UNIVERSITY OF CALIFORNIA

Los Angeles

Housing Analysis and Prediction in Melbourne Australia

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Applied Statistics

by

Zihao Huang

2020

ABSTRACT OF THE THESIS

Housing Analysis and Prediction in

Melbourne Australia

by

Zihao Huang

Master of Applied Statistics

University of California, Los Angeles, 2020

Professor Ying Nian Wu, Chair

Using Melbourne, Australia dataset from Kaggle.com, we analyzed the factors that determine the type of housing properties from Single-family house, Townhouse, and Apartment using multinomial logistic regression. We also predicted the property price using various machine learning algorithms such as Random Forest, Gradient Boosting, etc.

The thesis of Zihao Huang is approved.

Vivian Lew

Frederic R. Paik Schoenberg

Ying Nian Wu, Committee Chair

University of California, Los Angeles

2020

Contents

List of Figures

1 Introduction

Real estate and the housing market play an important role in the global economy. Take the United States, for example, as of 2018, spending on residential fixed investment was about $785 billion, accounting for about 3.3% of GDP; spending on housing services such as renter's rent and utility payment was about $2.6 trillion, accounting for about 11.6% of GDP; taking together, spending within the housing market accounted for nearly 15% of US GDP in 2018 [1]. At the individual level, owner-occupied real estate accounted for about 25% of households' net worth. Due to the scale and impact of the housing market, it is essential to understand the housing price trends and factors that relate to property type.

The data used in this study is the Melbourne, Australia Housing Market dataset downloaded from Kaggle.com [2]. The data includes listing price data from 2016 to 2018 in Melbourne, Australia. There are two major sections of this study: section 1 analyzes the factors such as region, age and selling method that indirectly relate to the type of property; section 2 utilizes all other factors such as number of rooms and property type to predict property price. By finishing these two analysis, there are interesting findings that will help user better understand the housing market in general.

2 Type of Property Analysis

2.1 Introduction

The purpose of this study is to identify the relationships between geographical, economic, and housing-specific variables to the type of housing (House, Condo, and Townhouse) purchased in the

Melbourne market. The geographical, economic, and housing-specific variables include: Region, Distance from CBD (Central Business District), the Median Income in the Zip that the Property was purchased, the method used to sell, and the Year Built.

By common sense, certain variables like number of bedrooms or square footage have a direct, obvious relationship to the type of housing. However, less is known about other variables such as when sold (first time or not), income class (upper, middle, lower), distance to CBD (close, medium, far), and region as it relates to the median income level. In this analysis, the main research question is: Can we predict the type of residence (h,u,t) as a function of above indirect variables?

During the analysis, the variables are studied and transformed into forms that are better suited to answer the research question and easier to interpret. A multinomial regression model is applied to predict the type of residence. ANOVA fuction is also applied to analyze the significance of the predictors. Finally, model is validated by a 80/20 train/test data for its accuracy, precision and recall.

Our study found that all these variables had a statistically significant relationship to the type of housing purchased. A property closer to the city center, or located in the South, or located in a High-Income area was more likely to be an apartment than a house. Additionally, a house built more recently was more likely to be a Townhouse than a House. This research provides valuable information on non-traditional variables with respect to how they relate to different types of Housing purchased.

2.2 Data Exploration and Processing

The variables that were used in this analysis is summarized in the Figure 1 on page 3.

Figure 1: Variables List

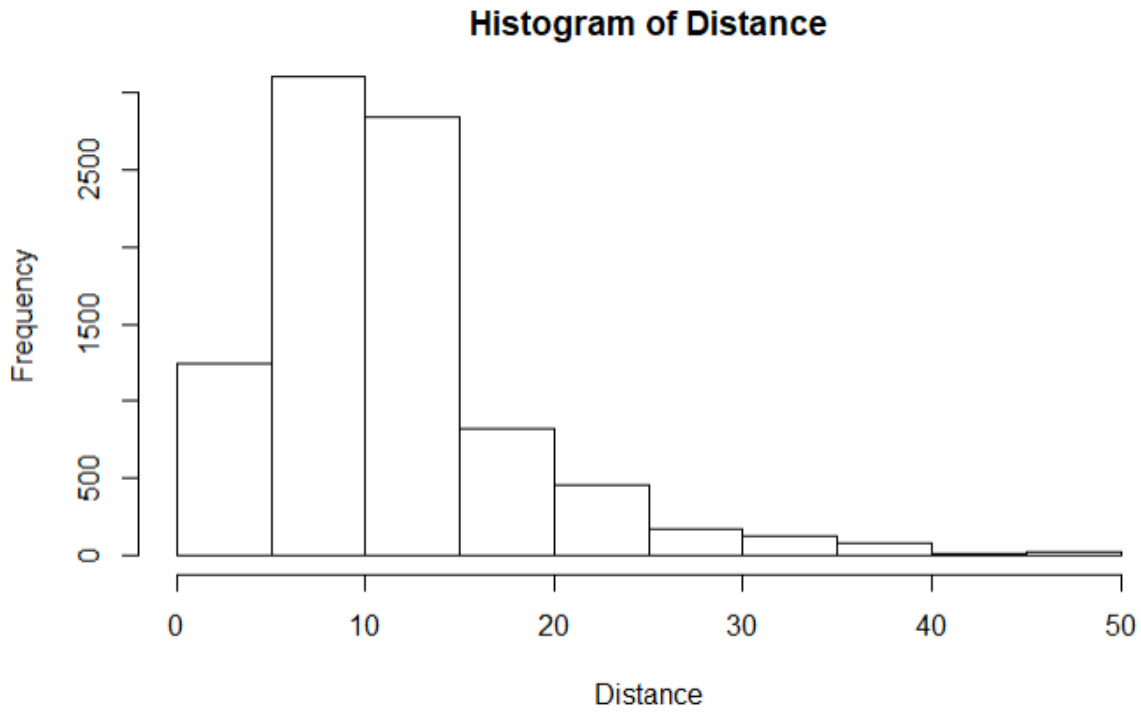
NUMERIC	# OF LEVELS	LEVEL NAMES	DESCRIPTION
Year Built	NA	NA	Year the residence was built
CATEGORICAL			
Type [outcome]	3	H, T, U	House, Townhouse, or Unit (apartment)
Method	2	1st time sold, Sold prior	Whether the residence was new and sold for the first time or if it had been sold before
Region	4	N, E, S, W	Cardinal directions
Distance	3	Far, Neither far nor close, Close	Binned distance to CBD into simpler categories
Income Class	3	Upper, Middle, Lower	Binned median income levels into simpler categories

The variables in this analysis are carefully chosen so that none of the variables have a direct relationship with the output variable. Variables such as number of bedrooms, number of bathrooms, property size, and land sizes are removed. The idea is that those excluded variables have a major impact on the type of property sold. Their effect would be dominant that the small relationship of the indirect variables would be covered, so removing those direct variables allow us to detect the slight relationship between the indirect variables with the outcome. And thus the result would provide a better understanding of the effect of indirect variables that is suited for this analysis.

There are total of 5 categorical variables and 1 numerical variable. Year Build is the the year the property is built. For Year Built, all houses were built before 1830, except one was built on 1100s, which it is considered as outlier and removed. Type is the outcome variable. There are 3 types: H, T, U, which corresponding to Single-Family House, Townhouse, and Unit (Condo/Apartment). There are 9 different methods originally in the data. To make the analysis more interpretable, the 9 methods are grouped into 2 method: First Time Sold vs Sold Prior. The

region variable came from the original data. It represents the cardinal directions of the house in relative to Melbourne. The distance variable represents the distance from Central Business District (CBD). The histogram of the distance as showned in the Figure 2 on page 4 is approximately uniformly distributed. The distance is cut into 3 even groups to enhance interpretability as well: far from CBD, neither far nor close to CBD, close to CBD. The medium income is the average income per post code, and it is cut into 3 groups which is weighted by the population of the post code.

