduction since our only main input was price. We did not use multiple features.

f. **Variable Selection(use LASSO or Stepwise or correlation, other variable selection?)**

Since we only used the housing price and unemployment rates, we did not have to use variable selections as there were too few features to be used.

g. **Regularization(does the model need to be regularized)**

We did not use regularization for our models because we only used one variable which was the housing prices in San Jose and it is not effective to use regularization in the time series model. It is also not necessary for us to normalize the data as that did not make a difference in our results.

h. Ensemble Learning

We tried to combine the results of the Holt Winters and ARIMA models, but were not too successful. However, we did compare both the models, and Holt Winters Triple Exponential Smoothing is the best method. We also used Time Series Split to validate data as well as Exogenous variables in SARIMAX to add variety.

4. Methodology Exploration (what are the candidate ML algorithms, not limited to algorithms taught in class, why do you choose this one, any ensemble technique applied...?)

Holt Winters Triple Exponential Smoothing is the algorithm we chose to use. We were considering either the ARIMA model or the Exponential Smoothing. We went with Exponential Smoothing because we have non-stationary data, and Time-Series is a good way to predict trends or seasonal trends in the data, as well as only focus on one aspect; in this case, price.*We decided to use 90% of the data for training, and 10% of the data for test*. Since the we have data from 323 months, which is a comparatively small dataset, we decided to use 90% for training. When we used 80% for training, our results were not as great. For example, our R^2 was about 0.5, compared to the ~0.8 we have now. Although we have non-stationary data, we did decide to involve SARIMA and SARIMAX in our code. We wanted to compare the two algorithms and see how much of a difference that would make, and if the different seasonalities in SARIMA would make a difference.

5. Required Assumption (e.g., the correlation between dependent and independent variable holds, or the data is still stationary in production, or the data remain same pattern in production,...)

The housing prices, average housing prices, unemployment rates, and interest rates are non-stationary, according to Augmented Dickey Fuller test. Because all of our data is non-stationary, we need to perform a cointegration test to see if either average housing prices, unemployment rates, or interest rates has long-term comovement with the housing prices. Both unemployment rates, the nation’s average house price, and interest rates do not pass the cointegration test, but we try to use unemployment rates or average US housing prices as an exogenous variable because they still have economic correlation to housing prices.

6. Model Equation (write out the model equation, if it is able to)

This is the math behind our Triple Holt Winters Exponential model. We used Professor’s notes from Week 12.

Triple Exponential Smoothing, Holt Winters Exponential Smoothing (Include Level, Trend and Seasonality smoothing component)

Holt proposed a method for seasonal data. His method was studied by Winters, and so now it is usually known as “Holt-Winters’ method”. Holt-Winters’ method is based on three smoothing equations—one for the level, one for trend, and one for seasonality.

K step forecast equation with **Additive** Trend and **Additive** Seasonality is :

$$\hat{Y}_{t+k|t} = L_t + kT_t + S_{t-m+1+(k-1)\,mod\,m}$$

Smoothing equation:

$$\begin{aligned} L_t &= \alpha(Y_t - S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1}) \\ T_t &= \beta((L_t - L_{t-1}) + (1-\beta)T_{t-1} \\ S_t &= \gamma(Y_t - L_{t-1} - T_{t-1}) + (1-\gamma)S_{t-m} \end{aligned}$$

$were\ \ S_t\ is\ seasonality\ at\ time\ t, m\ is\ length\ of\ seasonality, 0 < \alpha < 1\ , 0 < \beta < 1, 0 < \gamma < 1$

K step forecast equation with **Additive** Trend and **Multiplicative** Seasonality is :

