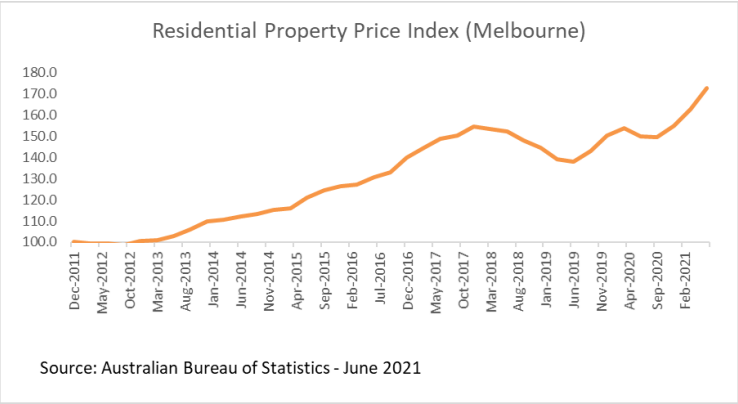
MAXIMIZING YOUR HOME INVESTMENT: A MACHINE LEARNING EXERCISE ON HOUSING

Fuentes, Pecundo, Villegas

INTRODUCTION

BACKGROUND

- Australians traditionally place a high degree of importance on homeownership, making houses in demand (Eslake, 2017).
- There has been a problem with housing in Australia though
    - Over time, rates of houses are rapidly increasing, especially in Melbourne compared to suburban areas



- The cause?
    - Low tax rates
    - Increasing demand but low supply

BACKGROUND

- The COVID-19 pandemic has worsened affordability
- Real estate agents have resumed operations post-lockdown while buyers have not fully recovered yet from financial losses

AUDIENCE

- First-time house buyers
- Families
- Professionals who Buy-and-rent (e.g. AirBnB)

SIGNIFICANCE OF THE STUDY

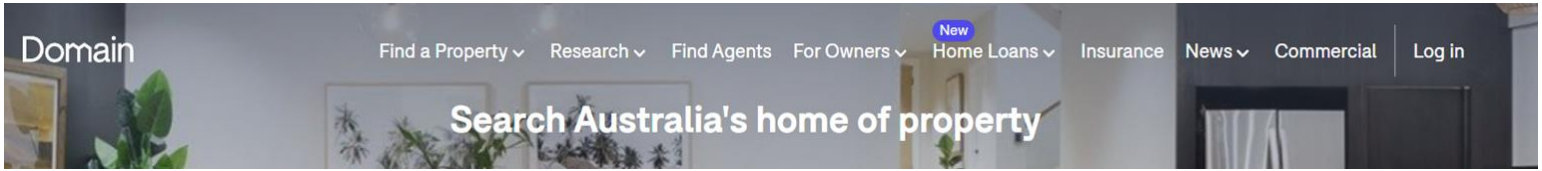
- Value is what buyers are looking for
- Value is dependent on
  - Lifestyle
  - Affordability in the context of Budget
- The study can help buyers regulate their expected expenses based on the predicted price of their preferred house characteristics
  - Choose the house that gives them the most bang for their buck through data-driven decisions
- Careful decision making is required to be able to maximize the value

OBJECTIVES

- Can we predict the price of a house by knowing the house's other characteristics?

- What properties of a house contribute the most to its value?

MELBOURNE HOUSING DATA (DOMAIN.AU)



- Publicly available on Kaggle
- Data scraped from Domain.au; an Australian real estate market website
- Around 34,000+ houses listed in the dataset

FEATURES

Location

Address, Region, Council, Longitude, Latitude

House Features

Rooms, Bathroom, Car Parking Size, Building Size

Setting

Land size, Property Count in the Area, Distance from the Central Business District

METHODOLOGY: PRE-PROCESSING

OBSERVATIONS WITH MISSING “PRICE” WERE REMOVED

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34857 entries, 0 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Suburb         34857 non-null  object
 1   Address        34857 non-null  object
 2   Rooms          34857 non-null  int64
 3   Type           34857 non-null  object
 4   Price          27247 non-null  float64
 5   Method         34857 non-null  object
 6   SellerG        34857 non-null  object
 7   Date           34857 non-null  object
 8   Distance       34856 non-null  float64
 9   Postcode       34856 non-null  float64
 10  Bedroom2       26640 non-null  float64
 11  Bathroom       26631 non-null  float64
 12  Car            26129 non-null  float64
 13  Landsize       23047 non-null  float64
 14  BuildingArea   13742 non-null  float64
 15  YearBuilt      15551 non-null  float64
 16  CouncilArea    34854 non-null  object
 17  Lattitude      26881 non-null  float64
 18  Longtitude     26881 non-null  float64
 19  Regionname     34854 non-null  object
 20  Propertycount  34854 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 5.6+ MB
```


