Middle Tennessee State University
DATA 6990 (Fall 2023)

Machine Learning in Real Estate

Christina Moody
cdc3m@mtmail.mtsu.edu

Robert Conner
rc5g@mtmail.mtsu.edu

Abstract

Building on previous research (Choy et al., 2023) that demonstrated the efficacy of machine learning in real estate pricing predictions, this paper extends the scope by incorporating additional models to further test accuracy. Alongside the previously used 24,317 housing transaction records, this study introduces Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multilayer Perceptron (MLP) to the existing suite of Extra Trees (ET), k-Nearest Neighbors (KNN), and Random Forest (RF) algorithms. Our methodology retains the unique feature of combining property age with square footage to better capture the effects of land depreciation. The primary objective is to evaluate whether these new models enhance predictive performance over both the traditional hedonic price model and the initially used machine learning algorithms. Preliminary results indicate a potential improvement in explanatory power and error reduction.

1 Introduction

The integration of advanced technologies in business operations, particularly those with large amounts of consumer data, has become a norm across various industries. This trend is notably evident in the real estate sectors, where firms are able to leverage significant consumer and household data to create complex models to assist in decision making at all levels of the business. The advent of machine learning (ML) has further revolutionized these industries. ML's capacity to analyze and interpret large datasets makes it an invaluable tool in the real estate market for tasks such as property valuation, management, and investment.

The original paper "The Use of Machine Learning in Real Estate Research" by Choy et al., demonstrated the effectiveness of machine learning in real estate by employing algorithms such as Extra Trees, k-Nearest Neighbors (k-NN), Random Forest, and Ordinary Least Squares (OLS). Their research highlighted the potential of ML in providing more accurate property price predictions compared to other traditional methods.

Our study extends this exploration by focusing on a different set of machine learning models: Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP). We aim to not only replicate but also build upon the original study's methodologies, applying these models to the same dataset used in the original paper to compare their performance. This paper is organized as follows. Section 2 presents the data collection and pre-processing. Section 3 describes the model implementation and methodology. Section 4 covers the model performance comparisons between the created models, namely KRR, SVR, and MLP. Section 5 explores a discussion of the results and section 6 provides the conclusions drawn from the project and a comparison of our results and the original paper. The references can be found following section 6. Section 7 details the contributions by the authors of this paper.

(LateX template borrowed from NeurIPS 2019)

1.1 Model Specification

In this study, we focus on three machine learning models: Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP). Each of these models brings unique strengths to the task of predicting property prices and offers interesting contrasts in their approach to learning from data.

1. **Kernel Ridge Regression (KRR)**: This regression technique combines ridge regression (linear least squares with l2-norm regularization) with the kernel trick. It thus learns a linear function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space. The key formula in KRR is:

$$f(x) = \sum_{i=1}^{n} \alpha_i K(x, x_i) + b$$

where $K(x, x_i)$ is the kernel function, x represents an input vector, x_i are the support vectors, α_i are the dual coefficients, and b is the bias. Common choices for the kernel function include polynomial, Gaussian, and sigmoid kernels.

Strengths:

- KRR can deal with multicollinearity (high correlation among features) by applying regularization.
- It is less prone to overfitting due to its ridge penalty.
- The kernel trick enables it to learn complex, non-linear functions.

Weaknesses:

- KRR can be computationally intensive for large datasets because it requires matrix inversion.
- Selecting the right kernel and tuning hyperparameters (like the regularization parameter and kernel-specific parameters) can be challenging.

2. **Support Vector Regression (SVR)**: SVR seeks to find a function that deviates from the targets y_i by a value less than ϵ for each training point x_i, while also being as flat as possible. It solves the following optimization problem:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

subject to $y_i - wx_i - b \leq \epsilon + \xi_i$, $wx_i + b - y_i \leq \epsilon + \xi_i^*$, and $\xi_i, \xi_i^* \geq 0$ for $i = 1, \ldots, m$.

Strengths:

- SVR is effective in high-dimensional spaces.
- It is robust to outliers and has a strong theoretical foundation that avoids the risk of overfitting.
- The use of kernel functions allows flexibility in modeling different types of data distributions.

Weaknesses:

- Like KRR, SVR can be computationally expensive for large datasets and requires careful tuning of hyperparameters.
- Interpretability can be challenging, especially with non-linear kernels.