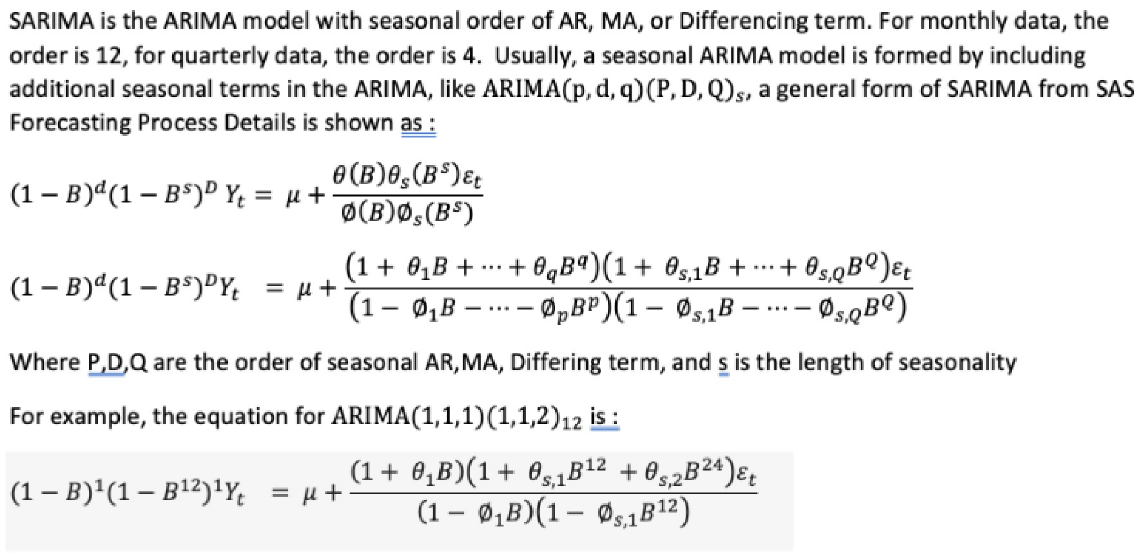
$$\hat{Y}_{t+k|t} = (L_t + kT_t)S_{t-m+1+(k-1)\,mod\,m}$$

Smoothing equation:

$$\begin{aligned} L_t &= \alpha(Y_t / S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1}) \\ T_t &= \beta((L_t - L_{t-1}) + (1-\beta)T_{t-1} \\ S_t &= \gamma[Y_t / (L_{t-1} + T_{t-1})] + (1-\gamma)S_{t-m} \end{aligned}$$

$were\ \ S_t\ is\ seasonality\ at\ time\ t, m\ is\ length\ of\ seasonality, 0 < \alpha < 1\ , 0 < \beta < 1, 0 < \gamma < 1$

The math behind SARIMA and SARIMAX. We used Professor’s notes from Week 12.



SARIMA is the ARIMA model with seasonal order of AR, MA, or Differencing term. For monthly data, the order is 12, for quarterly data, the order is 4. Usually, a seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA, like $\mathrm{ARIMA(p, d, q)(P, D, Q)}_s$, a general form of SARIMA from SAS Forecasting Process Details is shown as :

