Real Estate Sales: Variations in housing prices from 2001 to 2019

George Mason University
AIT – 580 | Prof. Harry Foxwell

Indu Priya Sharma
George Mason University
Fairfax, Virginia
isharma3@gmu.edu

Abstract

The research paper will be targeting the real estate sector in the state of Connecticut and its town. The aim of the research is to analyze the trends in the value of the properties from the different towns throughout the span of nearly a couple of decades. Recently, it has been noticed that the prices of homes have been increasing on a month to month basis which could probably be the after effects of Covid19. But since that data has not been compiled yet, the thought was to review the data that is available and that was from 2001 till 2019 which can give an overview of the real estate sales pre-Covid19 and later when the data is compiled, the findings can be compared. The dataset compiled "Real Estate Sales 2001-2019 GL"[1] was chosen from Data.gov. The hypothesis was that Year and Assessed value of a property would be significant factors impacting Sales value of the property. The first set of analysis such as cleaning and exploratory data analysis was conducted in Python, following which the refined data was used to gain insight with SQL. Lastly R was deployed for future forecasting by utilizing the prophet library from Facebook. Interestingly no attribute had a significant impact on Sales, it was little random, however there were interesting trend lines on monthly and daily basis.

I. Introduction

Real Estate has always been a fascinating aspect. Real estate market is quite a big market in the USA. According to [2], in 2020, there were more than 5.6 million existing units sold along with nearly 800,000 new units. In terms of brokerage, there were more than 100,000 real estate brokerage firms operational in the US. In 2019, almost two thirds of the families will have their own residence. According to [3], there has been a substantial rise in the real estate market in Connecticut since 2020, nearly rising by 18%. Covid-19 Pandemic has played a role in increasing the demand for housing. Interestingly, the number of houses built were fewer during the period from 2001 till 2020 in comparison to 1970 till 2000. According to the NAR, in order to close the deficit, the United States would need to build more than 2 million new units annually over the following ten years.

[3]

II. Literature Review

According to [4], we frequently hear complaints about the mismatch between the ease with which data is accessible and the challenge of utilizing it to provide swift, useful insights. Understanding where to buy property and when to start development has long been a goal for developers and investors. Owners of portfolios must maximize their assets and continuously monitor the circumstances that can prompt them to sell or realize a profit. Slow detection of subtle trends results in money being lost [4]. On the other hand, seizing an attractive (though possibly unnoticed) chance first gives you a substantial edge. Drawing precise conclusions and creating solid business cases are difficult due to the limitations of conventional analytical techniques and data sources. The finest chances have already passed by the time an investor can gather, analyze, and process the facts required to determine a course of action [4]. To take on this challenge head-on, businesses may start by using analytics to carry out their most important strategic imperatives, focusing data-cleansing efforts on the most beneficial use cases first, and developing clear procedures for data governance, interpretation, and decision-making. In the end, data analytics ought to have a distinct strategic course with long-term tasks and objectives beyond just a few pilot projects and use cases [4].

According to [5], more precise house value forecasts are the immediate benefit of predictive analytics for real estate brokers. When predicting trends in future value, predictive analytics takes into account previous data. This means you can give your buyer or seller client a good sense of what the home will be worth decades from now instead of just telling them what the home is currently worth. You now have a lot of negotiating power because of this. You have a negotiating tool to obtain your buyer a better deal if a home is priced competitively to sell on the present market, but its worth is expected to decline over time. If a house's value is anticipated to rise over time, a buyer should consider that information when making an offer, and a seller should consider it when determining the asking price [5].

Prophet, a forecasting toolkit for Python and R, was developed by the Facebook Core Data Science team to make scaling forecasting significantly simpler. Prophet was open sourced in 2017. Prophet's goal is to "enable both specialists and laypeople to produce high-quality predictions that keep up with the demand." Prophet is able to provide accurate forecasts with a minimum of manual input and consistently outperforms other conventional forecasting methodologies. It also allows for the use of domain expertise through intuitive parameter interpretation [5].

The prophet process is broken down to 4 processes: Data Preparation & Exploration, Box-Cox Transform, Forecasting, Inverse Box-Cox Transform. Prophet is rather resilient to missing data, but you should ensure that your time series doesn't have a large number of observations that are missing [5]. This library can help in predicting later the future sales prices and assessed values.

III. Methodology

The dataset chosen was from the Data.gov titled "Real Estate Sales 2001-2019 GL"[1] which contains data about the real estate of towns belonging to the state of Connecticut. The data set was publicly available in the format of .csv and was acquired in the same format. Python was first utilized for cleaning up the data and then the analysis moved on to SQL along with R.

A. Overview of Dataset

The dataset consists of 14 rows and has 997213 records. Here is an overview of the dataset:

Table1: Data Attributes, Types and Description

Attribute	NOIR	Description
Serial Number	Nominal	Unique random number, an id to classify each entry in the table
List Year	Interval	Year for the concerned entry, used as yearly intervals for prophet library forecasting
Date Recorded	Interval	Entry date for the concerned entry, month was extracted to be used as intervals
Town	Nominal	Town for the concerned real estate property
Address	Nominal	Actual address for the real estate property
Assessed Value	Ratio	The assessed value for the real estate property
Sale Amount	Ratio	Actual sale amount for that particular property
Sales Ratio	Ratio	% of Sale amount over assessed value, can be 0 too
	Ordinal	Column to be divided into ranked category
Property Type	Nominal	Tells about the property type - residential, condo etc
Residential Type	Nominal	Tells about the residential type- single or double family condo etc
Non Use Code	Nominal	Non usable scalable code for real estate management
Assessor Remarks	Nominal	Remarks the authority who assessed the property
OPM remarks	Nominal	Remarks from OPM
Location	Nominal	Coordinates for the real estate property

B. Python

The tool utilized for working on python is jupyter notebook as a part of Anaconda suite of software. The libraries utilized were Pandas for creating and analyzing dataframe. Numpy library was also utilized for statistical analysis. Additionally, seaborn, ggplot and matplot libraries were utilized for visualization purposes. Further sklearn library was utilized for testing out machine learning algorithms.