Figure 2: Distance Distribution



The Variable Proportion in the Figure 3 on page 5 shows the weighted proportions of the final variables. The total sample population is 8886. Of the Type category, House consists of 75%

Figure 3: Variables Proportion

Weighted Proportions of Analysis Samples (N=8886)

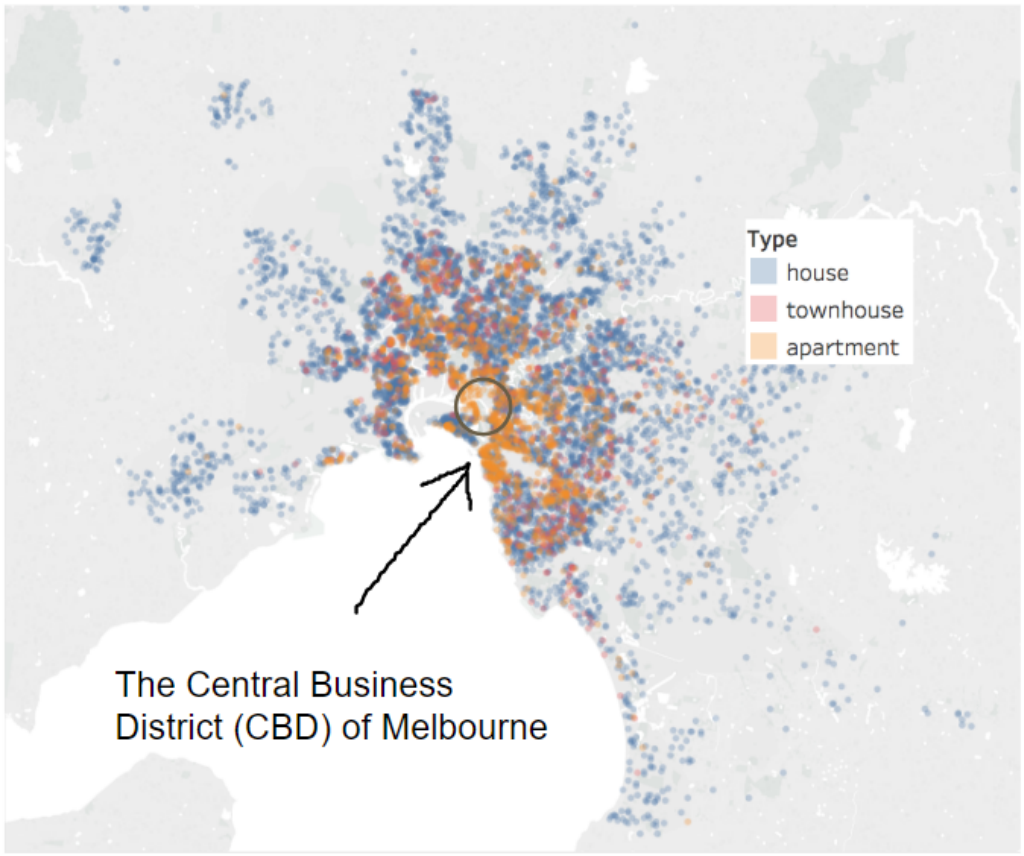
Variable	Proportion	Variable	Proportion
Type		*Income_class*	
House	0.75	Low	0.25
Townhouse	0.08	Mid	0.24
Apartment	0.17	High	0.51
Distance Group		*Method*	
Near	0.33	First Time Sold	0.63
Medium	0.33	Sold Prior	0.37
Far	0.33		
Regions		*Year Built**	1965
East	0.11		
North	0.30		
South	0.34		
West	0.23		

** Mean reported*

of the data; Apartment consists of 17%; Townhouse only consists of 8%. There is a imbalance data on the outcome variable Type, so when I evaluated the model performance, precision and recall are measured in additional to accuracy. Any single category in each variables, including Townhouse, contains more than 750 samples, so I concluded that there are enough sample size to proceed this analysis. For the variable Year Built, it ranges from 1830 to 2019, with mean equals 1965.

The Melbourne Housing Visualization in the Figure 4 on page 6 shows the mapping of properties by Type in Melbourne Australia. The center is the central business district (CBD); the colors in blue are the houses, red are townhouses, and yellow are apartments. From the visualization, the properties in the sample data distribute approximately even around the CBD, with higher density closer to CBD and less density the farther from CBD. Most of the apartment and town-

Figure 4: Melbourne Housing Visualization



house is located near the CBD, and houses are spread across the map. The apartments are much closer to the central business district with an average of 7.6km. Townhouses are on average 10.1km and houses are the furthest with 12.1 km. This intuitively makes sense as land is generally more expensive near CBD, more apartments, the smallest of the three property type, can be built with higher return on investment.

Geographically speaking, the Southest of CBD is the Port Phillip Bay, so the south of CBD is located around the ocean and still very close to the CBD. As a consequence, the southern region is more crowded with higher property density, and more apartments and townhouses are located in the south of CBD compare to other regions. By looking at the income by region, the

south region has the higest median income per post code compare to other three regions. It is the prime location so to speak of Melbourne, and that's why even though a house or townhouse is larger in size, people with more income decided to invest and purchase apartment properties rather than a more spacious residence.

2.3 Multinomial Logistic Regression

I have applied multinomial regression to predict the Types of Residences by distance_group, Regions, Income_Class, Method and YearBuilt. The outcome variables are type of properties which is a categorical variable with 3 types, so multinomial regression would fit this type of outcome variable. The priority of this analysis is to analyze the relationship between the predictor and outcome variable, so outcome interpretability is the key for this analysis. Compare to other multiple class prediction methods such as random forest, support vector machine, neural network, etc., multinomial regression can be better interpreted. Multinomial regression is an extension of binary logistic regression, and binary logistic regression is a sigmoid transformation of a linear function. Therefore, after taking a exponential transformation of the parameters, we could see the linear relationship between the input and output variables, such as increasing variable A by X would increase the chance of outcome variables by Y.

The fundation of multinomial logistic regression is linear regression and logistic regression.

2.3.1 Linear Regression

The dataset of linear regression consists of an $n \times p$ matrix $\mathbf{X} = (x_{ij})$, and a $n \times 1$ vector $\mathbf{Y} = (y_i)$. X is the predictor variables and Y is the out come variables. The final model is of the following

form:

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i,$$

for $i = 1, ..., n$, where $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$ independently. We are training the model to find the weights (β) acoording to a cost function. In linear regression, the cost function is the least square error e, which is:

$$e = (y_a - y_p)^2$$

which is the square of actual outcome minus the predicted outcome.

Since the relationship of the linear regression model is linear, the effect of the parameters can be intepreted by their weights. For example, if the weight of variable A is 2, this means for every 1 increase of variable A, the outcome variable would increase by 2.