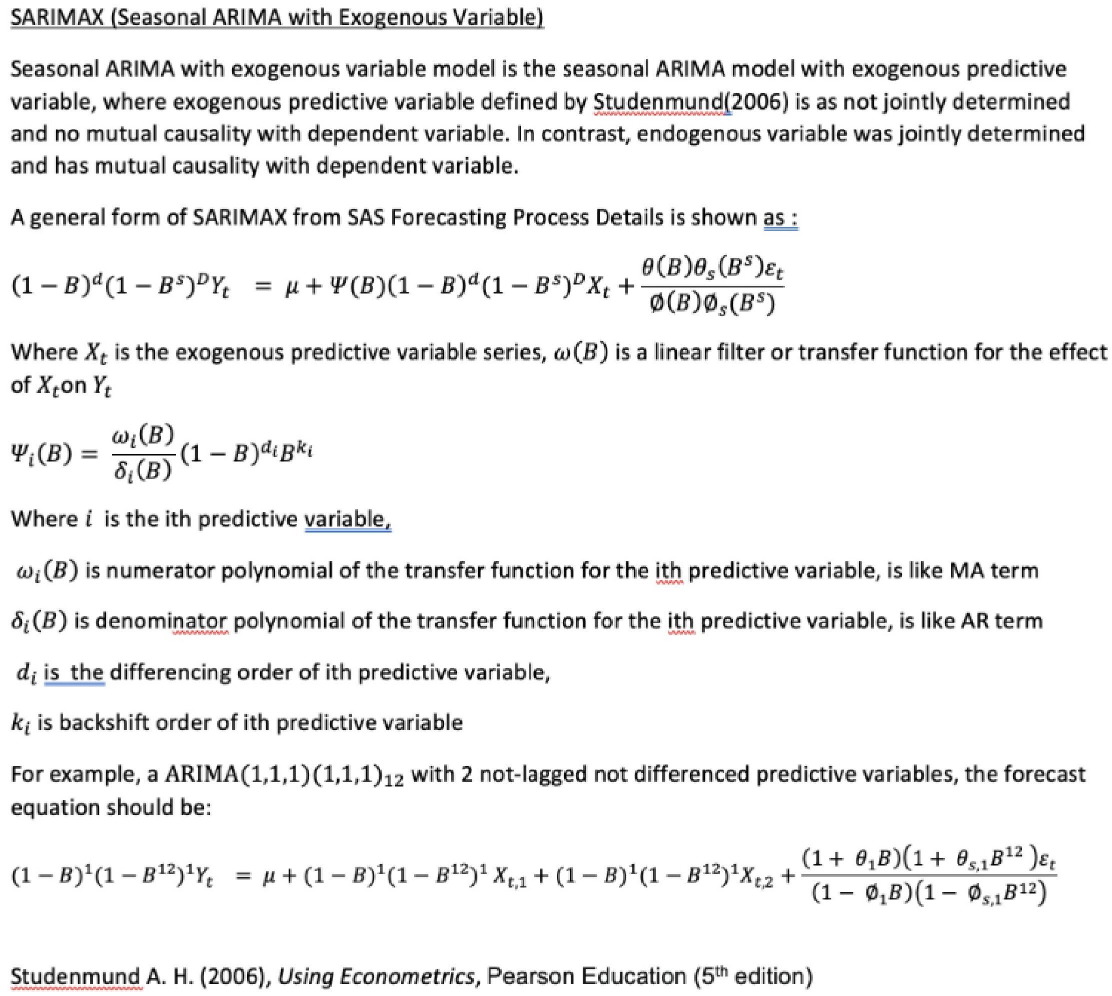
$$(1-B)^d(1-B^s)^D\,Y_t = \mu + \frac{\theta(B)\theta_s(B^s)\varepsilon_t}{\emptyset(B)\emptyset_s(B^s)}$$

$$(1-B)^d(1-B^s)^D Y_t \quad = \mu + \frac{(1+\theta_1 B + \cdots + \theta_q B^q)(1+\theta_{s,1}B + \cdots + \theta_{s,Q}B^Q)\varepsilon_t}{(1-\emptyset_1 B - \cdots - \emptyset_p B^p)(1-\emptyset_{s,1}B - \cdots - \emptyset_{s,Q}B^Q)}$$

Where P,D,Q are the order of seasonal AR,MA, Differing term, and s is the length of seasonality

For example, the equation for $\mathrm{ARIMA(1,1,1)(1,1,2)}_{12}$ is :

$$(1-B)^1(1-B^{12})^1 Y_t \quad = \mu + \frac{(1+\theta_1 B)(1+\theta_{s,1}B^{12} + \theta_{s,2}B^{24})\varepsilon_t}{(1-\emptyset_1 B)(1-\emptyset_{s,1}B^{12})}$$



SARIMAX (Seasonal ARIMA with Exogenous Variable)

Seasonal ARIMA with exogenous variable model is the seasonal ARIMA model with exogenous predictive variable, where exogenous predictive variable defined by Studenmund(2006) is as not jointly determined and no mutual causality with dependent variable. In contrast, endogenous variable was jointly determined and has mutual causality with dependent variable.

A general form of SARIMAX from SAS Forecasting Process Details is shown as :

$$(1-B)^d(1-B^s)^D Y_t \quad = \mu + \Psi(B)(1-B)^d(1-B^s)^D X_t + \frac{\theta(B)\theta_s(B^s)\varepsilon_t}{\emptyset(B)\emptyset_s(B^s)}$$

Where X_t is the exogenous predictive variable series, $\omega(B)$ is a linear filter or transfer function for the effect of $X_t$on Y_t

$$\Psi_i(B) = \frac{\omega_i(B)}{\delta_i(B)}(1-B)^{d_i}B^{k_i}$$

Where i is the ith predictive variable,

$\omega_i(B)$ is numerator polynomial of the transfer function for the ith predictive variable, is like MA term

$\delta_i(B)$ is denominator polynomial of the transfer function for the ith predictive variable, is like AR term

d_i is the differencing order of ith predictive variable,

k_i is backshift order of ith predictive variable

For example, a ARIMA$(1,1,1)(1,1,1)_{12}$ with 2 not-lagged not differenced predictive variables, the forecast equation should be:

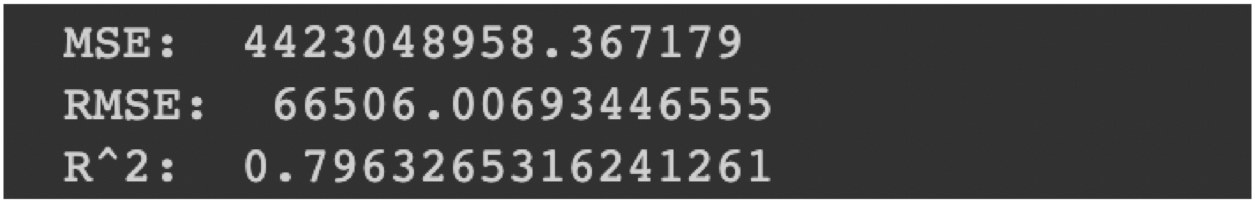
$$(1-B)^1(1-B^{12})^1 Y_t \quad = \mu + (1-B)^1(1-B^{12})^1 X_{t,1} + (1-B)^1(1-B^{12})^1 X_{t,2} + \frac{(1+\theta_1 B)(1+\theta_{s,1}B^{12})\varepsilon_t}{(1-\emptyset_1 B)(1-\emptyset_{s,1}B^{12})}$$

Studenmund A. H. (2006), *Using Econometrics*, Pearson Education (5th edition)

7. In-Sample Validation and Performance Metric(what performance metric do you use,how does the model perform in train and test (in-sample validation) data ?)

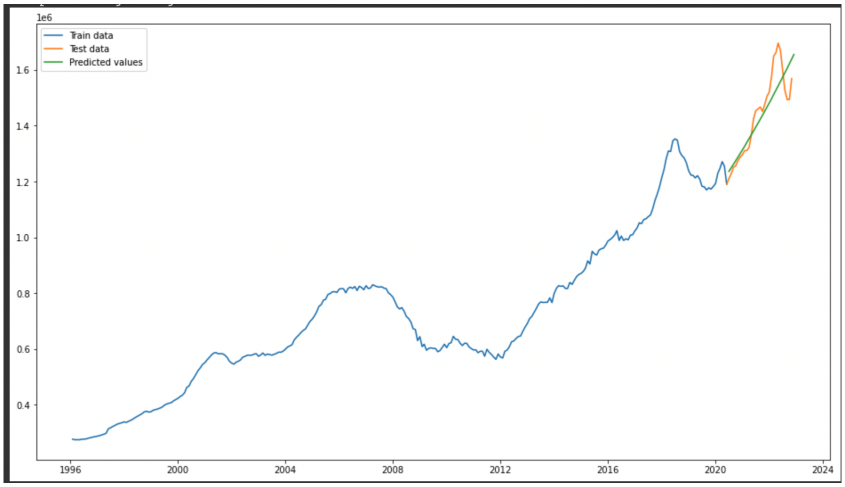
For the Exponential Smoothing model, we pick the set of parameters which yields the lowest mean-squared error and highest R-squared score.

Below is the metrics of our most optimal Exponential Smoothing model:

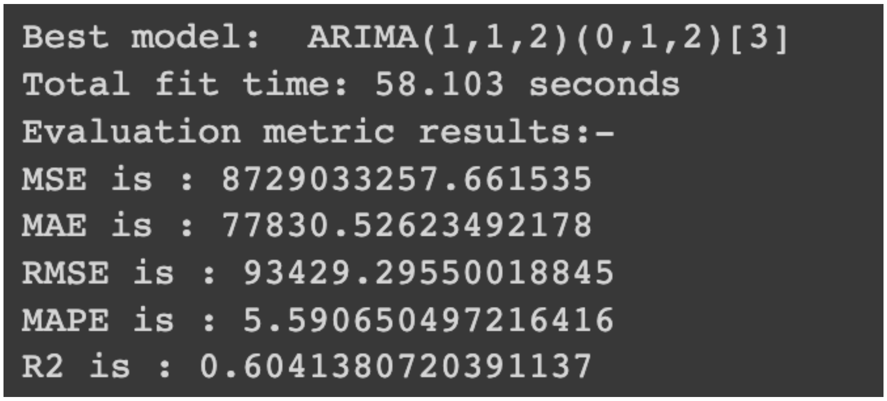


```
MSE:   4423048958.367179
RMSE:   66506.00693446555
R^2:   0.7963265316241261
```