The first step was to utilize pandas to read the csv file. As the first step of exploratory data analysis, the aim was to check the number of missing values. Here is the count of the null values in the dataset.

```
#checking for null values
print(df.isnull().sum())
```

```
Serial Number            0
List Year                0
Date Recorded            2
Town                     0
Address                 51
Assessed Value           0
Sale Amount              0
Sales Ratio              0
Property Type       382446
Residential Type    388309
Non Use Code        707532
Assessor Remarks    847349
OPM remarks         987279
Location            799516
dtype: int64
```

Image1: Count of null values per column

The column 'Address' was dropped as it was Personal Identifiable Information using drop() function. Following that the columns 'OPM remarks', 'Non Use Code', 'Assessor Remarks' and 'Location' were also removed as they would be non critical columns for further purposes. Another cause for removal was that it had a lot of missing records.

Then the correlation was done to check correlation between attributes. Here is the correlation using the sns heatmap plot:

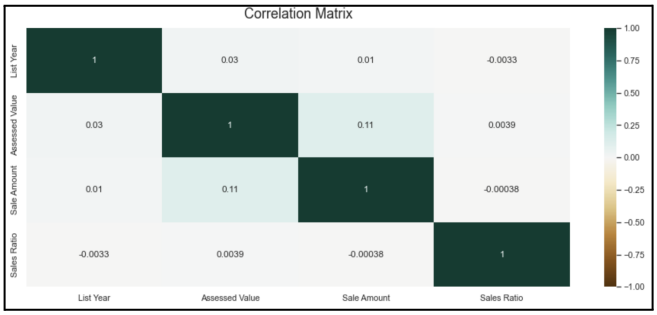

Image2: Correlation Plot

Using to_datetime() in the pandas library, the daterecorded was converted into date type and then month was extracted using the '.month' function. Here is the code for the same:

```
dfMLSample["Date Recorded"] = pd.to_datetime(dfMLSample["Date Recorded"])

dfMLSample['Month'] = pd.DatetimeIndex(dfMLSample['Date Recorded']).month
```

Image3: Code for converting to date and extracting month

As it can be observed from the above correlation plot that none of the numeric variables in the dataset are correlated to each other, all possible correlations are weak. So the chances of a regression model performing well would be quite low.

Prior to that an essential step was to continue exploratory data analysis which along with data cleaning was done in

python. To get a better understanding of the distribution of the data, visualizations were done. The type of plots that were utilized were bar plots, box plots, scatter plots and line plots. The visualizations would be available in the Section IV 'Results'. The currently available dataframe was converted into a .csv file for analysis on SQL which will be covered in the subsection III. C. 'SQL' using the following code:

```
dfMLSample.to_csv('RealEstateFinalClean.csv',index=False)
```

Image4: Exporting to .csv

It was noticed that the attributes 'Property Type' and 'Residential Type' had similar values for majority of the column, so the column 'Residential Type' was also dropped from the data frame using the df.drop().

```
dfMLNADrop = dfMLNADrop.drop(['Residential Type'], axis=1)
```

Image5: Dropping column

After reviewing the box plots for the 'Assessed Value', 'Sale Amount' and 'Sales Ratio', it was found that there are a lot of outliers in these columns so it had to be cleaned. Firstly, the Interquartile Range, Quartile 1 and Quartile 3 were utilized to filter the records which lie between Lower bound [(Quartile 1) - (1.5* (Interquartile Range))] and Upper bound [(Quartile 3) + (1.5* (Interquartile Range))]. The overview of the code for finding Quartile 1, Quartile 3, Interquartile Range and the bounds for one of the columns 'Assessed Value':

```
ASQ1 = np.percentile(dfMLNADrop['Assessed Value'], 25,
                     interpolation = 'midpoint')
ASQ3 = np.percentile(dfMLNADrop['Assessed Value'], 75,
                     interpolation = 'midpoint')
ASIQR = ASQ3 - ASQ1
ASupper = ASQ3+1.5*ASIQR
ASlower = ASQ1-1.5*ASIQR
```

Image6: Removing outliers using quartile and range

However, upon filtering for all the three concerned columns, there were no records left that suited the criteria. Since this method couldn't filter out records in the next attempt the attribute 'Sale Amount' was considered and the range considered was from 10th percentile to 90th percentile, which would also eliminate a lot of the outliers and this column had considerably higher outliers.

```
dfMLNADrop['Sale Amount'].quantile(0.1) #10th Percentile
```

```
85000.0
```

```
dfMLNADrop['Sale Amount'].quantile(0.9) #90th Percentile
```

```
629000.0
```

Choosing sample between 10th and 90th percentile to avoid outliers affecting the machine learning algorithm

```
dfMLSample = dfMLNADrop.loc[dfMLNADrop['Sale Amount'] >= 85000]
dfMLSample = dfMLSample.loc[dfMLNADrop['Sale Amount'] <= 629000]
```

Image7: Removing outliers by choosing bound between 10th and 90th percentile

This sample filtered out now had 488,750 records which was again now exported to .csv file for further work in R that will be covered in the subsection III. D. 'R'.

With the sample data, the next step was to perform a machine learning algorithm and the target variable was 'Sale Amount'. Since this is a numerical target variable, the algorithm to work upon would be the regression modeling. So, before deploying a linear regression model, an essential aspect is to check if there is any correlation between the dependent (target) and the independent variables. Below is the correlation heatmap:

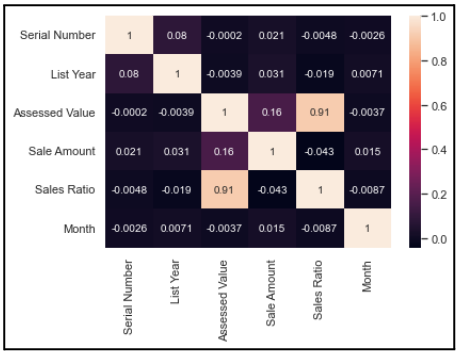

Image8: Correlation heat map

As we can see that the correlation still does not exceed 0.16 which again indicates that the 'Sale Amount' is not correlated with any attributes, thus the regression model would not prove a viable option. The only strong correlation is between 'Assessed Value' and 'Sales Ratio' of 0.91. However, in this project linear regression was still deployed to check if any useful insight can be gained.

