# Location Based Home Rent Value Prediction in Ontario

# with Machine Learning Algorithm

Sara Seyda Yenigun

501311857

Big Data Analytics and PA

Toronto Metropolitan University

May 2025

# Introduction

This capstone explores a live Ontario rental dataset consisting of 2,850 listings from Realtor.com. It has features such as property type, building size, rooms, price, city, and crime rate, among others. The procedure involves intense data cleaning, advanced feature engineering, and predictive modeling to learn about the association of these features with rent prices. Various visualizations (box plots, heatmaps, bar charts) were used to explore distributions and relationships, yielding insights into feature importance and model performance. New features in this project include using additional data (e.g., neighborhood crime), geospatial data, time-series dynamics, and explainability techniques (SHAP/LIME). For example, advanced feature engineering incorporates additional data sources, while temporal analysis considers seasonal price variations. The identical house rental prediction projects were described on the web (i.e., Kaggle analysis and machine learning tutorials), and crime data rely on resources like the Canadian Crime Report's Crime Severity Index.

This study mainly answers these primary questions:

- How do building size, bedrooms, and bathrooms affect rent?

- Does crime rate an influence that affects rental prices?

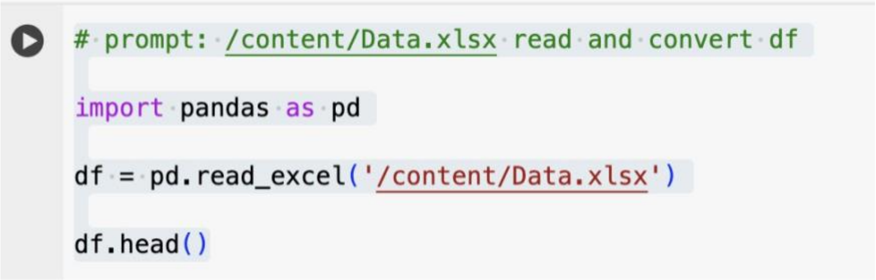
- Is it possible for a prediction model to project rent with high accuracy based on these variables?

# Methodology

The project approach was systematic, covering data preparation, exploration, and modeling. Key steps included:

## Data Collection

The dataset was provided in an Excel file and loaded into a Pandas Data Frame for analysis.

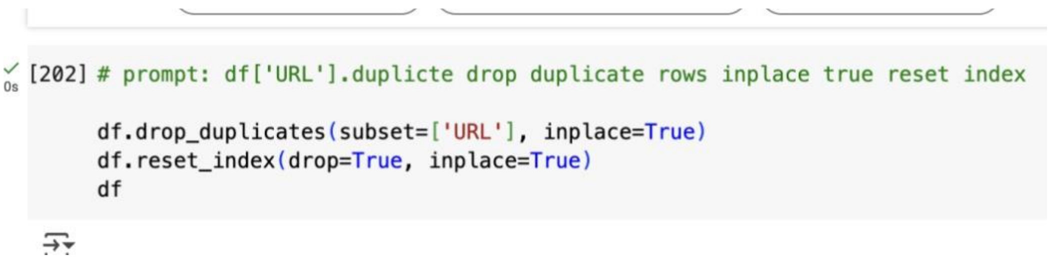


```
# prompt: /content/Data.xlsx read and convert df

import pandas as pd

df = pd.read_excel('/content/Data.xlsx')

df.head()
```

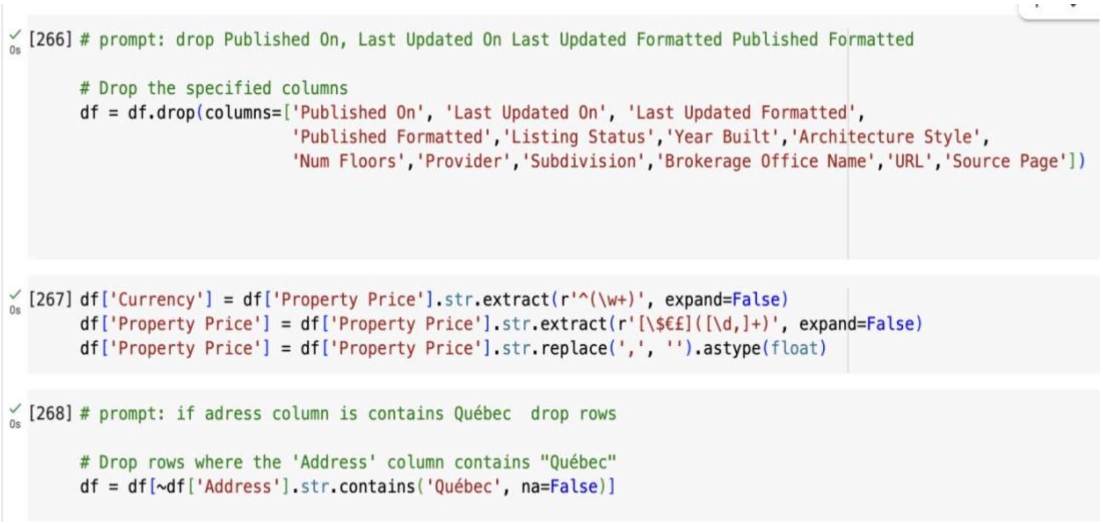
## Data Cleaning

Duplicates were removed based on the URL column. Date strings were standardized (removing irregular characters, mapping month names) and converted to datetime format. Irrelevant columns (e.g., raw date strings, miscellaneous text fields, Quebec addresses) were dropped. Numeric fields (currency, building size) were cleaned by extracting digits and converting to float. Missing values were handled by removing any remaining null rows and, where appropriate, imputing missing values using models (e.g., a Random Forest to predict missing building sizes and bedroom/bathroom counts from related features).

***Figure 1***: Python code used to convert raw date strings (e.g., “Published On”) into standardized datetime format, dropping irrelevant columns (e.g., 'Published On', 'URL'), extracting numeric currency from the price field, and removing Quebec addresses to clean the data and cleaning the “Building Size” column (removing commas, “sq ft”, etc.) and converting it to a float.



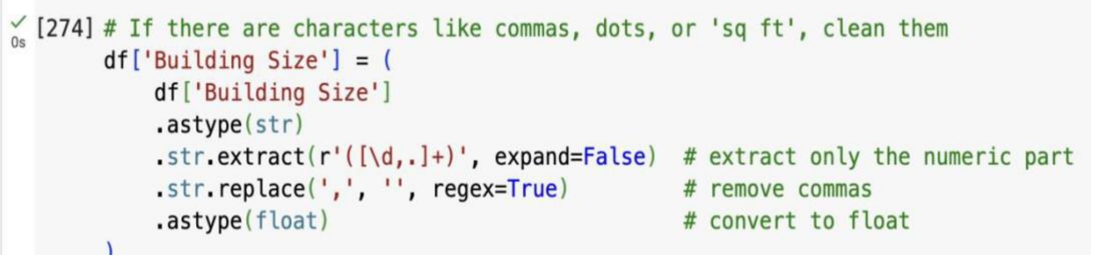
```
[202] # prompt: df['URL'].duplicte drop duplicate rows inplace true reset index

      df.drop_duplicates(subset=['URL'], inplace=True)
      df.reset_index(drop=True, inplace=True)
      df
```



```
[266] # prompt: drop Published On, Last Updated On Last Updated Formatted Published Formatted

      # Drop the specified columns
      df = df.drop(columns=['Published On', 'Last Updated On', 'Last Updated Formatted',
                            'Published Formatted','Listing Status','Year Built','Architecture Style',
                            'Num Floors','Provider','Subdivision','Brokerage Office Name','URL','Source Page'])
```