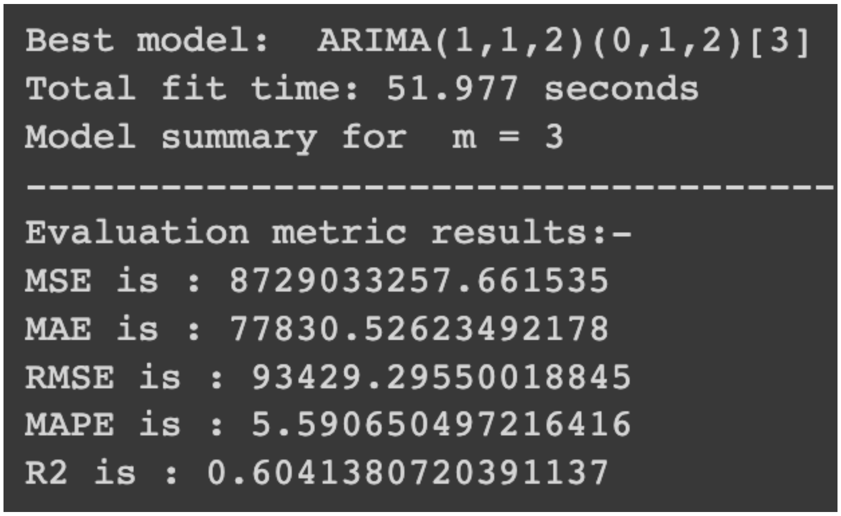
Our chart below shows that our model has a good performance as it is able to forecast the trend:



In terms of the SARIMA and SARIMAX model, we used the auto_arima function with the seasonality of 1, 3, 6, 12, 24 to help us find the best parameters. We use mean-squared error, R-squared score, root mean-squared error, mean absolute percentage error, and mean absolute error to validate the performance of the model. Below are the metrics for SARIMA and SARIMAX, respectively.



```
Best model:  ARIMA(1,1,2)(0,1,2)[3]
Total fit time: 58.103 seconds
Evaluation metric results:-
MSE is : 8729033257.661535
MAE is : 77830.52623492178
RMSE is : 93429.29550018845
MAPE is : 5.590650497216416
R2 is : 0.6041380720391137
```



```
Best model:  ARIMA(1,1,2)(0,1,2)[3]
Total fit time: 51.977 seconds
Model summary for  m = 3
-------------------------------------
Evaluation metric results:-
MSE is : 8729033257.661535
MAE is : 77830.52623492178
RMSE is : 93429.29550018845
MAPE is : 5.590650497216416
R2 is : 0.6041380720391137
```

Cross Validation:

Time Series' version of cross validation would be to use the method sklearn.model_selection.TimeSeriesSplit. In our code, we print out the test sets vs train sets, so we can see the divisions. We can split time series data samples that are observed at fixed time intervals.