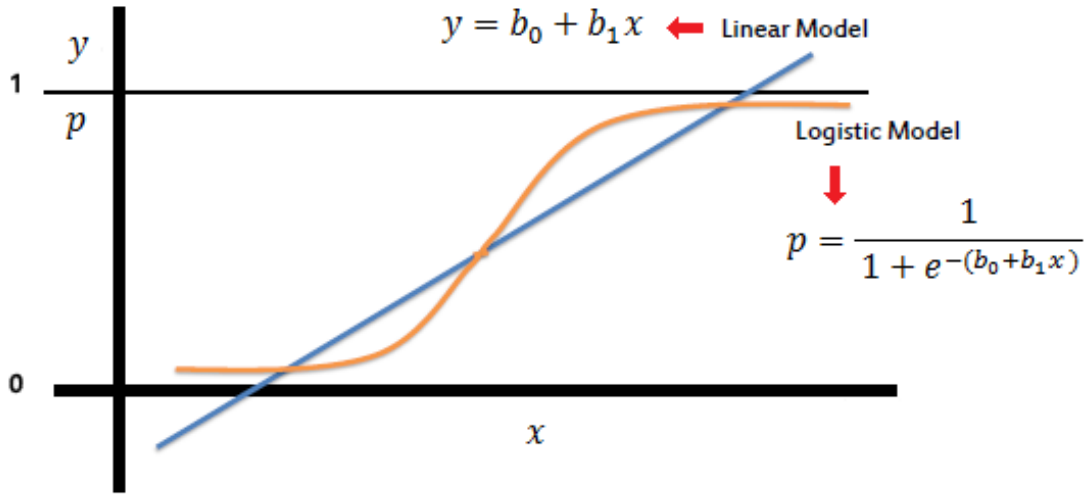
2.3.2 Logistic Regression

The outcome variable of linear regression is continuous variable which range from negative infinity to infinity. To predict a categorical variable with 2 types, we can take a sigmoid transformation of the outcome of linear regression. The following is the sigmoid transformation:

$$y_i = \mathrm{sigmoid}(s_i) = \frac{1}{1 + e^{-s_i}},$$

After the sigmoid transformatio, the outcome variable has range from 0 to 1. Then we can take 0.5 as the benchmark where we can classify y¡0.5 as type 1 and y¿=0.5 as type 2. This is the idea of logistic regression. The figure 5 on page 9 shows the outcome range of linear regression and logistic regression.

Figure 5: Linear Regression vs Logistic Regression



2.3.3 Multinomial Regression

The multinomial regression is an extension of logistic regression that is designed for categorical variable of multiple types instead of 2. We run a logistic regression against each caregory type and assign the out come variable to the type with highest probability.

2.4 Application and Interpretation

I have applied multinomial regression to the data using the multinom() function in the nnet library in R. Then, to analyze the significance of each variable, Anova() function is applied to the model and the figure 6 on page 10 is generated. As shown in the ANOVA table, each predictor variables, except method is significant, are highly significant, indicating all of the predictor variables are useful to explain some aspect of types of house.

Figure 6: ANOVA

Variable	Chi-Square	DF	Pr(>Chisq)
distance_group	773.83	4	<2.2e-16***
region	220.67	6	<2.2e-16***
income_class	37.01	4	1.79e-07***
Method	7.06	2	0.0293*
YearBuilt	2194.34	2	<2.2e-16***

The Multinomial Regression Result table in figure 7 on page 11 shows the details of the Multinomial Regression model of Townhouse and Apartment compared to House. The left is Townhouse vs House, and the right is Apartment vs House. For each model, I showed the OR (Odd Ratio) for the predictor variables compare to base, symbol of * to indicate significance level, 1/OR for better interpretation when OR < 1, and 95% Confidence Interval of the OR. The OR is generated by taking the exponential of the coefficients. To interpret OR, for example, an OR = 1.1 in medium income_class under Town House represents the chance of living in a Town House vs Single-Family House is 10% higher if you are in medium income class compare to low income class.

One finding is that the people who live far from CBD (Central Business District) compared the people who live near are 14 times more likely to be living in a House compared to Apartment, and 4 times more likely to be living in a House compared to a Townhouse. This makes sense because the major reason for living in an Apartment or Townhouse compared to House is that the space is limited, and businesses want to build more of them to earn more money. Another

Figure 7: Multinomial Regression Result

Multinomial Regression Analysis of Townhouse and Apartment compare to House

	Town House				Apartment			
	OR		1/OR	95% CI of OR	OR		1/OR	95% CI of OR
Distance Group (Base: Near)								
Medium	0.66	***	1.50	0.58 0.76	0.27	***	3.65	0.24 0.31
Far	0.21	***	4.84	0.19 0.23	0.07	***	13.89	0.06 0.08
Regions (Base: East)								
North	0.79	***	1.27	0.70 0.89	1.04		0.96	0.94 1.15
South	1.33	***	0.75	1.18 1.51	2.11	***	0.47	1.92 2.32
West	0.58	***	1.73	0.50 0.66	0.59	***	1.69	0.53 0.66
Income_class (Base: low)								
Medium	1.10		0.91	1.00 1.22	0.70	***	1.43	0.61 0.80
High	1.27	***	0.79	1.14 1.41	1.24	***	0.81	1.13 1.35
Method (Base: First Time Sold)								
Sold Prior	0.84	*	1.20	0.70 1.00	1.08		0.93	0.95 1.23
Year Built	1.07	***	0.94	1.07 1.07	1.03	***	0.97	1.03 1.03

Notes: OR = odds ratio, CI = confidence interval
*p < 0.05; **p < 0.01; ***p < 0.001.

finding is that the people who live in the South region compared to East region are 2.1 times more likely to be an Apartment compared to the House. By checking the median incomes and the maps, I found out that the South region is closer the Central Business District, closer to the beach, and have higher average median incomes when compared to other regions, so more people are willing to live in that region. Therefore, this creates a need for increased residences, which leads to a higher chance of being an Apartment than House in this region. This result can be generalized that for regions where there are higher needs for residence, the chances of being an Apartment is higher than average. I also found that people with high average income have 27% higher chance of living

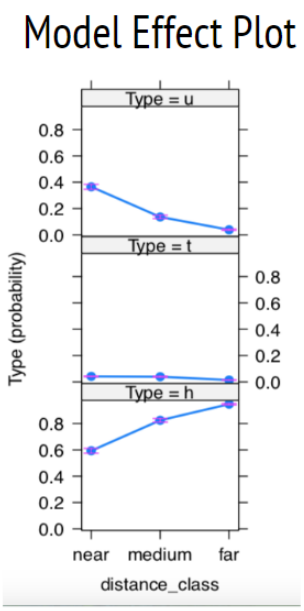
in a Townhouse and 24% higher chance of living in an Apartment compared to people with low average income. This is counterintuitive, but I proposed that people with higher income have more of a choice of where they live. Some of them choose to live in Townhouse or Apartment because the region they live in is nicer. On average, for every 10-year increase in YearBuilt, the chances of being a Townhouse compared to a House is 92% higher and 36% higher for Apartment, meaning that the concept of Townhouse and Apartment is kind of modern. Maybe they come as a result of the population growth of modern era.

Figure 8: Type vs Distance and Model Effect Plot

Actuals (Distance to City Center)

	near	medium	far
(u) apt	901 (30%)	456 (15%)	183 (6%)
(t) town house	221 (7%)	310 (11%)	191 (6%)
(h) house	1,852 (62%)	2,180 (74%)	2,592 (87%)

% is of the column total



The Type vs Distance table in figure 8 on page 12 shows a cross table of the actuals for the 3 outcome classes on the rows as well as the 3 values for distance to city center. The %s are the shares for each cell compared to the column that it is in. The model effect plot in Figure 6 shows the probability of housing outcome for properties near/medium/far from the CBD/City

Center. Here are the interpretation, holding all other variables constant:

1. the probability of being an Apartment (u) goes from 0.4 to 0.2 when going from near to medium from the city center and 0.2 to almost 0.0 when going from medium to far from the city center. This means apartments are more common when close to the Melbourne city center rare when far from the CBD.

2. The probability of being a Townhouse (t) stays very low (almost 0.0) when going from near to medium to far. This means distance from business center has relative low effect on townhouse proportion.

3. The probability of being a House (h) goes from 0.6 to 0.8 when going from when going from near to medium from CBD and 0.8 to almost 1.0 when going from medium to far from CBD.

In short, the above plots indicate the further you move from the city center, the less likely the property is an apartment and the more likely the property is a single-family house.