C. SQL

For this project, the SQL that has been used is Postgres (version 15) and the GUI used is pgAdmin (version 4). The schema for the respective table is:

Query | Query History

```
1  CREATE TABLE FinalProj
2  (
3      SerialNumber varchar,
4      ListYear int,
5      DateRecorded date,
6      Town varchar,
7      AssessedValue decimal,
8      SaleAmount decimal,
9      SalesRatio decimal,
10     PropertyType varchar,
11     ResidentialType varchar
12 );
```

Image9: Table Schema

Once the table was created, with the first exported .csv file, the table was made and using the import/export option available in the pgAdmin, the .csv was imported into the table.

By utilizing SQL, the aim was to figure out crucial information:

- Average and Maximum Sale amount for each year
- Average and Maximum Sale amount for each town
- Average and Maximum Sale ratio for each town
- Town which had maximum Assessed value property each year
- Town which observed the maximum Sale amount for one of it's property year wise
- Town which had maximum Sales Ratio each year

D. R

For the language R, the IDE used was RStudio. The second sample that was exported to .csv type was loaded into a dataframe. The libraries for the project done in R were 'tidyverse', 'ggplot2', 'lubridate' and 'prophet'. Once the .csv was read into a dataframe, first step was to check the summary of the dataframe which was done using the function str():

```
> str(df)
'data.frame':   488750 obs. of  9 variables:
 $ Serial.Number : int  20002 200212 2020180 20139 200086 2000381 201295 200032 200354 200527 .
 $ List.Year     : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
 $ Date.Recorded : chr  "2020-10-02" "2021-03-09" "2021-03-01" "2020-12-16" ...
 $ Town          : chr  "Ashford" "Avon" "Berlin" "Bethel" ...
 $ Assessed.Value: num  253000 130400 234200 171360 168900 ...
 $ Sale.Amount   : num  430000 179900 130000 335000 352000 ...
 $ Sales.Ratio   : num  0.588 0.725 1.802 0.511 0.48 ...
 $ Property.Type : chr  "Residential" "Residential" "Residential" "Residential" ...
 $ Month         : int  10 3 3 12 8 9 9 10 12 3 ...
```

Image10: About Dataframe

Next step was to convert the 'Date.Recorded' into date type using the as.Date() function:

```
> df$Date.Recorded <- as.Date(df$Date.Recorded)
> str(df)
'data.frame':   488750 obs. of  9 variables:
 $ Serial.Number : int  20002 200212 2020180 20139 200086 2000381 201295 200032 200354 200527 ..
 $ List.Year     : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
 $ Date.Recorded : Date, format: "2020-10-02" "2021-03-09" "2021-03-01" "2020-12-16" ...
 $ Town          : chr  "Ashford" "Avon" "Berlin" "Bethel" ...
 $ Assessed.Value: num  253000 130400 234200 171360 168900 ...
 $ Sale.Amount   : num  430000 179900 130000 335000 352000 ...
 $ Sales.Ratio   : num  0.588 0.725 1.802 0.511 0.48 ...
 $ Property.Type : chr  "Residential" "Residential" "Residential" "Residential" ...
 $ Month         : int  10 3 3 12 8 9 9 10 12 3 ...
```

Image11: Converted to Date Type and about the dataframe

Since the regression modeling had failed and it would not be a feasible option for forecasting, the prophet[6] library by Facebook was explored. Firstly, the main dataframe was splitted into three smaller dataframe which had these columns:

1. df1: 'Date.Recorded' and 'Assessed.Value'
2. df2: 'Date.Recorded' and 'Sale.Amount'
3. df3: 'Date.Recorded' and 'Sales.Ratio'

```
> df1 <- df[,c("Date.Recorded","Assessed.Value")]
> df2 <- df[,c("Date.Recorded","Sale.Amount")]
> df3 <- df[,c("Date.Recorded","Sales.Ratio")]
```

Image12: Splitting dataframe into three sub dataframes

Afterwards in each dataset, the respective values ('Assessed. Value', 'Sale.Amount', 'Sales.Ratio') were averaged out per day in order to get the mean value for each date entry. The code:

```
# mean assessed value
df1Ren <- df1 %>% group_by(Date.Recorded) %>%
  summarise(mean_assessedvalue=mean(Assessed.Value),
            .groups = 'drop')
```

Image13: Code for mean Assessed value per day into a new dataframe

For the prophet library to work, the requirement is to have the date column be renamed to 'ds' and the other column that is to be predicted be renamed as 'y' ie in all the three dataframes, the Date.Recorded was renamed as 'ds' and 'Assessed.Value', 'Sale.Amount', 'Sales.Ratio' were renamed to 'y' respectively. So, there would be three prophet models running for predicting the 3 'y's respectively.

For fitting prophet models, the function used is prophet(). Then using the function make_future_dataframe() function, the frequency and number of periods can be specified for prediction. Then the predict() function can be used for predicting and the same can be plotted using the plot() function. Prophet also has a built-in plot function prophet_plot_components() through which trends can be analyzed for daily, monthly, yearly as well. Below is the code:

```
> m <- prophet(df1Ren)
Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override thi
s.
> predFuture <- make_future_dataframe(m, periods = 365)
> forecastFuture <- predict(m, predFuture)
> plot(m, forecastFuture,xlabel = "Months", ylabel = "Assessed Value") + labs(title
="Assessed Value throughout the Years")
> prophet_plot_components(m, forecastFuture)
```

Image14: Prophet model and plot

IV. RESULTS

A. PYTHON

The below are the findings from the exploratory data analysis conducted in Python:

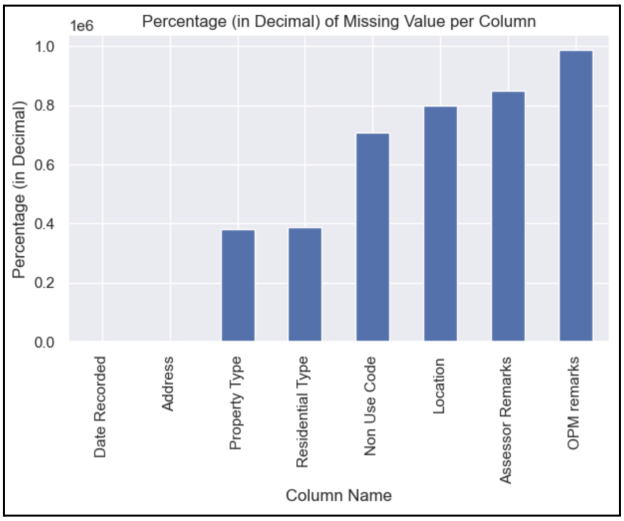

Image15: Percentage of Missing Value

It's very important to know about the missing values in the dataset that we use. The above representation shows us about the percentage of missing values where OPM remarks has the humongous amount of blank spaces, which is closely followed by Assessed remarks and Location, Non-Use Code like the stepping stone to success.

