CONNECTICUT REAL ESTATE DATA ANALYST PROJECT

TABLE OF CONTENTS

Business problem/Objective

Market Trends and Price Analysis:

- Identify and understand the overall trends in real estate within Connecticut sales over the years. Analyze changes to gain insights into the market dynamics and price fluctuations.
- This information can help businesses assess the competitiveness of different areas, track market trends, and make pricing decisions.

Property and Residential Type Analysis:

- Investigate the distribution of property types and residential types within the dataset. Determine which property types (e.g., residential, commercial, land) and residential types (e.g., single-family, multi-family, condominiums) are most prevalent in the dataset.
- This analysis can provide valuable information for developers, investors, and real estate agents to identify market niches and understand customer preferences.

Applying Models:

- Utilize various predictive modeling techniques such as Linear Regression, Decision Tree, and Logistic Regression to analyze the real estate sales data.
- This analysis can help businesses and stakeholders make informed decisions regarding the usage of models in pricing strategies, investment opportunities, and market forecasting.

DATA OVERVIEW

Name

Connecticut’s Real Estate Sales 2001-2020 GL

Description

The Connecticut Office of Policy and Management maintains a listing of all real estate sales with a sales price of $2,000 or greater that occur between October 1 and September 30 of each year.

Shape

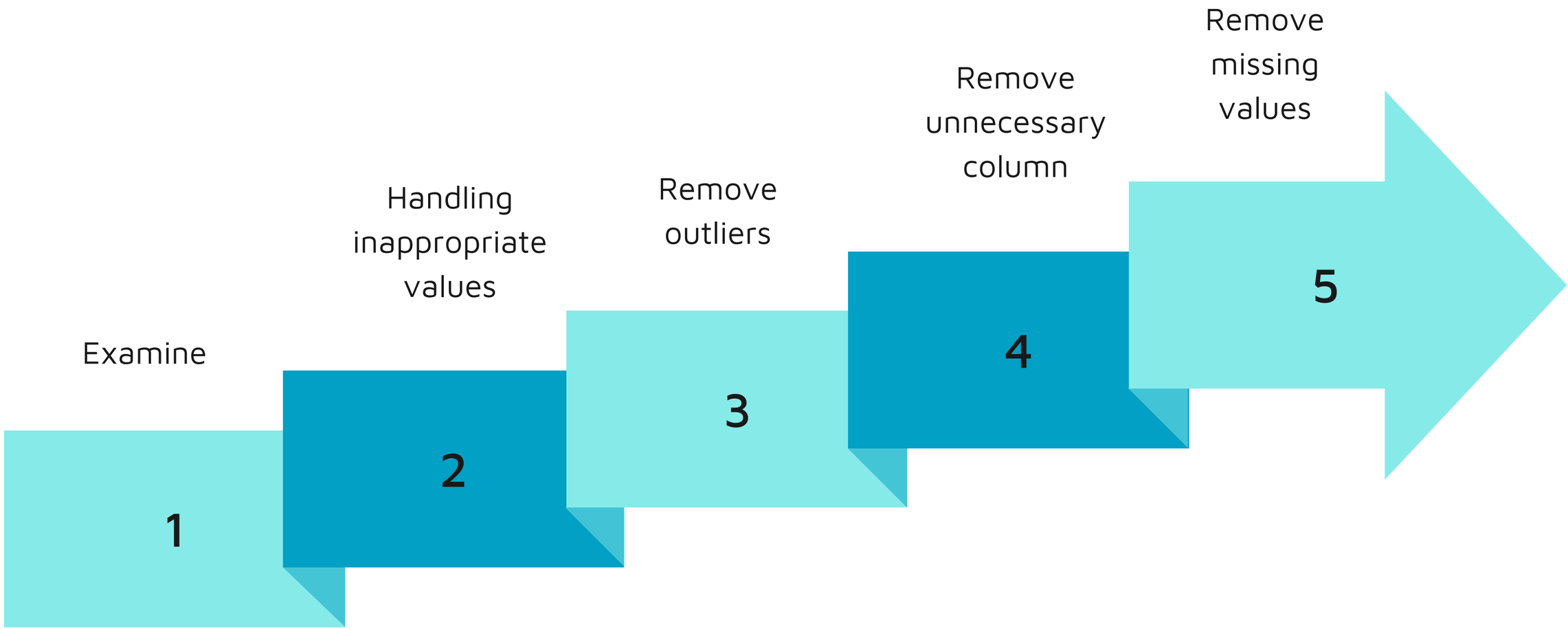
- 997213 rows
- 14 columns

```
df.head()
```

