Housing Rental Pricing Prediction

Pragyat Agrawal Ruxuan Ji Meng-Chuan Chang

Contents

1 Executive Summary

1.1 Background

The US rental market has been growing rapidly growing over time, making it one of the most sought after areas for investments. Be it from small home owners to big private equity firms, everyone seems to be after housing, expecting the values of these houses to rise and supplement their income by renting these places. Many people consider that location is the "only important" factor responsible for a house's value and the rent that can be expected of it, but this is far from the truth. There are a lot of other factors that need to be considered for determining housing valuations as we see a huge variation in prices in houses located in the same vicinity. There must be something about these houses which is causing such a big price change. Hence

we will be analyzing the data related to US Rental Listings in Summer of 2021, to find which of these factors, which consist of many in-house amenity components, impacts housing values the most.

These amenities range from simple aka micro-factors like the availability of Pools and Dishwashers in the house to major aka macro-factors i.e. cities. The data will also give us the opportunity to find in which cities are these factors playing the most impact. With over 27,000 values for each predictor in our data, we have a sufficient sample size to make a reasonable conclusion regarding the price of these summer rentals.

Graph: Rental vacancy rates in the United States from 2000 to 2021, by region (Source: Statista.com)

This shows how the US housing market has been more sought after year by year, making housing a form of valuable investment. This increase in demand has already pushed up the prices.

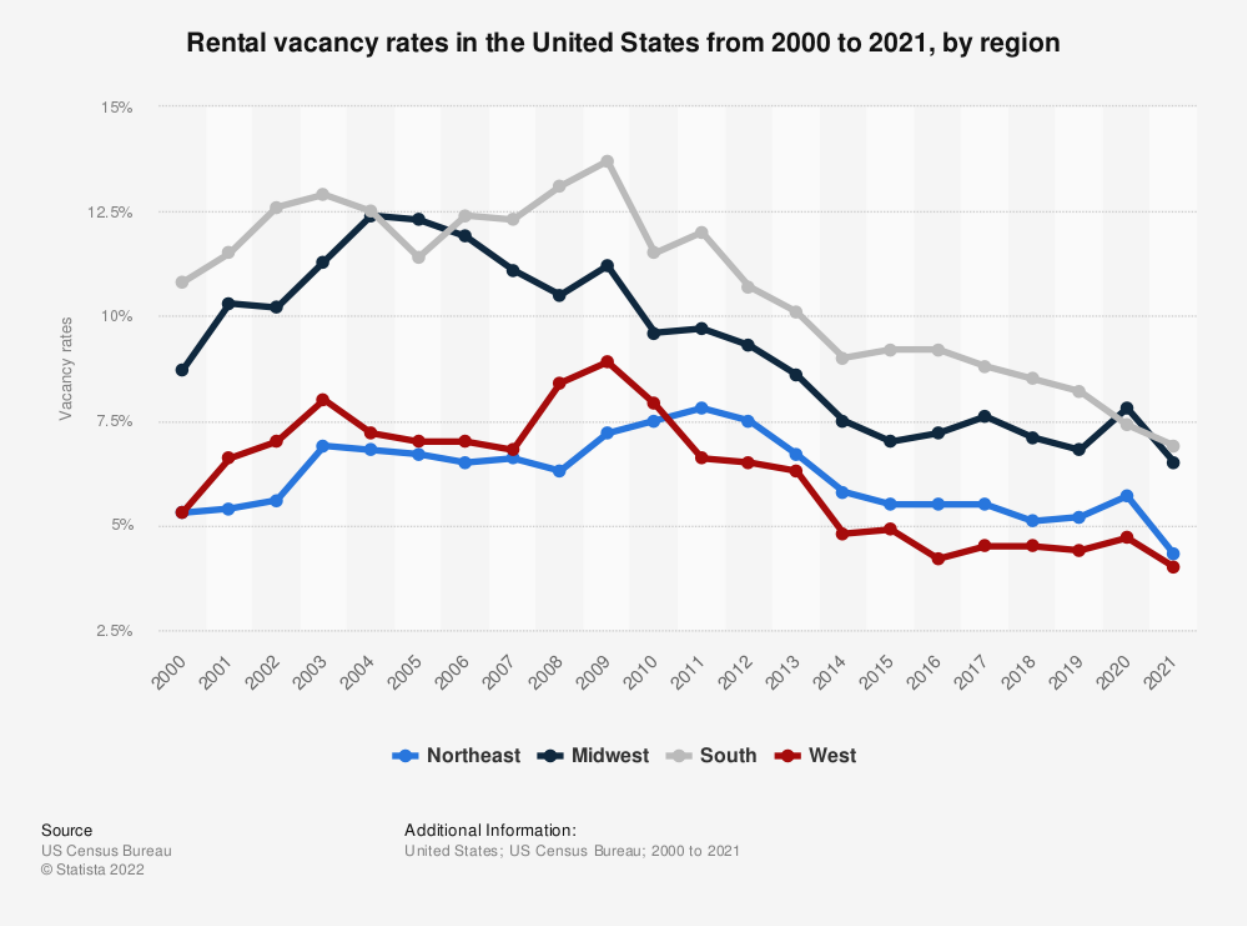


Figure 1: A caption

1.2 Description of Data

The data is gathered from Kaggle, a vast community published code and data repository. This data was pulled from Rentler.com on 7/12/2021, 8/12/2021, and 9/6/2021, and population density data was scraped by zip code from mapszipcode.com on 7/12/2021. The pull from Rentler.com resulted in 4 CSV files, which included the main rental listing, the list of amenities, the list of lease terms, and who was responsible for paying each utility. Many of the sparsely populated variables were dropped before denormalizing the dataset. The rental listing information was joined with the population and population density information from mapszipcode.com (Source: Kaggle). Many of the data columns in the dataset are embedded in binary format, with 0 representing the absence of the predictor attribute while 1 shows that the attribute is present. The

data file is massive at around half a Gigabyte. Working with this data size, we would have to use EDA or take a subset of the dataset for R to run effectively and not crash over the large size of the data. We will explore this idea in later sections. For now, the data is sufficient for analysis.

Our response variable is PRICE which represents the monthly price for the particular summer listing on rental.com.