```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27247 entries, 1 to 34856
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Suburb         27247 non-null  object
 1   Address        27247 non-null  object
 2   Rooms          27247 non-null  int64
 3   Type           27247 non-null  object
 4   Price          27247 non-null  float64
 5   Method         27247 non-null  object
 6   SellerG        27247 non-null  object
 7   Date           27247 non-null  object
 8   Distance       27246 non-null  float64
 9   Postcode       27246 non-null  float64
 10  Bedroom2       20806 non-null  float64
 11  Bathroom       20800 non-null  float64
 12  Car            20423 non-null  float64
 13  Landsize       17982 non-null  float64
 14  BuildingArea   10656 non-null  float64
 15  YearBuilt      12084 non-null  float64
 16  CouncilArea    27244 non-null  object
 17  Lattitude      20993 non-null  float64
 18  Longtitude     20993 non-null  float64
 19  Regionname     27244 non-null  object
 20  Propertycount  27244 non-null  float64
dtypes: float64(12), int64(1), object(8)
memory usage: 4.6+ MB
```

- Observations with missing “Price” were removed

CAR, BATHROOM, AND LANDSIZE FEATURES WERE CLEANED

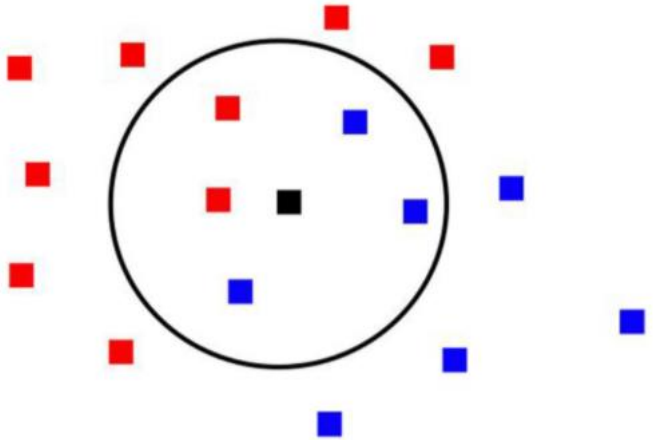
- Removed cases where “Bathroom” had larger number than “Room”