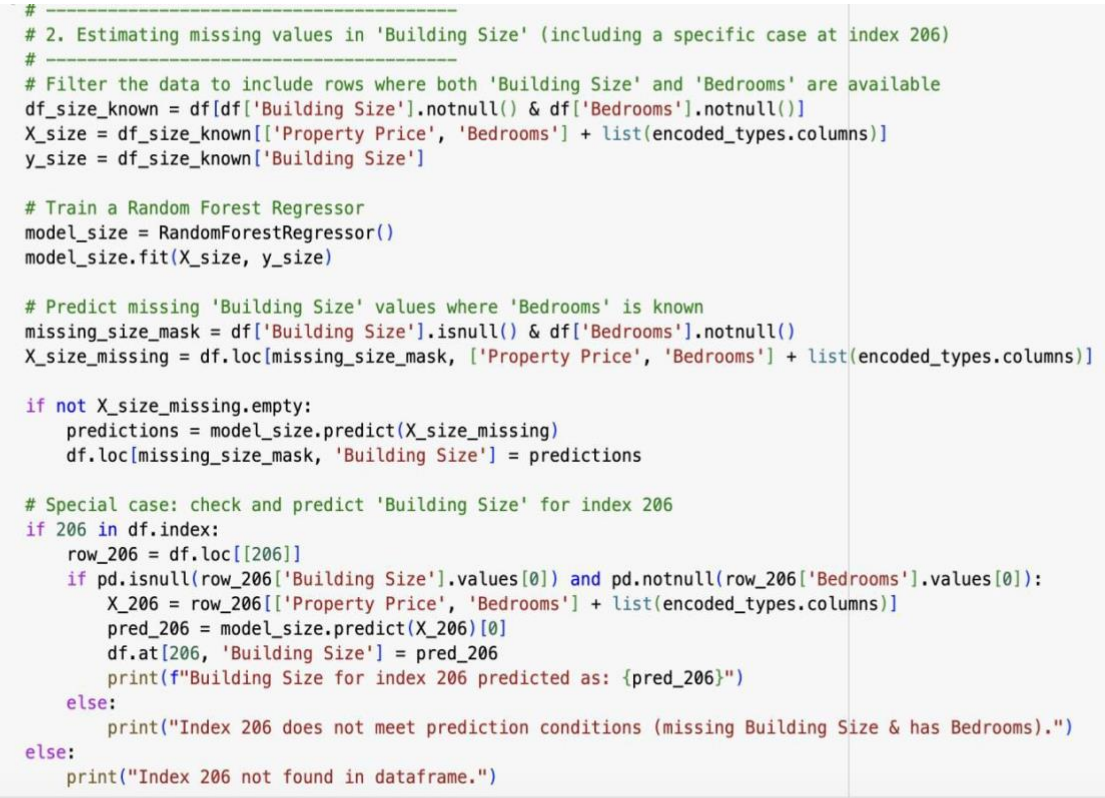
```
[267] df['Currency'] = df['Property Price'].str.extract(r'^(\w+)', expand=False)
      df['Property Price'] = df['Property Price'].str.extract(r'[\$€£]([\d,]+)', expand=False)
      df['Property Price'] = df['Property Price'].str.replace(',', '').astype(float)
```

```
[268] # prompt: if adress column is contains Québec  drop rows

      # Drop rows where the 'Address' column contains "Québec"
      df = df[~df['Address'].str.contains('Québec', na=False)]
```

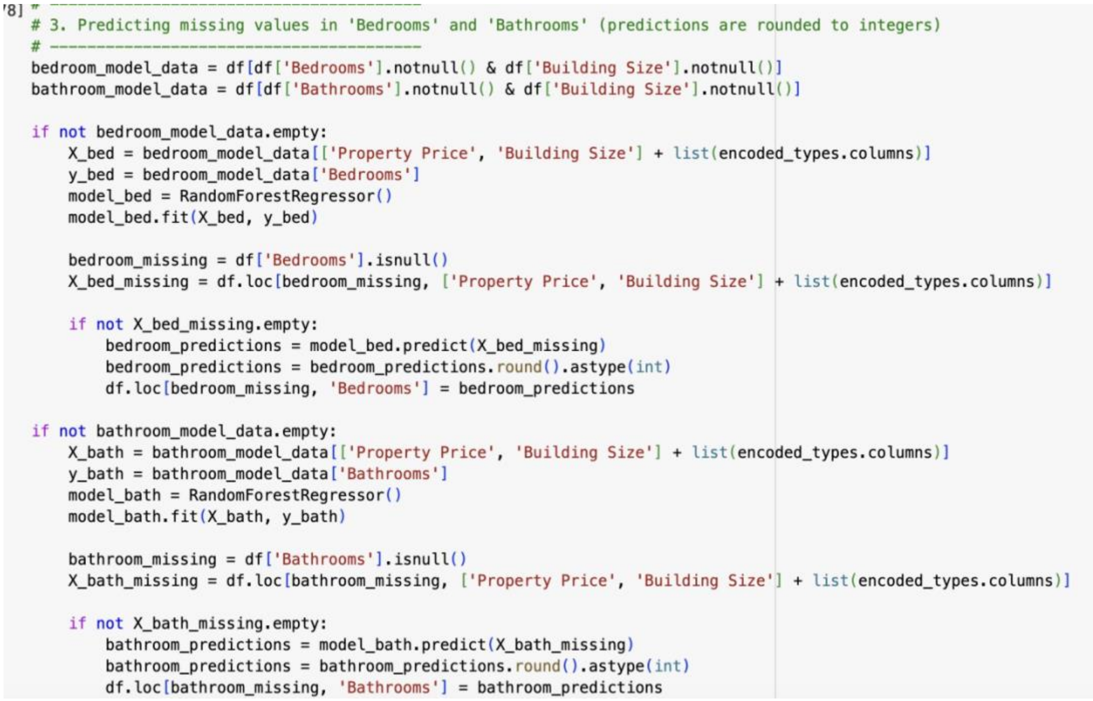


```
[274] # If there are characters like commas, dots, or 'sq ft', clean them
      df['Building Size'] = (
          df['Building Size']
          .astype(str)
          .str.extract(r'([\d,.]+)', expand=False)  # extract only the numeric part
          .str.replace(',', '', regex=True)         # remove commas
          .astype(float)                            # convert to float
      )
```

***Figure 2***: Python code using a Random Forest regressor to impute missing ‘Building Size’ values (focusing on index 206) and predicting missing ‘Bedrooms’ and ‘Bathrooms’ using Random Forest regression, followed by dropping any remaining null rows and resetting the index.



```
# ----------------------------------------
# 2. Estimating missing values in 'Building Size' (including a specific case at index 206)
# ----------------------------------------
# Filter the data to include rows where both 'Building Size' and 'Bedrooms' are available
df_size_known = df[df['Building Size'].notnull() & df['Bedrooms'].notnull()]
X_size = df_size_known[['Property Price', 'Bedrooms'] + list(encoded_types.columns)]
y_size = df_size_known['Building Size']

# Train a Random Forest Regressor
model_size = RandomForestRegressor()
model_size.fit(X_size, y_size)

# Predict missing 'Building Size' values where 'Bedrooms' is known
missing_size_mask = df['Building Size'].isnull() & df['Bedrooms'].notnull()
X_size_missing = df.loc[missing_size_mask, ['Property Price', 'Bedrooms'] + list(encoded_types.columns)]

if not X_size_missing.empty:
    predictions = model_size.predict(X_size_missing)
    df.loc[missing_size_mask, 'Building Size'] = predictions

# Special case: check and predict 'Building Size' for index 206
if 206 in df.index:
    row_206 = df.loc[[206]]
    if pd.isnull(row_206['Building Size'].values[0]) and pd.notnull(row_206['Bedrooms'].values[0]):
        X_206 = row_206[['Property Price', 'Bedrooms'] + list(encoded_types.columns)]
        pred_206 = model_size.predict(X_206)[0]
        df.at[206, 'Building Size'] = pred_206
        print(f"Building Size for index 206 predicted as: {pred_206}")
    else:
        print("Index 206 does not meet prediction conditions (missing Building Size & has Bedrooms).")
else:
    print("Index 206 not found in dataframe.")
```

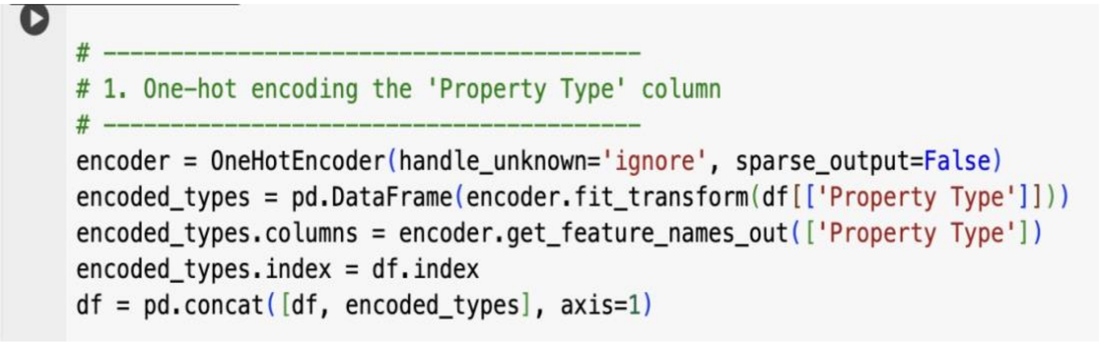


```
'8] # -----------------------------------------
    # 3. Predicting missing values in 'Bedrooms' and 'Bathrooms' (predictions are rounded to integers)
    # -----------------------------------------
    bedroom_model_data = df[df['Bedrooms'].notnull() & df['Building Size'].notnull()]
    bathroom_model_data = df[df['Bathrooms'].notnull() & df['Building Size'].notnull()]

    if not bedroom_model_data.empty:
        X_bed = bedroom_model_data[['Property Price', 'Building Size'] + list(encoded_types.columns)]
        y_bed = bedroom_model_data['Bedrooms']
        model_bed = RandomForestRegressor()
        model_bed.fit(X_bed, y_bed)

        bedroom_missing = df['Bedrooms'].isnull()
        X_bed_missing = df.loc[bedroom_missing, ['Property Price', 'Building Size'] + list(encoded_types.columns)]

        if not X_bed_missing.empty:
            bedroom_predictions = model_bed.predict(X_bed_missing)
            bedroom_predictions = bedroom_predictions.round().astype(int)
            df.loc[bedroom_missing, 'Bedrooms'] = bedroom_predictions

    if not bathroom_model_data.empty:
        X_bath = bathroom_model_data[['Property Price', 'Building Size'] + list(encoded_types.columns)]
        y_bath = bathroom_model_data['Bathrooms']
        model_bath = RandomForestRegressor()
        model_bath.fit(X_bath, y_bath)

        bathroom_missing = df['Bathrooms'].isnull()
        X_bath_missing = df.loc[bathroom_missing, ['Property Price', 'Building Size'] + list(encoded_types.columns)]

        if not X_bath_missing.empty:
            bathroom_predictions = model_bath.predict(X_bath_missing)
            bathroom_predictions = bathroom_predictions.round().astype(int)
            df.loc[bathroom_missing, 'Bathrooms'] = bathroom_predictions
```

