MQM DECISION 520Q Data Science for Business: Final Project Write-up

Section A Team 12: Erico Cuna, Yixin Wang, Erik Wicks, Rong Xiao, Yue Zheng

Business Understanding

According to ANAROCK Group, monthly residential rental demand across India's top seven cities have risen by at least 10-20% compared with the pre-Covid-19 period in 2019. Residential demand has suddenly spiked because people have returned from their hometowns. In fact, there are several cities across the country where demand has outstripped supply [1]. It is imperative for rental agents to build up a model to help them better price real estate to arbitrage from the blooming rental market. We are mainly concerned about the following two questions:

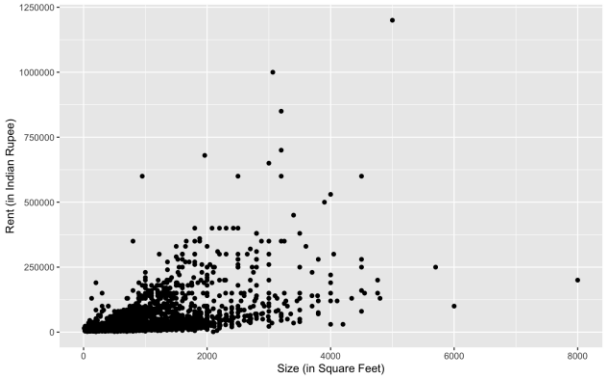
1. Which factors are highly correlated to the rent? 2. Which parameters are most valued by tenants and so that agents can renovate the apartments in styles that are appreciated by the market and offer tenants at a price suggested by our model to maximize agents' profits.

Given that we can forecast an apartment's rent based on a set of variables we collected, if we can acquire a rental house from the real estate owner at a cheaper price compared with the rental price we predict, there will be an arbitrage opportunity, assuming no other costs incurred.

If we take another way to look at it. After we build up the model, rental agents can figure out what factor is most critical in determining the rent of a real estate so that they can take the most cost-effective measure to maximize profits. For example, if tenants value the furnishing status of the apartment to a large extent (a very high coefficient) and are willing to pay a premium for that, and if the marginal rent tenants are willing to pay outweighs the marginal cost furnishing is going to incur, then rental agents should furnish the apartments to increase the profits. To put it another way, this pricing model serves as a guidance for rental agents to follow to help them run their businesses more efficiently.

Data Understanding

The data provided is at the house/apartment/flat level and contains information on features like BHK (number of bedrooms, hall, kitchen), size (in sq. ft), floor, area type (Super Area/Carpet Area/Build Area), city, furnishing status, tenants preferred, bedroom number, etc. The target/dependent variable is the apartment rent priced in Indian Rupee (INR). The dataset seems imbalanced, as shown from the two plots generated below, with most data points concentrated on the lower-left corner of the scatterplot (Figure 1), which represents relatively low rent. If we look at the boxplot, we can find that most Rent are below 23,000 and the median is close to 14,000 INR (Figure 2). Since we lack enough high rent data points (above 50,000 INR), the training result and forecasting accuracy might not be desirable for high rent.



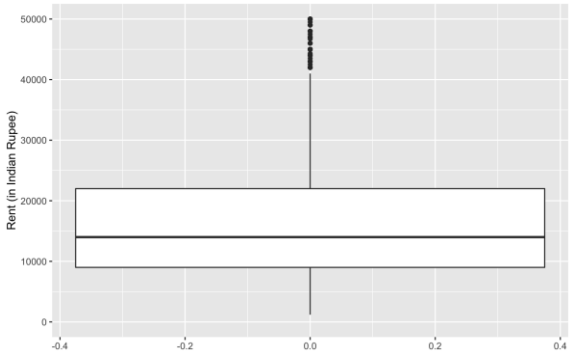


Figure 1: Rent VS Size Scatterplot

Figure 2: Boxplot of Rent

To get a deeper understanding of the relationship between pairs of variables, we generate a matrix plot (Figure 3). The factors that positively correlate with rent the most are bathroom number, BHK, floor number, and size. If the rental requires contacting the owner to make a contract, then rent would be negatively affected. We also found that the total floor number positively affected the rent. This pattern exists maybe because, in India, usually the larger the total floor number of the apartment the more luxurious it is. Thus, the rent is higher. Also, if the

real estate is located in big cities such as Mumbai, the rent also tends to be higher because the land resource tends to be scarcer in big cities. Thus, locating in big cities positively affects the rent. Our common sense confirms this pattern.