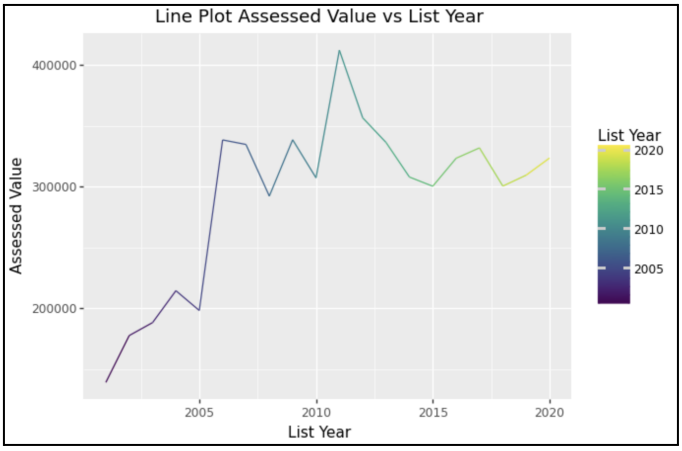

Image16: Assessed Value over Year

The graph gives us the description about the trend in assessed value of a property from 2000 to 2020. There have been many ups and downs in the graph with the lowest value

$300,000. 2011 recorded the highest number of all crossing $400,000 just like the tip of the iceberg.

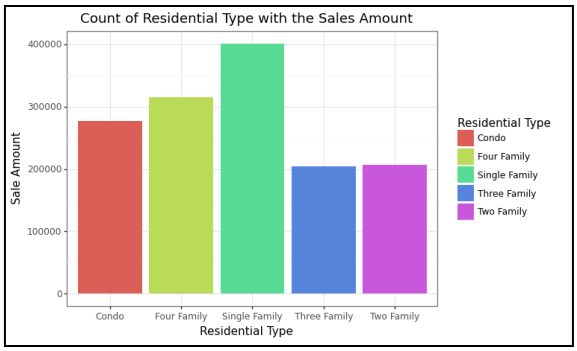

Image17: Count of Residential Type with the Sales Amount

As per the various property types available from the list, single family homes from residential type are a hit which is followed by a four family and a condo.

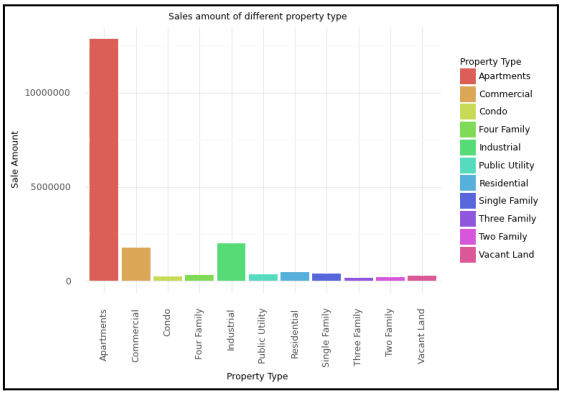

Image18: Sale Amount for property type

The sales amount of apartments is clearly booming out all its competitors which is followed by the industrial land and commercials.

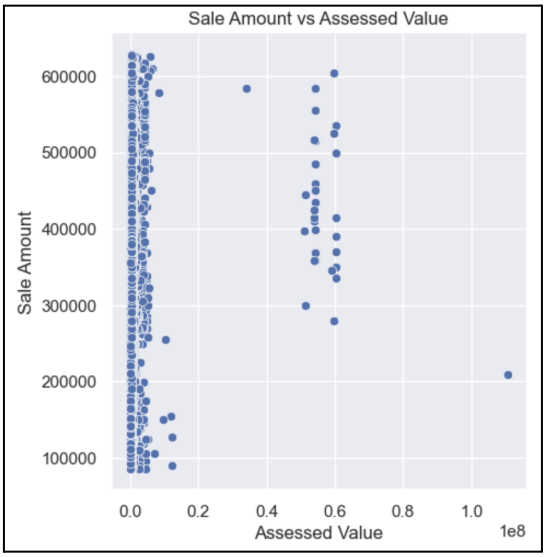

Image19: Sale Amount vs Assessed Value

Once the second sample was filtered which contained Sales Amount between 10th and 90th percentile (for outlier removals), the scatter plot was generated for Sale Amount vs Assessed value to see if there is any linear relation between them. However as can be observed, there is no linear relation as can be confirmed from the correlation plot in the image 8.

Image20: Sales Ratio vs Assessed Value

As there was a high correlation between Sales Ratio and Assessed Value (0.91) it indicates that increase in one would lead to increase in other which is evident from the scatter plot above that shows that there is sort of a linear relation between the two attributes.

B. SQL

The below are findings from the exploratory data analysis conducted in SQL:

Query 1:

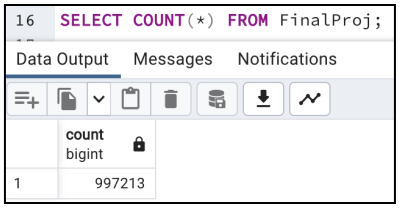

Image21: Query Count

The number of records in the dataset was 997,213. To ensure the consistency in the data frame in python and exported .csv file, this was done.

Query 2:

```
SELECT ListYear, MAX(SaleAmount) FROM FinalProj
GROUP BY ListYear ORDER BY ListYear ASC;
```

	listyear integer	max numeric
1	2001	89280000.0
2	2002	103631200.0
3	2003	85552581.0
4	2004	97750000.0
5	2005	87000000.0
6	2006	181700000.0
7	2007	124000000.0
8	2008	85177000.0
9	2009	90955593.0
10	2010	62000000.0
11	2011	70000000.0
12	2012	104900000.0
13	2013	152384149.0
14	2014	115500000.0
15	2015	105500000.0
16	2016	395500000.0
17	2017	136500000.0
18	2018	163000000.0
19	2019	230043624.0
20	2020	5000000000.0

Image22: Query to retrieve max sale amount per year

From the query that returns the max sale amount for each year, we can see that from the period of 2001 till 2008, the max sale amount has roughly been consistently around $85 million to $100 million dollars with an exceptionally big same amount in 2007 and 2008. Again there was a spike in 2016 ($395 million) compared to 2015 ($105 million) that dipped again in 2017. The year 2020 saw the biggest jump with the sale amount touching $5 billion.