## Feature Engineering

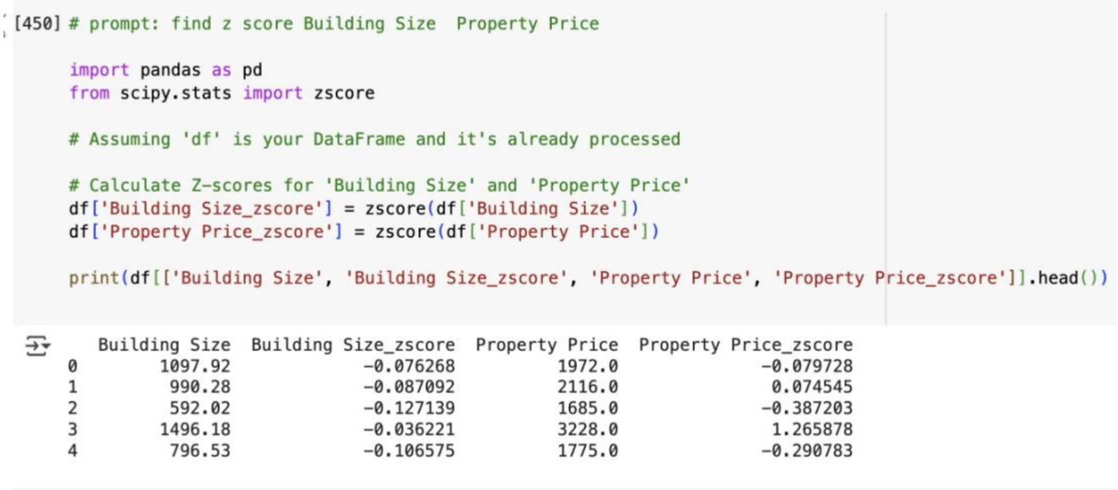
Categorical variables like “Property Type” were one-hot encoded to numeric columns. Z-scores were computed for features like Building Size and Property Price to identify outliers and normalize data. These engineered features prepared the data for effective analysis and modeling.

***Figure 3***: *One-Hot Encoding:* Categorical variables like 'Property Type' were transformed into numerical format through one-hot encoding, preparing them for use in machine learning models.



```
# ----------------------------------------
# 1. One-hot encoding the 'Property Type' column
# ----------------------------------------
encoder = OneHotEncoder(handle_unknown='ignore', sparse_output=False)
encoded_types = pd.DataFrame(encoder.fit_transform(df[['Property Type']]))
encoded_types.columns = encoder.get_feature_names_out(['Property Type'])
encoded_types.index = df.index
df = pd.concat([df, encoded_types], axis=1)
```

***Figure 4:*** *Z-Score Calculation:* Z-scores for 'Property Price' and 'Building Size' were computed to find and handle outliers effectively.



```
[450] # prompt: find z score Building Size  Property Price

      import pandas as pd
      from scipy.stats import zscore

      # Assuming 'df' is your DataFrame and it's already processed

      # Calculate Z-scores for 'Building Size' and 'Property Price'
      df['Building Size_zscore'] = zscore(df['Building Size'])
      df['Property Price_zscore'] = zscore(df['Property Price'])

      print(df[['Building Size', 'Building Size_zscore', 'Property Price', 'Property Price_zscore']].head())
```

```
   Building Size  Building Size_zscore  Property Price  Property Price_zscore
0        1097.92             -0.076268          1972.0              -0.079728
1         990.28             -0.087092          2116.0               0.074545
2         592.02             -0.127139          1685.0              -0.387203
3        1496.18             -0.036221          3228.0               1.265878
4         796.53             -0.106575          1775.0              -0.290783
```

Both "Building Size" and "Property Price" Zscores are largely centered at 0, which means that these values are near the mean of the dataset. For instance, the third one has a higher "Property Price" Z-score (1.265878), which implies that it is significantly above the average property price. Conversely, the second entry is lower in "Property Price" Z-score (-0.387203), which is less than average property price.

***Figure 5:*** Box plot of Property Price after cleaning. Most rents cluster around the median (~\$2000), with many high-end outliers.

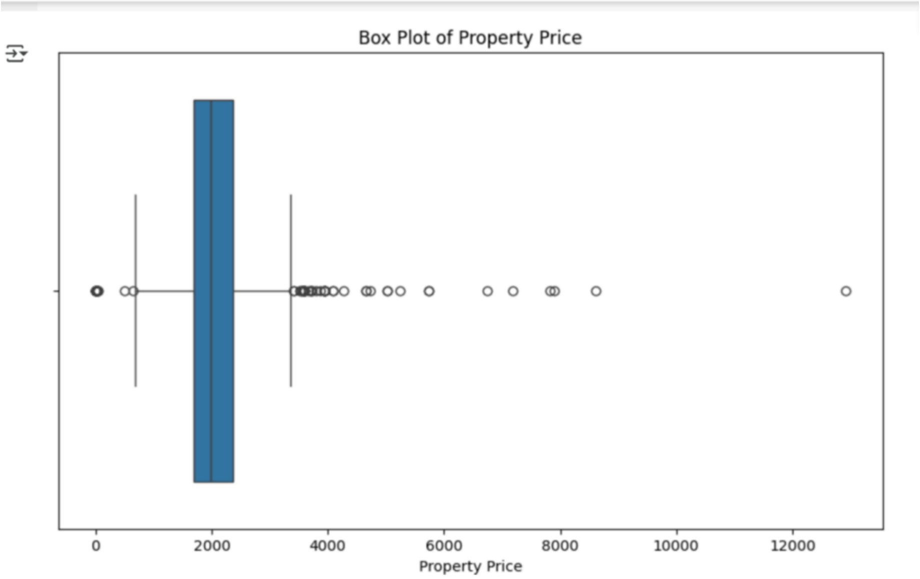
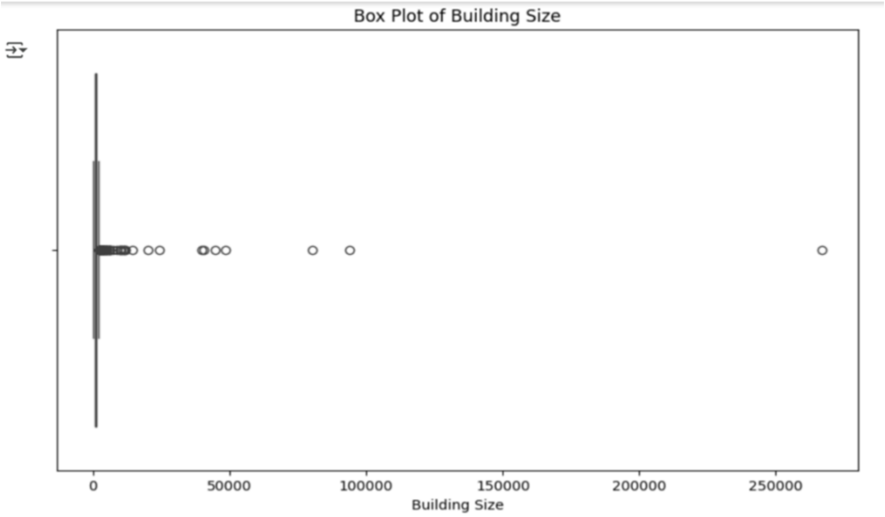


***Figure 6:*** Box plot of Building Size after cleaning, showing many small units (near 0–2000 sqft) and numerous large outliers.

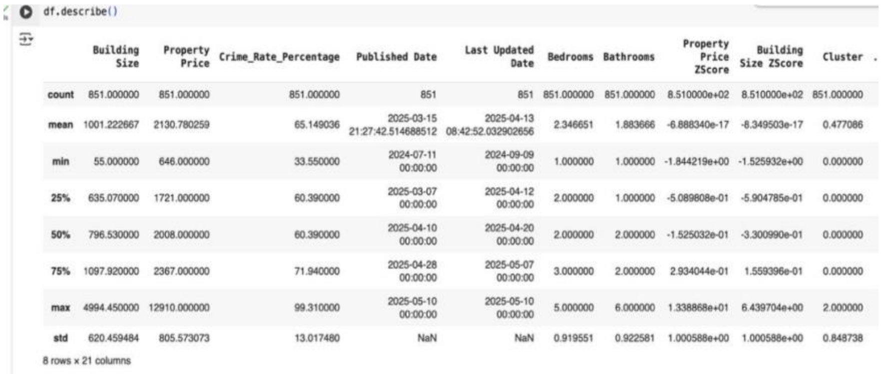


*As seen Figure 5 and 6* appears to be distributed quite normally for both building sizes and property prices, with most of the values clustering around the mean.
There is some variation, but no extreme-value outliers can be observed in the sample provided.