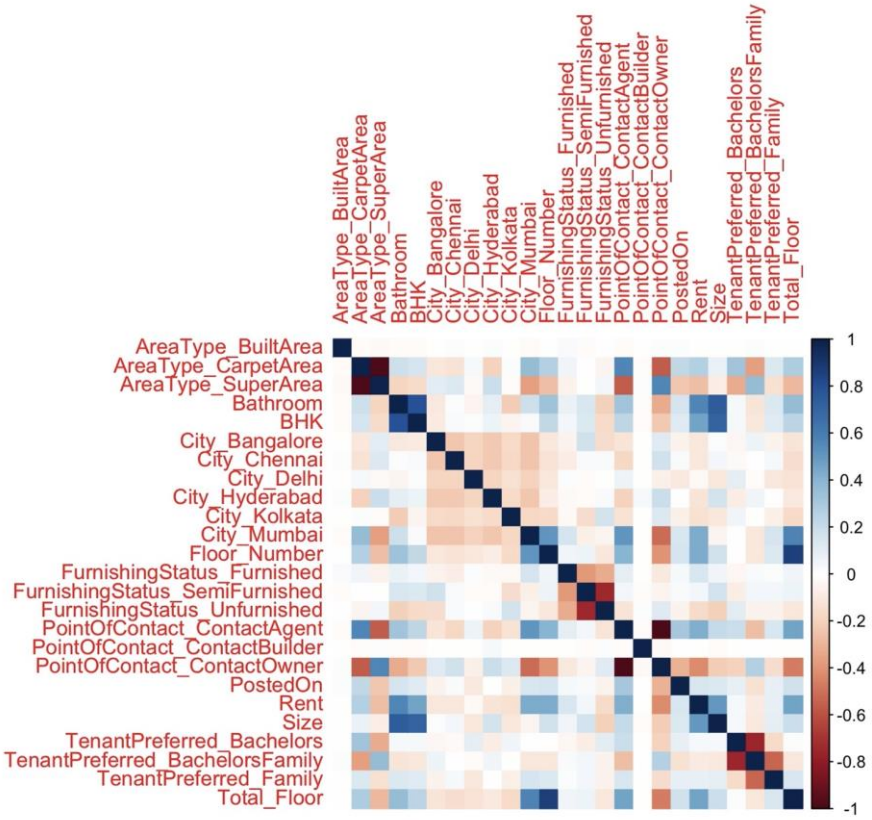


Figure 3: Correlation Matrix of variables

Data Preparation

We fixed several columns in our dataset to fit the formatting requirements for data mining:

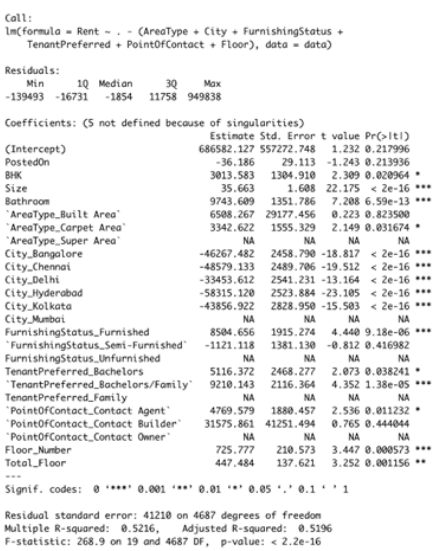
1. To analyze the data more conveniently, we renamed some variables to remove the "." in their variable names, for example, "Posted.On".
2. We converted some variables to dummy variables with the "fastDummies" package, such as "AreaType", "City", "FurnishingStatus", "TenantPreferred", and "PointOfContact". For example, the area type is separated into three columns – Built Area, Carpet Area and Super Area based on the size of the Houses/Apartments/Flats.

3. The original variable “PostedOn” is in Character format rather than in Date format. With the as.Date function, we transformed the variable into the Date format.
4. We split the column “Floor” into two columns - “Floor_Number” and “Total_Floor”, representing which floor number that the Houses/Apartments/Flats are located on and the total number of floors that the Houses/Apartments/Flats have. After this step, we replaced the ground floor with 0. Then we decided to add 1 to both “Floor_Number” and “Total_Floor” so that the ground floor could be the 1st floor rather than the 0 floor.
5. We removed NAs from our dataset, including those in the newly added “Floor_Number” and “Total_Floor” columns, because NAs only made up a small portion of our whole dataset. Also, we removed a significant outlier from the dataset, where the rent is $3,500,000.
6. We removed “AreaLocality” since we already have the variable “City” to indicate location.

