Exploring Nonlinear Relationships Between the Real Estate Housing Price And the Distance To Nearby MRT Station Using Smoothing Methods

OBJECTIVE-

The objective of this project is to model and analyze the nonlinear relationship between the real estate housing price and the distance to its nearest MRT station using smoothing methods such as Bin Smoothing, KNN Smoothing and Kernel Smoothing. The goal is to determine which smoothing technique provides the most accurate prediction based on cross-validation and test error comparison.

REPORT-

- ### DATA DESCRIPTION-

Dataset Name: REAL ESTATE VALUATION

This dataset contains real estate transaction records in Taipei, Taiwan. It includes structural and accessibility-related housing characteristics, including distance to the nearest MRT station and housing price per unit area. There are total **414 observations** in the dataset.

	VARIABLE NAME	VARIABLE TYPE	UNITS
Independent Variable (X)	Distance to the nearest MRT station	Continuous	Meter
Dependent Variable (Y)	House price of unit area	Continuous	10000 New Taiwan Dollar/Ping

Source: UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set

1. Why did you choose this dataset?

This dataset is openly accessible and very well-organized with a lot of non missing values available and it provides both the independent and dependent variables in continuous numeric form. Additionally, the relationship we are studying here that is how proximity to public transport like MRT stations influences housing prices in real estate market is practical, relevant, and frequently studied in real-world.

2. Why do we expect the relationship to be nonlinear?

The effect of accessibility is much stronger when a house is very close to an MRT station, resulting in noticeably higher prices. However, as the distance increases, the influence on price gradually weakens rather than decreasing at a constant rate. This suggests that the price decline does not follow a straight-line pattern, but instead displays a smooth, nonlinear trend.

---

3. What type of nonlinear relationship is suspected?

We suspect a decreasing non linear trend where the price drop at the beginning will be rapid but as distance increases the rate of drop in price will slow down.

DATA CLEANING-

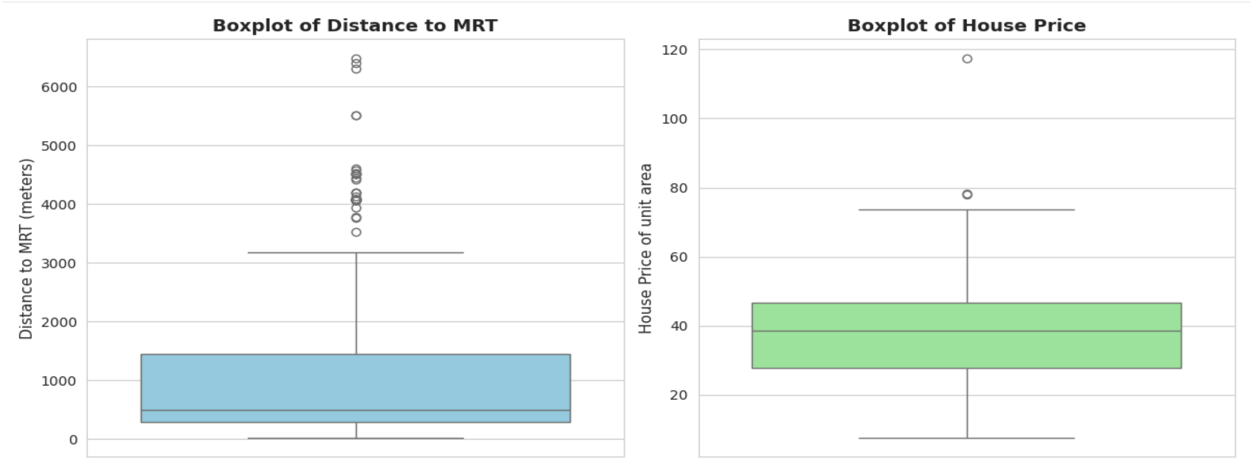
1. How many missing values were found and removed?

There were **no missing values** in either the distance to the nearest MRT station or the housing price variables, so no records were removed due to missing data.