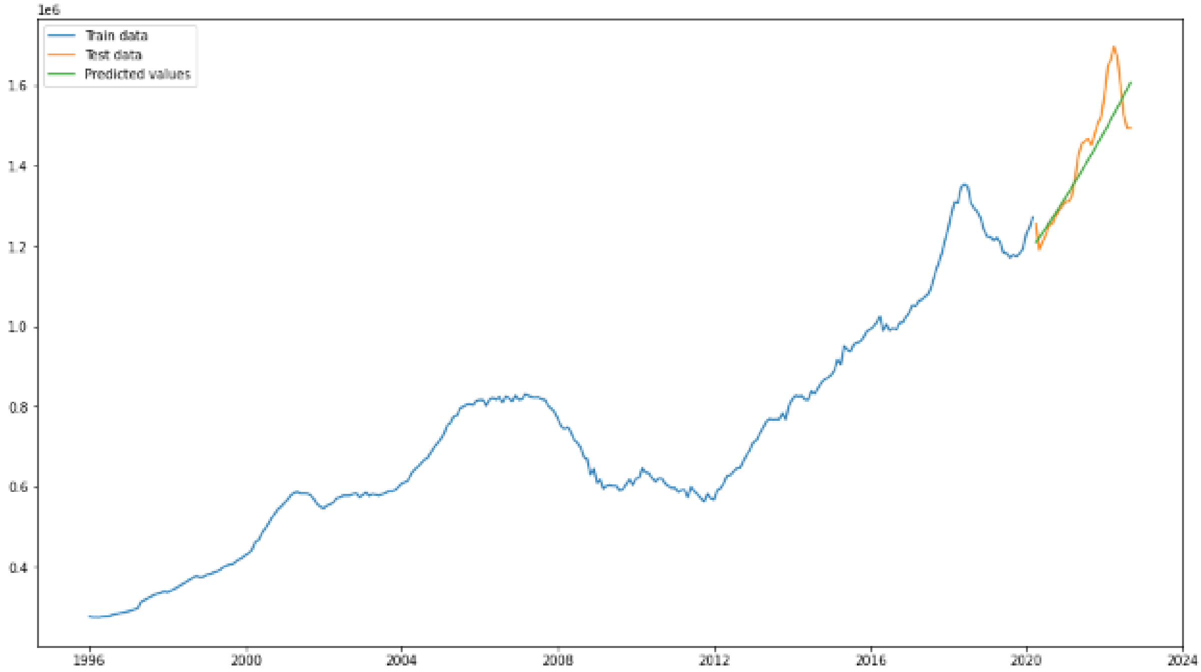
8. Monitoring and Maintenance Plan (after launch this model, how will you monitor and maintain this model, this section is highly related to section 5, e.g., check the data/feature stability across time)

One way we can monitor the model is to check on the variance and see if there are any extreme changes like during market crashes. We can prioritize more recent changes in the data during more volatile periods. We can use an index like the FHFA Housing Price Index (HPI) which can help the model to more accurately predict future prices based on more shorter term changes. Lastly, given more time, we can try to use regression and more variables to help us get more accurate results.

9. Conclusion

Holt Winter

The performance of our Holt Winter's model based on the MSE and R-squared values is fairly accurate. There is only an average of $70339.29 in error in our model based on the RMSE and the R-squared is almost .8 meaning around 80% of the variance in the data can be explained by the model.

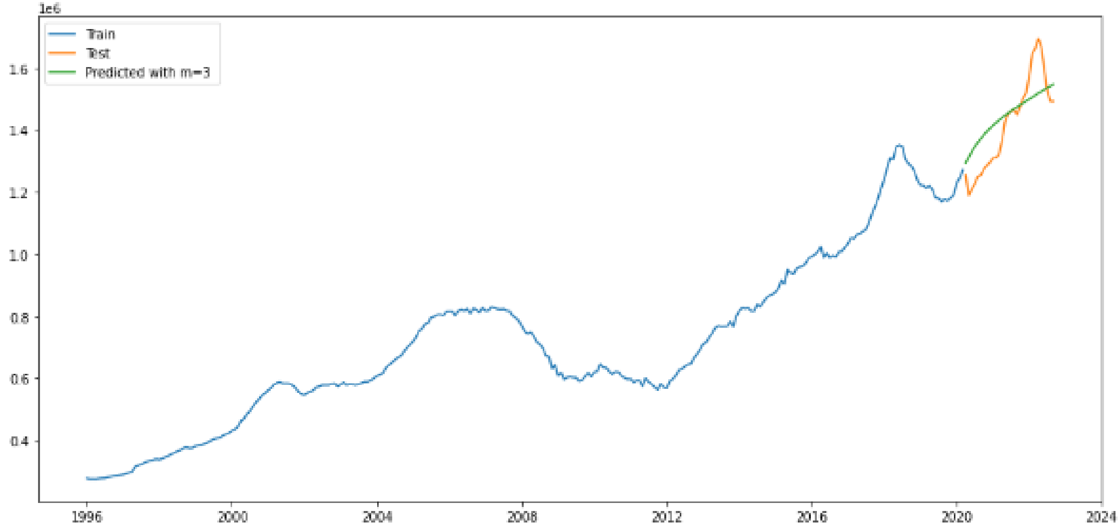


SARIMA

The performance of the SARIMA model did not perform as well as Holt's Winter.

Seasonality of length 3 is the best out of lengths 1, 3, 6, 12, and 24. The r^2 is 0.6, which is not bad.