3. **Multi-Layer Perceptron (MLP)**: As a class of feedforward artificial neural network, an MLP consists of multiple layers of nodes, each layer fully connected to the next one. The output y of an MLP for a given input x is a non-linear function of the weights W and biases b, typically represented as:

$$y = f(W_n \cdot f(W_{n-1} \cdot (\ldots f(W_2 \cdot f(W_1 \cdot x + b_1) + b_2)\ldots) + b_{n-1}) + b_n)$$

where f is a non-linear activation function, and n is the number of layers.

Strengths:

- MLPs are capable of modeling complex non-linear relationships and interactions between features.
- They can be scaled to larger datasets and benefit from deep learning optimizations and hardware accelerations.
- MLPs are flexible and can be adapted to various types of problems.

Weaknesses:

- MLPs require a lot of data to train effectively and are prone to overfitting without proper regularization.
- They have many hyperparameters to tune (layers, nodes, activation functions, etc.), which can make them complex to set up.
- MLPs are often considered "black boxes" due to their lack of interpretability.

Our approach involves a detailed evaluation of these models using various performance metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2), among others. The objective is to explore how these models perform in predicting property values and to understand their strengths and limitations in the context of real estate data analysis.

2 Data Collection/Preprocessing

In our study, we extended the analysis of the original research, which focused on four private housing estates (Grand Promenade, Kornhill Garden, Les Saisons, Taikoo Shing) in the Quarry Bay district of Hong Kong. These estates are recognized as “selected popular residential developments” by the Rating and Valuation Department, Hong Kong SAR Government. The original study’s dataset spanned from January 1997 through May 2021, encompassing 24,317 pooled cross-sectional data observations. Our dataset, provided without explict descriptions, includes a larger number of records, totaling 24,936 observations. However, without information on the date range, we cannot confirm if our dataset extends beyond the original study’s timeframe.

We assume our data were sourced from government records and compiled by the commercial company "EPRC", similar to the data used in the original study. These records included detailed information on building names, locations, dates of transactions and occupation permits, sums paid, square footage, and additional details such as the inclusion of a parking space with the property. In our exploration of the data, we found that the ’RP’ values are most likely already aligned with the original study, where property prices were deflated into real terms by dividing the popular housing estate price index compiled by the Rating and Valuation Department. We also found that records with incomplete or inaccurate information, as well as non-commercial property transactions (such as gifts), were previously excluded from our dataset.

Our data definitions, including the modifications and transformations applied, are summarized in Table 1. Histograms depicting the distribution of each variable, including the log-transformed RP and GFA, are presented in Figure 1, providing a visual estimation of their distributions. The correlation matrix (Figure 2) illustrates the linear relationships between variables. In our dataset, we observed high correlations similar to those in the original study, particularly between footage area and residential property prices (correlation coefficient of 0.84), floor level (0.36), age (-0.3), and proximity to mass transport railway (0.35). Additionally, Figure 3 offers data visualizations to showcase these relationships.

Table 1: Data Definitions

Variable	Definition
RP_i^t	Represents the total consideration of residential property i during time period t, measured in HK dollars, inflation adjusted.
GFA_i^t	Represents the gross floor area of residential property i, including the area of penthouse, bay windows, and balconies if any.
AGE_i^t	Represents the age of residential property i in years, which is calculated using the time elapsed between when the occupation permit was issued and when the homes were sold.
ZZ_i^t	Represents the multiplication of building structure and property age of residential property i.
FL_i^t	Represents the floor level of residential property i.
FR_i^t	Dummy variable that is set to be 1 if property i has a flat roof, 0 otherwise.
$ROOF_i^t$	Dummy variable that is set to be 1 if property i has a roof top, 0 otherwise.
CP_i^t	Represents the number of carpark(s) transacted with residential property i.
MTR_i^t	Dummy variable that is set to be 1 if it takes no more than ten minutes to walk from property i to the nearest mass transit railway station, 0 otherwise.
E_i^t, S_i^t, W_i^t, N_i^t, NE_i^t, SE_i^t, SW_i^t & NW_i^t	Represent eight possible directions in which property i could be facing. If a property is facing a specific direction, they are set to be 1; 0 otherwise. Northwest has been left out of the analysis so that these coefficients can be evaluated in relation to this category.

