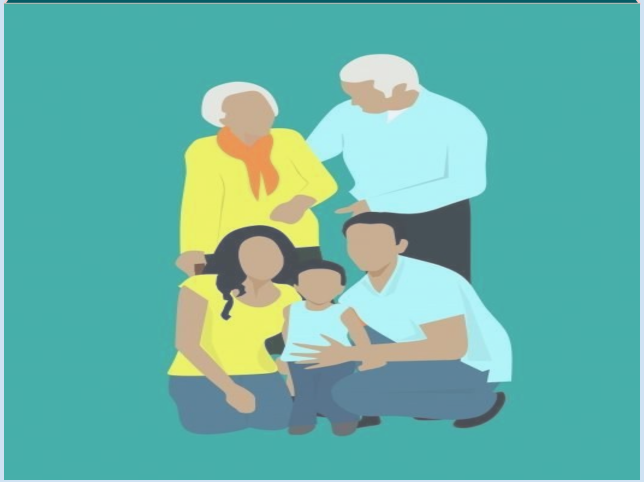
Home,
Home Alone



Common Real Estate Problems

Leveraging ML to Better Inform Real Estate Agents

Leveraging ML to Better Inform Potential Homeowners

Potential Impact

Potential Impact

- Home-Price Fit
- Increased Utility
- Faster Closing Process

Potential Impact

- Increased Satisfaction
- Targeted Customer Outreach
- Faster Closing Process
- Efficient Use of Inventory

Real Estate Agents

More Efficient Real Estate Market

Raw Dataset

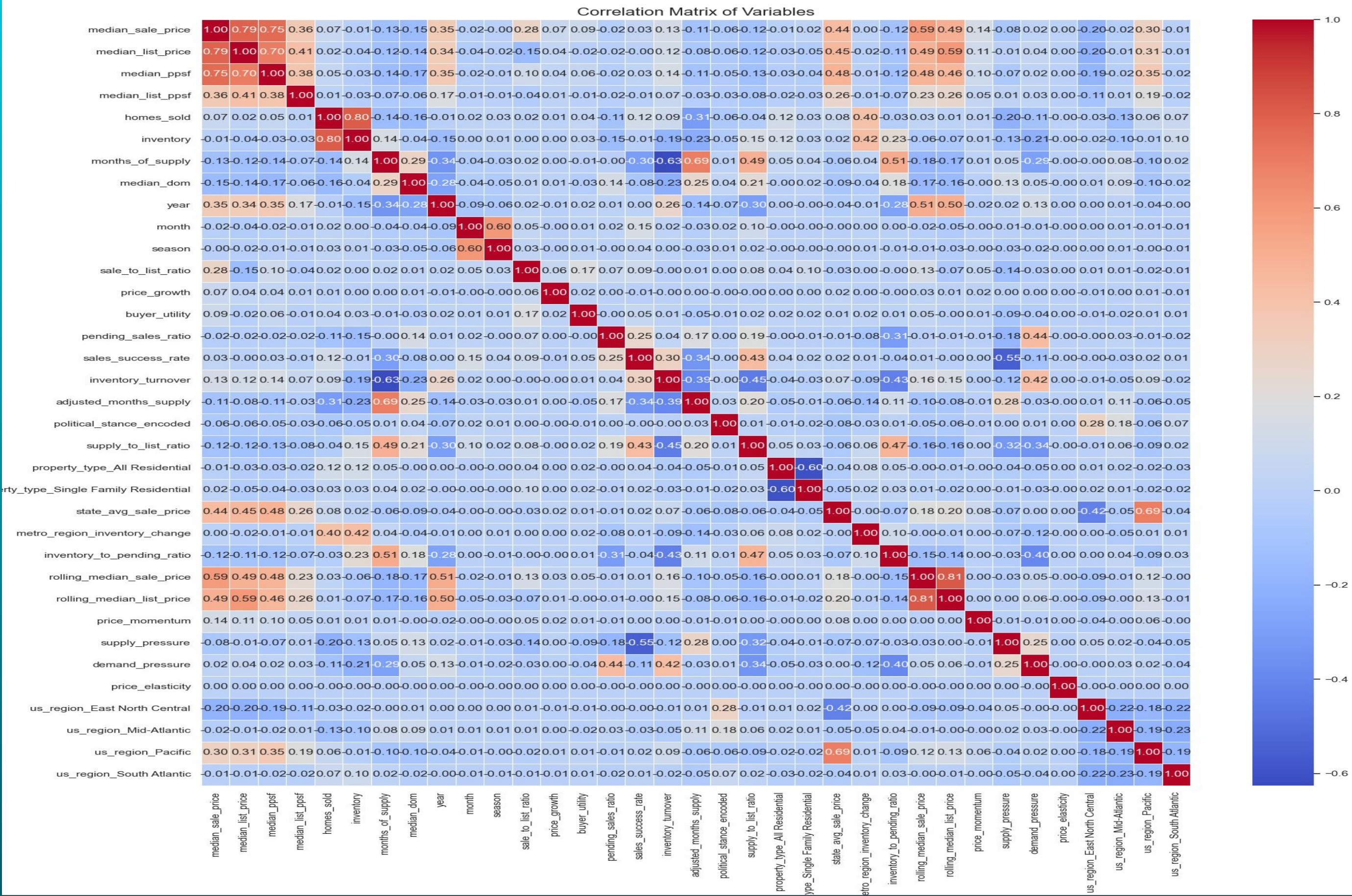
- 5,484,743 Rows, 58 Total Columns
- Kaggle data set, updated frequently
- 02/01/2012 - 05/31/2024
- Redfin
- 5,215,6152 null values
- https://www.senate.gov/senators/Senators1789toPresent.htm
- Data Dictionary