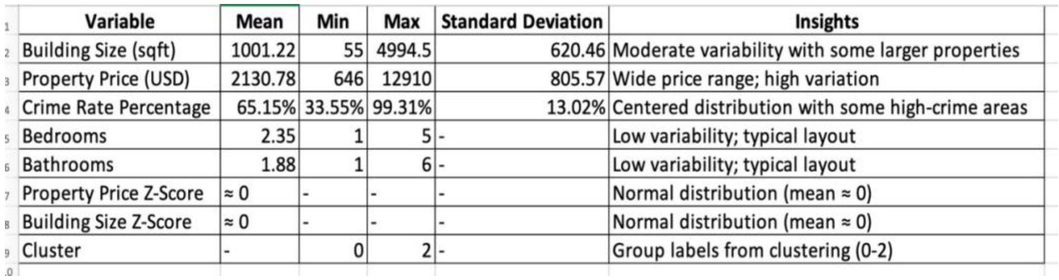
## Exploratory Data Analysis (EDA)

Summary statistics (e.g., count, mean, min, max) showed moderate variability in building sizes (mean ≈ 1001 sqft, max ~4994 sqft) and a wide rent range (mean ~\$2130, max \$12,910). Box plots highlighted above many outliers in property prices and building sizes after cleaning.

***Figure 7:*** Summary statistics were generated to understand information about the distribution of key features.



```
df.describe()

       Building    Property  Crime_Rate_Percentage  Published Date         Last Updated  Bedrooms    Bathrooms   Property      Building       Cluster  .
           Size       Price                                                        Date                            Price    Size ZScore
                                                                                                                   ZScore
count  851.000000   851.000000             851.000000             851                  851  851.000000  851.000000  8.510000e+02  8.510000e+02  851.000000
mean   1001.222667  2130.780259             65.149036         2025-03-15          2025-04-13    2.346651    1.883666  -6.888340e-17  -8.349503e-17    0.477086
                                                      21:27:42.514688512  08:42:52.032902656
min    55.000000    646.000000             33.550000          2024-07-11          2024-09-09    1.000000    1.000000  -1.844219e+00  -1.525932e+00    0.000000
                                                               00:00:00            00:00:00
25%    635.070000   1721.000000            60.390000          2025-03-07          2025-04-12    2.000000    1.000000  -5.089808e-01  -5.904785e-01    0.000000
                                                               00:00:00            00:00:00
50%    796.530000   2008.000000            60.390000          2025-04-10          2025-04-20    2.000000    2.000000  -1.525032e-01  -3.300990e-01    0.000000
                                                               00:00:00            00:00:00
75%    1097.920000  2367.000000            71.940000          2025-04-28          2025-05-07    3.000000    2.000000   2.934044e-01   1.559396e-01    0.000000
                                                               00:00:00            00:00:00
max    4994.450000  12910.000000           99.310000          2025-05-10          2025-05-10    5.000000    6.000000   1.338868e+01   6.439704e+00    2.000000
                                                               00:00:00            00:00:00
std    620.459484   805.573073             13.017480                 NaN                 NaN    0.919551    0.922581   1.000588e+00   1.000588e+00    0.848738
8 rows × 21 columns
```



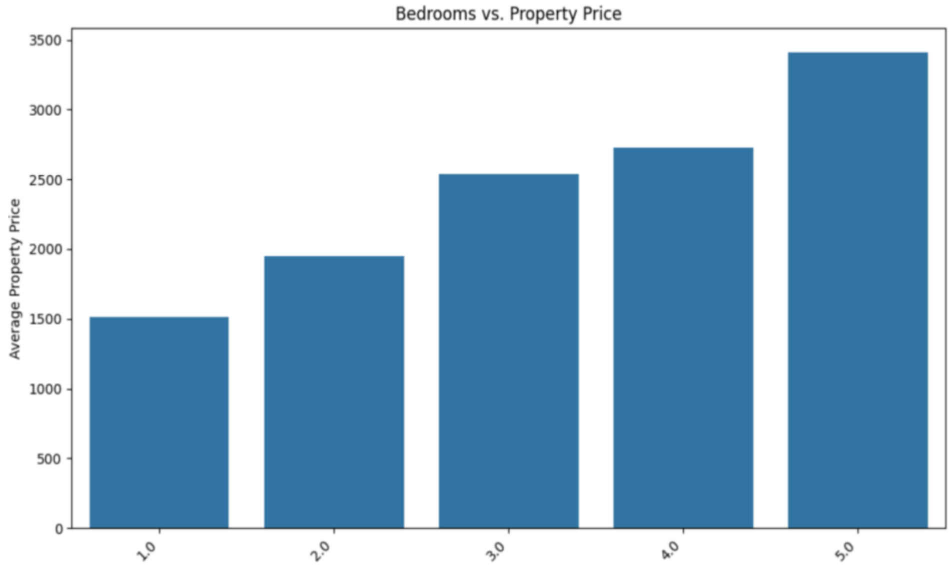
| Variable | Mean | Min | Max | Standard Deviation | Insights |
|---|---|---|---|---|---|
| Building Size (sqft) | 1001.22 | 55 | 4994.5 | 620.46 | Moderate variability with some larger properties |
| Property Price (USD) | 2130.78 | 646 | 12910 | 805.57 | Wide price range; high variation |
| Crime Rate Percentage | 65.15% | 33.55% | 99.31% | 13.02% | Centered distribution with some high-crime areas |
| Bedrooms | 2.35 | 1 | 5 | - | Low variability; typical layout |
| Bathrooms | 1.88 | 1 | 6 | - | Low variability; typical layout |
| Property Price Z-Score | ≈ 0 | - | - | - | Normal distribution (mean ≈ 0) |
| Building Size Z-Score | ≈ 0 | - | - | - | Normal distribution (mean ≈ 0) |
| Cluster | - | 0 | 2 | - | Group labels from clustering (0-2) |

The statistics reveal a variation in building size and cost, ranging from outliers to both high and low. The moderate standard deviation proves a mix of average and deviant buildings.

*Crime Rate*: The crime rate is reasonably even but varies throughout the data set, with a higher crime rate in some sections.

*Bedrooms and Bathrooms:* Most of the properties have around the same number of bedrooms and bathrooms, proving an average residential configuration.

***Figure 8:*** Relation between *bedrooms* and *property price* is linear.



Bar plots revealed that **rent increases with number of bedrooms** (e.g., 1-bedroom avg ~$1500, 5-bedroom ~$3400) and varies by property type (houses and "Other" types commanded higher rent than apartments).