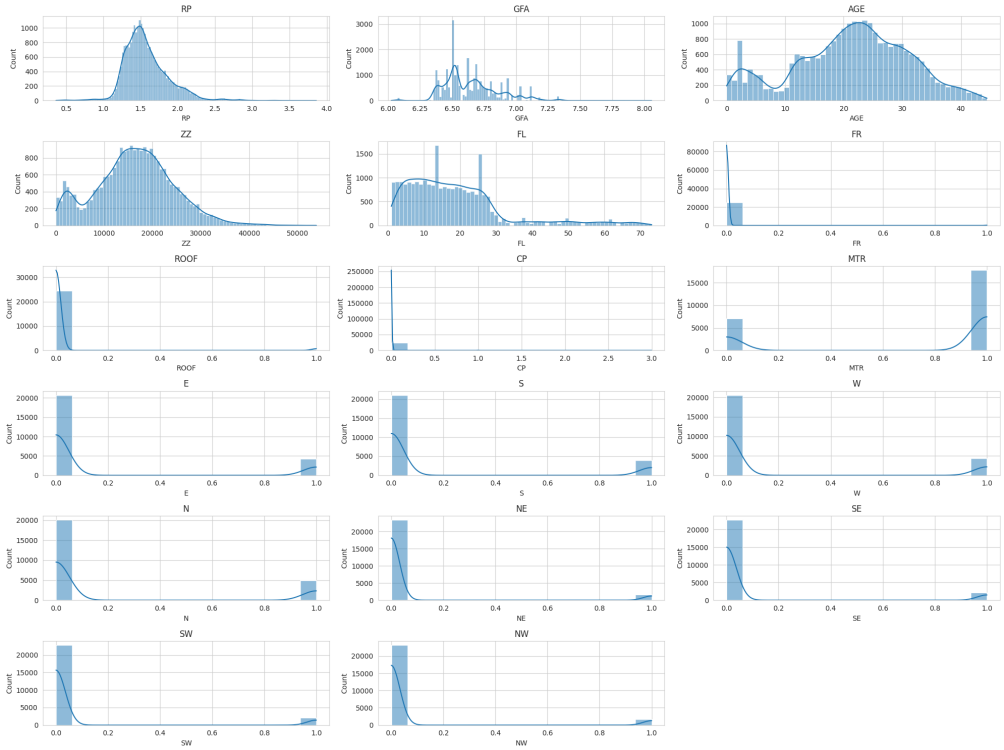

Figure 1: Histograms of Features, Distribution Analysis

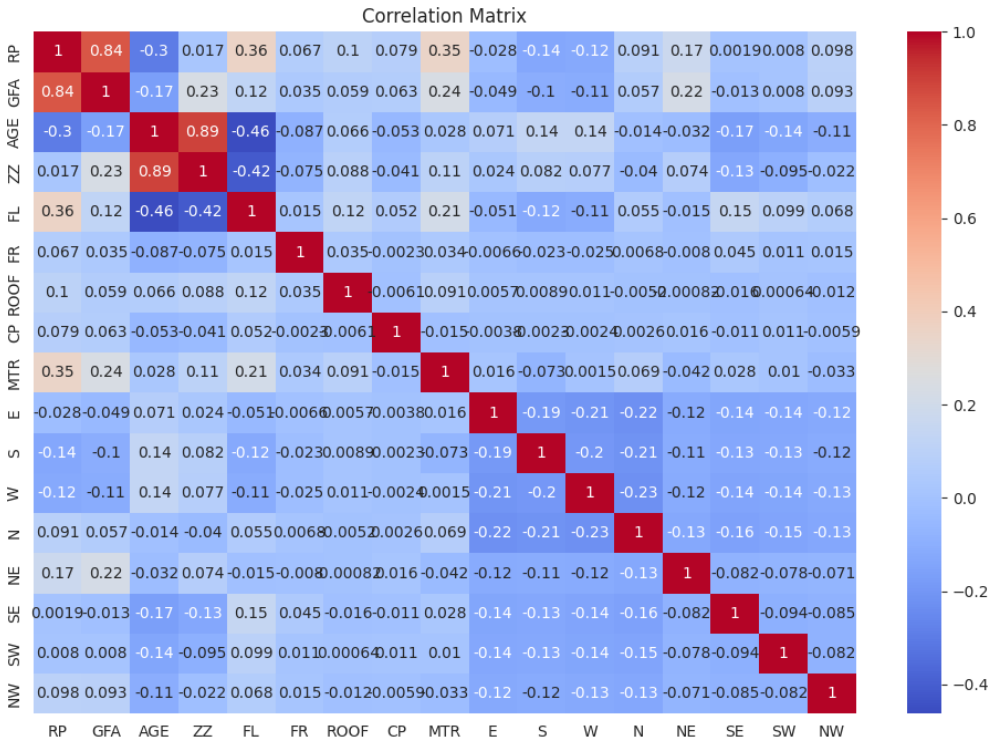

Figure 2: Correlation Matrix Heatmap

3 Model Implementation

In this project, the model implementation process is methodically designed to encompass a series of sophisticated machine learning regression techniques: Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP). The core of this implementation lies in data preprocessing, comprehensive hyperparameter optimization using Optuna, and rigorous model evaluation. The dataset undergoes a standardized split into training, validation, and test sets, with 'StandardScaler' employed for data normalization. This step ensures that the data is consistently scaled across all models, a critical aspect for unbiased model training and evaluation.

Next is hyperparameter tuning. Optuna, a robust hyperparameter tuning framework, is central to this process. It fine-tunes a range of parameters specific to each model — alpha, kernel, and gamma for KRR; C, epsilon, and kernel type for SVR; and various parameters like hidden layer sizes and solver for MLP. This optimization aims at minimizing the MSE on the validation set, guiding towards the most effective model configuration.

Post optimization, the models are retrained using the best hyperparameters on the combined training and validation data. Their performance is then evaluated on the test set using MSE and R-squared metrics. Additionally, the 'plot model diagnostics' function offers insightful visualizations, including actual vs. predicted values and residual analysis (Figures 3, 4, 5), with specific enhancements for SVR like displaying the number of support vectors(Figure 4).

The implementation includes robust features like dynamic naming for saving models and exception handling in the MLP function. This approach ensures resilience in the modeling process and flexibility in subsequent analyses. The output is comprehensive, providing optimized hyperparameters, details of the best trials, and the prepared datasets for further exploration or application in different scenarios.

Here are the figures associated with the models:.