Table 1: Variables and their descriptions

Variable	Description
V1	Numbering for the house number we are looking at (form of house identification)
pool	Binary data to show if pool exists in this house (1) or not (0)
dishwasher	Binary data to show if dishwasher exists in the house (1) or not (0)
washer-dryer	Binary data to show if washer-dryer exists in the house(1) or not (0)
ac	Binary data to show if air conditioning exists in the house (1) or not (0)
parking	Binary data to show if Parking exists in the house (1) or not (0)
zip	Zip code of the property
price	Monthly rent price for the property
city	City where the property is located
num_beds	Number of beds in the property
num_baths	Number of baths in the property
house_type	Type of house we are looking at
sqft	Square Feet in the property
smoking_ind	Does the rental allow smoking (Yes/No)
pets_ind	Does the rental allow pets (Yes/No)
acres	Number of acres rental includes
description	4000 character listing description of the rental
ZipCity	Primary city for the zip code
Population	Population in the zip code
PopulationDensity	Population density per square mile for zip code
security_deposit	Security deposit required

1.3 Goal

This study aims to analyze the most critical factors affecting housing rental prices in the US. We will be using the variables in the dataset to do so. In our preliminary market analysis, we found that many factors determine rental prices, and hence we will try to ascertain factors that have a significant impact on rental prices. Our goal is to build a model that will give us the value added or subtracted from a house with/without the presence of a variable factor. This observation will benefit people looking to rent properties in the US and help them get a better value for the kind of place they may be looking for.

1.4 Summary of Findings

When we started the analysis, we initially thought that the rental prices would be affected by a very few factors that may push the results higher. But this is not true. Instead, there are a plethora of factors affecting housing rental prices. From our research, we found a lot of significant factors affecting housing like the availability of a dishwasher, washer-dryer, number of baths, population density, etc. Some factors were expected due to the monetary values associated with them. Still, some factors, like if smoking is allowed or not or pets are allowed or not, were very unanticipated to be significant. Hence, through this research, we pinpoint the various factors that go into play in determining housing prices.

1.5 Issues and Limitations

The biggest issue we initially faced was concerning the file size, which turned out to be quite massive, even for R Studio. The raw data we started with was half a gigabyte big, which was very massive for any form of

extrapolation. Hence we had to shorten the data, choosing 20,000 data points for a particular seed to ensure reproducibility. After that, we also had some problems with knitting the data, because of which we had to keep clearing our knitr cache to ensure we get the correct data across on the PDF. Finally, the random forest part took some time as it had to run a data-heavy analysis. We had to wait for this part out as we wanted to give the highest quality output.

2 Exploratory Data Analysis

2.1 Data Preprocessing/Cleaning

2.1.1 Read the data

Firstly, we read the data from Kaggle - US Rental Listings Summer 2021

Then we read SOI Tax statistics in 2019 from IRS

Before joining those two dataset, we would use groupby to find the income level based on zip code

Here, `AGI_STUB` shows the level of adjusted gross income, and `N1` shows the number of returns of each level.

The following shows the level and corresponding income range

AGI_STUB	Range
1	$1~25,000
2	$25,000~$50,000
3	$50,000~$75,000
4	$75,000~$100,000
5	$100,000~$200,000
6	$200,000~

Based on the above two columns, we generate a new column `avg_level` from 1 to 6, showing the average level of gross income in each zip code region

2.1.2 Filter the data

The original dataset contains 276757 data, but we just need partial data. Before we randomly pick 20000 for further analysis, we can remove rows that is lack of important factors. The criteria is as follows

- sqft (squart feet) must be non-zero
- population and the density must be non-zero
- price must be non-zero

Then we drop several columns which is clearly not helpful for predicting the rental price

- link
- street_address
- full_address
- acres
- description
- zip city (duplicated data)