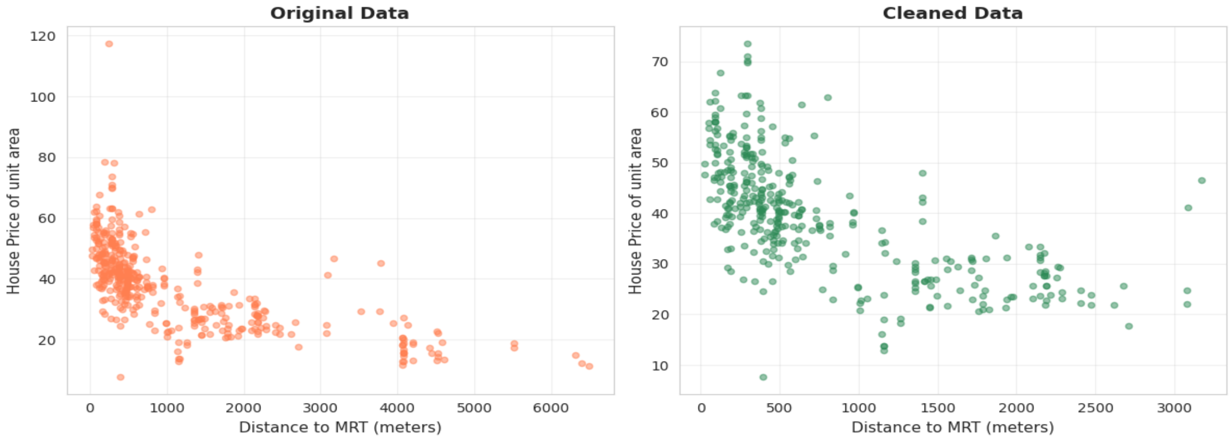
---

2. What criteria did you use to detect outliers?

Outliers were detected using a combination of **boxplot inspection and the IQR (Interquartile Range) rule**.
Values that fell **below Q1 − 1.5×IQR** or **above Q3 + 1.5×IQR** for either housing price or distance were considered outliers. Using this formula we noticed **37 outliers in diatance to MRT** values and **3 outliers in housing price** value. We removed this outlier values and after removing outliers we have **374 observations**.



We plotted the X vs Y scatterplot before and after cleaning the data.



3. After cleaning, how would you describe the visual pattern between X and Y?

The pattern resembles a **negative exponential curve** where housing prices drop sharply when moving slightly away from the MRT station but after a certain distance, the rate of decrease becomes slower and more gradual.

- **Houses located very close to MRT stations tend to have higher prices**,
- and housing prices **gradually decline** as the distance from the MRT station increases. The decline is **steep near shorter distances** and becomes **flatter** as distance grows, indicating a **smooth, curved relationship** rather than a straight-line form.

---

4. Do you think a parametric regression would fit this data? Why or why not?

A simple parametric regression (such as linear regression) would not fit this data well**,** because the relationship between distance and price is not linear**.**

Model Building and Cross-Validation-

We have split our entire dataset in 80% train(299 observations) and 20% test(75 observations).We have applied 3 smoothing techniques –

1. BIN SMOOTHING
2. KERNEL SMOOTHING(GAUSSIAN)
3. KNN SMOOTHING

For each technique we chose a range of hyper parameters and recorded the average MSE and MAE values for different hyperparameters using 5 fold cross validations.

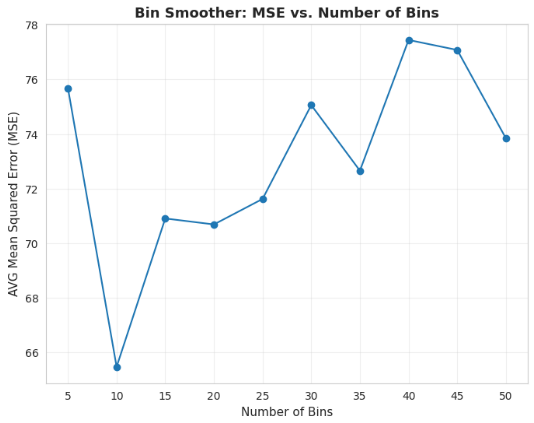
1. What hyperparameter(s) did you tune for each smoother?