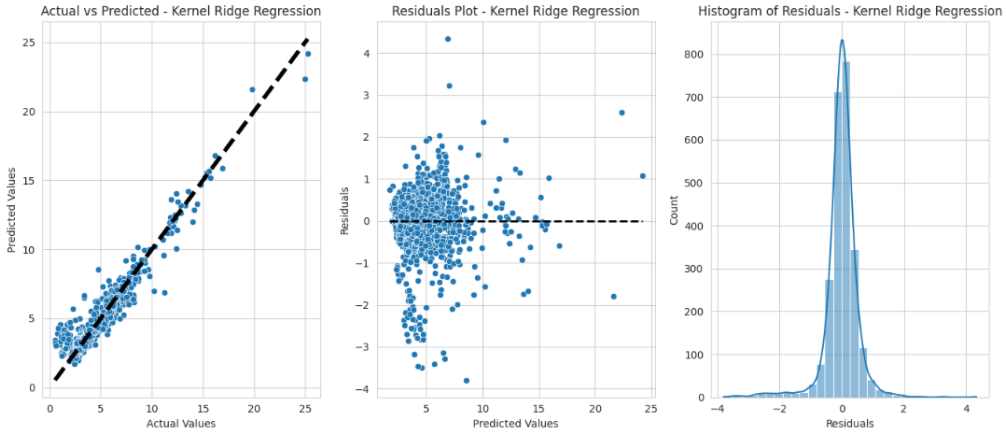

Figure 3: KRR actual vs predicted, residual analysis

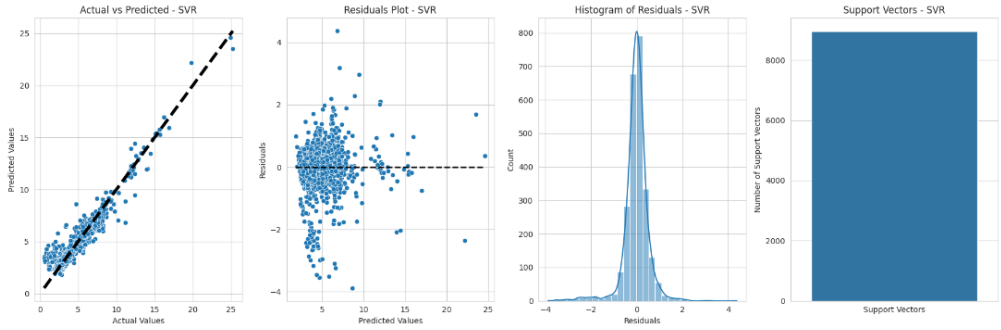

Figure 4: SVR Vector Analysis

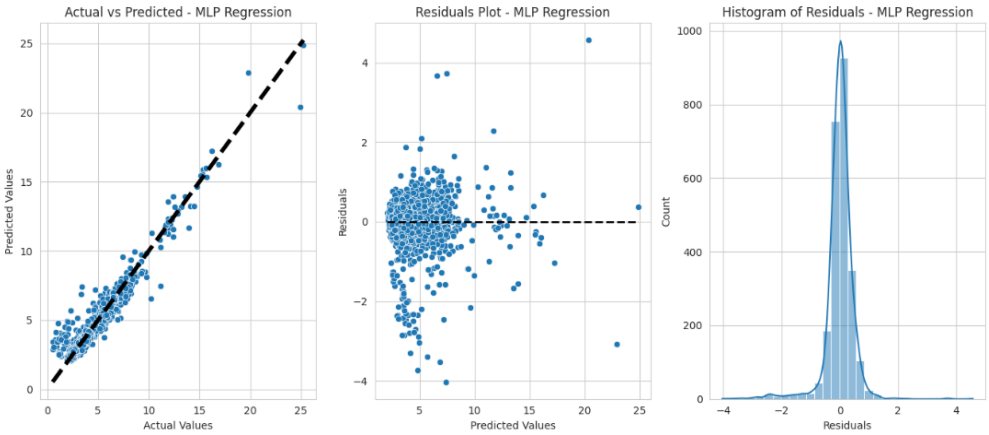

Figure 5: MLP actual vs predicted, residual analysis

4 Model Performance Comparison

Model	Test MSE	Test R^2	Test MAPE	Test RMSE
Kernel Ridge Regression	0.333114	0.909128	11.566388	0.577160
Support Vector Regression	0.328023	0.910517	11.009784	0.572733
MLP Regression	0.308698	0.915789	10.606829	0.555606
ET (Orig. Study)	0.305610	0.911640	9.046530	0.552820
KNN (Orig. Study)	0.362110	0.895300	10.395210	0.601760
Random Forest (Orig. Study)	0.279180	0.919280	8.889300	0.528370
Ordinary Least Squares (Orig. Study)	0.638900	0.814000	14.542680	0.799310

Figure 6: Model Performance Comparison

5 Discussion of Results

Here we compare their performance based on Mean Squared Error (MSE), R-squared (R^2), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) values obtained from the given dataset:

1. **Kernel Ridge Regression (KRR)**:
   - **Performance**: KRR displayed a test MSE of 0.3331, an R^2 of 0.9091, a MAPE of 11.57, and a RMSE of 0.5772.
   - **Analysis**: KRR combines ridge regression's penalty on large coefficients with kernel tricks to model non-linear relationships. While the R^2 value indicates a significant level of explanatory power, it is marginally eclipsed by other models in terms of MSE, R^2, and