Finally, we fill all NA with 0. The columns having NA is as follows

- pool
- dishwasher
- washer-dryer
- ac

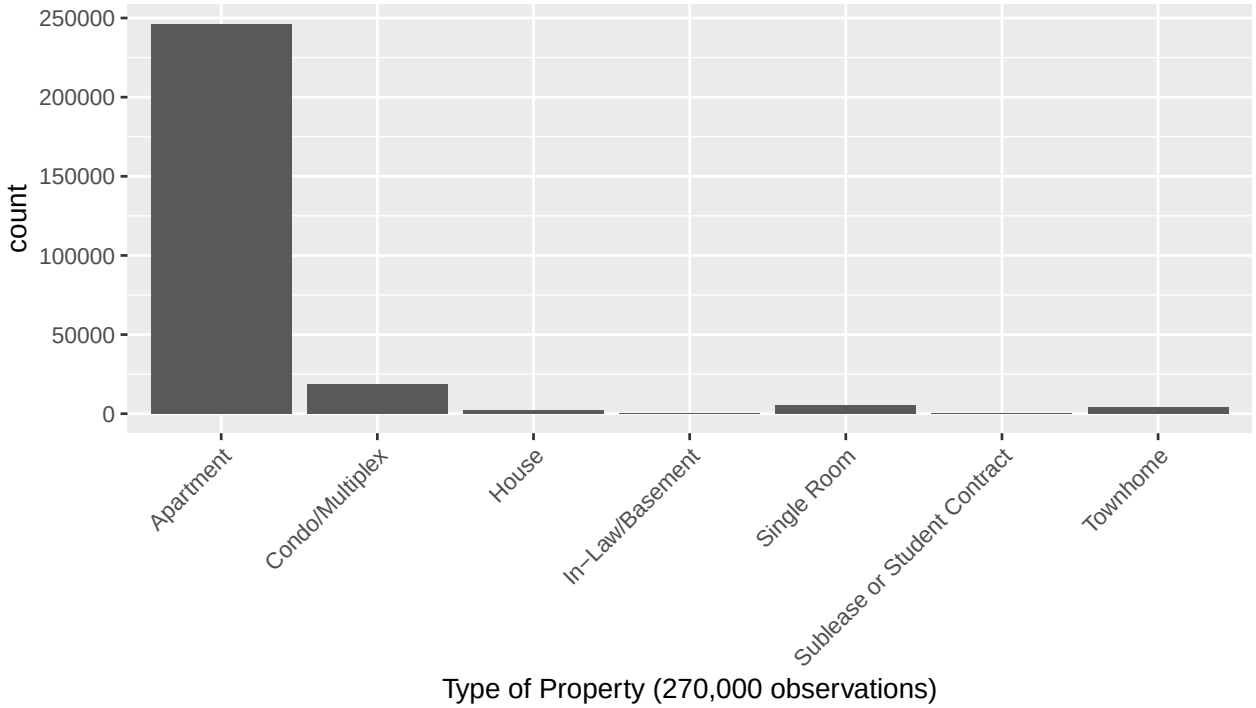
- parking

Then we export the cleaned dataframe to csv

2.2 Data Transformations and Plots

2.2.1 House Type

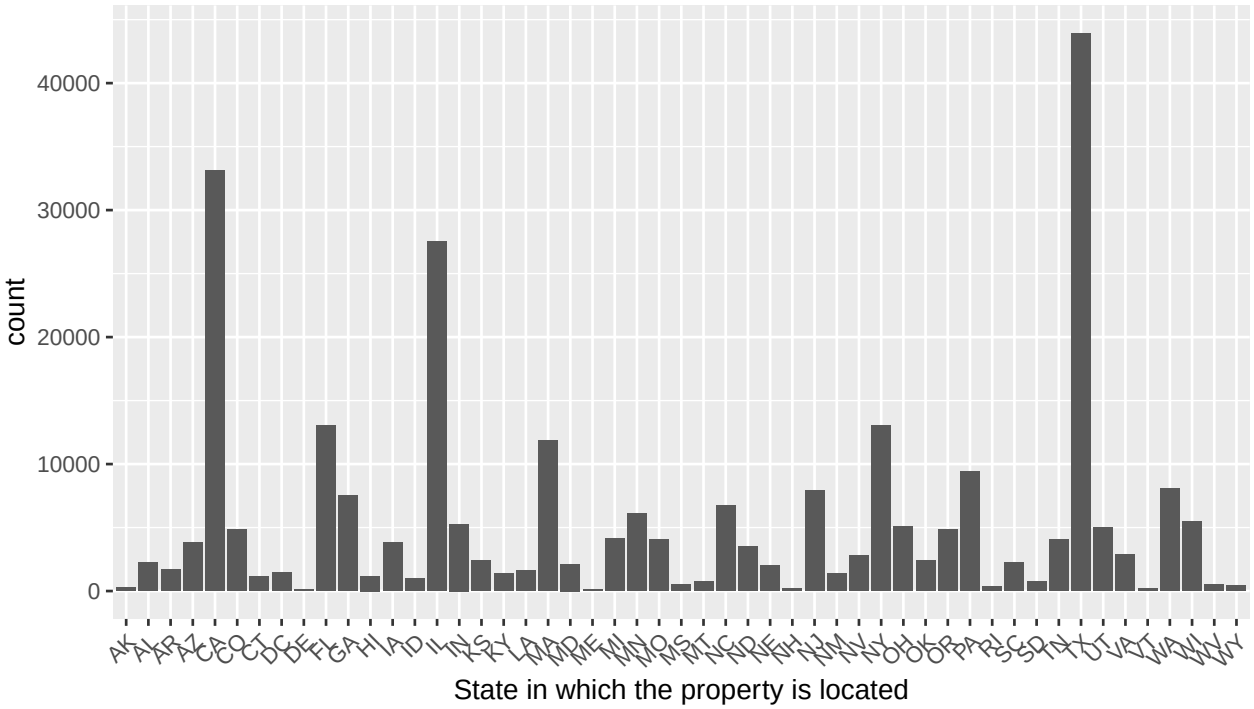
house_type	n	n_all
Apartment	17749	246365
Condo/Multiplex	1519	18510
House	219	1967
In-Law/Basement	26	230
Single Room	211	5195
Sublease or Student Contract	1	26
Townhome	275	4155



n is the number of count with 17,000 observations, and n_all is that with 270,000 observations. Looking at the data we see that the properties we will most be evaluating will be Apartment style places with 17749 observations. Other places are lesser in number but still there. The second largest group is the Condo/Multiplex group that we are looking at with 1519 observations. Other than that the smallest group we see is the sublease or student contract group which only has 1 observation.

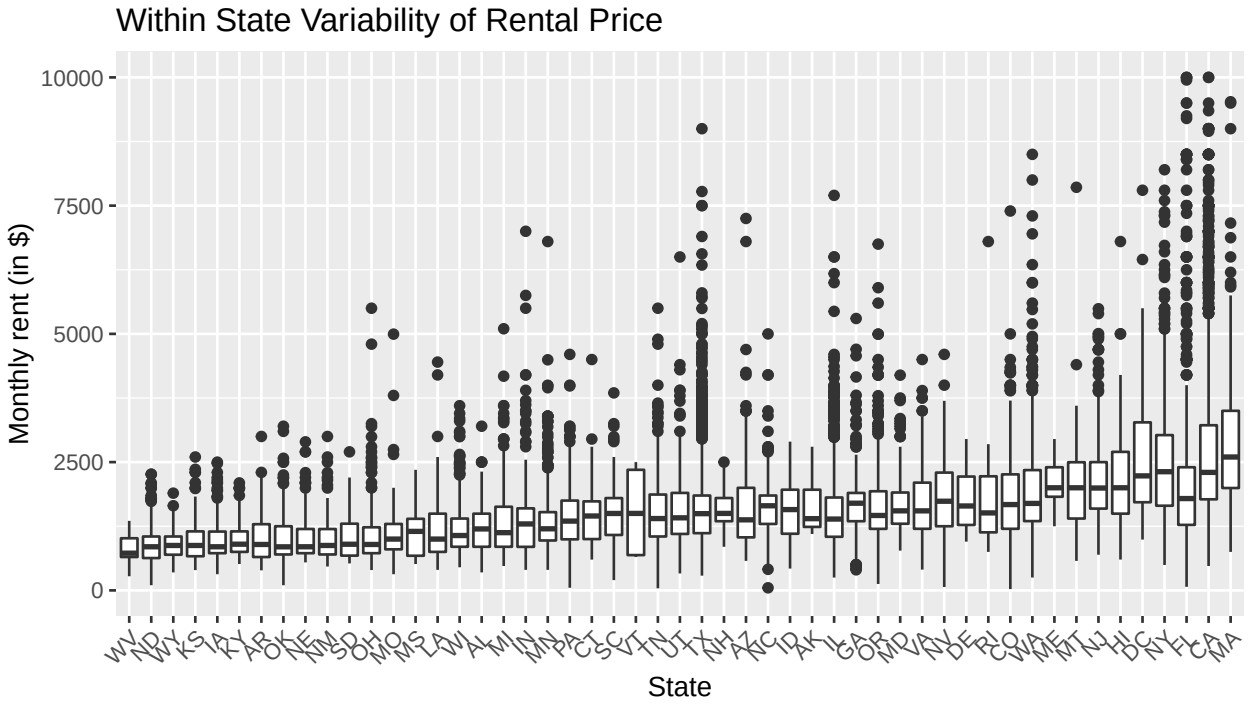
Also, compared n with n_all, we find the distribution is almost the same. Therefore, it is safe to use those 17,000 samples for further analysis.