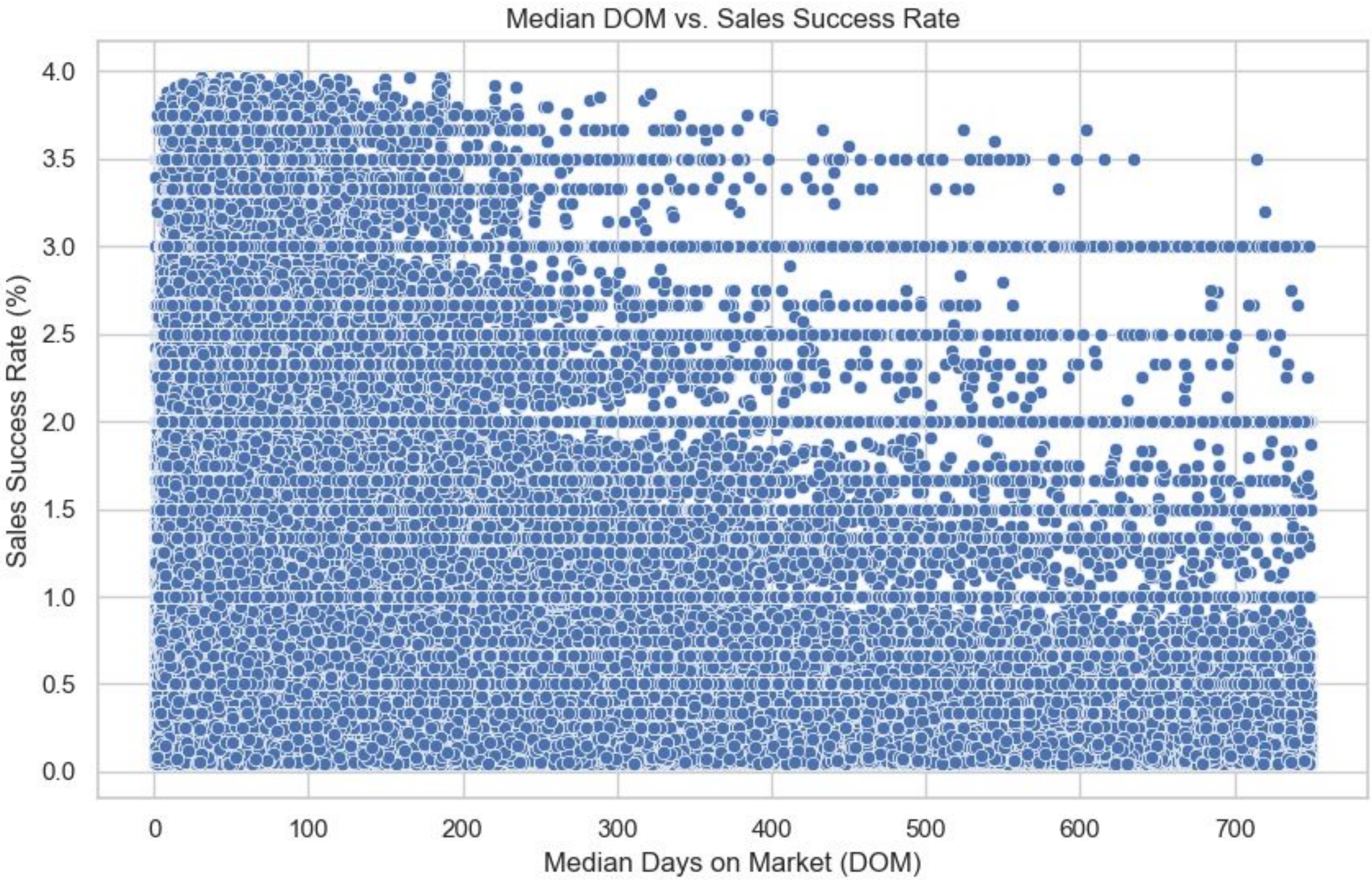
Processed Dataset

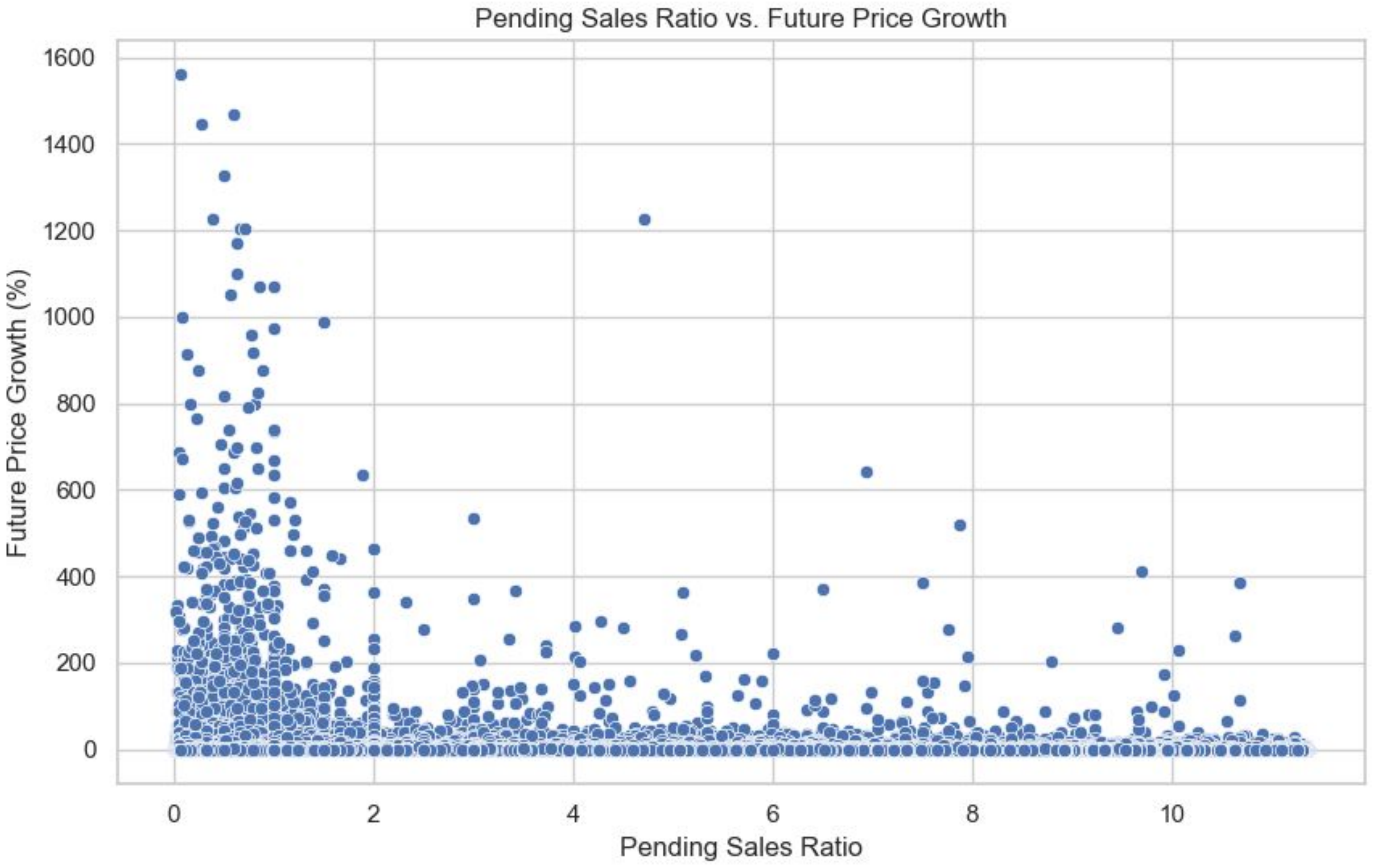
```
1   median_sale_price                        float64
2   median_list_price                        float64
3   median_ppsf                              float64
4   median_list_ppsf                         float64
5   homes_sold                               float64
6   inventory                                float64
7   months_of_supply                         float64
8   median_dom                               float64
9   year                                     float64
10  month                                    float64
11  season                                   float64
12  sale_to_list_ratio                       float64
13  price_growth                             float64
14  buyer_utility                            float64
15  pending_sales_ratio                      float64
16  sales_success_rate                       float64
17  inventory_turnover                       float64
18  adjusted_months_supply                   float64
19  political_stance_encoded                 float64
20  supply_to_list_ratio                     float64
21  property_type_All Residential            float64
22  property_type_Single Family Residential  float64
23  state_avg_sale_price                     float64
24  metro_region_inventory_change            float64
25  inventory_to_pending_ratio               float64
26  rolling_median_sale_price                float64
27  rolling_median_list_price                float64
28  price_momentum                           float64
29  supply_pressure                          float64
30  demand_pressure                          float64
31  price_elasticity                         float64
32  us_region_East North Central             float64
33  us_region_Mid-Atlantic                   float64
34  us_region_Pacific                        float64
35  us_region_South Atlantic                 float64
dtypes: float64(35), int64(1)
memory usage: 1.2 GB
```