```
df_clean.loc[df_clean['Rooms'] < df_clean['Bathroom']].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 36 entries, 400 to 32868
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Suburb         36 non-null     object
 1   Address        36 non-null     object
```

- Observations with missing “Car”, “Bathroom”, and “Landsize” were imputed using KNN Imputer
  - Following features were considered in KNN Imputer: Rooms, Type, Car, Bathroom, Land size
  - Imputation was done separately for training and test set

K-NEAREST NEIGHBORS IMPUTATION



- Maps observations as points on a space, based on the features selected, and locates the nearest neighbors based on common features

- Missing data is predicted based on average value of nearest neighbors; count of neighbors chosen is decided by a parameter k, selected for the model

REMAINING MISSING AND ATYPICAL FEATURES WERE REMOVED

- Observations with missing “Distance” were removed
  - This was done both for training and test dataset
  - 1 observation was removed (1 from training dataset)

- Observations with missing “Propertycount” were removed
  - This was done both for training and test dataset
  - 3 observations were removed (2 from training dataset and 1 from test dataset)

- Total observations were reduced from 34,857 to 27,208 (28% reduction)
  - Most reductions were from missing “Price”

FINAL FEATURES WERE SELECTED

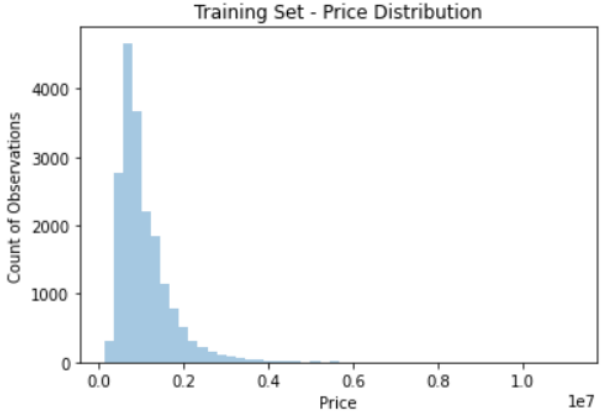
- Input Features:
  - Number of rooms
  - Type of house (dummy encoded)
  - Distance from city center
  - Property count within area
  - Region (dummy encoded)
  - Number of bathrooms
  - Number of car parking spaces
  - Land size

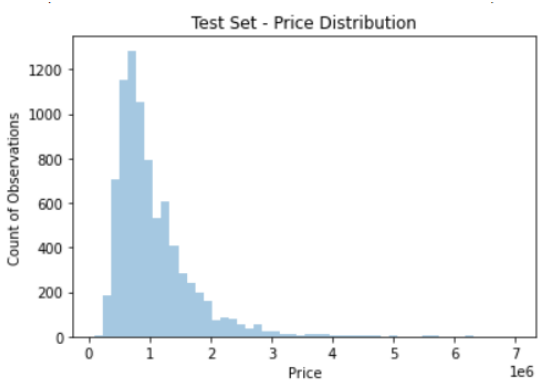
- Target Feature:
  - Price

METHODOLOGY: MODEL BUILDING

MODEL TRAINING

- For model training, the data was split into training and test set
  - Model was trained on the training set (80% of the data)
  - Model was evaluated on the test set (20% of the data)