2.2.2 Count based on state



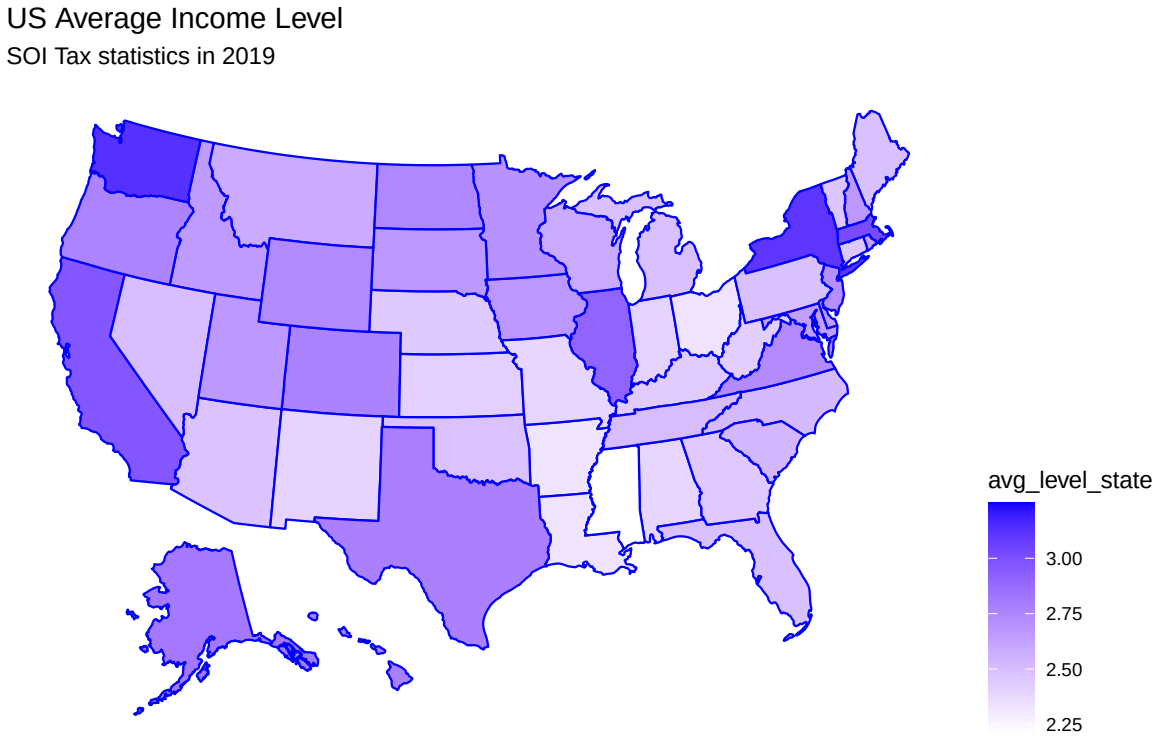
The data represents all states, some more than others. Texas is the most represented state with the least being Vermont at 4 listings. The sample is representative of all states in the US. We are choosing roughly 270,000 data points so this, but in our model training we only randomly choose 20,000 observations. The plot with 20,000 data may fluctuate if we change the data seed but the overall tendency should be the same.

2.2.3 Rental Price Variation among US States



We see the prices remaining fairly same for the start with the prices rising with states like New York, California, Florida and Massachusetts. This is evidently true for these "more expensive" places as states on the right end of the spectrum like California, and Massachusetts are known for their high per capita incomes which tends to push housing prices up. There is a lot of variation in data as well as we move to states with higher housing prices, indicated through the presence of excessive outliers on the right end of the boxplot.

2.2.4 Average Income Level among US States



For average income level (from 1 to 6) among all state, we first group by the state, take average of `avg_level`, and then draw the us plot to see the distribution of income among US state. The average income level in NY, WA, CA and TX are first tier, which is reasonable and meet our expectation. Hence we choose to use the data for further model training.

3 Model Training

3.1 Dataset Preparation

To do the further model analysis, we transform some of the data. Firstly, convert the following columns from characters to binary data

- house_type
- smoking_ind
- pets_ind
- state

Also, we need to predict the rental price which is continuous. In order to implement random forest, we transform the price to binary outcome, with the threshold of median of rental price, which is (1500).

Unlike state, the amount of unique city is far more than the state. In our sample data, we find there are 2395 unique cities, more than 1/10 of the observation. Therefore, we choose to discard this factor for our further model analysis.