The Type vs Income table in figure 9 on page 14 shows a cross table of 3 property type vs 3 classes of Median Income based on the post code. The %s are the shares for each cell compared to the column that it is in. The model effect plot on the right shows the probability of housing outcome for properties in low/middle/high income areas. Here are the interpretation, holding all other variables constant:

1. The probability of being an Apartment (u) goes from (apprx) 0.15 to 0.15 when going from low to middle income areas and from 0.15 to almost 0.2 when going from medium to high income areas. This indicates as people have more income, there are high probabilities for them to live in an Apartment compare to a house.

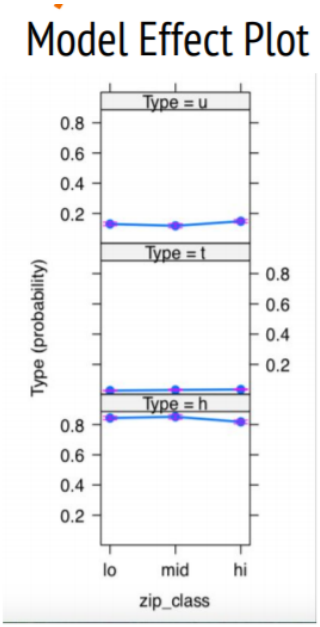
2. The probability of being a Townhouse (t) stays very low (almost 0.0) when going from

Figure 9: Type vs Income and Model Effect Plot

Actuals (Income Areas)

	lower	middle	upper
(u) apt	242 (11%)	190 (9%)	1108 (25%)
(t) town house	157 (7%)	175 (8%)	390 (9%)
(h) house	1,849 (82%)	1,776 (83%)	2,999 (67%)

% is of the column total



low to middle to high income areas, which indicates income class does not have much impact on whether the property being a Townhouse or not.

3. The probability of being a House (h) stays around 0.8 when going from low to middle to high income areas, which indicates income class has little impact on whether the property being a house or not.

In short, the above plots indicate that as you go from low/middle to high income areas, it is more likely that the property is an apartment. This conclusion reaffirms that people with higher income may choose to live in apartment for their convenience such as environment and commute, as we concluded from the Melbourne Housing Visualization,.

2.5 Model Validation

The original data has been randomly splitted into 80% train data and 20% test data. The 20% test data is used for model validation. The House vs Townhouse and House vs Apartment confusion matrix are built based on the test data. For each confusion matrix, we filtered down to just the samples that had those 2 actual labels. From there, I chose the "predicted" label by taking the predicted label with the highest probability. The result for the 20% test data has been shown below:

Figure 10: Confusion Matrix

House vs Townhouse

		Actual			
		house	townhouse	Accuracy	88%
Prediction	house	1,262	125	Precision	29%
	townhouse	47	19	Recall	13%

House vs Apartment

		Actual			
		house	apartment	Accuracy	83%
Prediction	house	1232	205	Precision	61%
	apartment	77	119	Recall	37%

These are the definition of Accuracy, Precision, and Recall: the accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined; Precision is of Positive Predictions, how many did you get correct; Recall is of positive actuals, how many did you predict positive. In mathematical formula:

$$Accuracy = \frac{Correct\,Cases}{Total\,Cases}$$

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

$$where\, tp = true\, positive, fp = false\, positive, fn = false\, negative$$

The accuracy is quite high, but due to there is class imbalance in our data, we cannot simply treat accuracy as our final metric. The precision and recall for house vs apartment is not low but for house vs townhouse is quite low. This is due to there are much more houses than townhouses and apartments. The result is acceptable because the focus is to analyze the relationship of variables to type of property. If the focus is to improve the precision or recall of the model, we can modify the cost function in the multinomial model to put higher panality on wrong prediction of the type with smaller sample size.

2.6 Type of Analysis Conclusion

This project was interesting because there are variables that have very high prediction power (land-size, number of bedrooms/bath). I decided to not use these typical predictors and threw them out of the model with a goal to get a more "interesting" model rather than the most accurate model. After all, I can predict the type of residence using the indirect variables: method, region, income class, distance group, and Year Built to find that they are all significantly related to type of housing. Most of our findings confirm our intuitions which are, that if a place has a better habitat such as better natural resource (ocean, lakes, or mountains) or better community, more people would choose to live in a smaller residence such as an Apartment or Townhouse rather than a more spacious Home. And because of these needs, more Apartments and Townhouses are built. Therefore, the ultimate factors of that determine the types of house is Supply and Demand, and that is generally true for

most of the products in the modern market.

The results of this analysis are important because people who would like to purchase a property will need to decide what type of residence to invest in. This analysis can help them to make informed decision on what are some factors they can investigate and what are their impact. By understanding how income impact choices of residency people make, we can make better purchasing decision based on the income we have now. Also, there is an intuition of having a larger income related to purchasing a larger residence, but this study shows us that people with higher income may choose to live in a smaller residential type like condo or townhouse because that is more convenient to them. Lastly, residence types reflect the consumers preferences and needs. If we find significant results, we will be able to advise and predict which type of residence the consumers want, and we can sell these predictions to housing construction companies to help them construct the type of houses that suited customers' need.

3 Price Prediction Analysis

3.1 Introduction

The key metric that anyone interests in a residential property is the price of the property. In this analysis, we will be using property related factors such as type of property, number of rooms, region to predict the price of property in Melbourne Australia. In doing so, we can provide an objective property value for both the buyer and seller in real estate trading that would help them to make better informed decision when buying/selling a property. Although the final selling price are only determined by the buyer and seller, the price prediction could help them to set a fair

expectation even when they plan to buy/sell the property. This could help them make better budget plans too.

In the following section, the Melbourne Housing dataset from Kaggle will be explored and important features will be extracted for the analysis. Then, multiple models will be running on the both 70% training data and 30% test data. Models include multiple linear regression, random forest, adaboost regression, neural network and more. The model with the highest coefficient of determination (R^2) on test data will be selected. Finally, this analysis will be wrapped up with interpretation and conclusion.

3.2 Data Exploration and Feature Engineering

The data used in this Price Prediction Analysis is the Melbourne Housing dataset from Kaggle.com [2]. The data orginally contains 63043 rows of data, but we only need the property with price listed. After dropping all rows with missing price, there are 48433 rows of data remained. The remaining rows of data has no missing rows.

There are additional variables that are derived from the data parameter. They are month, year, year and month, and number of days from the minimal date of the data. Those parameters are extracted to capture the relationship between the date in various form with the price of property.

From the Price vs Type in figure 13 on page 20, we can see that in average, house (h) has higher price than townhouse (t), and townhouse (t) has higher price than apartment (u). This intuitively makes sense that apartment is the cheapest among the three property type, since house owns the yard, fixture, and land; townhouse owns fixture and land and shares the yard; and apartment only owns the interior fixture and shares all common spaces such as exterior wall and yard

Figure 11: Variable Counts

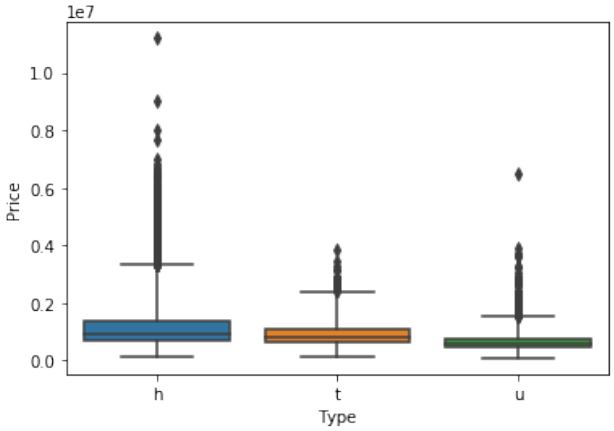
```
In [25]: df.count()

Out[25]: Suburb          48433
         Rooms           48433
         Type            48433
         Price           48433
         Method          48433
         Postcode        48433
         Regionname      48433
         Propertycount   48433
         Distance        48433
         CouncilArea     48433
         Year            48433
         Month           48433
         YearMonth       48433
         DayDelta        48433
         dtype: int64
```

All remaing variables have no missing data.