Modeling

We used linear regressions, LASSO, Post-LASSO, and random forest models to forecast rent. We started by putting all variables into our linear regression model (Figure 4) and then we removed the insignificant variables to build up a new linear regression model. (Figure 5)

According to the attached model summary, the adjusted R-squared increased slightly (by 0.0001) than the original linear regression model and all variables are significant.

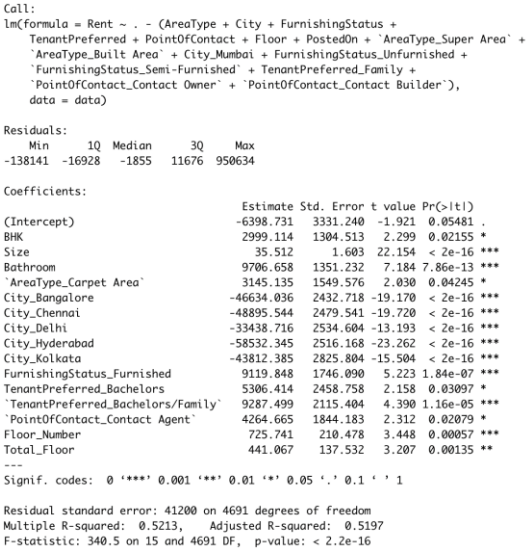


```
Call:
lm(formula = Rent ~ . - (AreaType + City + FurnishingStatus +
    TenantPreferred + PointOfContact + Floor), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-139493  -16731   -1854   11758  949838

Coefficients: (5 not defined because of singularities)
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                     686582.127 557272.748   1.232 0.217996
PostedOn                           -36.186     29.113  -1.243 0.213936
BHK                               3013.583   1304.910   2.309 0.020964 *
Size                                35.663      1.608  22.175  < 2e-16 ***
Bathroom                          9743.609   1351.786   7.208 6.59e-13 ***
`AreaType_Built Area`             6508.267  29177.456   0.223 0.823500
`AreaType_Carpet Area`            3342.622   1555.329   2.149 0.031674 *
`AreaType_Super Area`                   NA         NA      NA       NA
City_Bangalore                  -46267.482   2458.790 -18.817  < 2e-16 ***
City_Chennai                    -48579.133   2489.706 -19.512  < 2e-16 ***
City_Delhi                      -33453.612   2541.231 -13.164  < 2e-16 ***
City_Hyderabad                  -58315.120   2523.884 -23.105  < 2e-16 ***
City_Kolkata                    -43856.922   2828.950 -15.503  < 2e-16 ***
City_Mumbai                             NA         NA      NA       NA
FurnishingStatus_Furnished         8504.656   1915.274   4.440 9.18e-06 ***
`FurnishingStatus_Semi-Furnished`  -1121.118   1381.130  -0.812 0.416982
FurnishingStatus_Unfurnished            NA         NA      NA       NA
TenantPreferred_Bachelors          5116.372   2468.277   2.073 0.038241 *
`TenantPreferred_Bachelors/Family` 9210.143   2116.364   4.352 1.38e-05 ***
TenantPreferred_Family                  NA         NA      NA       NA
`PointOfContact_Contact Agent`     4769.579   1880.457   2.536 0.011232 *
`PointOfContact_Contact Builder`  31575.861  41251.494   0.765 0.444044
`PointOfContact_Contact Owner`          NA         NA      NA       NA
Floor_Number                        725.777    210.573   3.447 0.000573 ***
Total_Floor                         447.484    137.621   3.252 0.001156 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41210 on 4687 degrees of freedom
Multiple R-squared:  0.5216,    Adjusted R-squared:  0.5196
F-statistic: 268.9 on 19 and 4687 DF,  p-value: < 2.2e-16
```



```
Call:
lm(formula = Rent ~ . - (AreaType + City + FurnishingStatus +
    TenantPreferred + PointOfContact + Floor + PostedOn + `AreaType_Super Area` +
    `AreaType_Built Area` + City_Mumbai + FurnishingStatus_Unfurnished +
    `FurnishingStatus_Semi-Furnished` + TenantPreferred_Family +
    `PointOfContact_Contact Owner` + `PointOfContact_Contact Builder`),
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-138141  -16928   -1855   11676  950634

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                     -6398.731   3331.240  -1.921  0.05481 .
BHK                              2999.114   1304.513   2.299  0.02155 *
Size                               35.512      1.603  22.154  < 2e-16 ***
Bathroom                         9706.658   1351.232   7.184 7.86e-13 ***
`AreaType_Carpet Area`           3145.135   1549.576   2.030  0.04245 *
City_Bangalore                 -46634.036   2432.718 -19.170  < 2e-16 ***
City_Chennai                   -48895.544   2479.541 -19.720  < 2e-16 ***
City_Delhi                     -33438.716   2534.604 -13.193  < 2e-16 ***
City_Hyderabad                 -58532.345   2516.168 -23.262  < 2e-16 ***
City_Kolkata                   -43812.385   2825.804 -15.504  < 2e-16 ***
FurnishingStatus_Furnished       9119.848   1746.090   5.223 1.84e-07 ***
TenantPreferred_Bachelors        5306.414   2458.758   2.158  0.03097 *
`TenantPreferred_Bachelors/Family` 9287.499   2115.404   4.390 1.16e-05 ***
`PointOfContact_Contact Agent`   4264.665   1844.183   2.312  0.02079 *
Floor_Number                      725.741    210.478   3.448  0.00057 ***
Total_Floor                       441.067    137.532   3.207  0.00135 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 41200 on 4691 degrees of freedom
Multiple R-squared:  0.5213,    Adjusted R-squared:  0.5197
F-statistic: 340.5 on 15 and 4691 DF,  p-value: < 2.2e-16
```