	Serial Number	List Year	Date Recorded	Town	Address	Assessed Value	Sale Amount	Sales Ratio	Property Type	Residential Type	Non Use Code	Assessor Remarks	OPM remarks	Location
0	2020348	2020	2021-09-13	Ansonia	230 WAKELEE AVE	150500.0	325000.0	0.4630	Commercial	NaN	NaN	NaN	NaN	NaN
1	20002	2020	2020-10-02	Ashford	390 TURNPIKE RD	253000.0	430000.0	0.5883	Residential	Single Family	NaN	NaN	NaN	NaN
2	200212	2020	2021-03-09	Avon	5 CHESTNUT DRIVE	130400.0	179900.0	0.7248	Residential	Condo	NaN	NaN	NaN	NaN
3	200243	2020	2021-04-13	Avon	111 NORTHINGTON DRIVE	619290.0	890000.0	0.6958	Residential	Single Family	NaN	NaN	NaN	NaN
4	200377	2020	2021-07-02	Avon	70 FAR HILLS DRIVE	862330.0	1447500.0	0.5957	Residential	Single Family	NaN	NaN	NaN	NaN

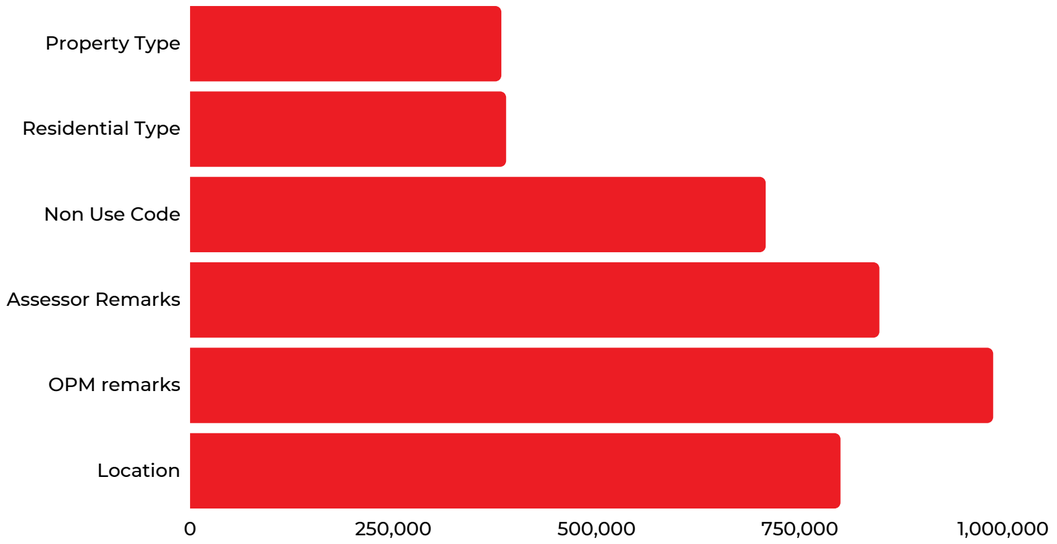
Data Wrangling

5 Steps Process



Examination Result

Missing Values:




```
Column: Property Type, Number of missing values: 382446
Column: Residential Type, Number of missing values: 388309
Column: Non Use Code, Number of missing values: 707532
Column: Assessor Remarks, Number of missing values: 847349
Column: OPM remarks, Number of missing values: 987279
Column: Location, Number of missing values: 799516
```

Intertwined Data

```
df[df['Property Type'] == 'Residential'][['Property Type', 'Residential Type']].head()
```