***Figure 9:*** *In Ontario,* Houses are more expensive compared to all property types.

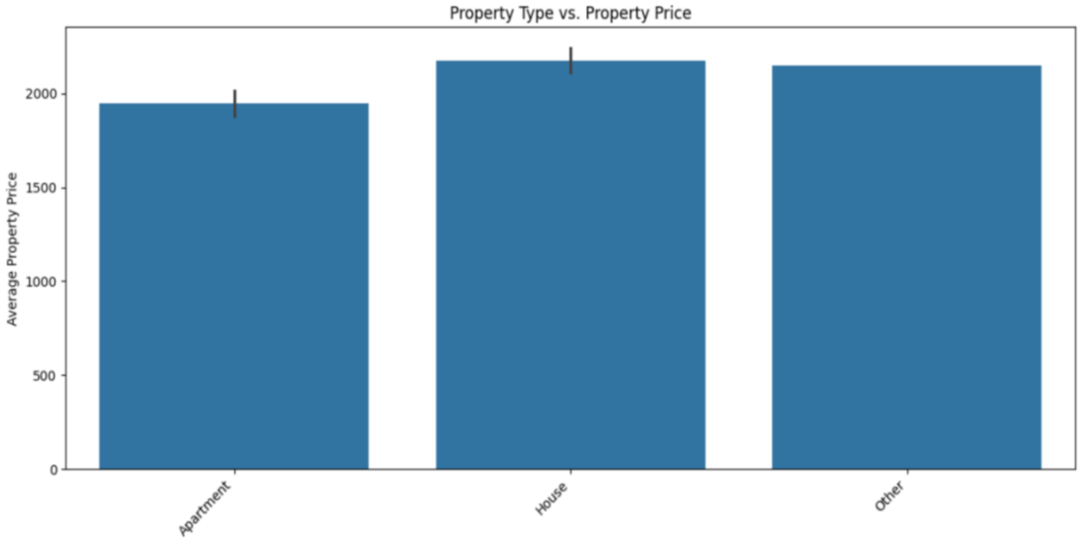


***Figure 10:*** *Toronto* has the most expensive, and *Burlington* is least expensive city among Ontario cities.

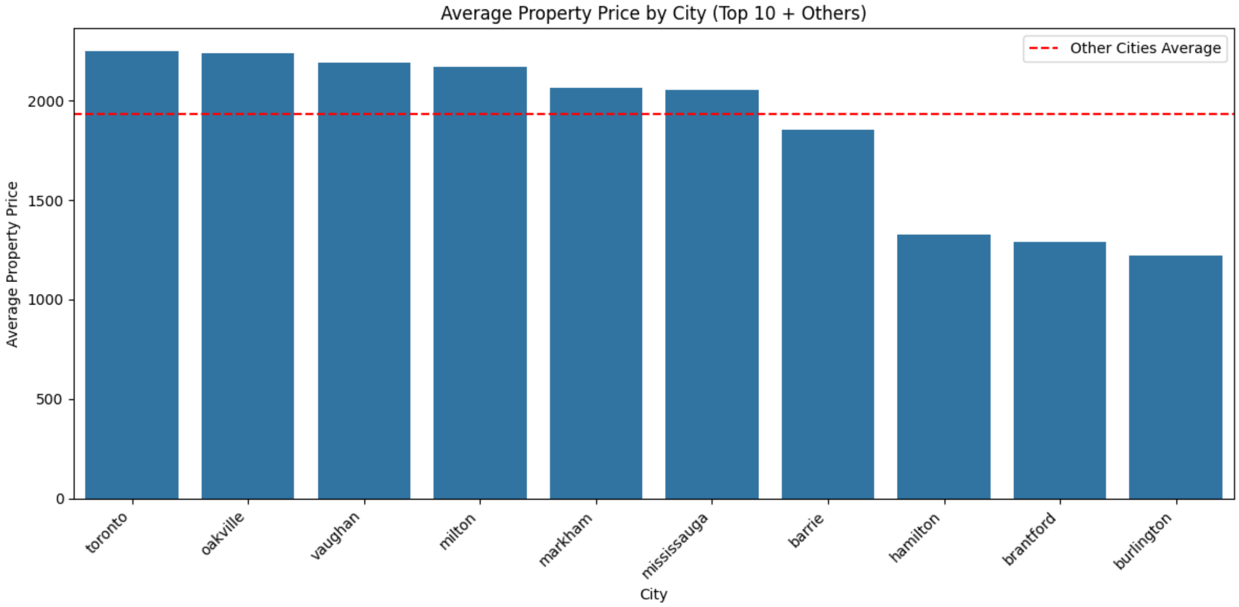
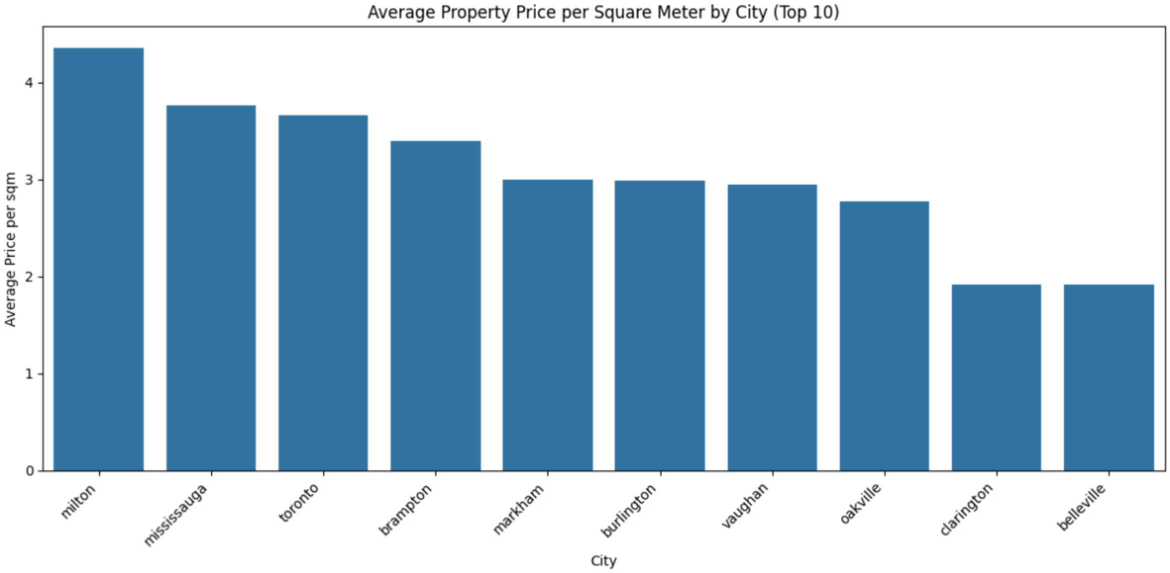
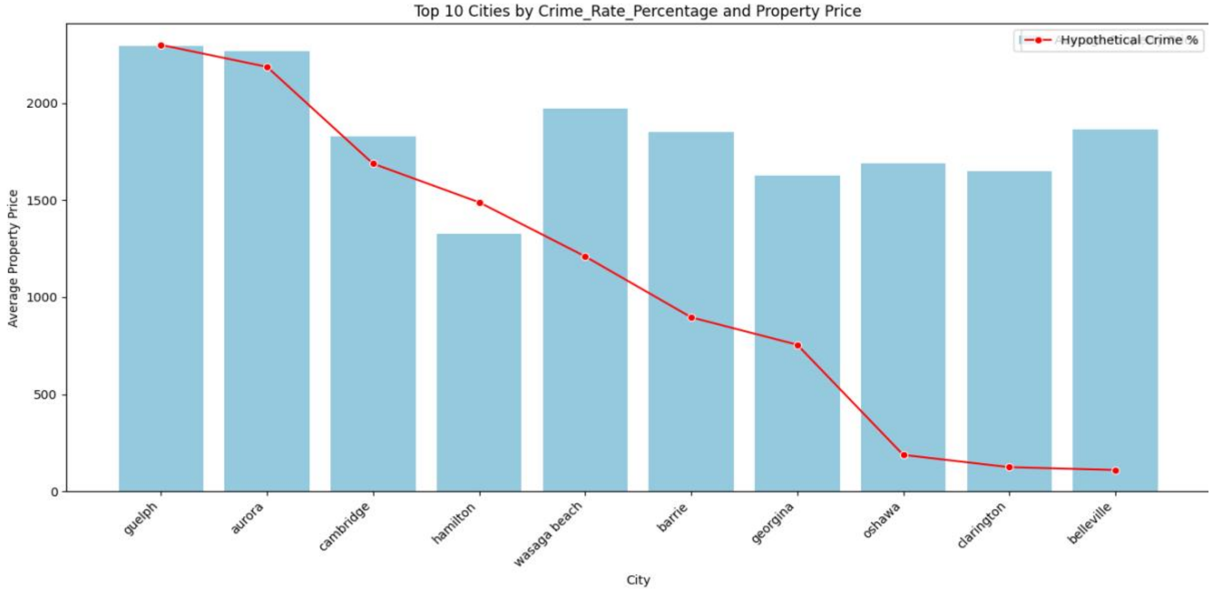


***Figure 11:*** *Milton is* the most expensive city for the average price per square fit.



City-level analysis showed *Toronto* had the highest average rent while *Burlington* was the lowest. A bar chart of price per square foot indicated Milton and *Mississauga* had the highest ratios.

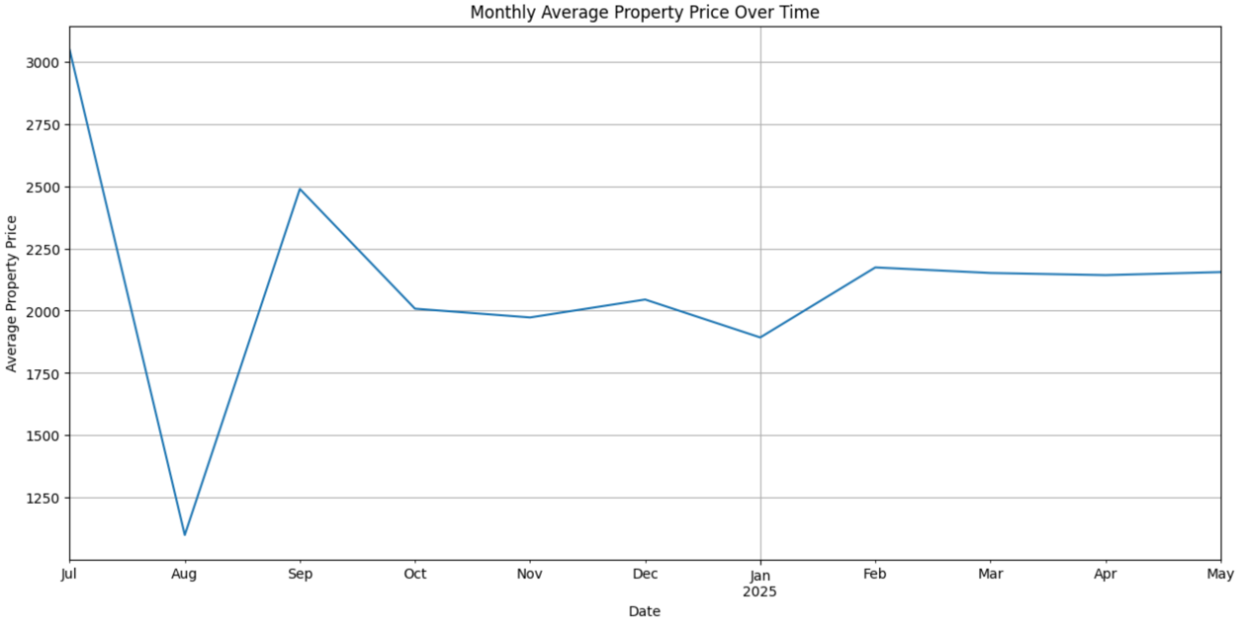
***Figure 12:***



There appears to be an inverse correlation between crime rates and the value of properties. Urban centers with high property values like Aurora and Guelph often witness proportionally increased crime rates.

Urban towns which experience lowered property values, like Belleville and Clarington, tend to have fewer crimes.