We split the data for choosing and validating and model

- train: 13000
- test: 5000
- validation: the rest

3.2 Linear Regression

3.2.1 Normal Linear Regression

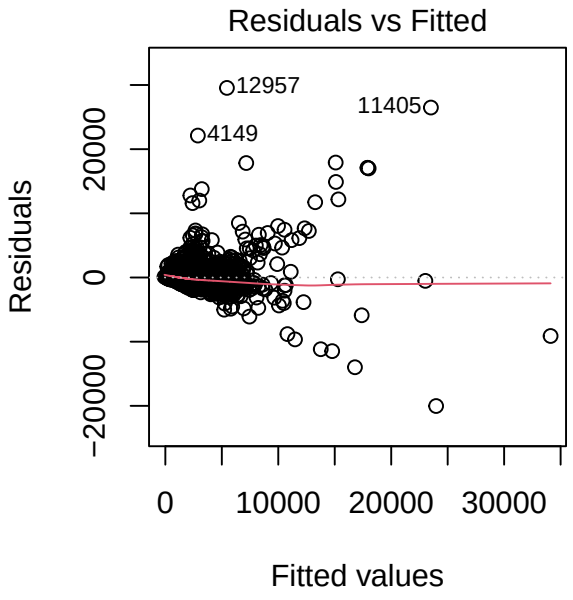
In this part, we fit the linear regression model `price` vs. other variables as `fit1` directly, then select significant variables to fit the final fit, `fit2`.

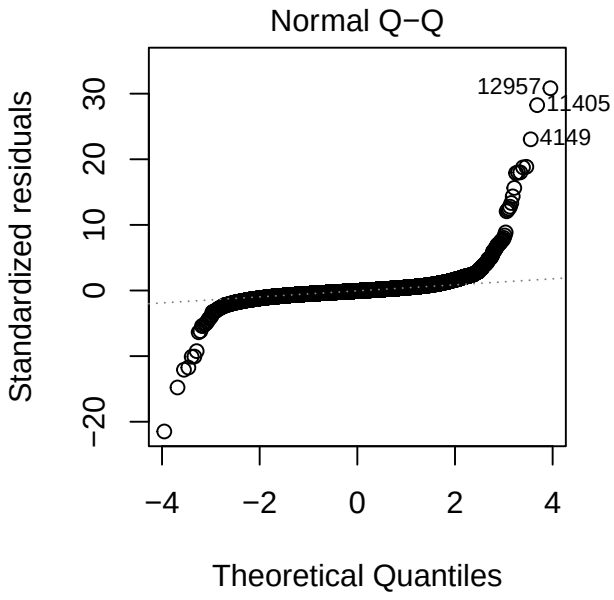
1. Since the `description` and `price` features are useless for linear model, we drop them.
2. From the summary of fit1, we select pool, dishwasher, washer-dryer, ac, parking, state, num_beds, num_baths, house_type, sqft, Population, population density and security deposit as the variables of our final model. All the variables are significant at 0.001. The result of this model is showed as follow.
3. Training MSE: 965083.3, Testing MSE: 1304949

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-642.3	54.49	-11.79	6.592e-32
pool	250.4	27.29	9.173	5.286e-20
dishwasher	219.2	22.23	9.862	7.351e-23
washer-dryer	71.18	22.45	3.171	0.001523
ac	-210.1	27.48	-7.646	2.227e-14
parking	120.1	30.08	3.992	6.575e-05
state	-6.649	0.5756	-11.55	1.018e-30
num_beds	2.659	12.27	0.2167	0.8284
num_baths	212.8	15.87	13.41	9.872e-41
house_type	-75.81	10.54	-7.195	6.594e-13
sqft	0.5339	0.02304	23.17	2.046e-116
Population	-0.0003867	0.0005111	-0.7565	0.4494
PopulationDensity	0.0189	0.000838	22.56	1.482e-110
security_deposit	0.3984	0.004556	87.44	0
avg_level	400.2	16.28	24.58	1.932e-130

Table 5: summary of linear regression model

Observations	Residual Std. Error	R^2	Adjusted R^2
13000	960.9	0.6046	0.6042

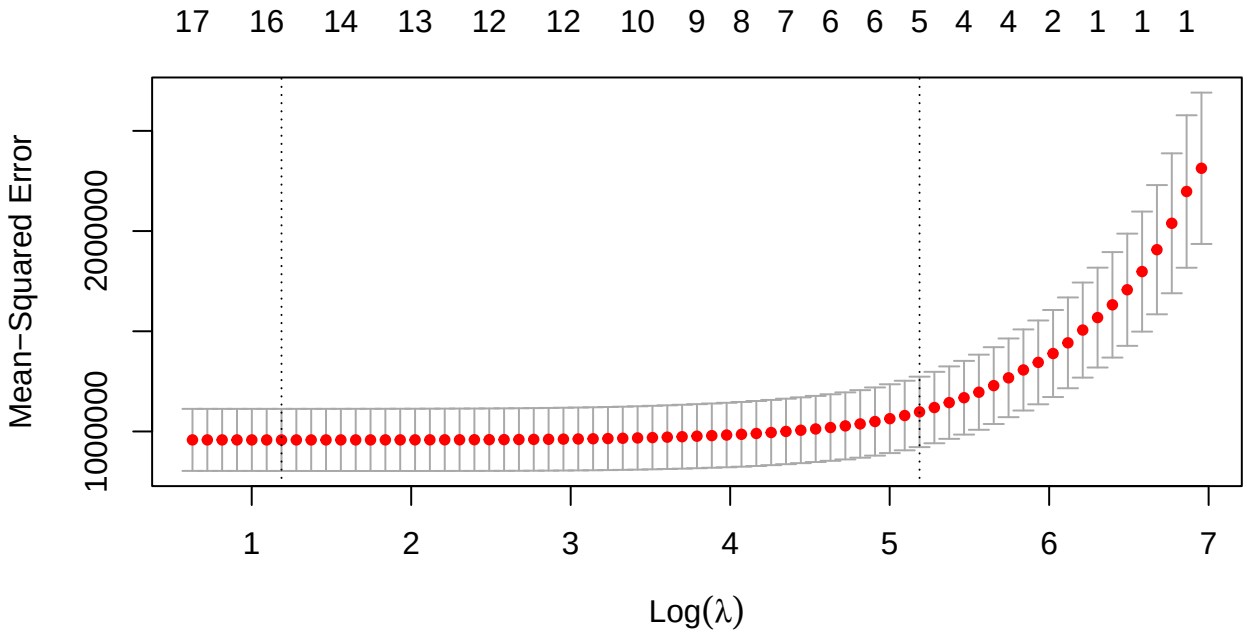




3.2.2 LASSO Model

In this part, we introduce LASSO for a linear regression model with fewer variable since in many case not all the variables are useful for a model. So using lasso can help to reduce computation load significantly with maintaining a good result. We build the model with lasso following 3 steps.