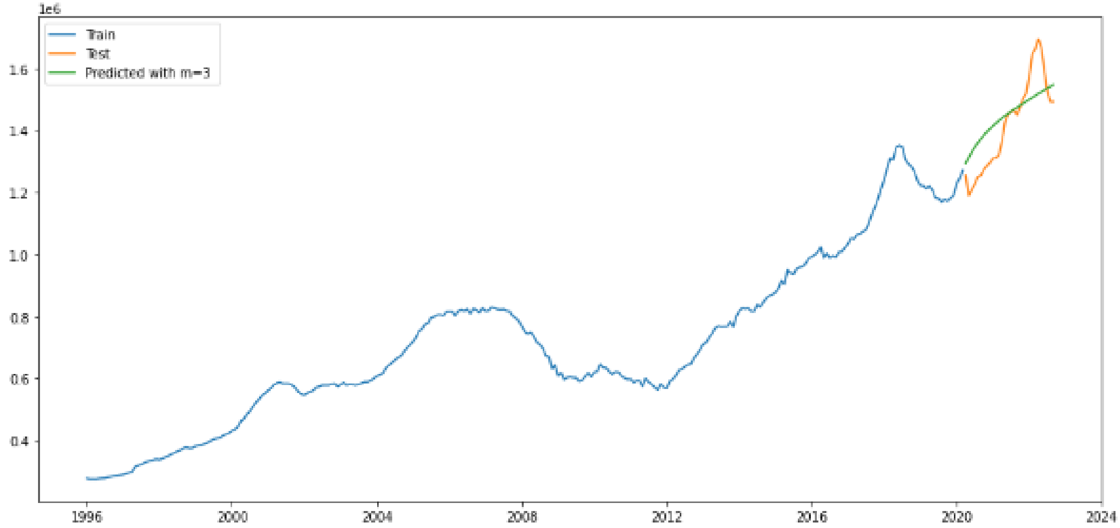
There is an average of $93429.29 in error in our model based on the RMSE, which is also not too bad considering how high the prices are to start with.



SARIMAX

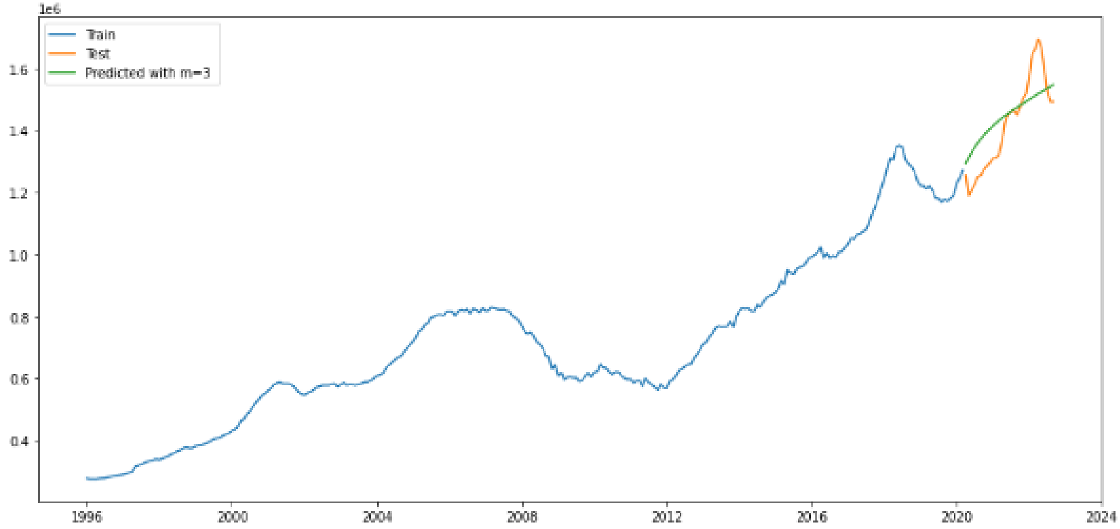
Exogenous variable: unemployment rates

Seasonality of length 3 is the best out of lengths 1, 3, 6, 12, and 24. The r^2 is 0.6, which is not bad.There is an average of $93429.29 in error in our model based on the RMSE, which is also not too bad considering how high the prices are to start with.