Figure 12: Price Distribution

The property price is right skewed, with mean = $997,898, median = $830,000 and maximum = $11,200,000.

Figure 13: Price vs Type Boxplot

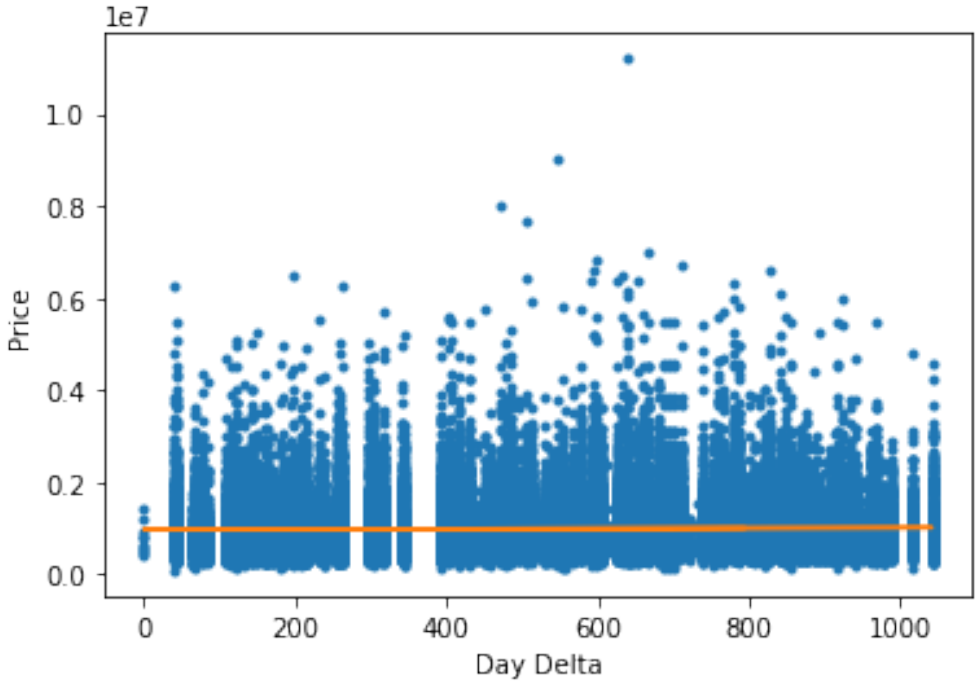


and does not own the land. The price range of house is also larger than the other two property type. The Interquartile range (IQR) for house is $659000, for townhouse is $440000, for apartment is $275000. Lastly, we notice that there are a lot of property price are considered outliers in all 3 type of housing. The property price is considered an outlier when the price lies 3*IQR above the third quartile or 3*IQR or more below the first quartile. This is partially due to we have a right skewed distribution in Price.

From the Figure 14 on page 21, there is a slight positive trend between Price and Daydelta, where Daydelta is defined as the day difference between the date the property is sold and the minimum date in the data which is 2016-01-28. The slope of the regression line = 36, which means keeping other variables constant, for each day increased from 2016-01-28, the price of the property increases by $36. This yields to a price increase of $13,149 per year. The increases is about 1.3% year of year taking the average property price around $1 million, so the difference in time has no prominent effect on the price of property.

From the Figure 15 on page 22, we can see as the distance increases, the price is generally decreased. The correlation between Price and distance is -0.254. Also, as number of rooms

Figure 14: Price vs Daydelta

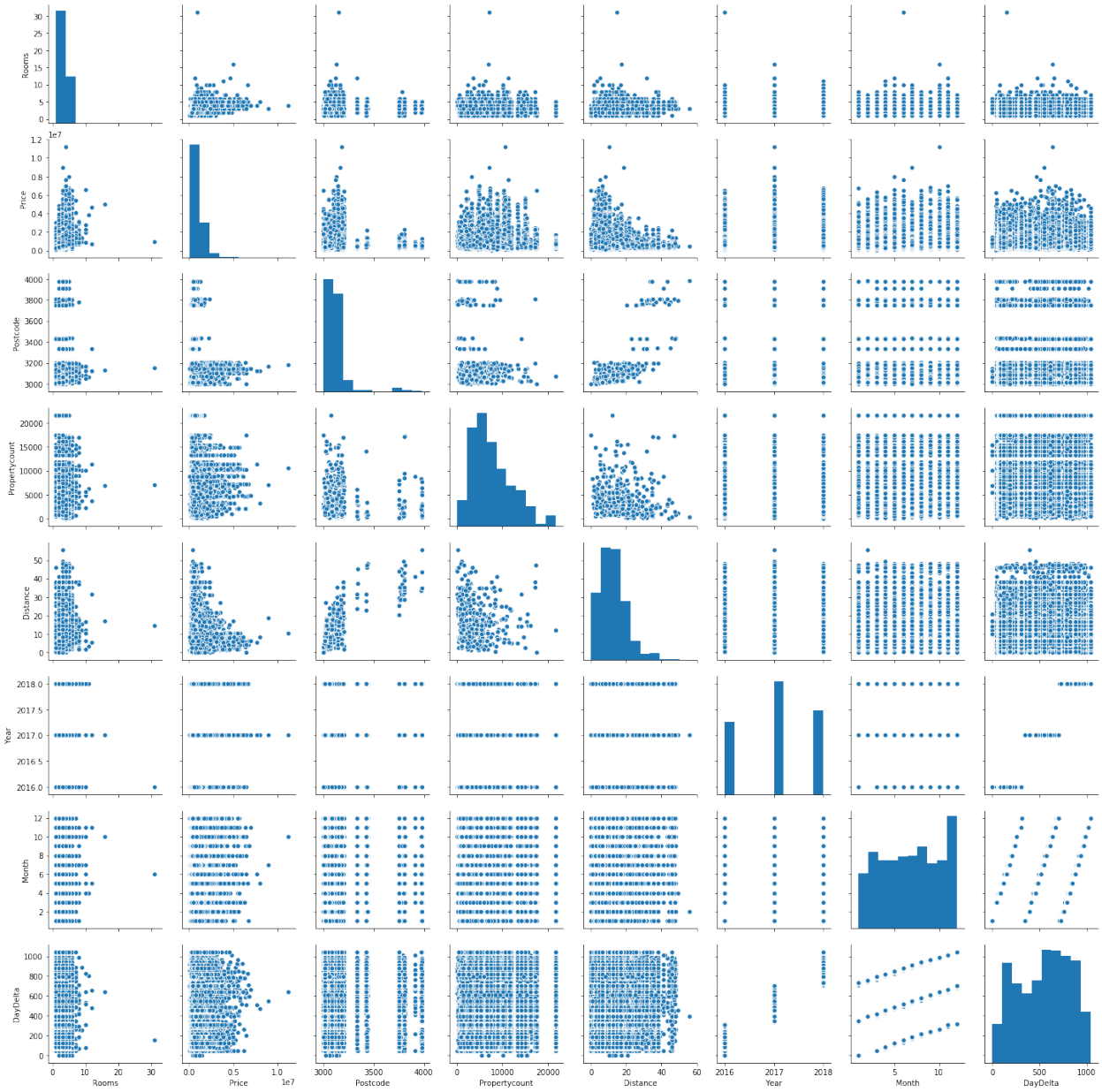


increase, the property price increases as well. The correlation between Price and number of rooms is 0.412. All other numeric variables have little correlation with Price. These findings match our intuition such that as number of rooms increase in a property, the property size is generally increased, so will the property price. Also, the distance is defined as the distance from the Central Business District, so the closer the distance should mean the better the location of the property, so will the property price increase.