	Property Type	Residential Type
1	Residential	Single Family
2	Residential	Condo
3	Residential	Single Family
4	Residential	Single Family
5	Residential	Single Family

```
df[df['Property Type'] == 'Single Family'][['Property Type', 'Residential Type']].head()
```

	Property Type	Residential Type
67926	Single Family	Single Family
67927	Single Family	Single Family
380637	Single Family	Single Family
380640	Single Family	Single Family
380641	Single Family	Single Family

Handling inappropriate values

```
#Fixing Property Column
df.loc[df['Property Type'] == 'Residential', 'Property Type'] = df.loc[df['Property Type'] == 'Residential', 'Residential Type']
```

Before

```
df[['Property Type', 'Residential Type']].head()
```

	Property Type	Residential Type
0	Commercial	NaN
1	Residential	Single Family
2	Residential	Condo
3	Residential	Single Family
4	Residential	Single Family

After

```
df[['Property Type', 'Residential Type']].head()
```

	Property Type	Residential Type
0	Commercial	NaN
1	Single Family	Single Family
2	Condo	Condo
3	Single Family	Single Family
4	Single Family	Single Family

Remove outliers for numerical columns

Objective:

By identifying and removing outliers, we aim to improve the accuracy and reliability of subsequent data analysis or modeling tasks.

Method:

- The code uses the IQR method to detect outliers.
- Outliers are identified based on values outside the range of Q1 - 1.5 * IQR to Q3 + 1.5 * IQR.
- The IQR captures the middle 50% of the data and is less sensitive to extreme values.

- Rows with outlier values in the specified columns are filtered out.
- Outlier rows are stored separately in the outliers DataFrame.
- The original DataFrame df is updated to exclude the outlier rows.

Remove unnecessary columns/missing values

```
# Drop the columns from 'Residential Type' to 'Location'
df = df.drop(columns=df.columns[df.columns.get_loc('Residential Type'):df.columns.get_loc('Location')+1])
# Drop the Serial number
df = df.drop("Serial Number", axis=1)
# Drop the Address
df = df.drop("Address", axis=1)
# Drop the Recorded date
df = df.drop("Date Recorded", axis=1)
```

```
# Remove rows with missing values
data_clean = df.dropna()
```

Data Wrangling: Results

```
In [59]: data_clean.head()
```

Out[59]:

	List Year	Town	Assessed Value	Sale Amount	Sales Ratio	Property Type
0	2020	Ansonia	150500.0	325000.0	0.4630	Commercial
1	2020	Ashford	253000.0	430000.0	0.5883	Single Family
2	2020	Avon	130400.0	179900.0	0.7248	Condo
7	2020	Berlin	412000.0	677500.0	0.6081	Single Family
9	2020	Bethel	171360.0	335000.0	0.5115	Single Family

```
In [60]: data_clean.shape
```

```
Out[60]: (492842, 6)
```

Numerical Analysis

Data

	List Year	Assessed Value	Sale Amount	Sales Ratio
count	492842.000000	492842.000000	492842.000000	492842.000000
mean	2013.894516	167776.762758	255215.867511	0.685746
std	4.530202	84902.194562	130361.174181	0.187036
min	2006.000000	190.000000	2000.000000	0.038854
25%	2010.000000	107940.000000	160000.000000	0.577300
50%	2015.000000	151360.000000	230000.000000	0.670700
75%	2018.000000	214620.000000	329000.000000	0.782143
max	2020.000000	432500.000000	702500.000000	1.233120

	List Year	Assessed Value	Sale Amount	Sales Ratio
List Year	1.000000	-0.030959	0.021106	-0.093850
Assessed Value	-0.030959	1.000000	0.850676	0.172135
Sale Amount	0.021106	0.850676	1.000000	-0.296908
Sales Ratio	-0.093850	0.172135	-0.296908	1.000000

Insights

- Data prior to 2006 was not sufficient
- There are significant variations among Assessed Value/Sale Amount
- A strong positive relationship between the "Assessed Value" and "Sale Amount" columns

Categorical Analysis

Cross-Tabulation:

Town	***Unknown***	Andover	Ansonia	Ashford	Avon	Barkhamsted	\
Property Type							
Vacant Land	0	6	7	8	16	11	
Two Family	0	5	521	13	2	4	
Three Family	0	0	125	0	2	1	
Single Family	1	459	1690	599	2143	500	
Public Utility	0	0	0	0	0	0	
Industrial	0	0	1	0	1	0	
Four Family	0	0	28	0	0	0	
Condo	0	0	81	23	1329	0	
Commercial	0	0	6	2	3	1	
Apartments	0	0	2	0	0	0	

Town	Beacon Falls	Berlin	Bethany	Bethel	...	Willington	\
Property Type					...		
Vacant Land	5	20	9	10	...	15	
Two Family	13	94	2	134	...	8	
Three Family	2	9	0	10	...	1	
Single Family	734	2416	740	2055	...	578	
Public Utility	0	0	0	0	...	0	
Industrial	1	1	0	1	...	0	
Four Family	0	2	0	6	...	1	
Condo	242	650	3	985	...	52	
Commercial	0	14	0	11	...	1	
Apartments	0	0	0	0	...	0	

Town	Wilton	Winchester	Windham	Windsor	Windsor Locks	Wolcott	\
Property Type							
Vacant Land	17	11	11	13	0	14	
Two Family	4	111	186	95	79	21	
Three Family	0	38	103	13	7	1	
Single Family	629	1380	2310	3912	1747	2359	
Public Utility	0	0	0	0	0	0	
Industrial	0	1	0	0	4	1	
Four Family	0	26	12	3	1	1	
Condo	326	210	108	953	632	263	
Commercial	0	7	11	8	4	8	
Apartments	0	0	2	1	0	0	

Town	Woodbridge	Woodbury	Woodstock
Property Type			
Vacant Land	10	8	29
Two Family	35	10	9
Three Family	0	1	1
Single Family	1262	1113	1411
Public Utility	0	0	0
Industrial	1	0	0
Four Family	1	1	0
Condo	11	489	132
Commercial	0	1	4
Apartments	0	1	0

Chi-square test p-value:
0.0

Town	
Waterbury	13266
Bridgeport	13126
Stamford	11454
Norwalk	11202
West Hartford	11197
	...
Bridgewater	185
Canaan	178
Roxbury	176
Union	100
Unknown	1