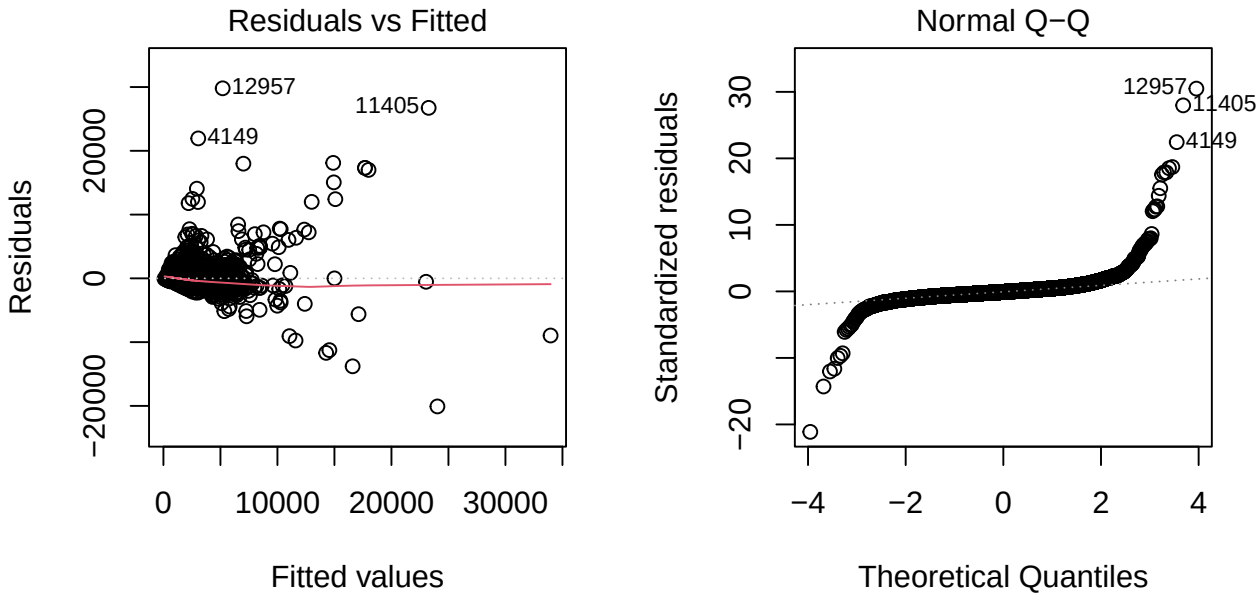
1. Description and price_binary column are drop since they make no sense for this model.
2. `price` is set to be X and the other column of training data are set to be Y, and they are used to fit the first model, `model1`, with alpha = 1 and nfolds = 10.
3. We use LASSO to select several useful variables and refit the linear regression model, named `model2_lasso`, finding that all the vaiables are significant at 0.001, including dishwasher, number of baths, sqft, population density and security deposit.



	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-958.8	46.25	-20.73	5.942e-94
num_baths	228.5	15.15	15.08	5.537e-51
sqft	0.5328	0.02021	26.36	2.989e-149
PopulationDensity	0.01949	0.0007999	24.37	2.782e-128
security_deposit	0.3987	0.004514	88.31	0
avg_level	449	15.84	28.35	1.277e-171

Table 7: summary of lasso regression model

Observations	Residual Std. Error	R^2	Adjusted R^2
13000	978.7	0.5895	0.5894



3.3 Logistic Regression

After using linear regression models to create the predictors for specific price of rental, we use logistic regression model to predict whether one rental house or department's price is higher or lower than the median. In this part, `price_binary` that we create early is used as the result for prediction, and we build the model following 4 steps.

1. Description and price column are drop since they make no sense for this model.
2. Price_binary is set to be X and the other column of training data are set to be Y, and they are used to fit the first model, `fit.log.cv`, with alpha = 1 and nfolds = 10.
3. We use LASSO to select several useful variables and refit the logistic regression model, named `fit.logit2`, finding that `num_beds` is not significant in this model. So we drop this column and refit the model again, named `fit.logit.final`. 4.Finally, the logistic regression model includes pool, dishwasher, washer_dryer, ac, parking, state, num_baths, house_type, sqft, smoking_ind, pets_ind, PopulationDensity and security_deposit. All of the variables are significant at 0.01. Among those, ac, dishwasher and Pool play relatively important roles when the prices are setting.