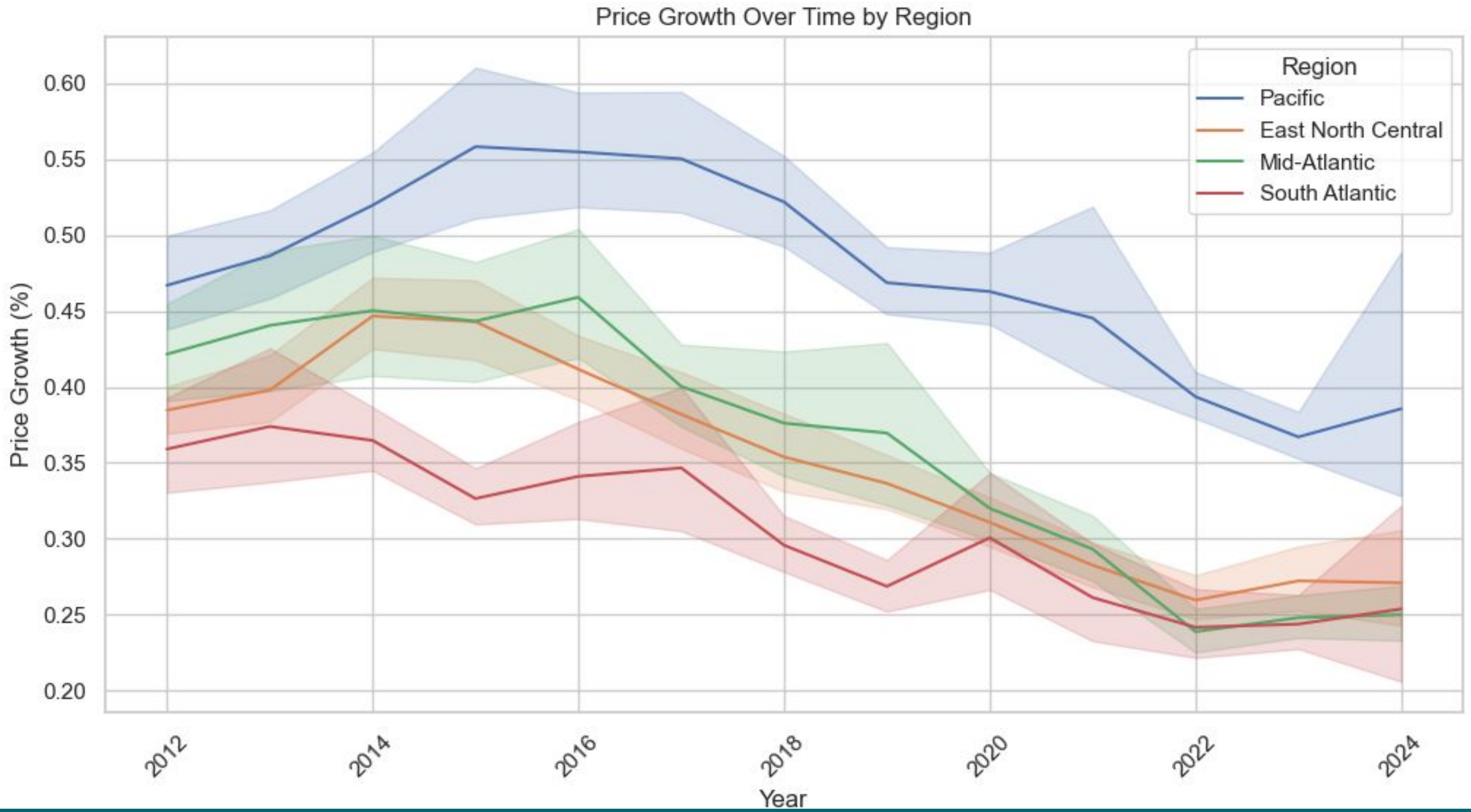
Preprocessing

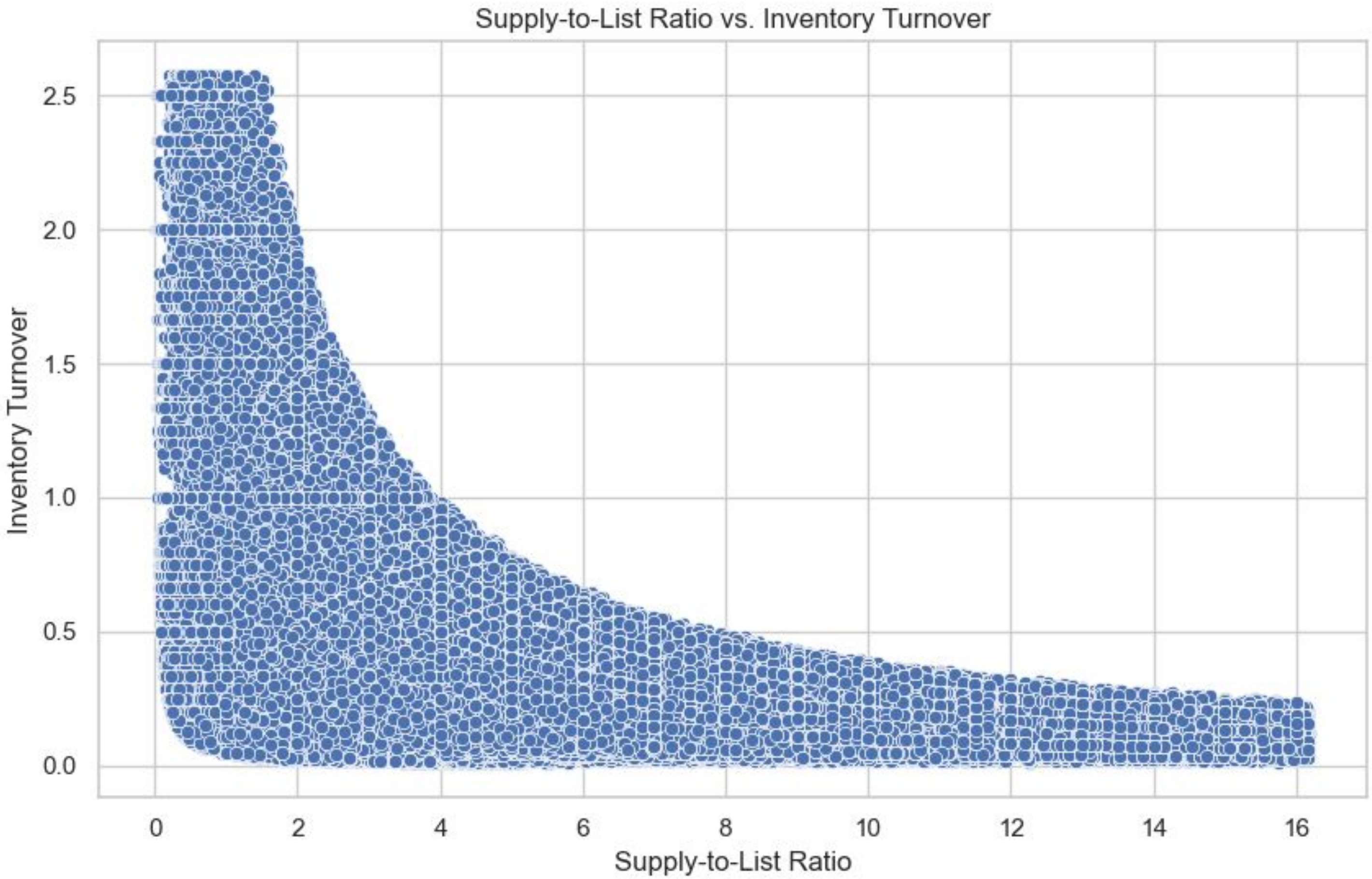
- Nulls
- Combining Features
- Duplicates
- Adjusted Prices
- Binning by Regions
- Outliers, Irrelevant Features
- Highly correlated features