Query 3:

```
SELECT ListYear, AVG(SaleAmount) FROM FinalProj
GROUP BY ListYear ORDER BY ListYear ASC;
```

listyear integer	avg numeric
2001	246235.035160445757
2002	296357.123705639891
2003	327217.932922368032
2004	380297.014169125345
2005	364030.126083568715
2006	475379.225384851901
2007	435713.379734396496
2008	325831.792393462655
2009	355250.327161945987
2010	331657.472574721567
2011	391684.320746821181
2012	395477.676013120952
2013	413516.239641489122
2014	401421.941219659827
2015	345883.763949325845
2016	507761.249271693488
2017	393251.314693484494
2018	383727.664935218600
2019	420296.971308138549
2020	604963.871050876982

Image23: Query to retrieve average sale amount per year

From the query that returns the average sale amount for each year, we can see that from the period of 2001 till 2004, the average amount kept increasing, then from 2006 till 2008 there was a dip in the average. From 2008 it was again an increase till 2014. Since 2018 it has again been on the rise.

Query 4:

```
SELECT Town, MAX(SaleAmount) FROM FinalProj GROUP
BY Town ORDER BY MAX(SaleAmount) DESC LIMIT 10;
```

	town character varying	max numeric
1	Willington	5000000000.0
2	Stamford	395500000.0
3	Waterbury	230043624.0
4	Greenwich	181700000.0
5	Manchester	161238793.0
6	Norwalk	157000000.0
7	Hamden	136500000.0
8	Westport	130000000.0
9	Danbury	124000000.0
10	Hartford	113250000.0

Image24: Query to retrieve average sale amount per year

The towns with the top 10 max sales amount since 2000 to 2020 are shown above. Willington has seen the biggest sale amount and in comparison to that the 10th city (Hartford) on the list is just 2.3% of the highest ever sale.

Query 5:

```
SELECT Town, AVG(SaleAmount) FROM FinalProj GROUP
BY Town ORDER BY AVG(SaleAmount) DESC LIMIT 10;
```

Output Messages Notifications

town character varying	avg numeric
Willington	4620008.458795562599
Greenwich	2071224.808641748131
Darien	1534504.405194805195
New Canaan	1512124.767171314741
Westport	1392323.992808219178
Weston	931835.775703794370
Wilton	894222.435051048821
Stamford	876235.225245165852
Washington	854023.997269417476
Ridgefield	735996.218220338983

Image25: Query to retrieve average sale amount per Town

Again we can observe that the town Willington has the highest average sale amount crossing $4.6 million which is not surprising as it had generated the highest sale amount by quite a lot of margin. Interestingly the second spot was taken over by Greenwich town.

Query 6:

```
SELECT Town, MAX(SalesRatio) FROM FinalProj GROUP
BY Town ORDER BY MAX(SalesRatio) DESC LIMIT 10;
```

Output | Messages | Notifications

town character varying	max numeric
Salisbury	1226420.0
New Fairfield	611900.0
Westport	594000.0
Brookfield	519130.0
Newtown	473780.0
Monroe	368680.0
Beacon Falls	241910.0
Bethany	224940.0
Guilford	198960.0
Farmington	196980.0

Image26: Query to retrieve max sale ratio per town

The result for sales ratio is quite surprising, as the max sales ratio is now for the town of Salisbury which had not been in the part of earlier two queries.

Query 7:

```
SELECT Town, AVG(SalesRatio) FROM FinalProj GROUP
BY Town ORDER BY AVG(SalesRatio) DESC LIMIT 10;
```

Output | Messages | Notifications

town character varying	avg numeric
Salisbury	799.7715616397812500
Bethany	261.0102860135154703
Beacon Falls	204.5847323229645503
New Fairfield	148.7174227720519073
Brooklyn	100.0483792687682281
Newtown	96.0614462888927699
Brookfield	91.5800349817206216
Thompson	84.7598597522404850
Middlebury	83.6256131016014109
Westport	65.4971161076847141

Image27: Query to retrieve average sale ratio per town

Unsurprisingly in the average sales ratio again Salisbury town is leading indicative of better investment return/yield on real estate.

Query 8:

```
SELECT DISTINCT(ListYear), Town, SaleAmount FROM
FinalProj WHERE (ListYear, SaleAmount) IN
(SELECT ListYear, MAX(SaleAmount) FROM FinalProj
 GROUP BY ListYear )ORDER BY ListYear DESC;
```

listyear integer	town character varying	saleamount numeric
2020	Willington	5000000000.0
2019	Waterbury	230043624.0
2018	Stamford	163000000.0
2017	Hamden	136500000.0
2016	Stamford	395500000.0
2015	Windsor	105500000.0
2014	Stamford	115500000.0
2013	Stamford	152384149.0
2012	West Hartford	104900000.0
2011	Stamford	70000000.0
2010	Stamford	62000000.0
2009	Trumbull	90955593.0
2008	Danbury	85177000.0
2007	Hamden	124000000.0
2006	Greenwich	181700000.0
2005	Norwalk	87000000.0
2004	Greenwich	97750000.0
2003	Stamford	85552581.0
2002	Stamford	103631200.0
2001	Stamford	89280000.0

Image28: Query to retrieve the Town with max sale amount each year

The aim was to figure out for each year which city peaked the sale amount. As we can observe Willington had peaked in 2020. Out of the two decades, the town which peaked the most was Stamford, it peaked for 9 years.