- 2 Models were tested on the data:
  - Model 1: Linear Regression
  - Model 2: ElasticNet Regression

MODEL 1: LINEAR REGRESSION

Identify *w* and *b*

$$\hat{y} = wX + b$$

While minimizing

$$C(\vec{w}) = \sum_{i=1}^{n} (\hat{y}_i - \vec{y}_i)^2 .$$

Where

$\hat{y}$ = predicted target variable

X = vector of input features

w = vector of coefficients

b = y-intercept

- Model is fit using least-squares method
- Minimizes the difference between predicted $\hat{y}$ and actual y

MODEL 2: ELASTICNET REGRESSION

Identify *w* and *b* $\hat{y} = wX + b$

While minimizing $$C(\vec{w}) = \sum_{i=1}^{n} (\hat{y}_i - \vec{y}_i)^2 + \alpha_1 \sum_{j=1}^{m} |w_j| + \alpha_2 \sum_{j=1}^{m} w_j^2.$$

Where

$\hat{y}$ = target variable

X = vector of input features

w = vector of coefficients

b = y-intercept

- Minimizes the difference between predicted $\hat{y}$ and actual y while considering additional penalty of coefficients.
- Regularization penalty α applied to the weights is added to the cost function

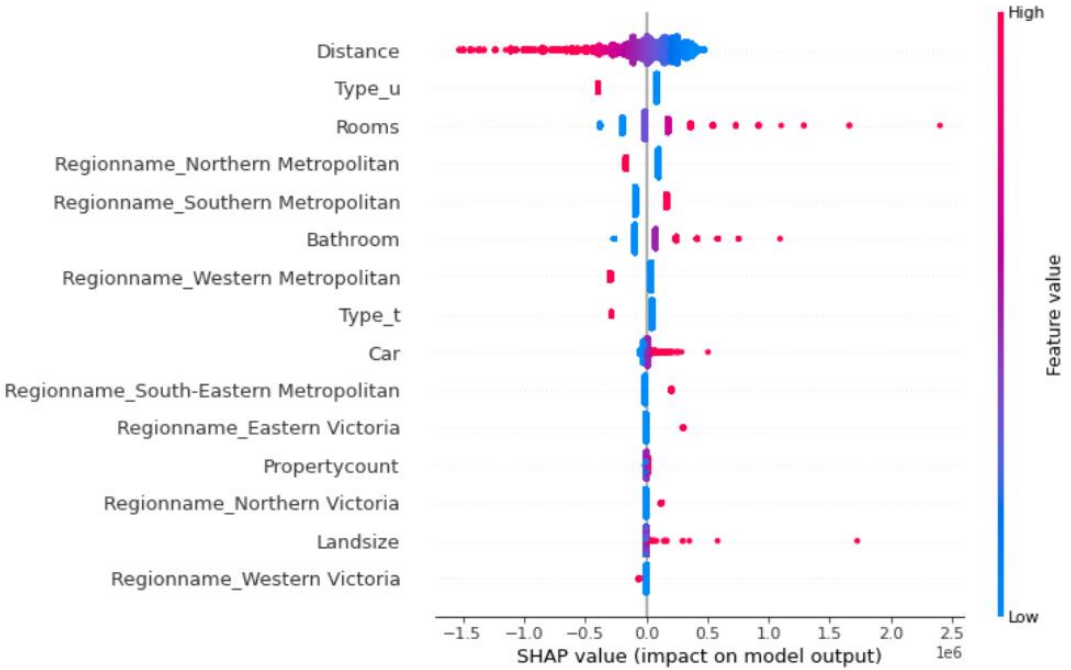
MODEL RESULTS

Model	Training R2	Testing R2
Linear Regression	0.564	0.551
ElasticNet (alpha = 0.00001)	0.564	0.551

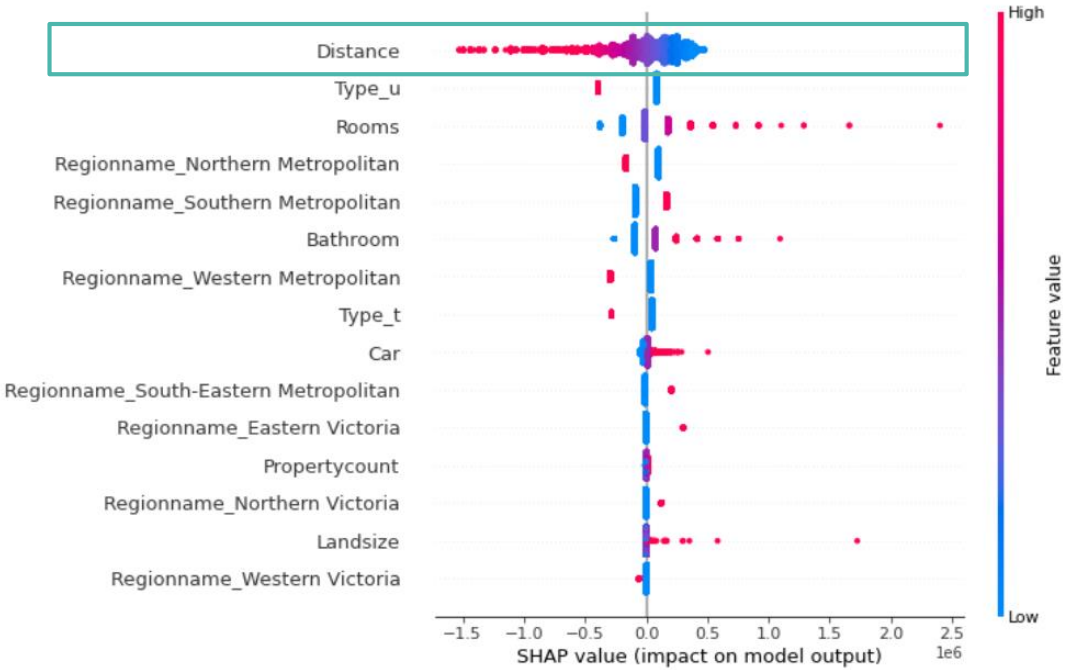
- Both models relatively the same performance; also likely due to alpha being small
- Final model chosen was ElasticNet

INSIGHTS

FEATURE IMPORTANCE (SHAP)

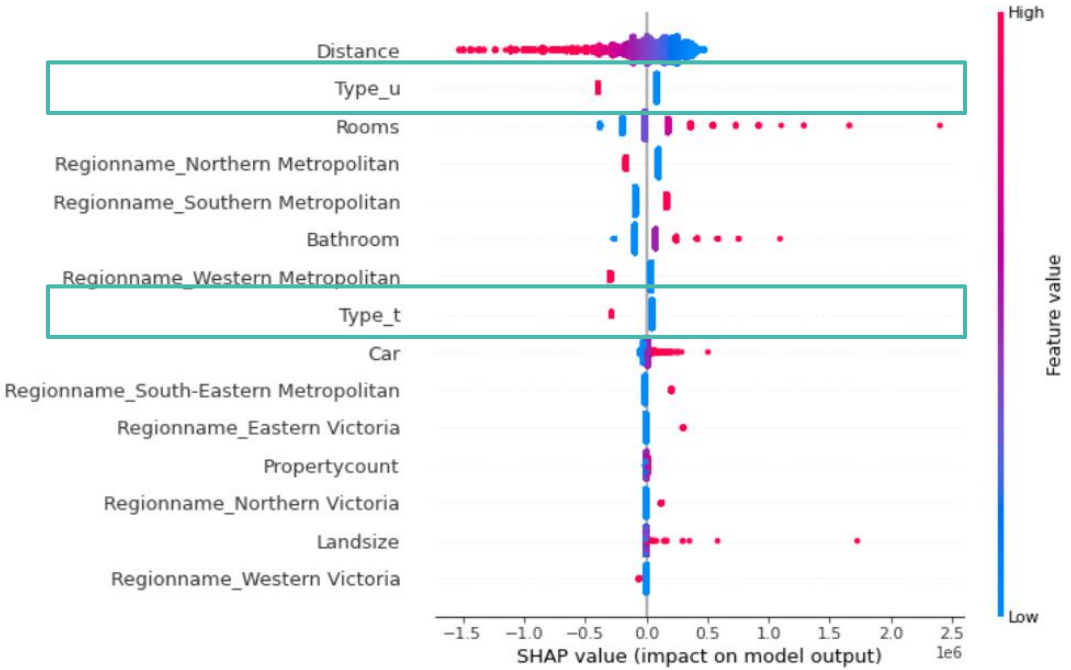


FEATURE IMPORTANCE (SHAP)



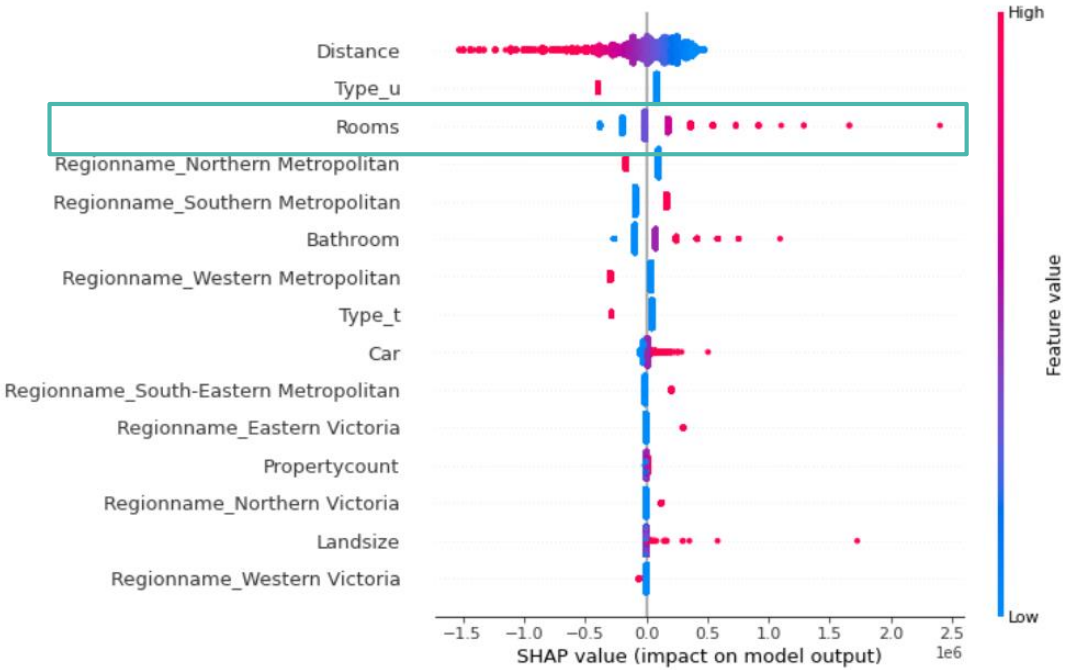
The farther the house is from the central business district, the cheaper the pricing is.

FEATURE IMPORTANCE (SHAP)



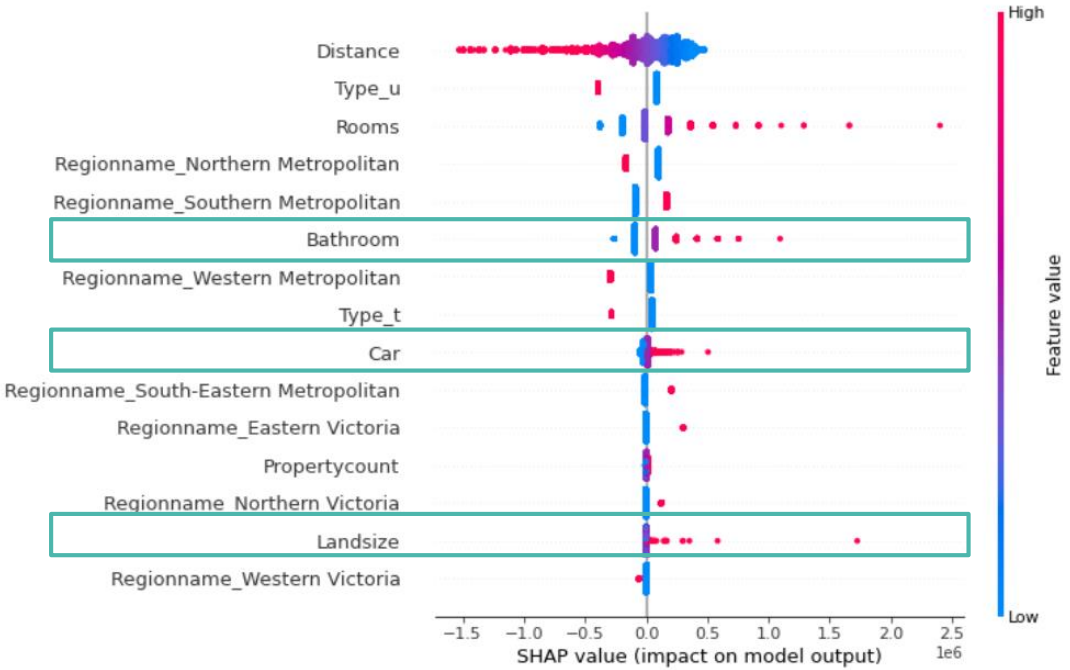
Units and townhouses tend to have Lower Prices.

FEATURE IMPORTANCE (SHAP)



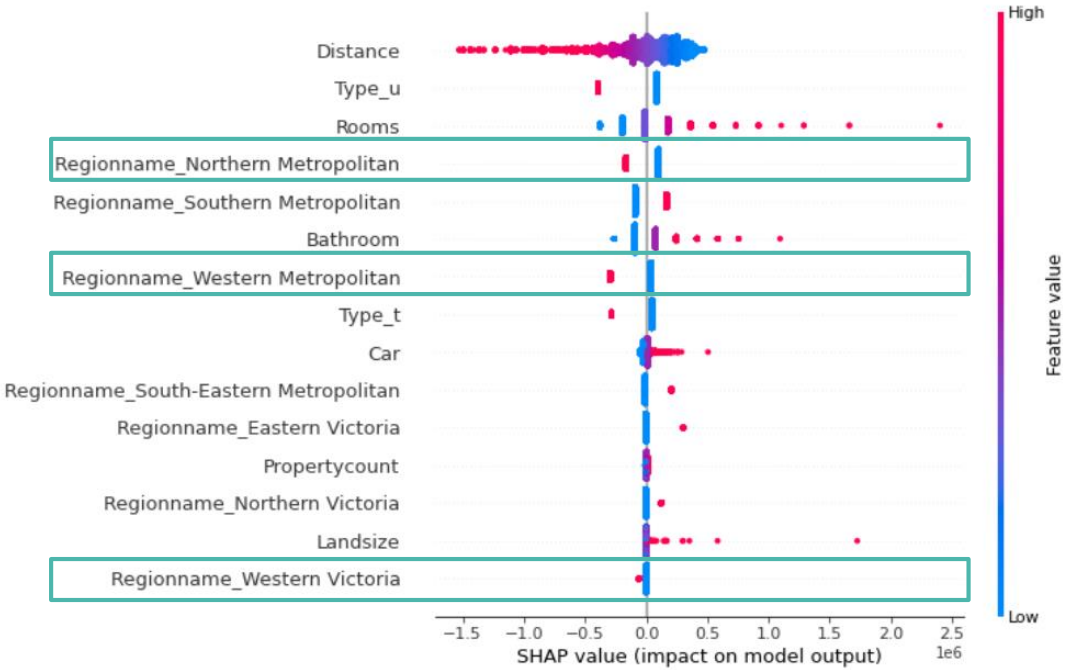
Houses with more rooms have higher prices.

FEATURE IMPORTANCE (SHAP)



Houses with higher bathroom count, higher car parking spot count, and larger land size tend to have higher prices.

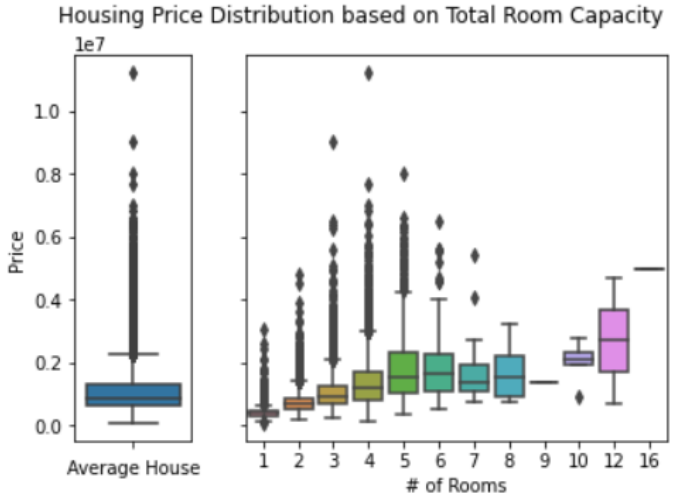