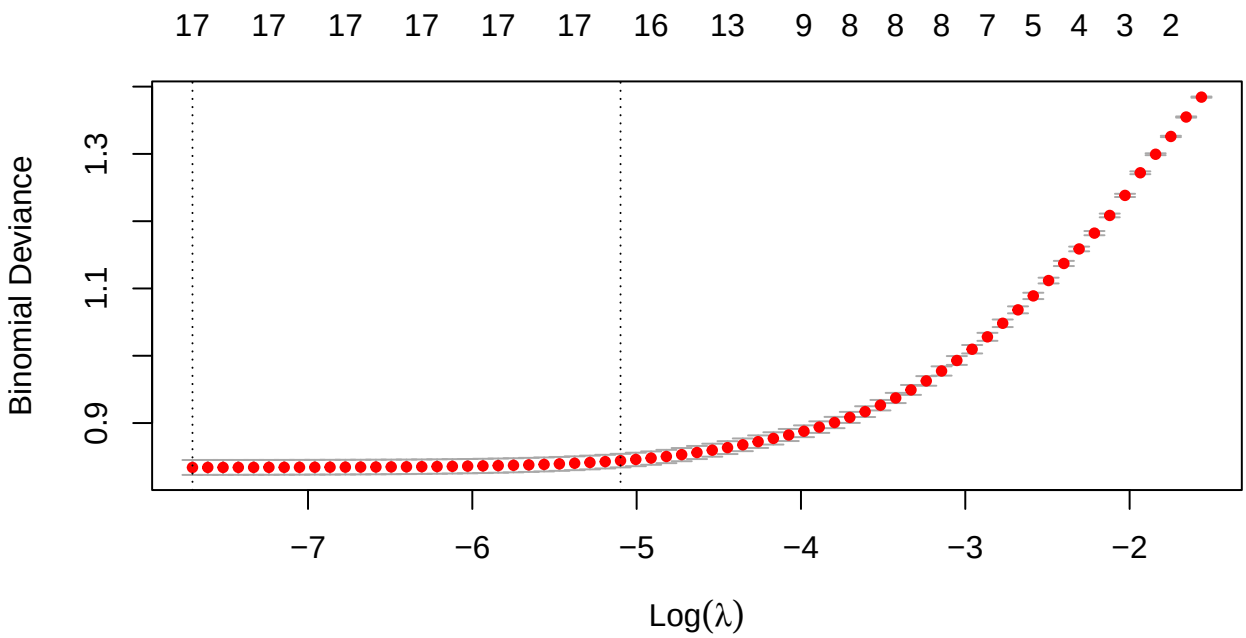


Table 8: summary of logistic regression model (binary classification)

	LR Chisq	Df	Pr(>Chisq)
pool	36.02	1	1.954e-09
dishwasher	136.3	1	1.685e-31
washer_dryer	17.16	1	3.444e-05
ac	36.64	1	1.42e-09
parking	18.69	1	1.539e-05
state	114.6	1	9.699e-27
num_baths	144.9	1	2.222e-33
house_type	18.76	1	1.486e-05
sqft	990.4	1	2.204e-217
smoking_ind	11.56	1	0.000675
pets_ind	34.09	1	5.263e-09
PopulationDensity	772.6	1	4.815e-170
security_deposit	594.9	1	2.145e-131
avg_level	967.9	1	1.736e-212

After We get the final model of logistic regression, we predict the result in test dataset and compute the confusion matrix. From the confusion matrix, we find the misclassification error is 0.2156, which is a little bit higher.

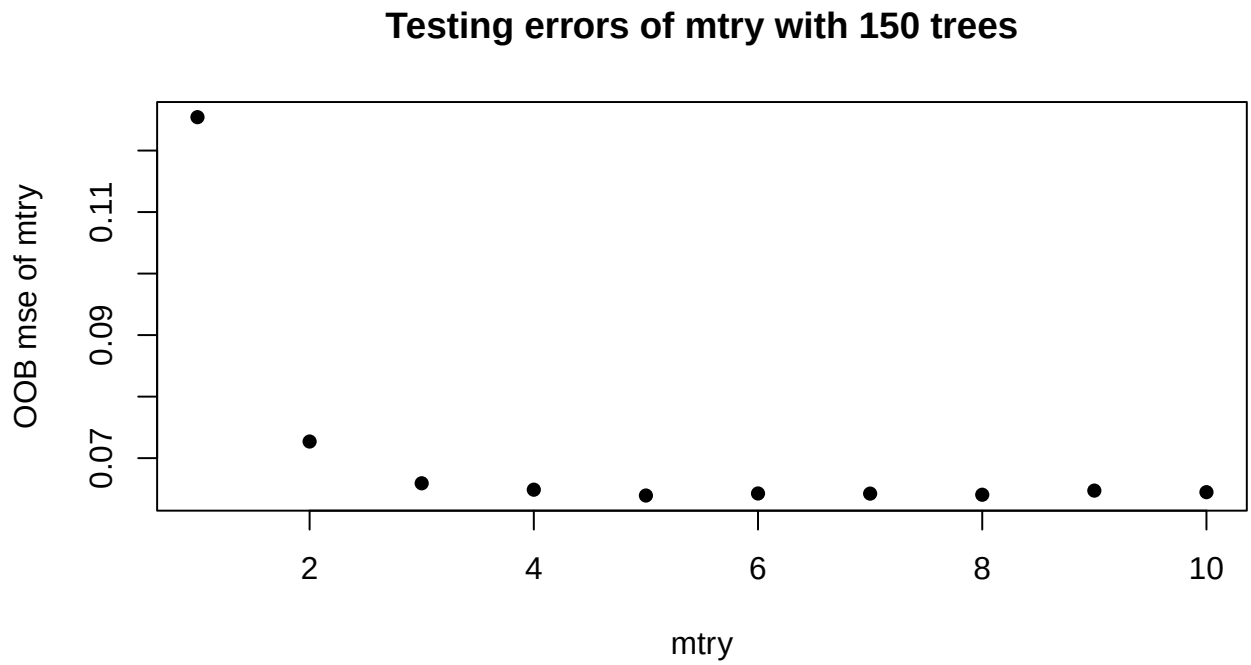
However, after we integrate the second dataset, the misclassification error drops to 0.1922, indicating that the income would improve the final result.

3.4 Random Forest

In random forest model, we just need the binary data of the price, so we remove `price` column, and we need to rename the column `washer-dryer` for the random forest package