Query 9:

```
SELECT DISTINCT(ListYear), Town, AssessedValue FROM
FinalProj WHERE (ListYear, AssessedValue) IN
(SELECT ListYear, MAX(AssessedValue) FROM FinalProj
 GROUP BY ListYear ) ORDER BY ListYear DESC;
```

listyear integer	town character varying	assessedvalue numeric
2020	Stamford	114924210.0
2019	Waterbury	142858700.0
2018	Stamford	105438300.0
2017	Hamden	881510000.0
2016	New Britain	131072830.0
2015	Suffield	138958820.0
2014	Stamford	70180430.0
2013	Westport	78206200.0
2012	Greenwich	56112000.0
2011	West Haven	89465210.0
2010	Hamden	110670208.0
2009	Greenwich	62895000.0
2008	Greenwich	62895000.0
2007	Greenwich	62895000.0
2006	Greenwich	122935400.0
2005	Norwalk	50095400.0
2004	Milford	65145150.0
2003	Greenwich	38350900.0
2002	Hartford	75041970.0
2001	Stamford	39819770.0

Image29: Query to retrieve the Town with max assessed value each year

The next aim was to figure out for each year which city peaked the assessed value. We can see that Stamford had peaked in 2020. Out of the two decades, the town which peaked the most was Greenwich, it peaked for 6 years.

Query 10:

```
SELECT DISTINCT(ListYear), Town, SalesRatio FROM
FinalProj WHERE (ListYear, SalesRatio) IN
(SELECT ListYear, MAX(SalesRatio) FROM FinalProj
 GROUP BY ListYear ) ORDER BY ListYear DESC;
```

listyear integer	town character varying	salesratio numeric
2020	Stratford	679.5008
2019	Bridgeport	988.0726
2018	Beacon Falls	241910.0
2017	Plymouth	10328.0
2016	Madison	1672.8
2015	Wallingford	4516.083916
2014	Bethany	1083.474
2013	Stratford	2724.858889
2012	Norwalk	3467.128
2011	Bridgeport	2537.222667
2010	Stratford	1511.673333
2009	Bridgeport	6343.056667
2008	Middletown	1011.843845
2007	Norwalk	1731.299517
2006	Salisbury	1226420.0
2005	New Fairfield	611900.0
2004	Westport	594000.0
2003	Thompson	83000.0
2002	Hartford	165060.0
2001	Stamford	41303.5

Image30: Query to retrieve the Town with max sale ratio each year

Final aim was to figure out for each year which city peaked the sales ratio. We can see that Stratford had peaked in 2020. Out of the two decades, the town which peaked the most was Bridgeport, it peaked for 3 years.

C. R

In R, firstly, the sales ratio was analyzed in the form of grouped intervals which were ranked, hence also utilized in the form of Ordinal Type, where Rank 1 meant better performance than Rank 2 then Rank 3 and so on, Rank 5 had the lowest Sales Ratio. The intervals were decided after reviewing the summary of the column by using the summary() function in the R. For this step, two new data frames were created which had Average and Max Sales Ratio per Year. This was done by using the same aggregate function in the image 13 with mean and max function. Then the next step was to divide into intervals for Ranks. Rank1 would be the best rank with the highest possible Sales Ratio(either mean or max) and Rank 5 with the minimum. The cut function was also used to create mentioned intervals which took the aggregated sales ratio column and divided it into the ranked intervals.

We can see from the image below that there was a steady increase in the average sales ratio from 2006 till 2011 then it dipped till 2018 has remained stable since then.

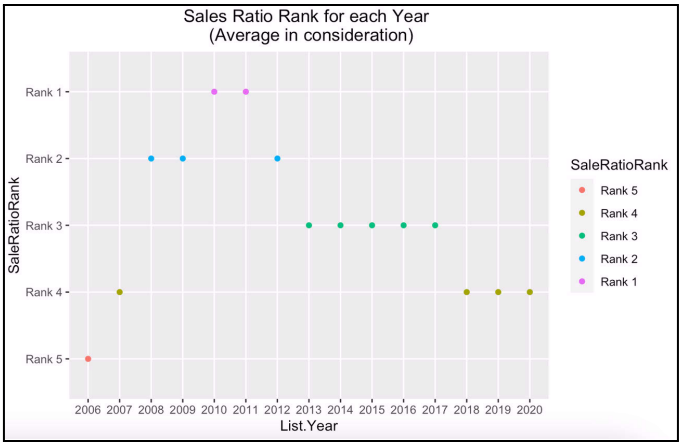

Image31: Trend in Assessed Value

However, if we take the max sales ratio into consideration we can see that almost each year, it changes and follows somewhat of a wave pattern with increase and decrease with each passing year. 2010 and 2020 were the best years.

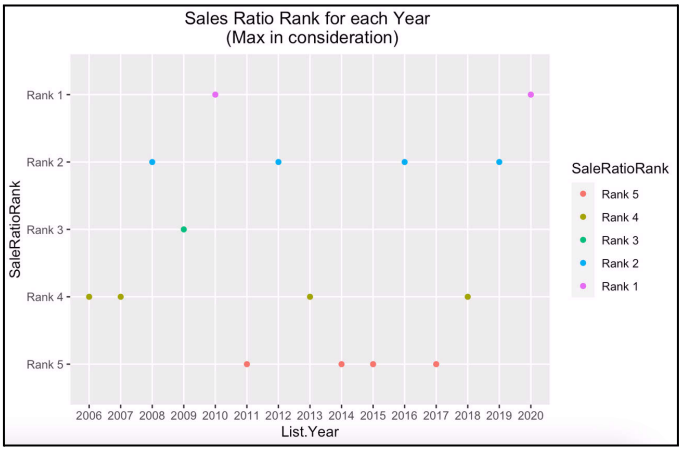

Image32: Trend in Assessed Value

The next part of analysis can be segregated into three sections; each for 'Assessed Value', 'Sale Amount' and 'Sale Ratio' respectively. Below are the findings for them:

1. 'Assessed Value'

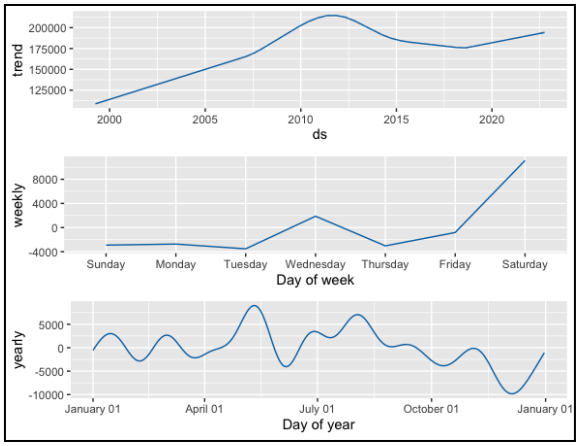

Image33: Trend in Assessed Value

In the sample that was analyzed, Wednesdays, Friday end of day to Saturday predicts more crowds going to purchase/sell. While 2011 had the highest assessed value, May end to mid June remains to be a good fit for many.