FEATURE IMPORTANCE (SHAP)



House Prices are cheaper in some regions.

CONSIDERATIONS FOR MAXIMIZING HOME VALUE

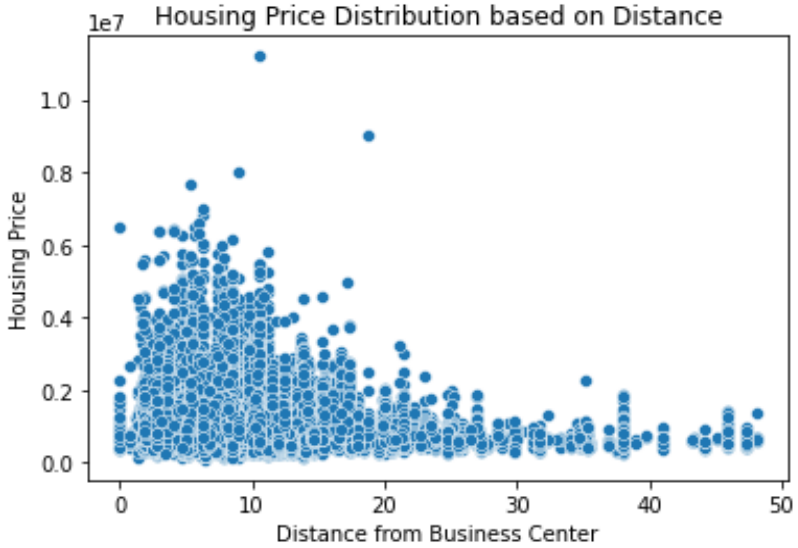
- Having more rooms needed will increase the housing price requirement



- Larger families are disadvantaged with regards to price

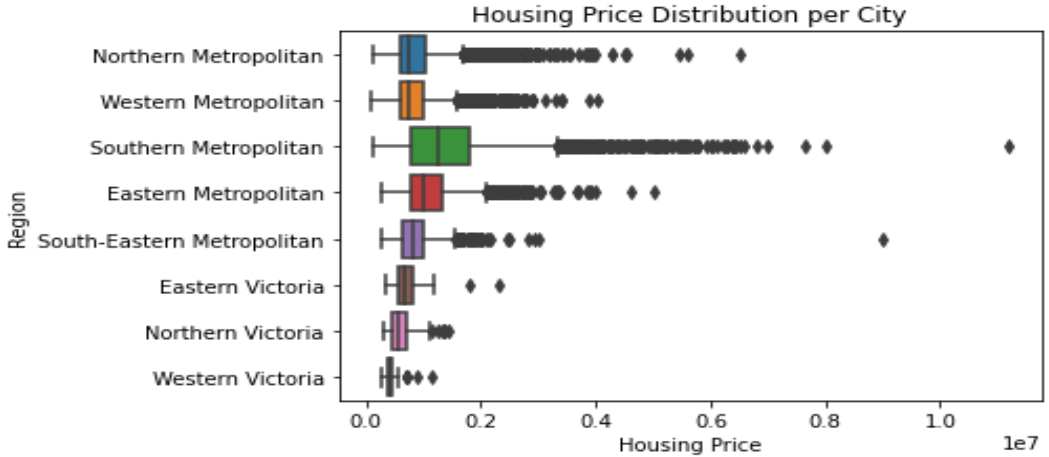
CONSIDERATIONS FOR MAXIMIZING HOME VALUE

- People with larger families who still want lower house prices can consider:
  - Looking into areas farther from the business center



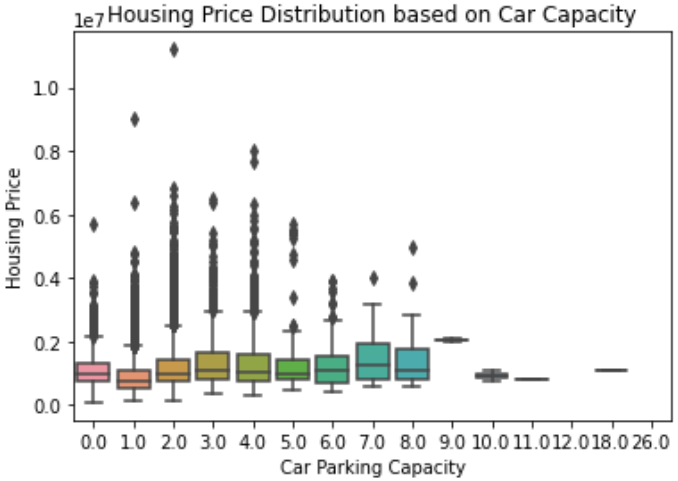
CONSIDERATIONS FOR MAXIMIZING HOME VALUE

- People with larger families who still want lower house prices can consider:
  - Living in cheaper regions such as Northern Metropolitan, Western Metropolitan, and Western Victoria.



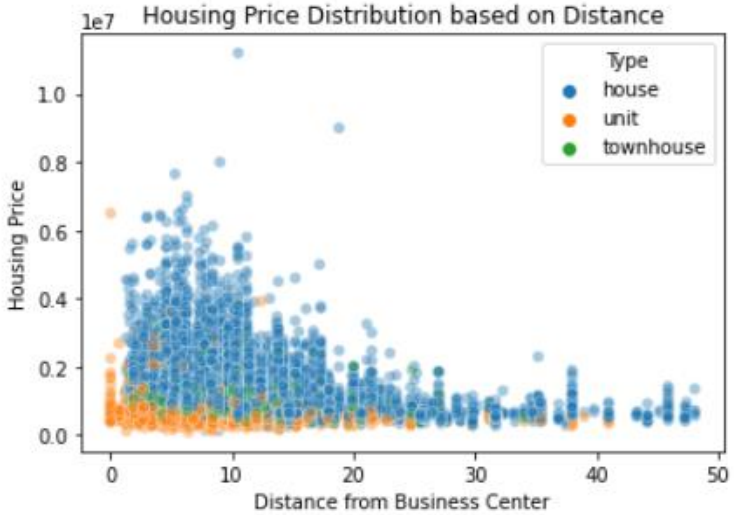
CONSIDERATIONS FOR MAXIMIZING HOME VALUE

- People with larger families who still want lower house prices can consider:
  - Looking for houses that do not have as much car capacity, if any.