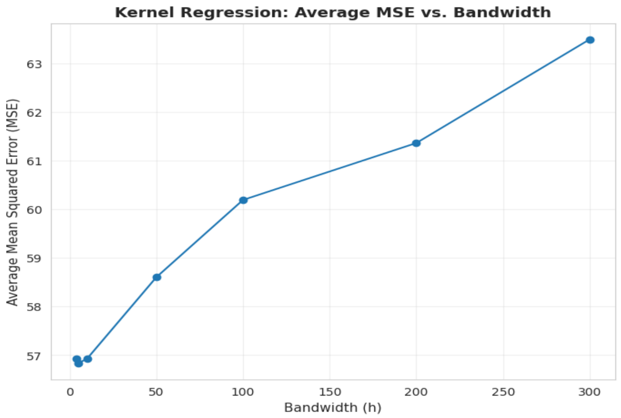
Smoother	Hyperparameter Tuned
Bin Smoother	:Number of bins (**B**)
KNN Smoother	:Number of neighbors (**k**)
Kernel Smoother	:Bandwidth (**h**)

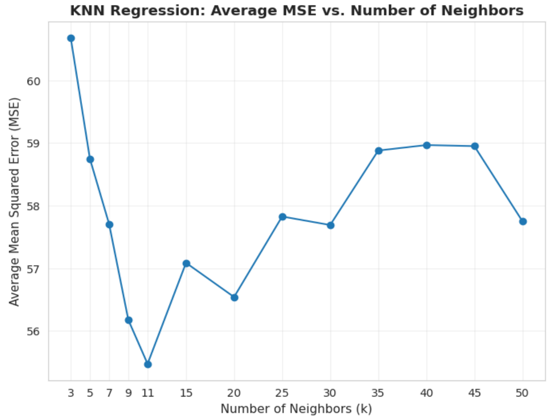
2. How did the validation error change across the hyperparameter range?

The validation error followed a **bias–variance trade-off pattern**.

- **Small values** of hyperparameters led to **over fitting**, producing a curve that was too rough resulted in high validation error as it tried to fit all the train observations that is decreased bias and increased variance.
- **Large values** caused **under fitting**, over smoothing the relationship which also increased validation error that is increased bias and decreased variance.
- The **lowest validation error** occurred at a **moderate hyperparameter value**, indicating an optimal balance between bias and variance.







3. What number of folds did you use in cross-validation, and why?

I used **5-fold cross-validation** because it provides stable and reliable performance estimates while keeping computational time and cost reasonable as it is a medium sized dataset (299 observations) which leaves ~**59 validation samples in each fold.**

4. Which error metric(s) did you use and what are the pros and cons of your choice?

I used **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** as error metrics.

Metric	Pros	Cons
MSE	Heavily penalizes larger error values so that we can see noticeable difference in MSE values for different hyperparameters across different smoothers.	Very sensitive to outliers, MSE increases largely for some outlier values which decreases overall model performance even if the model chosen is a good model.
MAE	Does not get affected by outliers much and it has easier calculation simpler interpretation.	Does not penalize large errors as strongly as MSE .Also the MAE values do not differ much from one another in hyperparameter tuning.

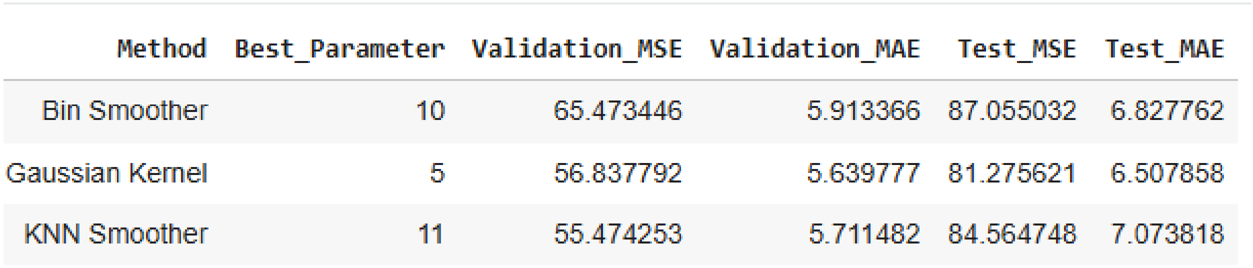
We used both for more balanced evaluation and comparison across smoothers.