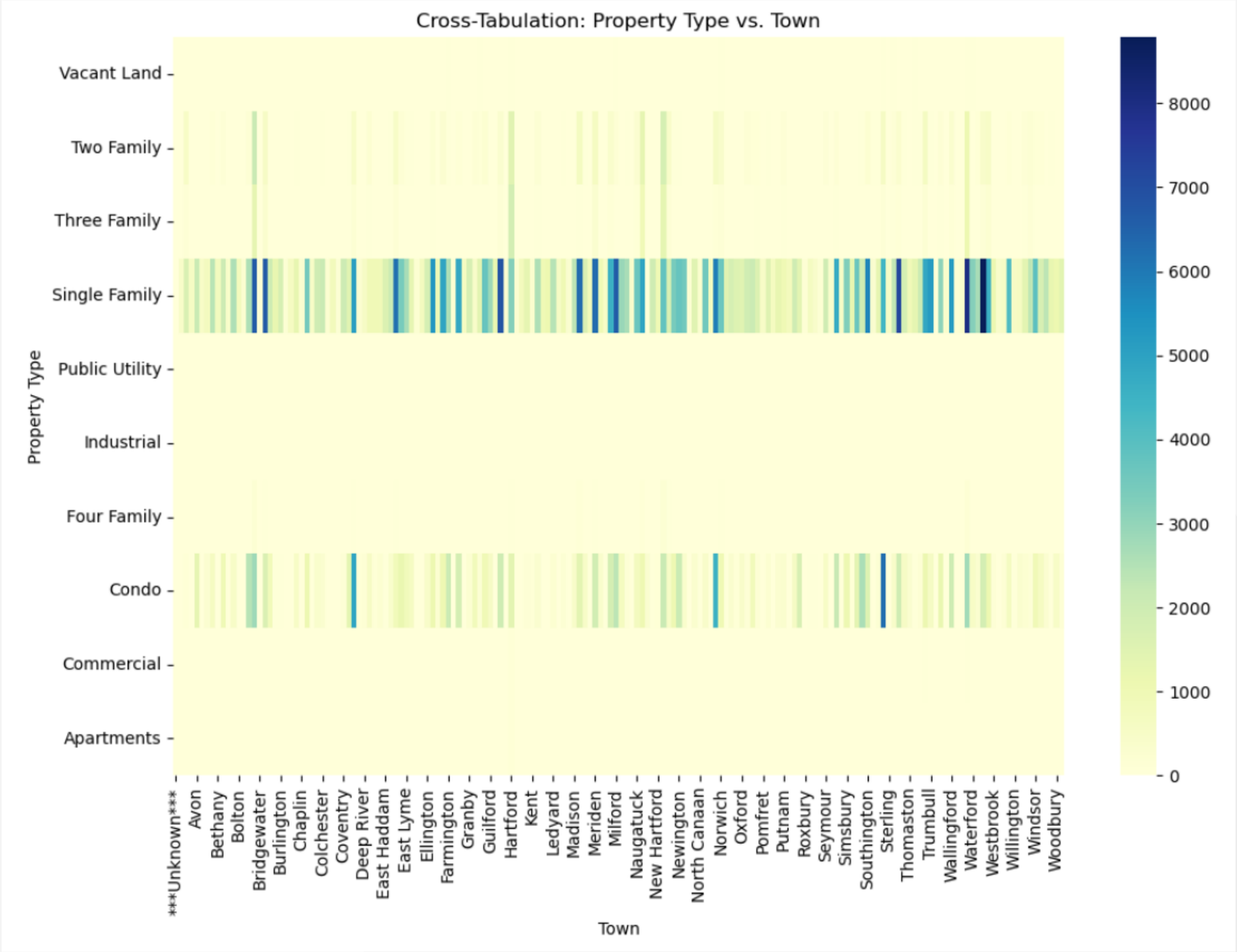
Name: count, Length: 170,

Property Type	
Single Family	350890
Condo	104389
Two Family	22107
Three Family	10307
Vacant Land	2151
Four Family	1681
Commercial	949
Apartments	279
Industrial	86
Public Utility	3