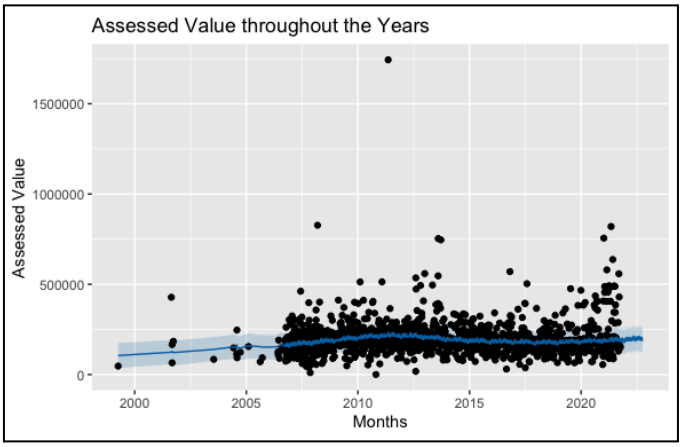

Image34: Assessed Value throughout the years

It is particularly evident that the years 2006 to 2020 had stepped up in purchases of assets, which is continuously seen in the forecasted years. Although the value ranges between $0 to $500,000 mostly. Investment in real estates has a profit and also serves as a pension for the rest of the life. The blue line that protrudes at the right end is the forecasting from prophet library which shows the assessed value of the properties would increase in the coming years.

2. 'Sale Amount'

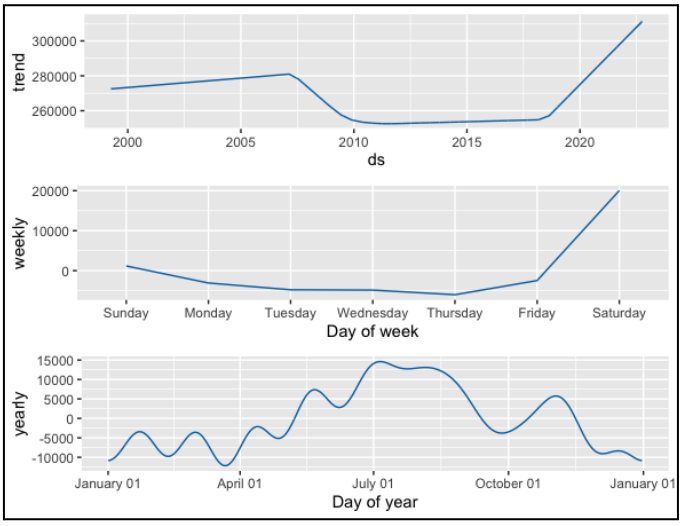

Image35: Trend in Sale Amount

According to the yearly trend 2007 to 2010 there was a consistent increase in the price at which the sales took place, which then stepped down to <260,000 and was the same until 2019, post that there has been an increment, the future predictions also show advancement in the amount. People are interested in purchasing on the weekends more than on the weekdays. Also, summer is the time when people go out for shopping than the rest of the year.

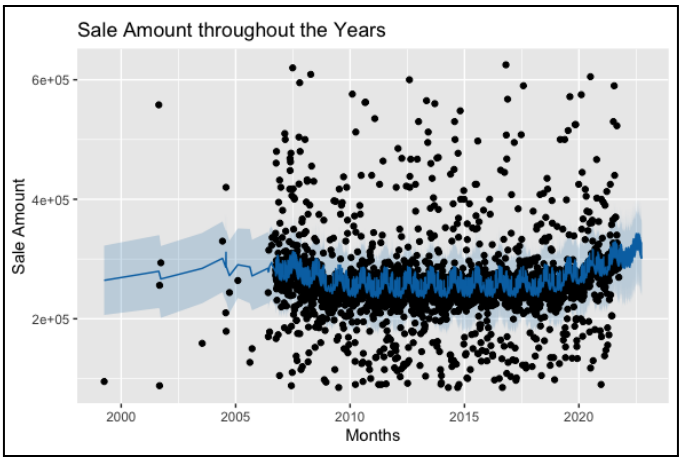

Image36: Sale amount throughout the years

The above graph depicts the sales amount throughout the years specified in the dataset. We can see that there has been an increment in the number of sales post 2005 ranging above $200,000 to $300,000. The properties have been sold as low as $100,000 to as high as $600,000, the future prediction specifies that there would be an increase in the amount the land being sold to approximately $300,000 as minimum amount. The blue line that protrudes at the right end is the forecasting from prophet library which shows the Sale amount of the properties would increase in the coming years but in the zigzag trend as it had been.

3. 'Sale Ratio'

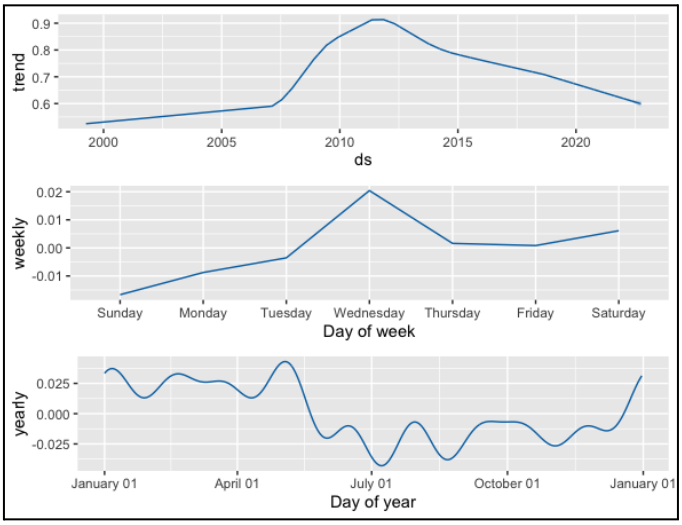

Image37: Trend in Sales Ratio

Although there is a boost in the sales amount and the assessed value of a property the overall ratio seems to decrease on the other hand. On a weekly basis, Wednesdays seem a perfect time for individuals to go looking for houses. Whereas on yearly basis May end to mid June has high chances of people looking for apartments/properties apart from that during the year end/start.