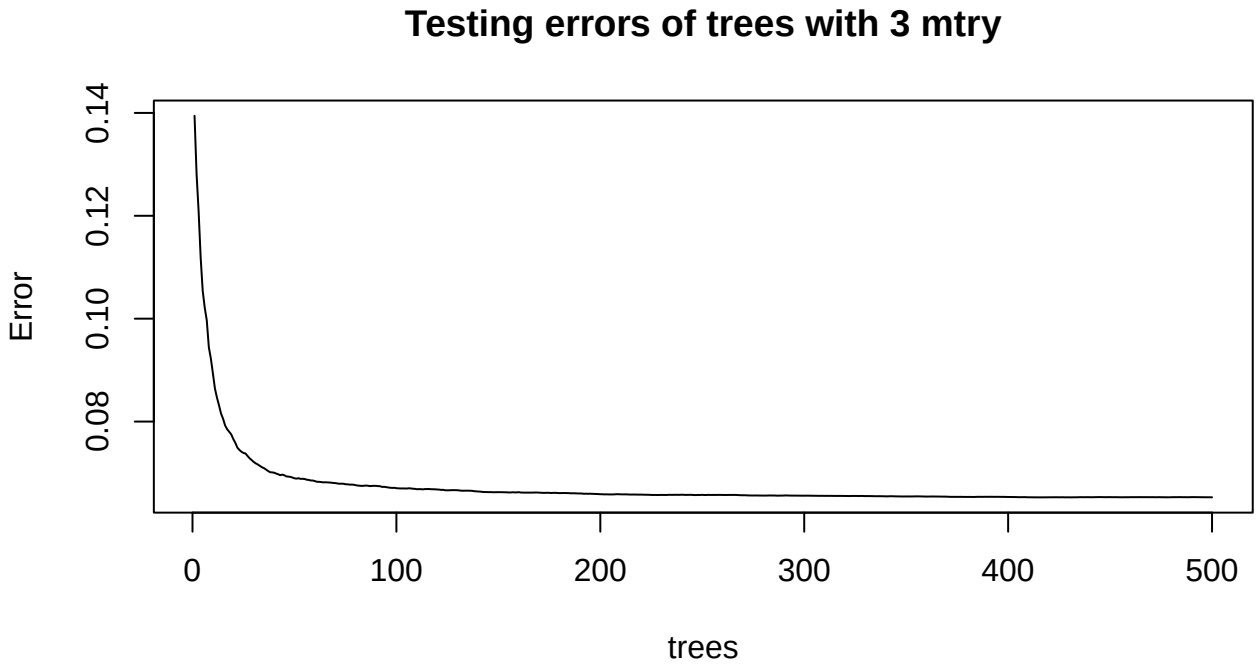
3.4.1 Tune Hyperparameter

Firstly, we use OOB to find the testing error for given parameter, and we choose mtry from 1 to 10, with 150 tree.



According to the above plot, we choose 3 as mtry based on elbow rule.

Then we set ntree to be 500, to find the optimal number of tree.



Also based on the above plot, we choose 100 as ntree

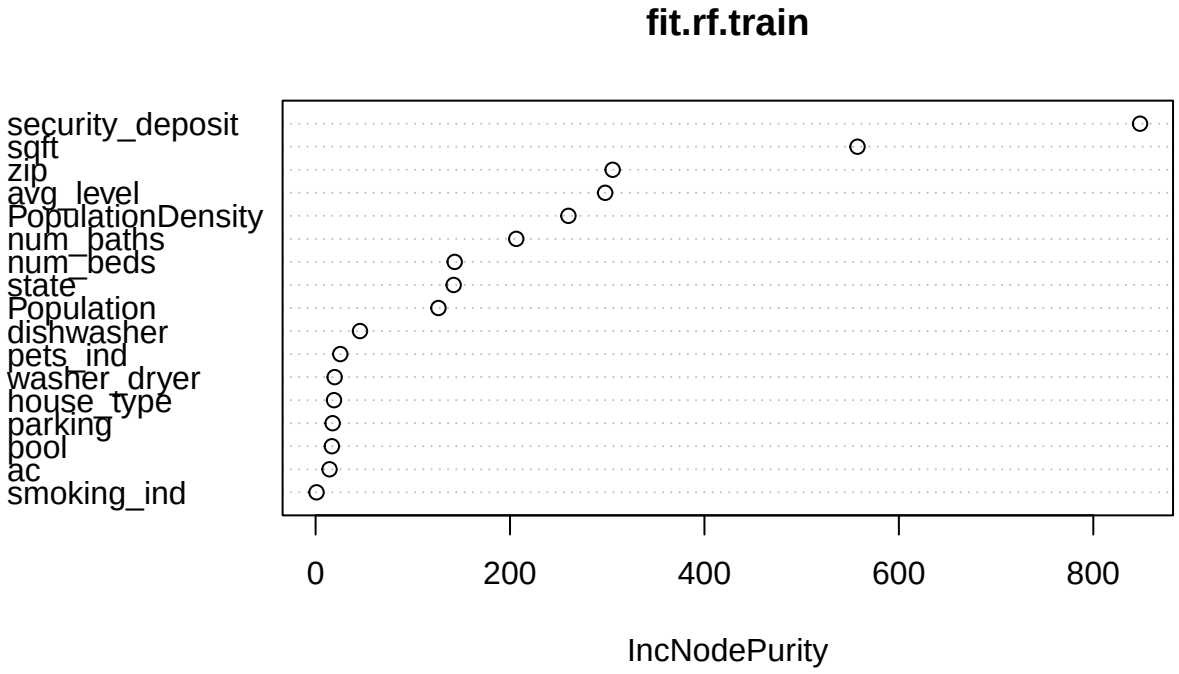
3.4.2 PCA

Apply PCA to dataset would help us to remove correlated data and speed up the runtime of the program. However, in this analysis, the runtime to train 20000 data is fast enough (about several minutes). Therefore, we decide not to implement PCA here to keep the information in the dataset.

3.4.3 Performance of Random Forest Model

Finally, given fixed mtry and ntree, we can compute the testing error of our random forest model. After we do prediction on the testing dataset, we set the threshold as 0.5. If the result is greater than 0.5, we would categorize the result as 1. On the other hand, result less than 0.5 is 0. Finally we compte the error to be 0.082

We draw the importance of variables from the final random forest model. Here, we can see that security deposit is the most important factor, and the area is the second important factor. It is reasonable that those two factors are highly correlated to the rental price. As for the `avg_level`, it is also regarded as a crucial factor, indicating that the financial condition actually related to the rental price. Surprisingly, the facilities of the house is not important compared to the security deposit, area, localtion and the level of income.



4 Conclusion

In conclusion, various factors affect rental prices beyond the simple square feet and location. The various micro-factors like availability of a pool, washer-dryer, dishwasher, etc., play an essential role and shouldn't be forgotten when considering a housing rental price. Our data is most applicable to Apartment and Condo/Multiplex forms of apartments. However, with repeated testing, our results can be extended to House, Townhome, Single Room and Sublease, or Student contract form of rentals. There was also considerable variability in the state where the houses are located, with states like New York being more expensive than mid-western states or southern states. The per capita incomes can also explain this in these places, drastically differing. We also ran a random forest model to predict the values with reasonable accuracy. We find that random forest is the best model to predict housing prices, combined with simple logistic regression.