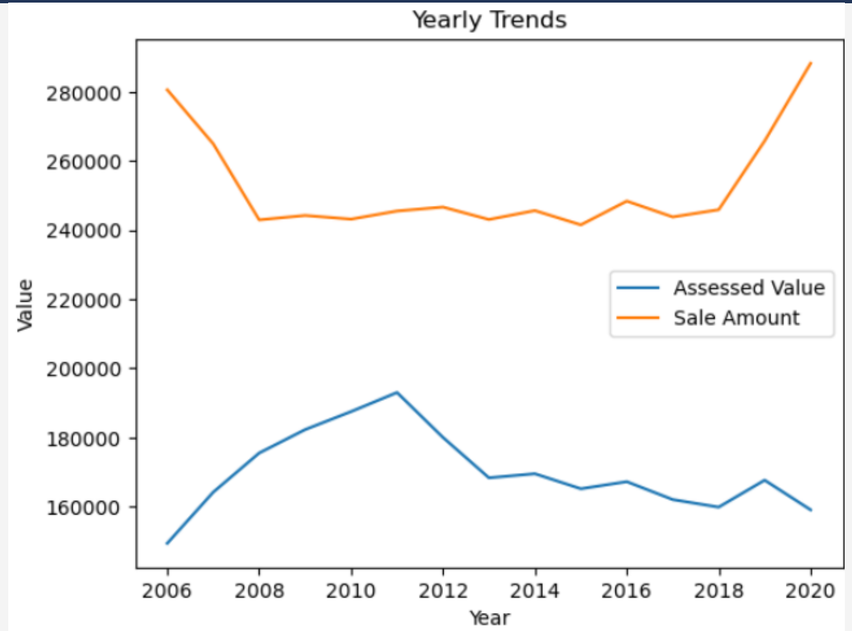
Insights

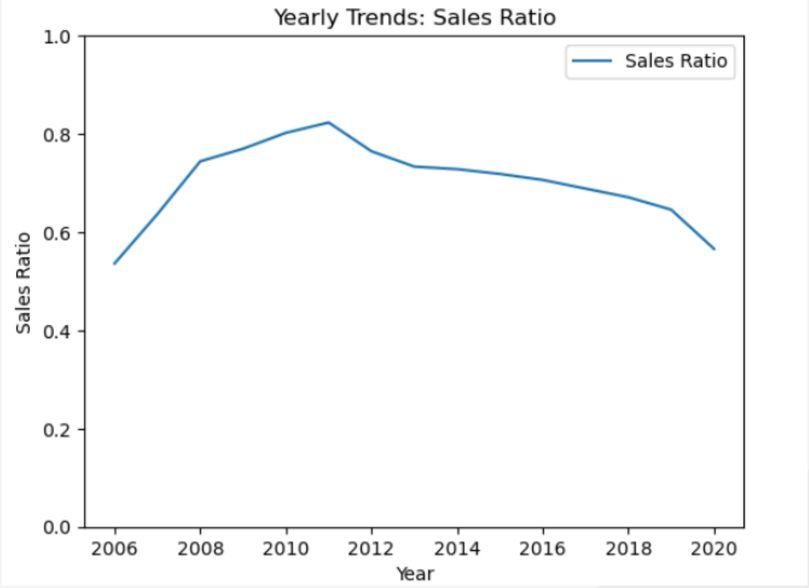
- Top five towns with the highest number of properties are Waterbury, Bridgeport, Stamford, Norwalk, and West Hartford.
- "Single Family" is the most common property type across all towns.
- Each town has a unique distribution of property types.
- A p-value of 0.0 suggests a non-random distribution of property types across towns.
- The clustering of specific property types in certain towns indicates underlying reasons or factors at play.

Categorical Visualizaiton



Numerical Visualizaiton





- Market Appreciation:
  - The fact that sale amounts consistently exceed the assessed values suggests that buyers are willing to pay higher prices for properties, indicating an overall upward trend in property values.
- Competitive Market:
  - When sale amounts consistently exceed assessed values, it suggests that buyers are willing to pay more than what the properties are officially valued at. This indicates a high level of competition among buyers.
- Seller's Advantage:
  - When sale prices consistently exceed assessed values, sellers have the opportunity to negotiate higher prices and potentially achieve better returns on their investments.
- Potential Underassessment:
  - Assessed values are typically used for tax purposes and may not always reflect the current market conditions accurately.
  - Reduced likelihood due to the sale ratio following a the same path as the Assessed value.

The Stages of Analysis with models

1 **PREPARING DATA**

```
#Create second data copy for later use
data_log = data_clean.copy()

#Removing Town column as this point due to no longer needed.
data_clean = data_clean.drop("Town", axis=1)
```

2 **HANDLING CATEGORICAL ATTRIBUTES**

- Town
- Property Type

3 **CREATING TRAINING/TESTING DATA**

4 **APPLYING MODELS**

- Linear Regression
- Decision Tree
- Logistics Regression

5 **CONCLUSION**

Interpret the results

Linear Regression

Create Training/Testing data

Applying Linear Regression Model for training and predicting

Regression coefficients / RMSE

Results:

	0
List Year	182.941033
Assessed Value	1.422931
Sales Ratio	-317714.089600
apartments	28778.128878
commercial	5862.398445
condo	-6549.406588
four family	5709.362123
industrial	18849.169524
public utility	-38260.568899
single family	-3036.640865
three family	-1622.003103
two family	-1033.732966
vacant land	-8696.706549

```
In Sample RMSE:  35325.44614118062
```

```
Out of Sample RMSE:  35410.47420528373
```

Decision Tree

Result:

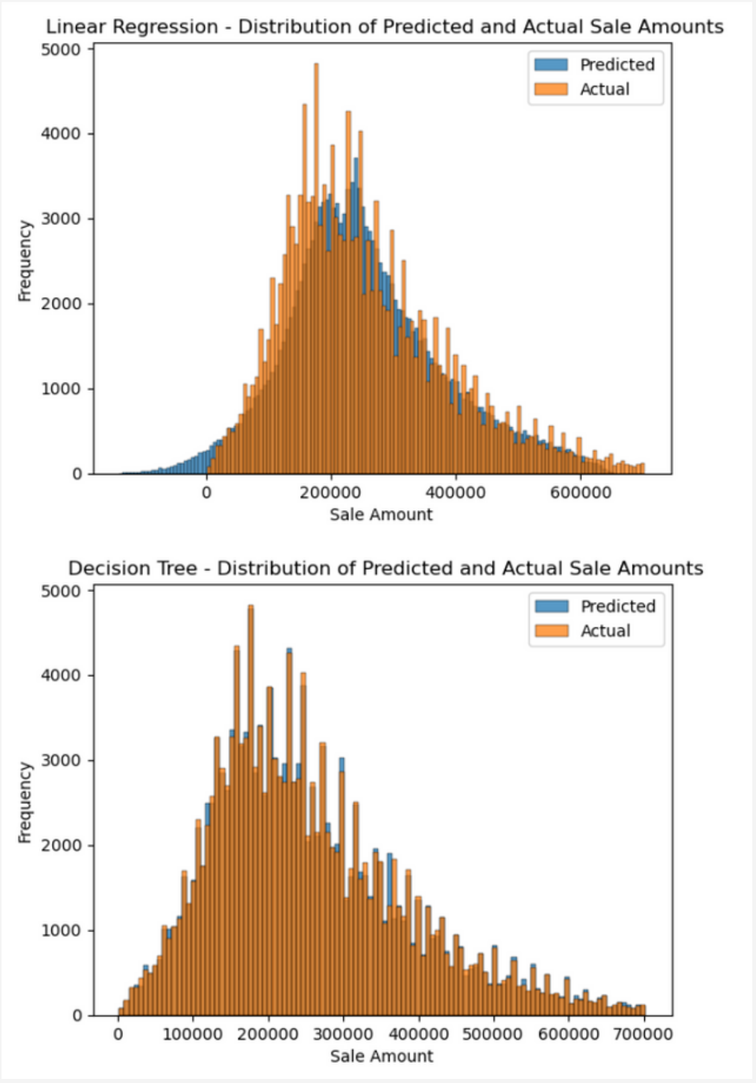
```
In Sample RMSE:  1.4924405340245084

Out of Sample RMSE:  1570.8234183711927
```

Interpretation:

Decision Tree’s RMSE suggests that it performs better on the training data compared to the linear regression model. The lower RMSE indicates that the decision tree model has a smaller average prediction error on the training data.

Comparing between Linear Regression model and Decision Tree model



Insights

- The decision tree model's predictions closely follow the actual data, indicating a better fit to the training data compared to the linear regression model.
- The observed performance suggests that the decision tree model may be a more suitable choice for capturing the complexities in the data and making predictions that closely align with the actual values.

Logistics Regression

Create Training/Testing data

Applying Logistics Regression Model for training and predicting

Generate classification report

This model use the data_log dataframe which still has the Town columns(encoded)

Results:

	precision	recall	f1-score	support
Non-Residential	0.00	0.00	0.00	986
Residential	0.99	1.00	1.00	146867
accuracy			0.99	147853
macro avg	0.50	0.50	0.50	147853
weighted avg	0.99	0.99	0.99	147853

Logistics Regression

Interpretation:

- The model accurately predicts Residential properties with high precision (0.99) and recall (1.00), indicating excellent performance for this class.
- It fails to identify any Non-Residential properties, with both precision and recall at 0.00, suggesting the model is not effective for classifying Non-Residential properties.
- The overall model accuracy is 0.99, primarily due to the dominance of Residential properties in the dataset, which skews performance metrics.
- The macro averages for precision, recall, and F1-score are all 0.50, highlighting the model's significant imbalance in classifying different property types.
- The significant imbalance between Residential (146,867 instances) and Non-Residential (986 instances) properties in the dataset likely contributes to the model's biased performance towards Residential properties.