***Figure 13:*** Average Property Price on Monthly Basis



The graph displays the changes in house prices over one year. The initial fluctuation could represent seasonal factors or market corrections, but the stabilization that follows means a more balanced market.

***Figure 14:*** Elbow plot showing the inertia vs. number of clusters (K). The "elbow" around K=3 suggests three clusters is optimal.

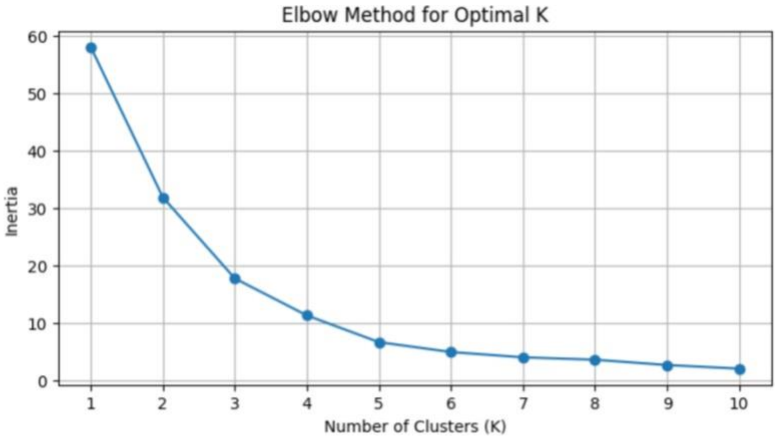


***Figure 15:*** A scatter plot of city clusters (K=3) based on scaled rent price on the x-axis and scaled crime rate on the y-axis. Three clusters emerge: (0) high-price/low-crime, (1) low-price/high-crime, (2) moderate/moderate.

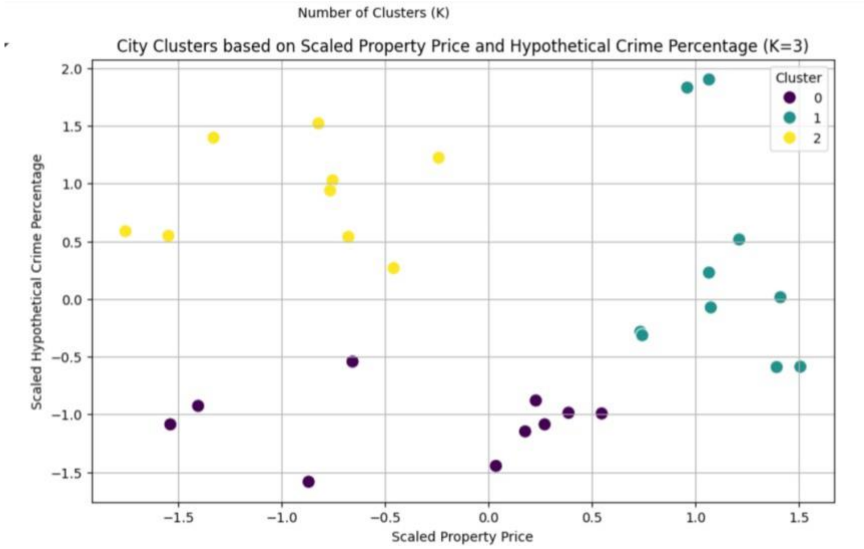
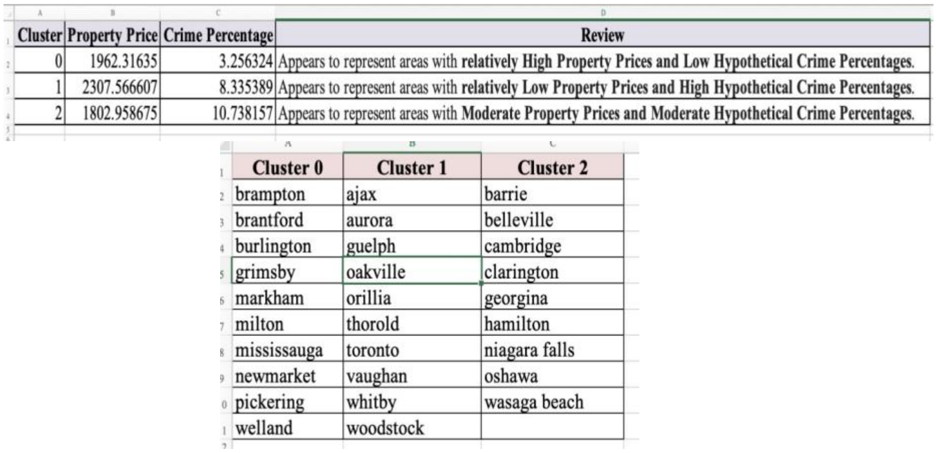


***Figure 16:*** Table of cities grouped by cluster label (0, 1, 2) as obtained above. Summary table of each cluster's average property price and crime percentage, with a brief interpretation of each cluster's characteristics.



| Cluster | Property Price | Crime Percentage | Review |
|---|---|---|---|
| 0 | 1962.31635 | 3.256324 | Appears to represent areas with **relatively High Property Prices and Low Hypothetical Crime Percentages.** |
| 1 | 2307.566607 | 8.335389 | Appears to represent areas with **relatively Low Property Prices and High Hypothetical Crime Percentages.** |
| 2 | 1802.958675 | 10.738157 | Appears to represent areas with **Moderate Property Prices and Moderate Hypothetical Crime Percentages.** |

| Cluster 0 | Cluster 1 | Cluster 2 |
|---|---|---|
| brampton | ajax | barrie |
| brantford | aurora | belleville |
| burlington | guelph | cambridge |
| grimsby | oakville | clarington |
| markham | orillia | georgina |
| milton | thorold | hamilton |
| mississauga | toronto | niagara falls |
| newmarket | vaughan | oshawa |
| pickering | whitby | wasaga beach |
| welland | woodstock | |

Clustering is a technique used to classify similar data points into clusters based on their features, without referencing any pre-defined labels. It enables the identification of inherent patterns, structures, or segments within a dataset.

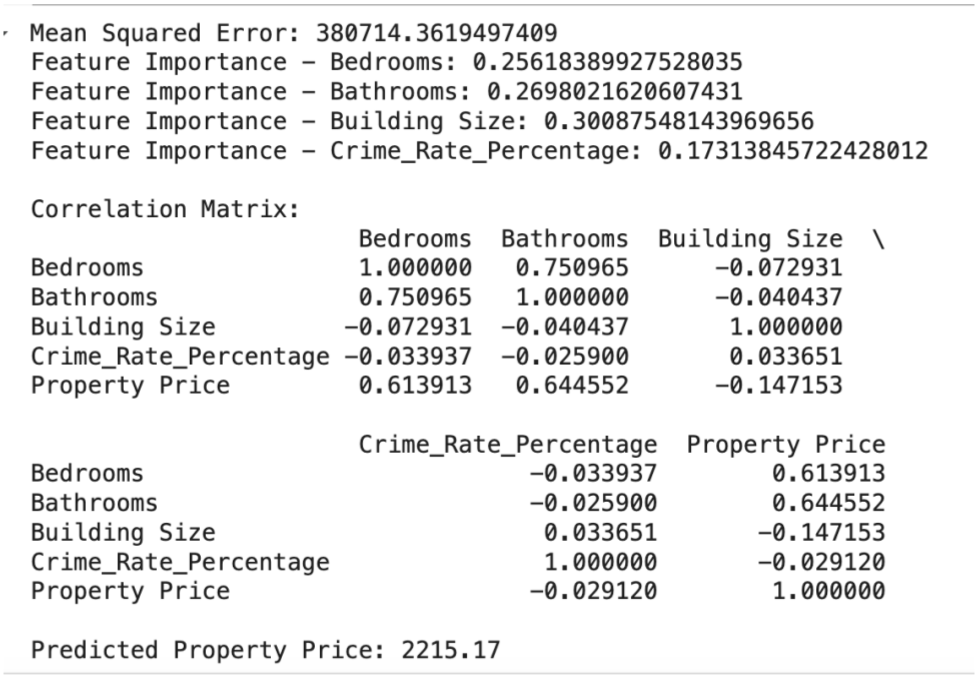
Unsupervised clustering (K=3) of scaled crime rate and price revealed three groups of cities (high-price/low-crime, low-price/high crime, moderate-price/crime). Figures (below) illustrate these findings.

## Predictive Modeling

Correlation analysis found moderate positive correlations between price and Bedrooms/Bathrooms (r≈0.61–0.64) and low correlation between price and crime rate. Multiple regression models were evaluated.

The **Random Forest regressor** achieved the best performance (lowest MSE and RMSE, highest R²) among tested models. In contrast, Support Vector Regression had the highest errors and even a negative R². Classification metrics (for a classification of price bracket) showed overall accuracy ~81%.