the additional metrics of MAPE and RMSE. Its performance suggests effective handling of multicollinearity among features and an ability to capture complex patterns in the dataset, albeit slightly less efficiently than some alternatives.
2. **Support Vector Regression (SVR)**:

- **Performance**: SVR yielded a test MSE of 0.3280, an R^2 of approximately 0.9105, a MAPE of 11.01, and a RMSE of 0.5727.
- **Analysis**: SVR relies on defining a margin of tolerance (epsilon) and attempts to fit the data within this margin while penalizing instances that fall outside. The higher R^2 value, coupled with lower MSE, MAPE, and RMSE compared to KRR, indicates that SVR may be capturing the dataset's variance more effectively. This suggests that the chosen kernel and regularization parameters are well-suited for the dataset's structure.

3. **Multi-Layer Perceptron (MLP) Regression**:

- **Performance**: MLP showed a test MSE of 0.3087, the highest R^2 at 0.9158, a MAPE of 10.61, and the lowest RMSE at 0.5556.
- **Analysis**: As a type of neural network, MLP can capture complex non-linear relationships through multiple layers and neurons. Its superior performance across all metrics suggests that the depth and breadth of the network are well-suited to the dataset. The lower MSE and RMSE indicate better prediction accuracy, and the higher R^2, along with the lowest MAPE, suggest a strong fit to the variance in the data.