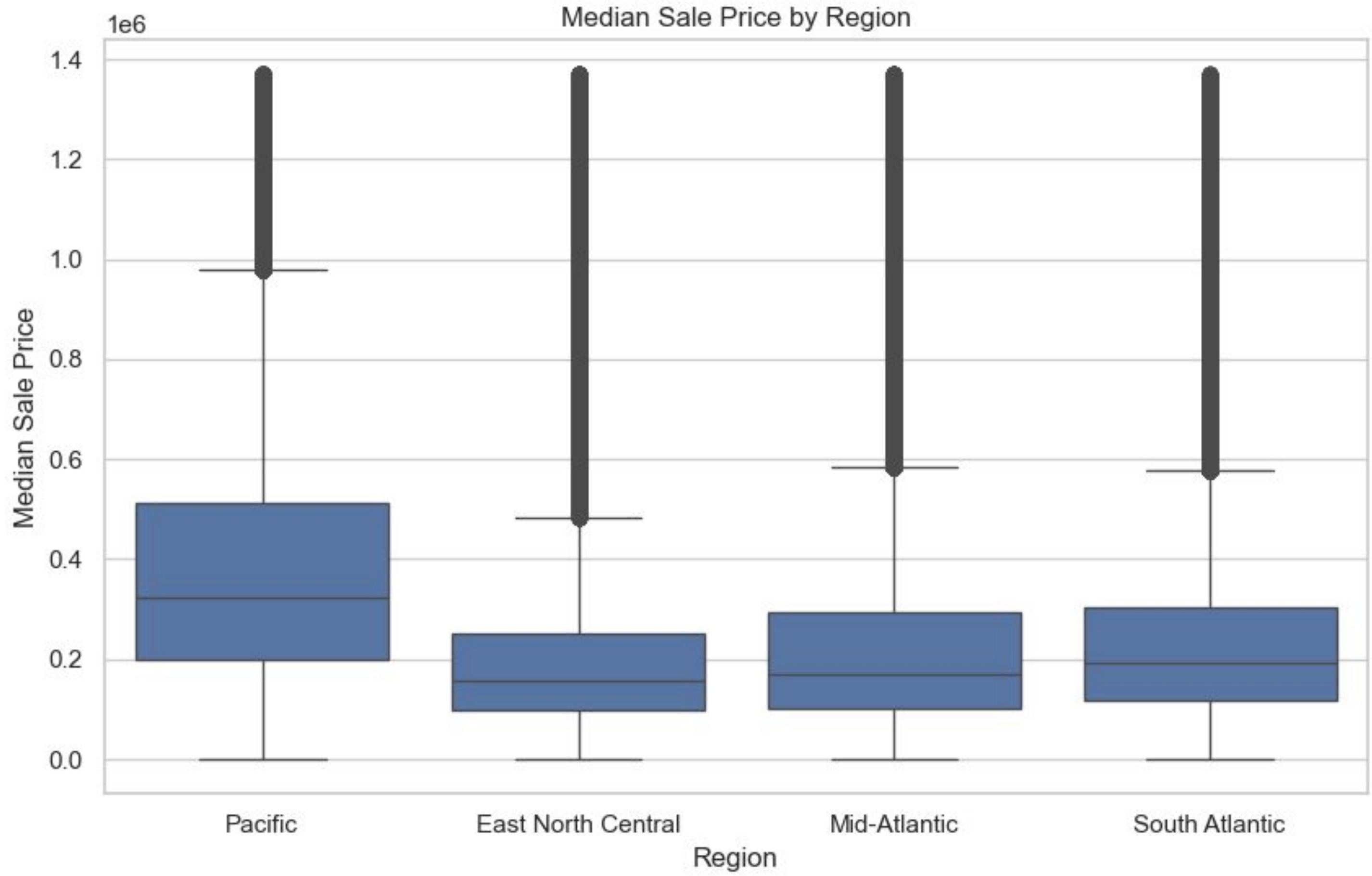

Correlation Matrix of Variables
median_sale_price
median_list_price
median_ppsf
median_list_ppsf
homes_sold
inventory
months_of_supply
median_dom
year
month
season
sale_to_list_ratio
price_growth
buyer_utility
pending_sales_ratio
sales_success_rate
inventory_turnover
adjusted_months_supply
political_stance_encoded
supply_to_list_ratio
property_type_All Residential
erty_type_Single Family Residential
state_avg_sale_price
metro_region_inventory_change
inventory_to_pending_ratio
rolling_median_sale_price
rolling_median_list_price
price_momentum
supply_pressure
demand_pressure
price_elasticity
us_region_East North Central
us_region_Mid-Atlantic
us_region_Pacific
us_region_South Atlantic
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Median DOM vs. Sales Success Rate
Sales Success Rate (%)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
700
Median Days on Market (DOM)