3.3 Implementation and Validation

The modeling of Price Prediction Analysis is implemented in Python using the sklearn package. The original data was split into 70% train data and 30% validation data. The models are trained on the train data. The trained model is then fit on both train data and test data to predict the price.

Figure 15: Correlation Matrix



Correlation Matrix for all numeric variables in data

The predicted price is compared with the actual price to obtain the score for the model (coefficient of determination R^2).

The Figure 16 on page 24 shows the training scores and validation scores for all 19 models used in this analysis. The training score comes from the training data and the validation score comes from the validation data. The outcome variable price is a continuous variable, so regression type of models could be applied to predict the outcome. We have applied multiple linear regression, Ridge regression, Lasso Regression, Support Vector Machine, Random Forest, Adaboost regression, Gradient Boosting, and Neural network with the sklearn package in python. The goal of this analysis is to predict the most accurate price. Therefore, we will choose the the model that produces the highest accuracy, which is represented by scores here. We valued the validation score over the training score because we would like to see the performance of the model on unseen data.

The model with the highest validation scores are Random Forest with $R^2 = 0.75$, Neural Network with $R^2 = 0.73$, and Gradient Boosting with $R^2 = 0.72$. $R^2 = 0.75$ means that the model Random Forest explains 75% of the variance of the original data. This means the Random Forest model works great on predicting the price of the property.

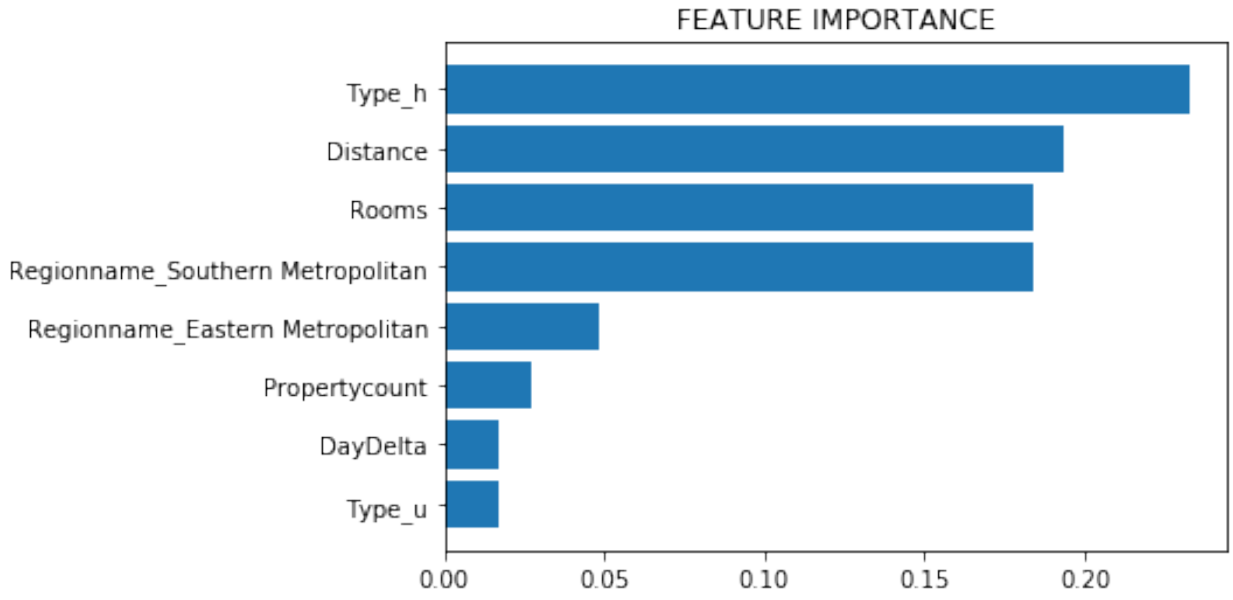
The model's performance makes sense because machine learning models in general would produce a more accurate prediction compare to non machine learning models. Tree base model works well on medium-size dataset, and Random Forest as an esemble of thousands of tree models, would produce the best outcome on a dataset with around 40,000 rows and 10 variables. The model gradient boosting and neural network would require more data to reach a better performance. If I could increase my datasize by 100 times, I would expect neural network would produce the best outcome among all other models.

Figure 16: Model Scores

	Model Name	Train Score	Validation Score
0	Multiple linear regression	0.64	0.64
1	Ridge Regression with C = 0.01	0.64	0.65
2	Ridge Regression with C = 0.1	0.64	0.65
3	Ridge Regression with C = 1	0.64	0.65
4	Ridge Regression with C = 10	0.64	0.65
5	Ridge Regression with C = 100	0.63	0.65
6	Lasso Regression with C = 0.01	0.64	0.65
7	Lasso Regression with C = 0.1	0.64	0.65
8	Lasso Regression with C = 1	0.64	0.65
9	Lasso Regression with C = 10	0.64	0.65
10	Lasso Regression with C = 100	0.64	0.65
11	SVM Regression with C = 0.01	0.42	0.42
12	SVM Regression with C = 0.1	0.42	0.42
13	SVM Regression with C = 1	0.42	0.42
14	SVM Regression with C = 10	0.42	0.42
15	SVM Regression with C = 100	0.43	0.43
16	Random Forest	0.76	0.75
17	Adaboost	0.41	0.4
18	Gradient Boosting	0.71	0.72
19	Neural Network	0.71	0.73

Model training scores and validation scores for all models.

Figure 17: Random Forest Feature Importance



The feature importance of Random Forest model, sorted from high to low, feature importances < 0.01 are excluded

The Figure 17 on page 25 shows the feature importance of the Random Forest Model. The feautures are sorted from high to low by their importance, and importances < 0.01 are excluded. Whether the property is a house contribute the most to the model, follow by distance, rooms and region. As we saw in the data exploration section, property being an house in general has a higher selling price than townhouse and apartment. From the correlation matrix in Figure 15 on page 22, we saw that as distance increases, the selling price decreases, and we also notice that as number of rooms increases, the selling price increases as well. Also, from the Type of Analysis, we noticed that the southern region is considered the welthier region that is closer to both the Central Business District and the beach. Although the DayDelta variable shows in the feature importance figure, its importance $= 0.0166$ matches our finding in Figure 14 on page 21 that the difference in day has no prominent effect on the price of Property.

3.4 Price Prediction Analysis Conclusion

The Random Forest model provides a fairly accurate model with validation $R^2 = 0.75$ in predicting the price of property. The variables that contribute most to the model are intuitively make sense, which are number of rooms, the distance from CBD, and region of the property. Therefore, we can draw a conclusion on this analysis that as the property size (infer from number of rooms increase) and the region (infer from distance and region) are the 2 most important parameters that affects the property price. I believe this conclusion not only applied in the Melbourne Australia, but also in housing market in general.

The result of the price prediction can be fairly for buyer and seller. For example, buyer can check against the selling price of property against the price prediction result and identify the chance to save money on those lower than market value property. Seller, on the other hand, could happily sell his/her property if the sale price is higher than predicted. Or if the market is not hot and selling price is lower than expected, the seller could see how much the property price is he/she could normally sell and see if he/she accept the gap. If the seller's budget is not tight, he/she could wait a few more months for the possibility to sell the property at a predicted price. The price gap should accommodate the interest payment during the waiting period. In short, this price prediction model is essential in the real estate business.

There are some improvements that could be done to improve the model accuracy. I believe the most effective ways are increase data size and data enrichment. If we can increase the data size to millions of rows, the performance on Neural Network and Gradient Boosting model should be improved. We could also enrich the data with property size, average household income of the region, property condition, land size, the age of the property, etc. so that the data could be

better used to predict the property price.