Exogenous variable: US mid-tier homes average prices

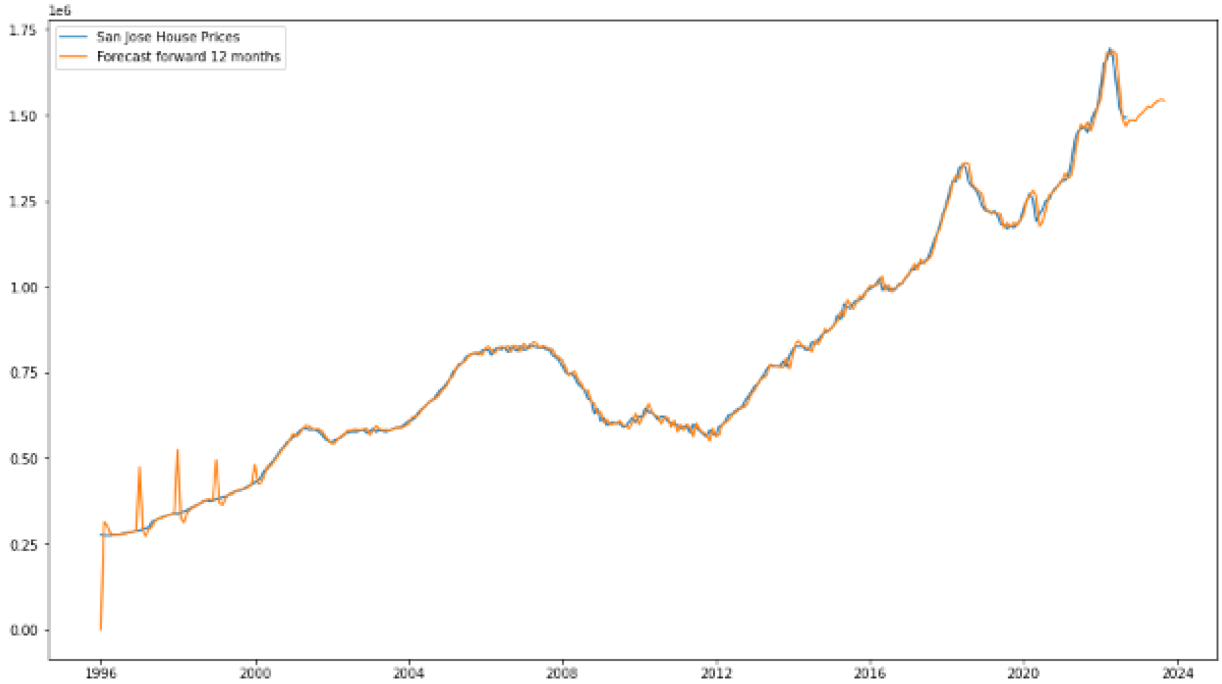
Seasonality of length 3 is the best out of lengths 1, 3, 6, 12, and 24. The r^2 is 0.6, which is not bad. There is an average of $93429.29 in error in our model based on the RMSE, which is also not too bad considering how high the prices are to start with.



Overall, the exogenous variables we used did not make a difference in making our predictions better, which may mean the data may not be highly correlated.

Future Forecast:

The SARIMAX library allows us to forecast the future. Using SARIMAXResults.get_prediction, we predicted a forecast 12 months in advance, as seen in the model below. The prices in San Jose may slightly increase.



What we would work on in the future given more time:

We would like to add in an exogenous variable with interest rate data in the future. We also would like to figure out ways to have the exogenous variables make a difference in results. We want to find the right variables and datasets to make our accuracy better. Given more time, we would like to add in different aspects like interest rates and employment rates in a regression model.

10. Final Results and Winner!

Holt's Winter is the obvious choice as it has the best accuracy, and the prediction model matches the best. Based on the performance of our model, we can accurately predict future housing prices in order for house appraisers to more accurately price houses in the future. Our model does not take into account other external factors, so the prediction may not be accurate; however, based on the current data, average housing prices in San Jose will probably continue to increase even if they drop in the short term.

11. Appendix:

We used W11L2_Univariate_Time_Series_Expo_Smooth.ipynb and W12L2_SARIMA_SARIMAX.ipynb as reference for our code.

This helped us understand feature engineering.

This helped us chose a model.

Ensemble Learning Source.

We used Google Colab and Google Docs to collaborate on code and this report:

https://colab.research.google.com/drive/1yffgvkney64YmJ9hQ46kHMk2ldxcsIGN?usp=sharing

How you can connect our data table to the code: Comment cells 3 and 4, and use this data table to run the notebook. Data table for house values we used. Unemployment data table we used.