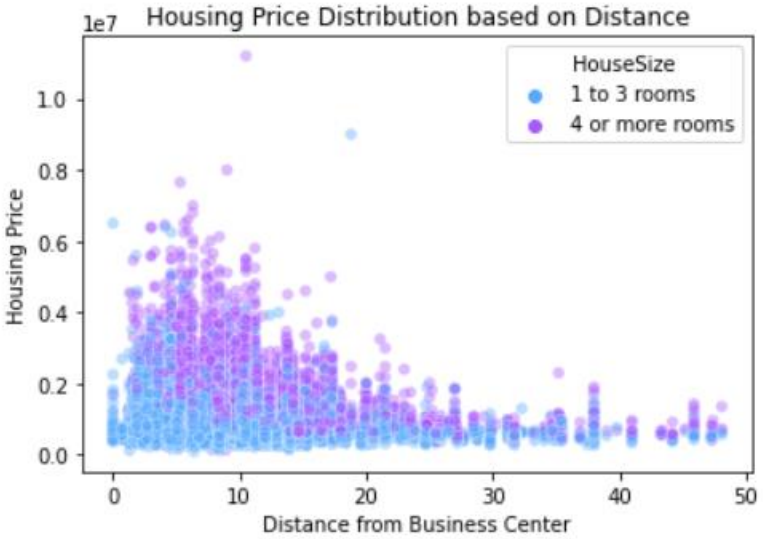
CONSIDERATIONS FOR MAXIMIZING HOME VALUE

- People with larger families who still want lower house prices but need to be near the city center:
  - Can consider looking into townhouses or units instead; low price despite close distance to business center



CONSIDERATIONS FOR MAXIMIZING HOME VALUE

- Single people or smaller families who do not need as many rooms can consider smaller houses while still being near the center



CONCLUSION

CONCLUSION

- We can predict the housing price using the limited features available before the purchase.

- Some of the important features from the Trained Elastic Net model are Distance from business center, Type, and Count of Rooms.

- Larger families are at a disadvantage in terms of pricing requirement due to room count
  - Trade-offs may need to be considered if lower housing price is needed
  - Smaller families have more freedom in other aspects due to low room count requirements