4 Conclusion

Using the Melbourne Housing dataset from Kaggle.com, the Type of Property analysis and Price Prediction analysis are performed on the data. There are many interesting conclusions are drawn from the two analysis. There are certain caveats of this analysis. Some of the conclusions in this analysis could only be applied to Melbourne Australia, some could be genearlized to similar housing market like Melbourne, some could be applied everywhere. For example, in Type of Property analysis, the multinomial model indicates the southern region of Melbourne has a higher chance of being an Apartment compare to a Single-Family House. This conclusion is Melbourne specific, since the southern region of Melbourne is closer to the Central Business District and the beach. Demand is higher than demand in the southern region which causes more apartment constructions that could fit more residents per square feet. But this conclusion can be generalized to a similar housing market like Melbourne where space is limited and demand is rising. In such housing market, the region with nicer environment such as better commute or more natural resource should have more apartments compare to other regions. Simliarly, in the Price Prediction analysis, we concluded that difference in time has little effect on the property price. This conclusion is limited to the Melbourne Australia housing market from 2016 to 2018 only. For a fast growing housing market, time should be considered as an important parameter to predict property price. Lastly, we concluded that the number of rooms and the distance from central business district is important in predicting the property price. This conclusion could be generalized to everywhere that property size and the community of the property are important in price prediction. In average,

the larger the property and the better the community the property located should mean the higher the property price.

References

[1] Introduction to U.S. Economy: Housing Market, October 2, 2019

https://fas.org/sgp/crs/misc/IF11327.pdf

[2] Melbourne Housing Data, Tony Pino

https://www.kaggle.com/anthonypino/melbourne-housing-market

[3] Machine Learning Note, Yinian Wu

Appendix

Type of Property Analysis:

```{r cars}
library(Hmisc)
library(dplyr)
library(ggplot2)
library(nnet)
library(car)

setwd("D:\\Document\\Study Related\\UCLA\\Thesis\\Type of Property")
df <- read.csv("melbourneHousePrices.csv",stringsAsFactors = FALSE)
df <- df %>%  filter(df$YearBuilt >= 1830)
df <- df %>% mutate(MedianIncome = as.numeric(gsub(",","",MedianIncome)),
                    Individuals =  as.numeric(gsub(",","",Individuals)))
df %>% group_by(Type) %>%  summarise(n())
summary(df$YearBuilt)
# Variables
# Medium incomes (Cut into categories)
# Can make a âclassâ variable from median incomes
# Distance from center (Cut into categories)
# Year Built
# Sale Type
# Region (Metropolitan vs Victoria)
# https://liveinmelbourne.vic.gov.au/discover/melbourne-victoria/metropolitan-melbourne
# Metropolitan: capital of Victoria state
# Individual

#distance for each Type of housing.
#APT/UNIT
df_apt <- df %>% filter(Type=='u')
hist(df_apt$Distance)
summary(df_apt$Distance)

#TOWNHOME
df_townhome <- df %>% filter(Type=='t')
hist(df_townhome$Distance)
summary(df_townhome$Distance)

#HOUSE
df_house <- df %>% filter(Type=='h')
```

```
hist(df_house$Distance)
summary(df_house$Distance)

#Method
#S means first time sold, aka new house. Other means sold before, passed in, etc.
df$Method <- as.factor(ifelse(df$Method=="S",as.character(df$Method),'Sold Prior'))
table(df$Method)

hist(df$Distance, main="Histogram of Distance", xlab="Distance")

# Create variable at the zip level
# zip_class is the bottom/middle/top third at the zipcode level
MedIncDistinct <- df %>% select(Postcode,MedianIncome,Individuals) %>% distinct() %>%
arrange(MedianIncome)
ggplot(MedIncDistinct,aes(x=MedianIncome)) + geom_histogram(aes(weight = Individuals)) +
ggtitle('zips by income')
thirds <-
wtd.quantile(MedIncDistinct$MedianIncome,probs=seq(.333,1,by=0.333),weights=MedIncDistin
ct$Individuals)
thirds_list <-
c(min(MedIncDistinct$MedianIncome)-1,thirds[[1]],thirds[[2]],max(MedIncDistinct$MedianIncom
e)+1)
class_list <- c('lo','mid','hi')
MedIncDistinct$zip_class <- cut(MedIncDistinct$MedianIncome,thirds_list,labels = class_list)
MedIncDistinct <- MedIncDistinct %>% select(Postcode, zip_class)
MedIncDistinct %>% group_by(zip_class) %>% summarise(n())

# Join Back
df <- left_join(df,MedIncDistinct,by='Postcode')

# Create variable at the listings level
# class is the bottom/middle/top third income at the listing level
# thirds <- quantile(df$MedianIncome,probs=seq(.333,1,by=0.333))
# thirds_list <- c(min(df$MedianIncome)-1,thirds[[1]],thirds[[2]],max(df$MedianIncome)+1)
# class_list <- c('lo','mid','hi')
# df$class <- cut(df$MedianIncome,thirds_list,labels = class_list)
# df %>% group_by(class) %>% summarise(n())

# Some differences between class and zip_class (but not a ton)
# df %>% group_by(class,zip_class) %>% summarise(n())

# Do same breaks for listings at the listings level
thirds <- quantile(df$Distance,probs=seq(.333,1,by=0.333))
```

```
thirds_list <- c(min(df$Distance)-1,thirds[[1]],thirds[[2]],max(df$Distance)+1)
class_list <- c('near','medium','far')
df$distance_class <- cut(df$Distance,thirds_list,labels = class_list)
df %>% group_by(distance_class) %>% summarise(n())

# Category for Regionname
df$region_class <- substr(df$Regionname,1,1)
#df %>% group_by(region_class,Regionname) %>% summarise(n())

# Category for YearBuilt
df$yearbuilt_class <- ifelse(df$YearBuilt>=1970,'1970 or after','1969 or before')
df %>% group_by(yearbuilt_class) %>% summarise(n())

df <- df %>% select(c(Type,Method,Distance,Postcode,YearBuilt,Regionname,MedianIncome,
                      zip_class,distance_class,region_class,yearbuilt_class))

#Type ~ Region
m3 = multinom(Type~region_class,  data = df)
exp(coef(m3))


#Model1:
m1 = multinom(Type~distance_class+region_class+zip_class+Method+YearBuilt, data = df)
Anova(m1)
z <- summary(m1)$coefficients/summary(m1)$standard.errors
p <- (1-pnorm(abs(z),0,1))*2
One_over_OR <- 1/t(exp(coef(m1)))[c(-1,-11)]
round(t(exp(coef(m1))),2)
t(round(p,4))
round(exp(confint(m1)),2)

#Model2:
m2 =
multinom(Type~distance_class+region_class+zip_class+Method+YearBuilt+distance_class*regi
on_class, data = df)
Anova(m2)
round(t(exp(coef(m2))),2)

```

Price Prediction Analysis:

"""

```
Spyder Editor
"""

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn import svm
from sklearn import ensemble
from sklearn.neural_network import MLPRegressor

#
================================================================================
=====
# Data Preparation
#
================================================================================
=====

#obtain data
#df = pd.read_csv(r'D:\Document\Study Related\UCLA\Thesis\Price
Prediction\MELBOURNE_HOUSE_PRICES_LESS.csv')
df = pd.read_csv(r'D:\Document\Study Related\UCLA\Thesis\Price
Prediction\Melbourne_housing_FULL.csv')


#data exploration:
df.describe()
df['Type'].value_counts()
df['Method'].value_counts()
df['Regionname'].value_counts()
df['Date'].value_counts()

#only include row with Price, since Price is the output
#df = df.dropna(subset=['Price'])
df = df.dropna()

#no missing data after price is removed
df.count()
df.dtypes
```

```
#convert date to date format and postcode to categorical variable
df['Date'] = pd.to_datetime(df['Date'])
df['Postcode'] = df['Postcode'].astype('category')

#Get year, month, weekday of the date
df['Year'] = df['Date'].apply(lambda x: x.year)
df['Month'] = df['Date'].apply(lambda x: x.month)
df['YearMonth'] = df['Date'].apply(lambda x: str(x.strftime('%Y-%m')))

#Get day delta, how many days from 2016-01-28
mindate = df['Date'].min()
df['DayDelta'] = df['Date'].apply(lambda x: (x - mindate) / np.timedelta64(1, 'D'))


#drop address, seller, and date
df = df.drop(['Address', 'SellerG', 'Date'], axis=1)
df = df.drop(['Lattitude','Longtitude'], axis=1)


#
================================================================================
=====
# Graphics
#
================================================================================
=====

#plt.hist(df['Type'])
#
#n, bins, patches = plt.hist(x=df['Price'], bins='auto', color='#0504aa',
#                            alpha=0.7, rwidth=0.85)
#plt.grid(axis='y', alpha=0.75)
#plt.xlabel('Value')
#plt.ylabel('Frequency')
#plt.title('My Very Own Histogram')
#plt.text(23, 45, r'$\mu=15, b=3$')
#maxfreq = n.max()
## Set a clean upper y-axis limit.
#plt.ylim(ymax=np.ceil(maxfreq / 10) * 10 if maxfreq % 10 else maxfreq + 10)
#
#
#df['Price'].plot.hist(grid=True, bins=20, rwidth=0.9,
#                      color='#607c8e')
```

```
#plt.title('Commute Times for 1,000 Commuters')
#plt.xlabel('Counts')
#plt.ylabel('Commute Time')
#plt.grid(axis='y', alpha=0.75)
#
#
#
## Basic correlogram
#sns.pairplot(df)
#plt.show()


#
# ============================================================================
=====
# Prepare train and test data and result table
#
# ============================================================================
=====

df_X = pd.get_dummies(df.drop(['Price'], axis=1))
df_Y = df['Price']

X_TRAIN, X_TEST, Y_TRAIN, Y_TEST = train_test_split(df_X, df_Y, \
                                   test_size=0.30, \
                                   random_state=2019)

SCORE_TABLE = pd.DataFrame(columns=['Model Name', 'Train Score', 'Validation Score'])
def ROW_COUNTER():
   def add():
      counter[0] = counter[0] + 1
      return counter
   counter = [0]
   return add
ROW_COUNT = ROW_COUNTER()

#
# ============================================================================
=====
# Intial Model: multiple linear regression
#
# ============================================================================
=====
```

```
M1 = linear_model.LinearRegression().fit(X_TRAIN, Y_TRAIN)
M1_TRAIN_SCORE = M1.score(X_TRAIN, Y_TRAIN) #R^2 = 0.289 on training data
#M1_prediction = M1.predict(X_TEST)
#plt.scatter(Y_TEST,M1_prediction)
M1_VALID_SCORE = M1.score(X_TEST, Y_TEST) #R^2 = 0.295 on validation data
SCORE_TABLE.loc[0] = ['Multiple linear regression', M1_TRAIN_SCORE, M1_VALID_SCORE]

#
================================================================================
=====
# Model #2, ridge regression
#
================================================================================
=====

C = [0.01, 0.1, 1, 10, 100]
for c in C:
   M2 = linear_model.Ridge(alpha=c).fit(X_TRAIN, Y_TRAIN)
   M2_TRAIN_SCORE = M2.score(X_TRAIN, Y_TRAIN)
   M2_VALID_SCORE = M2.score(X_TEST, Y_TEST)
   SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Ridge Regression with C = {c}',
M2_TRAIN_SCORE, M2_VALID_SCORE]

#
================================================================================
=====
# Model #3, lasso regression
#
================================================================================
=====

for c in C:
   M3 = linear_model.Lasso(alpha=c).fit(X_TRAIN, Y_TRAIN)
   M3_TRAIN_SCORE = M3.score(X_TRAIN, Y_TRAIN)
   M3_VALID_SCORE = M3.score(X_TEST, Y_TEST)
   SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Lasso Regression with C = {c}',
M3_TRAIN_SCORE, M3_VALID_SCORE]

#
================================================================================
=====
# Model #4, SVM regression, SVR
```

```
#
================================================================================
=====

M4 = svm.SVR(C=1, gamma='scale').fit(X_TRAIN, Y_TRAIN)

for c in C:
   M4 = svm.SVR(C=c, gamma='scale').fit(X_TRAIN, Y_TRAIN)
   M4_TRAIN_SCORE = M4.score(X_TRAIN, Y_TRAIN)
   M4_VALID_SCORE = M4.score(X_TEST, Y_TEST)
   SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'SVM Regression with C = {c}',
M4_TRAIN_SCORE, M4_VALID_SCORE]

#
================================================================================
=====
# Model #5, random forest
#
================================================================================
=====

M5 = ensemble.RandomForestRegressor(n_estimators=100, min_samples_leaf=20,
random_state=1992).fit(X_TRAIN, Y_TRAIN)
M5_TRAIN_SCORE = M5.score(X_TRAIN, Y_TRAIN)
M5_VALID_SCORE = M5.score(X_TEST, Y_TEST)
SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Random Forest', M5_TRAIN_SCORE,
M5_VALID_SCORE]

#
================================================================================
=====
# Model #6, gradient boosting tree
#
================================================================================
=====

M6 = ensemble.GradientBoostingRegressor(n_estimators=100, min_samples_leaf=20,
random_state=1992).fit(X_TRAIN, Y_TRAIN)
M6_TRAIN_SCORE = M6.score(X_TRAIN, Y_TRAIN)
M6_VALID_SCORE = M6.score(X_TEST, Y_TEST)
SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Gradient Boosting', M6_TRAIN_SCORE,
M6_VALID_SCORE]
```

```
#
================================================================================
=====
# Model #7, Adaboost
#
================================================================================
=====

M7 = ensemble.AdaBoostRegressor(n_estimators=100, learning_rate=0.1,
random_state=1992).fit(X_TRAIN, Y_TRAIN)
M7_TRAIN_SCORE = M7.score(X_TRAIN, Y_TRAIN)
M7_VALID_SCORE = M7.score(X_TEST, Y_TEST)
SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Adaboost', M7_TRAIN_SCORE,
M7_VALID_SCORE]

#
================================================================================
=====
# Model #8, Neural Network
#
================================================================================
=====

M8 = MLPRegressor(hidden_layer_sizes=(100, 10), max_iter=500).fit(X_TRAIN, Y_TRAIN)
M8_TRAIN_SCORE = M8.score(X_TRAIN, Y_TRAIN)
M8_VALID_SCORE = M8.score(X_TEST, Y_TEST)
SCORE_TABLE.loc[ROW_COUNT()[0]] = [f'Neural Network', M8_TRAIN_SCORE,
M8_VALID_SCORE]

SCORE_TABLE.to_csv('SCORE_TABLE .csv')

#
================================================================================
=====
# Final testing with test data using Ridge regression and Gradient boost
#
================================================================================
=====

print(f'The R^2 for gradient boost on test data is {round(M6.score(X_TEST, Y_TEST),4)}')
```

```
#
================================================================================
=====
# Interpretion of gradient boost model
#
================================================================================
=====

#need exponential to transform log_installs back to number of installs
FEATURE_IMPORTANCE = list(zip(X_TRAIN.columns, M6.feature_importances_))
FEATURE_IMPORTANCE = [i for i in FEATURE_IMPORTANCE if i[1]>0.01]
FEATURE = [i[0] for i in FEATURE_IMPORTANCE]
IMPORTANCE = [i[1] for i in FEATURE_IMPORTANCE]
plt.barh(FEATURE, IMPORTANCE)
plt.title("FEATURE IMPORTANCE")
#plt.xticks(rotation=90)
plt.show()
```