In our study, Kernel Ridge Regression (KRR) emerges as a straightforward, less parameter-intensive model, albeit with limitations in modeling complex non-linear patterns, unlike the Multi-Layer Perceptron (MLP). The latter excels in capturing intricate data relationships, as indicated by its leading performance metrics (lowest MSE and RMSE, highest R^2, and lowest MAPE), but at the cost of increased computational demands and potential overfitting risks. Support Vector Regression (SVR) finds a middle ground, balancing complexity with interpretability and demonstrating versatility with varied dataset structures. As far as computational demand required by these models, we found that the KRR model required the longest computational training times of 4 hours, while the SVR model required 46 minutes and the MLP model required 1 hour and 29 minutes. This significant variance in computational training times underscores the importance of considering resource allocation and efficiency when selecting models, particularly in scenarios where time constraints or computational resources are limited.

Our results show slight deviations from the original studies, particularly in the performance of models like Random Forest and KNN. These differences might stem from our dataset's larger size (24,936 records compared to 24,317), potentially impacting model generalization and learning capabilities.

When selecting a model for regression tasks, one must weigh factors such as dataset characteristics, computational resources, and non-linear relationship modeling. While KRR offers simplicity, SVR provides flexibility, and MLP demonstrates superior capability in handling complex patterns. This decision-making is further nuanced by comparing our results with the original studies, where models like Random Forest and KNN showed robustness in handling non-linearities. The varying dataset sizes between our study and the original research underscore the need to consider data volume's impact on model performance and suitability.

6 Conclusion

This study extends the work of Choy et al. by introducing and analyzing Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP) in real estate valuation, offering new insights into the application of machine learning in this field. By implementing and comparing three distinct machine learning models – Kernel Ridge Regression (KRR), Support Vector Regression (SVR), and Multi-Layer Perceptron (MLP) – our research contributes to a growing body of work on the use of ML techniques in the Real Estate industry.

Our findings highlight the unique strengths and limitations of each model in handling real estate data. KRR, though offering straightforward implementation and efficiency in smaller datasets, required the longest computational training time and was slightly less adept at modeling complex non-linear patterns compared to MLP. SVR struck a middle ground, demonstrating flexibility and effectiveness in high-dimensional spaces, but also required considerable computational time. MLP, on the other hand, showcased superior performance in terms of accuracy, handling non-linear relationships with the highest efficacy, as indicated by its leading metrics, though it necessitated substantial computational resources and posed a risk of overfitting.

These insights underscore the importance of considering dataset characteristics, computational demands, and model interpretability when selecting machine learning tools for real estate valuation. The choice between KRR, SVR, and MLP should be contextually based, taking into account the specific requirements of the real estate application. Notably, our expanded dataset's results differ from the original study, particularly in the performance of models like Random Forest and KNN, highlighting the potential impact of dataset size on model efficacy.

The usefulness of these models should not be discounted. They achieved high accuracy rates in predicting the sale price of the homes in Hong Kong. This information can help home buyers and sellers make informed decisions when performing real estate transactions. Companies can make use of these types of models to help drive decisions for greater profits by discouraging the purchase of overpriced homes. These models also have the potential to help an individual find deals in the market of homes that are underpriced. The integration of machine learning in real estate research offers a promising future for more accurate, efficient, and sophisticated analysis. As the field continues to evolve, it is imperative to continually assess and refine these models, ensuring they remain effective tools for professionals in the dynamic landscape of real estate.

References

[1] Choy, L.H.T.; Ho, W.K.O. The Use of Machine Learning in Real Estate Research. Land 2023, 12, 740. https://doi.org/10.3390/land12040740

7 Contribution

This section describes the individual contributions of each member of our team, comprising two members: Robert Conner and Christina Moody. Our collaborative efforts were important in both the development and documentation of the project.

- **Robert:** The primary role of Robert involved the development and execution of all Python code used in the project. This encompassed the creation of the algorithms, data analysis, and implementation of machine learning models. Additionally, Robert was a part of the iterative editing and refining of the final document.
- **Christina:** Christina's contribution was instrumental in the iterative process of code development and refinement. Christina collaborated closely with Robert, providing insights and suggestions to enhance the functionality and output of the Python code. Furthermore, Christina was responsible for drafting the initial versions of the paper, laying down the foundational structure and content upon which the final manuscript was built.

The synergistic collaboration between our group members was crucial in developing the final document. While the tasks were separated to appropriately fit our group size of 2, we were both greatly involved with each others work to produce a collaborative and in-depth project conclusion.