***Figure 17:*** Correlation analysis was performed to examine correlations among features such as 'Bedrooms', 'Bathrooms', ‘Crime Rates’ and 'Property Price'.



```
Mean Squared Error: 380714.3619497409
Feature Importance - Bedrooms: 0.25618389927528035
Feature Importance - Bathrooms: 0.2698021620607431
Feature Importance - Building Size: 0.30087548143969656
Feature Importance - Crime_Rate_Percentage: 0.17313845722428012

Correlation Matrix:
                       Bedrooms  Bathrooms  Building Size  \
Bedrooms               1.000000   0.750965      -0.072931
Bathrooms              0.750965   1.000000      -0.040437
Building Size         -0.072931  -0.040437       1.000000
Crime_Rate_Percentage -0.033937  -0.025900       0.033651
Property Price         0.613913   0.644552      -0.147153

                       Crime_Rate_Percentage  Property Price
Bedrooms                           -0.033937        0.613913
Bathrooms                          -0.025900        0.644552
Building Size                       0.033651       -0.147153
Crime_Rate_Percentage               1.000000       -0.029120
Property Price                     -0.029120        1.000000

Predicted Property Price: 2215.17
```

***Model Performance:*** Mean Squared Error (MSE) is 380,714.36 measures the average squared difference between predicted and actual property prices. A high MSE indicates that the model's predictions deviate significantly from the actual values, suggesting room for improvement in the model.

***Feature Importances:*** Feature importance scores indicate how much each variable contributes to the prediction of property prices:

- Building Size: 0.3009 (30.09%)
    - The most significant predictor of property prices.
- Bathrooms: 0.2698 (26.98%)
    - The second most important feature.
- Bedrooms: 0.2562 (25.62%)
    - The third most important feature.
- Crime Rate Percentage: 0.1731 (17.31%)
    - While less significant than the others, it still influences property prices.

***Correlation Matrix:*** The correlation matrix provides insights into the relationships between variables:

- Bedrooms and Bathrooms:
    - Correlation: 0.7509
    - Strong positive correlation, indicating that properties with more bedrooms tend to have more bathrooms.
- Bedrooms and Property Price:
    - Correlation: 0.6139
    - Moderate positive correlation, meaning more bedrooms generally lead to higher property prices.
- Bathrooms and Property Price:
    - Correlation: 0.6446
    - Moderate positive correlation, similar to bedrooms.

- Building Size and Property Price:
    - Correlation: -0.1472
    - Weak negative correlation, which is unexpected and may require further investigation.
- Crime Rate Percentage and Property Price:
    - Correlation: -0.0291
    - Very weak negative correlation, indicating that crime rate has minimal direct impact on property prices in this dataset.

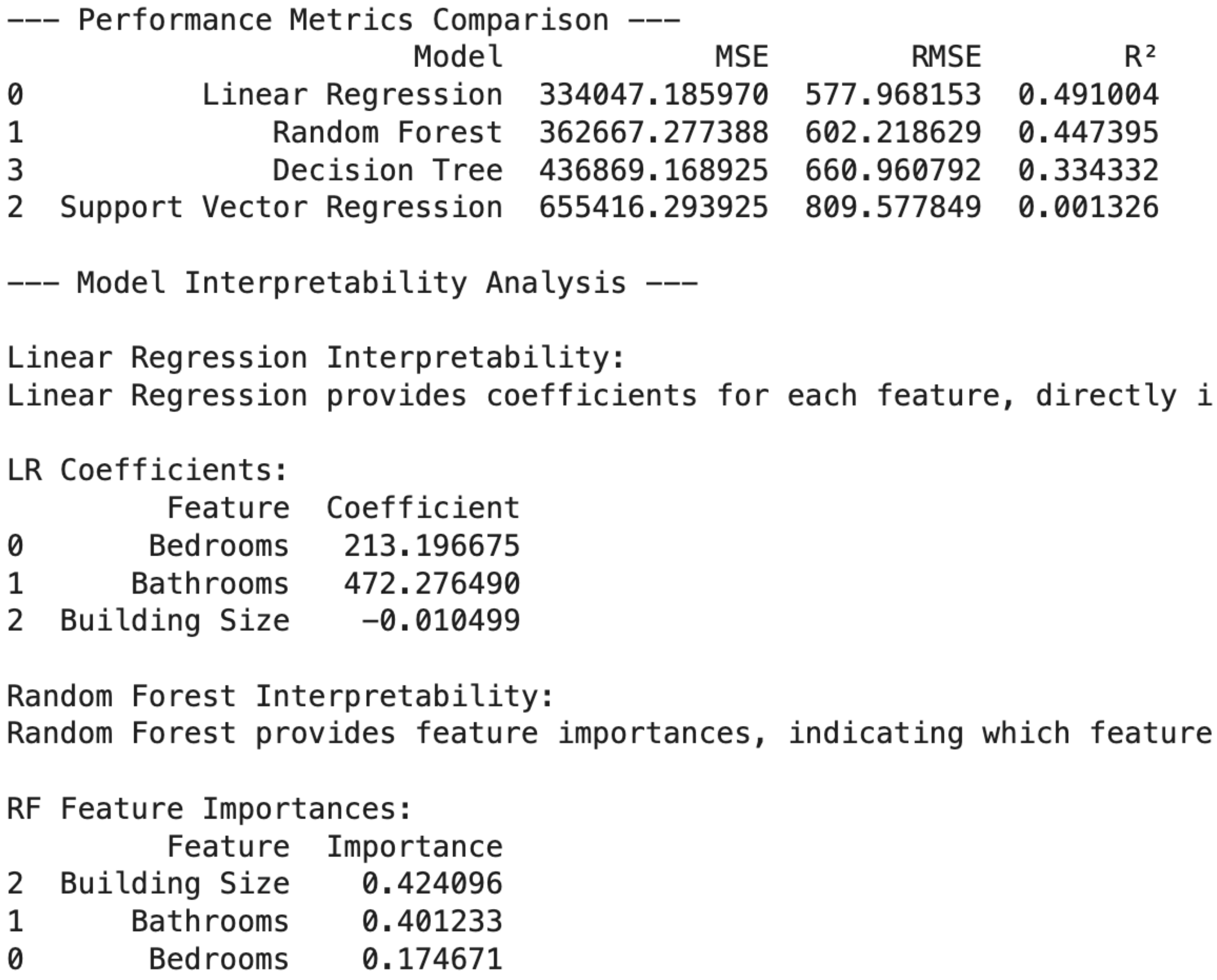
**4. Predicted Property Price**

- Value: 2,215.17
    - This is the predicted price for a property based on the input features provided to the model.
    -

***Overall Conclusion*:**

The model identifies building size, bathrooms, and bedrooms as the most influential factors for property price predictions. The weak relationship found between the architectural measurements and the property value estimation could indicate a possible inconsistency in the dataset or might require further feature engineering. The high MSE suggests the model's predictions are not yet highly accurate and could benefit from optimization or additional data preprocessing.

***Figure 18:*** Model Performance was evaluated in terms of metrics like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) for accuracy.



```
--- Performance Metrics Comparison ---
                       Model            MSE        RMSE        R²
0          Linear Regression  334047.185970  577.968153  0.491004
1              Random Forest  362667.277388  602.218629  0.447395
3              Decision Tree  436869.168925  660.960792  0.334332
2  Support Vector Regression  655416.293925  809.577849  0.001326

--- Model Interpretability Analysis ---

Linear Regression Interpretability:
Linear Regression provides coefficients for each feature, directly i

LR Coefficients:
         Feature  Coefficient
0       Bedrooms   213.196675
1      Bathrooms   472.276490
2  Building Size    -0.010499

Random Forest Interpretability:
Random Forest provides feature importances, indicating which feature

RF Feature Importances:
         Feature  Importance
2  Building Size    0.424096
1      Bathrooms    0.401233
0       Bedrooms    0.174671
```

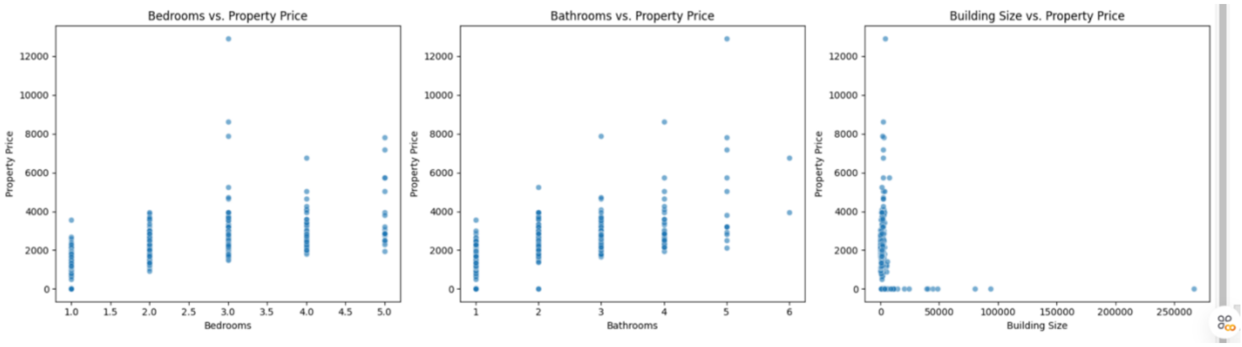
**Observations on Performance:**

- Random Forest consistently exhibits the best performance across all metrics (lowest MSE, RMSE, highest R²). This suggests it is the most accurate model for predicting Property Price on this dataset.
- Random Forest R² (0.4474) indicates that it explains a significant portion of the variance in Property Price.
- Linear Regression shows moderate performance, with a higher MSE/RMSE and lower R² compared to Random Forest, suggesting that a purely linear model is not the best fit.
- Support Vector Regression (SVR) performs poorly, indicated by the highest MSE/RMSE and a negative R². A negative R² suggests the model performs worse than simply predicting the mean of the target variable, indicating it is not suitable for this data in its current configuration.
- Decision Tree performance is better than LR and SVR but worse than Random Forest, which is expected as RF is an ensemble of Decision Trees.