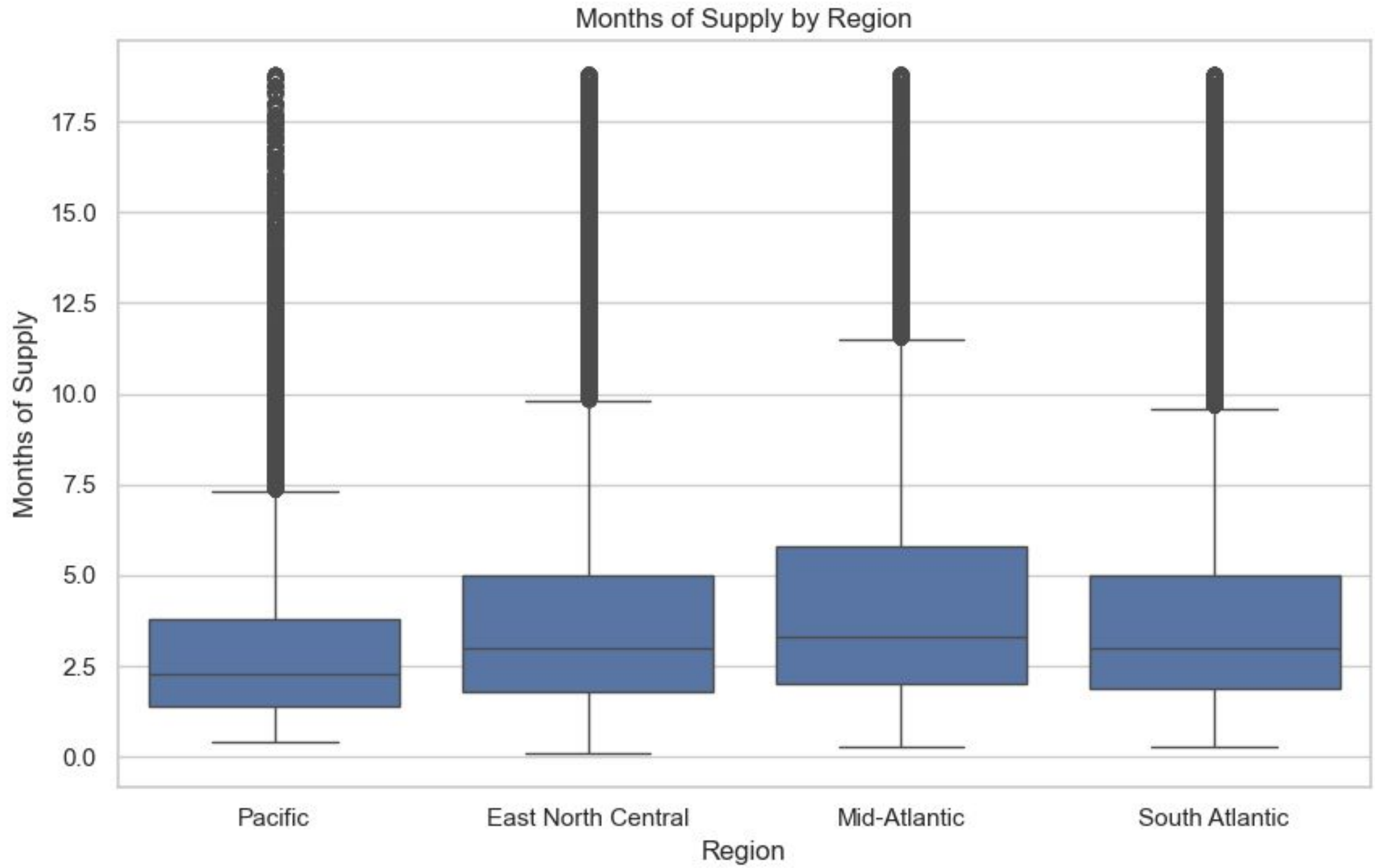

Pending Sales Ratio vs. Future Price Growth
Future Price Growth (%)
1600
1400
1200
1000
800
600
400
200
0
0
2
4
6
8
10
Pending Sales Ratio


Price Growth Over Time by Region
Region
Pacific
East North Central
Mid-Atlantic
South Atlantic
Price Growth (%)
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
2012
2014
2016
2018
2020
2022
2024
Year


Supply-to-List Ratio vs. Inventory Turnover
Inventory Turnover
2.5
2.0
1.5
1.0
0.5
0.0
0
2
4
6
8
10
12
14
16
Supply-to-List Ratio



Median Sale Price by Region
1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Median Sale Price
Pacific
East North Central
Mid-Atlantic
South Atlantic
Region


Months of Supply by Region
Months of Supply
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Pacific
East North Central
Mid-Atlantic
South Atlantic
Region

```python
# Feature selection - selecting columns that are likely to affect homes sold
features = ['median_sale_price', 'median_list_price', 'inventory', 'months_of_supply',
            'price_growth', 'buyer_utility', 'pending_sales_ratio', 'sales_success_rate',
            'inventory_turnover', 'adjusted_months_supply', 'supply_to_list_ratio',
            'property_type_All Residential', 'state_avg_sale_price', 'price_momentum',
            'supply_pressure', 'demand_pressure', 'price_elasticity']

X = df[features]  # Predictor variables
y = df['homes_sold']  # Target variable

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```python
# Initialize and train the linear regression model
lr_model = LinearRegression()
lr_model.fit(X_train_scaled, y_train)

# Predict homes sold using the test set
y_pred_lr = lr_model.predict(X_test_scaled)

# Evaluate the model
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)

print(f'Linear Regression - MSE: {mse_lr}, R-squared: {r2_lr}')
```

```
Linear Regression - MSE: 100.73826222540995, R-squared: 0.755139756428168
```

```python
X = df.drop(columns=['median_sale_price', 'Unnamed: 0'])  # Predictor variables
y = df['median_sale_price']  # Target variable

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```python
# Initialize the Linear Regression model
lr_model = LinearRegression()

# Fit the model to the training data
lr_model.fit(X_train, y_train)

# Predict buyer utility on the test data
y_pred = lr_model.predict(X_test)
```

```python
# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Calculate R-squared (R²)
r2 = r2_score(y_test, y_pred)
print(f'R-squared: {r2}')
```

```
Mean Squared Error: 5983453454.910419
R-squared: 0.847707192781659
```

- Increase accuracy of models
- Random Forest?
- KNN clustering?


THANK YOU
FOR WATCHING