5 . Which smoother performed best on the test data, and why might that be?

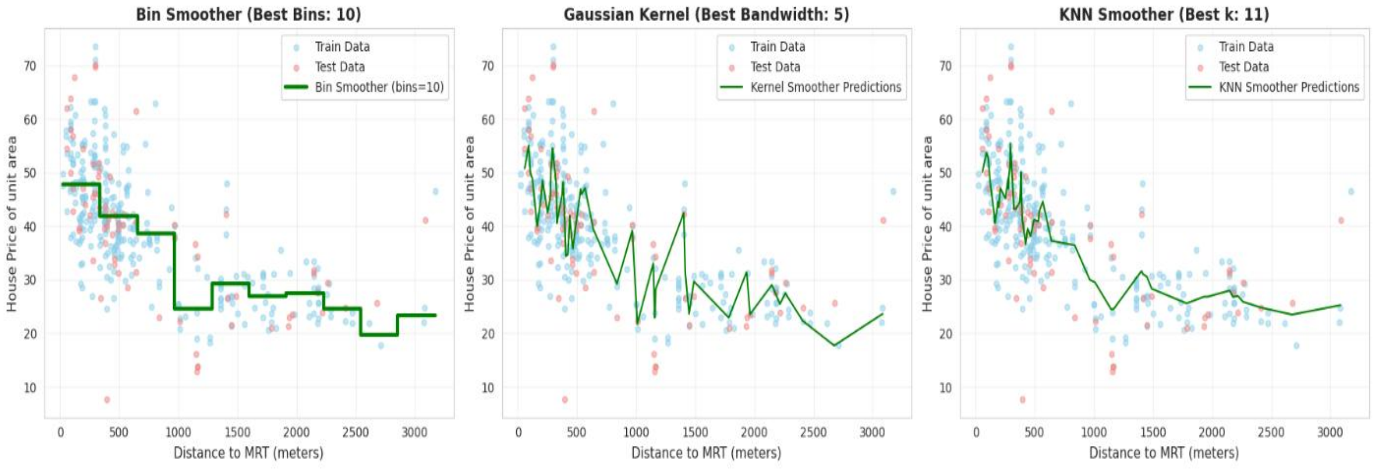
The Kernel Smoother performed the best, achieving the lowest Test MSE**.**

The Kernel Smoother produces a smooth curve across distances, avoiding sudden changes. The bandwidth helps model steep declines in housing prices near MRT stations while maintaining gradual slopes for distant houses. Additionally, the kernel approach weights neighboring observations continuously, which prevents the sharp step patterns seen in the bin smoother and avoids the fixed neighborhood boundaries used in KNN smoothing.

Model Comparison and Discussion-



Method	Best_Parameter	Validation_MSE	Validation_MAE	Test_MSE	Test_MAE
Bin Smoother	10	65.473446	5.913366	87.055032	6.827762
Gaussian Kernel	5	56.837792	5.639777	81.275621	6.507858
KNN Smoother	11	55.474253	5.711482	84.564748	7.073818



1. Which method achieved the lowest test error?

The **Kernel Smoother (Gaussian)** achieved the lowest test error**(TEST MSE-81.27,TEST MAE-6.50**) among all the smoothing methods applied with **the best bandwidth-5**.

2. Was the best-performing model also the smoothest visually?

No. While the Kernel Smoother provided the best predictive performance, it was **not the smoothest curve** as it tried to capture the patterns well and did not over smooth the curve.

3. In what situations could another smoother outperform this one?

KNN or the Bin Smoother might have performed better if the data exhibited **more localized fluctuations** within the neighboring range or within the bin length.

4. What does this project teach you about the importance of hyperparameter tuning in nonparametric regression?

This project demonstrates that hyperparameter tuning is crucial in nonparametric regression. The performance of smoothers depends heavily on the choices of hyperparameters such as the number of bins, number of neighbors, or bandwidth.

- **Too little value of hyperparameter** leads to **overfitting**, where the model tries to capture all the train observations causing increased test mse.
- **Too much value of hyperparameters** leads to **underfitting**, where it over smooths the curve and cannot capture the pattern of the data well.

Finding the **optimal balance through cross-validation** is essential to obtain a model that generalizes well to test data.