**Random Forest (Chosen as Best Performer in the analysis):**

- Random Forest is an ensemble method based on decision trees. It was likely chosen because it is robust to outliers and capable of capturing non-linear relationships and interactions between features.
- The scatter plots (Bedrooms vs. Price, Bathrooms vs. Price, Building Size vs. Price) show that the relationships are not perfectly linear and have considerable spread, which aligns with why a non-linear model like Random Forest would perform well.
- The observed lower MSE/RMSE and higher R² confirm that Random Forest was effective in modeling the complex relationships in the data.

*Figure 19:* Visual Exploration for Non-linearity



**Linear Regression:**

- Linear Regression was included as a simple, interpretable baseline model that assumes a linear relationship between features and the target.

- Its performance was better than SVR but worse than Random Forest and Decision Tree, suggesting that while there might be some linear components in the relationships, they are not purely linear, or there are significant non-linear interactions.

- The coefficients provide insights into the direction and magnitude of linear associations, which is useful for understanding the data even if the model isn't the best predictor.

**Support Vector Regression (SVR):**

- SVR is a non-linear regression model that aims to find a hyperplane with a maximal margin, using kernel functions to handle non-linearity.

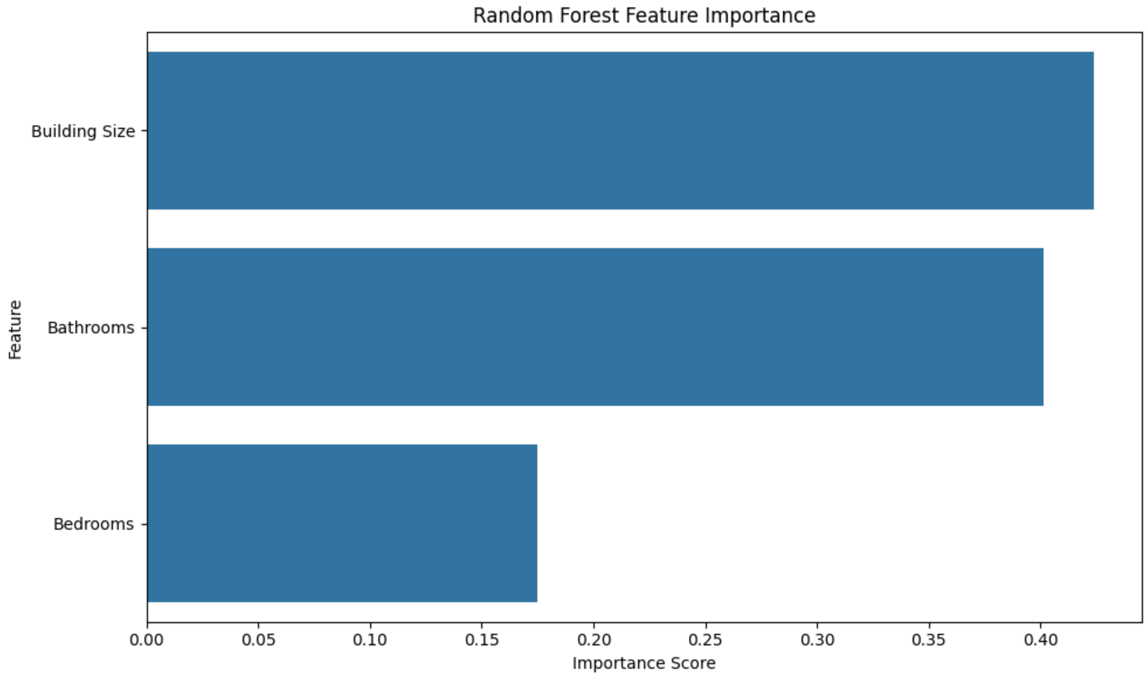
- While SVR can be powerful, its performance is highly dependent on hyperparameter tuning (like the choice of kernel and regularization parameters, C and epsilon).

- The observed poor performance of SVR (high errors, negative $R^2$) strongly suggests that the default parameters or the chosen kernel were not suitable for this specific dataset and the observed relationships.

- The data might not have clear margins or structures that SVR is designed to exploit effectively without specific tuning, or the noise level might make it difficult for SVR to find a good fit.

Random Forest model was the most suitable algorithm for this task based on the performance metrics (MSE, RMSE, R²), likely because it effectively captured the non-linear and potentially complex relationships between the number of bedrooms, bathrooms, building size, and the property price that were evident in the data's scatter plots and not well-modeled by simpler linear methods or an unoptimized SVR.

***Figure 20:*** Random Forest Feature Importance Score



While K-NN or median imputation are simpler and more transparent, the choice of `Random Forest Regressor` for imputation in the provided code was likely driven by the need to leverage the relationships between features, capture potential non-linear patterns, and ultimately improve the quality of the imputed data for better downstream model performance. The complexity is accepted because the benefit of more accurate, context-aware imputation outweighs the simplicity of less sophisticated methods, especially when features are known to be highly correlated and interact in non-trivial ways.

# Conclusion

**Investment and Development of Real Estate**

- For Luxury/Premium Development: Focus on cities in Cluster 0 (High Price, Low Crime). These cities likely have a market for high-end properties and security is of utmost importance to residents.
- For Affordable Housing/Redevelopment: Consider cities in Cluster 1 (Low Price, High Crime). While possibly riskier, these cities may have lower entry points and potential for value-integrate prospects, possibly with the assistance of revitalization programs in urban areas.
- Urban areas classified under Cluster 2, which exhibit modest degrees of both crime and pricing, can be good investment options that can appeal to a wide demographic.

**Urban planning and Public Policy**

- In the Low Price and High Crime urban areas in Cluster 1, policy interventions must be geared towards enhancing public safety, improving community engagement activities, enticing businesses to spur the local economy, and investing in infrastructural development to enhance the attractiveness of these areas.
- For Cities in Cluster 0 (High Price, Low Crime): Planning could entail managing growth, maintaining infrastructure quality, maintaining the character of the area. In addition, solving issues like housing unaffordability can help if there are high costs that are keeping certain segments out. In the case of Cluster 2 (Moderate) Cities, priority could be given to promoting balanced development, investing in public facilities like schools and recreation spaces, and implementing strategies to prevent decline or to adjust to rapidly shifting demographic trends or crime patterns.

### Commercial and Marketing Strategy

- Firms selling High-End Goods/Services put target advertising and store locations in Cluster 0 (High Price, Low Crime) cities.
- Organizations that focus on Value/Affordability or Security Products/Services can see increased profitability in urban areas classified under Cluster 1 (Low Price, High Crime).
- For General Services and Retailers, urban areas identified under Cluster 2 (Moderate) reflect a diverse market that may be suited to standard business models.

### Protocols of Residential Procurement and Relocation

- People who value security and are willing to pay a higher cost
- Find the metropolitan areas classified under Cluster 0.
- For buyers who prioritize affordability: Seek cities in Cluster 1 but be ready for potentially higher crime rates. For buyers looking for a balance, Cluster 2's urban locations might offer a compromise between affordability and protection.

# Result And Discussion

- Building Size, number of Bathrooms, and number of Bedrooms are the most crucial predictors of rent prices.
- Toronto was the most expensive city on average, while Hamilton was the least.
- The Random Forest model produced the best prediction accuracy (lowest error, highest $R^2$).
- Clustering cities by price and crime suggests tailored strategies for each group (e.g., luxury development in high-price/low-crime areas vs. affordable housing initiatives in high-crime areas).

- The analysis revealed clear patterns: larger buildings and more rooms command higher rent, with Toronto exhibiting the highest prices among cities.

- The data cleaning and feature engineering steps (e.g., one-hot encoding, z-scoring) enabled effective modeling. The Random Forest model outperformed others, suggesting non-linear interactions in the data.

- Clustering of cities by price and crime rate offered actionable insights for urban planning: high-price low-crime cities (Cluster 0) are targets for luxury investment, while low-price high-crime areas (Cluster 1) might benefit from redevelopment initiatives. These findings align with related work (e.g., cluster-based policy suggestions in rental markets).

- The methodologies used—such as advanced feature engineering and model explainability—provide a framework that can guide future projects and real estate decision-making.

# References

Canada Crime Report. (n.d.). *Crime Severity Index*. Retrieved from https://canadacrimereport.com/crime- severity-index

GeeksforGeeks. (n.d.). *House Price Prediction using Machine Learning in Python*. Retrieved from https:// www.geeksforgeeks.org/house-price-prediction-using-machine-learning-in-python/

Kaggle. (n.d.). *Exploratory Data Analysis – House Rent Prediction* [Code notebook]. Retrieved from https:// www.kaggle.com/code/rkb0023/exploratory-data-analysis-house-rent-prediction

Senthilkumar, V. (2023). *Enhancing House Rental Price Prediction Models for the Swedish Market: Exploring External Features, Prediction Intervals and Uncertainty Management in Predicting House Rental Prices* (master's thesis, KTH Royal Institute of Technology).

Realtor.com. (n.d.). *Ontario Rent Listings* [Dataset]. Retrieved from https://www.realtor.com/international/ca/ontario/rent/p1