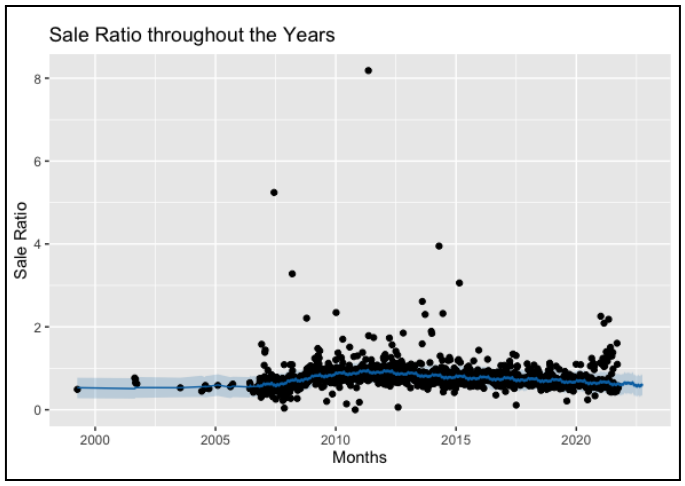

Image38: Sales Ratio throughout years

The year 2012 had the highest sales ratio where as the least almost 0% was observed in 2007, 2011, 2013, 2017 this indicates that there are some specific months, places and properties that haven't had much profit selling them. Surprisingly there has not been any difference in the sales ratio during the covid times during the year 2019 to 2020. There has been a significant increase in the percentage of sales from 2006 which then becomes stable/decreases during covid times. Looking at the future predictions of our graph it is seen that the ratio is decreasing. The blue line that protrudes at the right end is the forecasting from prophet library which shows the Sales Ratio of the properties would increase in the coming years.

C. Machine Learning

For the machine learning the linear regression was utilized just to check if it was possible to get any insight despite the low correlation as available in the image 8.

Linear Regression

```
lm = LinearRegression()
lm.fit(train_X, train_y)
lm.score(test_X, test_y)
```

```
# Predict using test data
yPred = lm.predict(test_X)
```

```
from sklearn.metrics import r2_score
acqSc =r2_score(test_y, yPred)
acqSc
```

```
0.09307071857183669
```

Image39: Linear Regression Model Output

The sklearn library was utilized in the python and as we can see that the accuracy received was just 9.3% indicating the model has not fit and cannot be used for predictions. The target variable was 'Sale Amount' and the remaining numeric columns were the independent variables.

V. Limitations

The biggest limitation for the dataset would be the fact that the machine learning model failed to successfully deploy because of the variations in the data. There were cases where the sales ratio exceeded 1000% and it seems that the data has not been completely curated by the officials as it still has some missing information or possibly incorrect information. However, as it is the real estate sector, there is also a possibility that the properties can gain value significantly to achieve a higher sales ratio. It would have been even more interesting if the data from the covid phase was also available to gain even more insights on how the pandemic affected the real estate sector in Connecticut.

The dataset had a significant number of missing values as well which in case were provided might have made the other attributes more significant and could have helped with the regression modeling. So, there needs to be further analysis that can be done after getting even more refined data from official government authorities as the framework for research has been constructed in this project. The dataset doesn't contain information about the sector of the plot that can assist in the sale amount such as School in neighborhood or Commercial market area. These further attributes could also help in better understanding the real estate market of Connecticut.

VI. Discussion/Conclusion

Though the regression model couldn't be utilized, the prophet library along with the exploratory data analysis did provide valuable insights. Once the data was filtered, strong correlation between Sales Ratio and Assessed Value could be established. However, with lack of correlation from either with Sales amount meant exploring different options for forecasting. As is evident the prices have surged in the last two decades.

The towns 'Willington', 'Salisbury' and 'Greenwich' town were the favorites for the real estate sector investment. We could ascertain that with the current dataset available, there is no correlation between the Assessed Value and Sales Amount. Sales Ratio has more or less been stable throughout the course of two decades with slight gradual increase. Apartments are being sold most out of all the types of real estate with the highest sales ratio. The years 2010 and 2020 were the best for the real estate sector.

The article [3] had mentioned an increase in real estate sector value post 2020 which is also evident from the predictions made by the prophet library. Data analytics plays a pivotal role in the real estate sector and would continue to do so. The framework can let potential customers know when to sell and buy property. The project is also a stepping stone for future work once Pandemic data is available as well.

References

[1] "Real estate sales 2001-2019 GL," *Catalog*, 29-Nov-2021. [Online]. Available: https://catalog.data.gov/dataset/real-estate-sales-2001-2018. [Accessed: 15-Oct-2022].

[2] "Quick Real Estate Statistics," www.nar.realtor, 11-Nov-2020. [Online]. Available: https://www.nar.realtor/research-and-statistics/quick-real-estate-statistics. [Accessed: 15-Oct -2022].

[3] Stacker, "10 statistics about Connecticut's real estate market," *Stacker*, 22-Dec-2021. [Online]. Available: https://stacker.com/connecticut/10-statistics-about-connecticuts-real-estate-market. [Accessed: 03-Dec-2022].

[4] G. M. Asaftei, S. Doshi, J. Means, and A. Sanghvi, "Getting ahead of the market: How big data is transforming real estate," McKinsey & Company, 30-Mar-2021. [Online]. Available: https://www.mckinsey.com/industries/real-estate/our-insights/getting-ahead-of-the-market-how-big-data-is-transforming-real-estate. [Accessed: 30-Oct-2022].

[5] E. Sires, "How to increase real estate sales with Predictive Analytics," *Rapid Insight*, 02-Jun-2021. [Online]. Available: https://www.rapidinsight.com/blog/how-to-increase-real-estate-sales-with-predictive-analytics/. [Accessed: 30-Oct-2022].

[6] "Forecasting in python with prophet: Reports - mode," *Mode Resources*, 01-Mar-2018. [Online]. Available: https://mode.com/example-gallery/forecasting_prophet_python_cookbook/. [Accessed: 30-Oct-2022].