Figure 4: Summary of original OLS results Figure 5: Summary of new OLS results

One of the pros of linear regression in this situation is that it is easy to implement and train. Also, many analysts have applied linear regression to predict housing prices. However, the linear regression models did not perform extremely well with our data (Adj. R^2 = 0.5197). This might be because linear regression is prone to noises, and it is sensitive to outliers.

Next, we employed the LASSO technique with the "cv.glmnet" function. Through this step, we found a lambda.min of 78.80972 and a lambda.1se of 10914.25. We built two separate models – LassoMin and Lasso1se using lambda.min and lambda.1se respectively. We used the "set.seed" function twice to keep the results reproducible and to guarantee that we produce the same random values every time we run the code.

- For the LassoMin model, "PostedOn", "BHK", "Size", "Bathroom", "Floor_Number", "Total_Floor" and some factors of the dummy variables "AreaType", "City", "FurnishingStatus", "TenantPreferred" and "PointOfContact" are included as our independent variables.

- For the Lasso1se model, “Size”, “Bathroom”, "City_Mumbai”, “PointOfContact_ContactAgent” and “Total_Floor” are included as our independent variables.

One of the most significant pros of LASSO is that it can provide more accurate results through automatic feature selection. During this step, the features that are not crucial for our analysis will be excluded automatically. Also, the problem of overfitting and multicollinearity will be circumvented. However, because of its automatic feature selection, some crucial features might also be ignored, resulting in an inaccurate result.

Considering the bias of LASSO as a shrinkage estimator, we further employed the post-LASSO estimator to better estimate the coefficients, which is only using LASSO for variable screening and then discarding the regression coefficients of LASSO, and then performing the OLS (ordinary least squares) regression on the variables. We built PostLassoMin and PostLasso1se based on LassoMin and Lasso1se, respectively. Lastly, we employed random forest technique. Random Forest can produce very high-dimensional data without feature selection. At the same time, it is less prone to overfitting. By using ensemble learning methods for regression, it can make more accurate predictions on the rent than a single model.

In the validation process, we found Random Forest performs best (refers to Evaluation), so we decided to interpret the importance of the variables from the perspective of Random Forest. We used the ggplot package to facilitate visualizing the importance of each variable through a metric called IncNodePurity. This is a measure of variable importance based on the Gini impurity index used for calculating the splits in trees. The higher the value of mean decrease accuracy or mean decrease gini score, the higher the importance of the variable to our model. We can tell that among all the variables that might have an impact on the rent, we found 5 of them are really the

"Big Cheese" in determining the rent, which are Total Floor, City Mumbai, BHK (number of bedrooms, hall and kitchen), bathroom and finally the size. However, of the 5 important features, Total Floor, City Mumbai and the size of the real estate are unchangeable in any sense, so if a savvy rental agent really wants to improve the financial value of their in-stock rental real estate, it would be advised to work on the bathroom and BHK. For instance, if a rental agent is willing to divide the living room into several small bedrooms, he/she can earn more revenue.

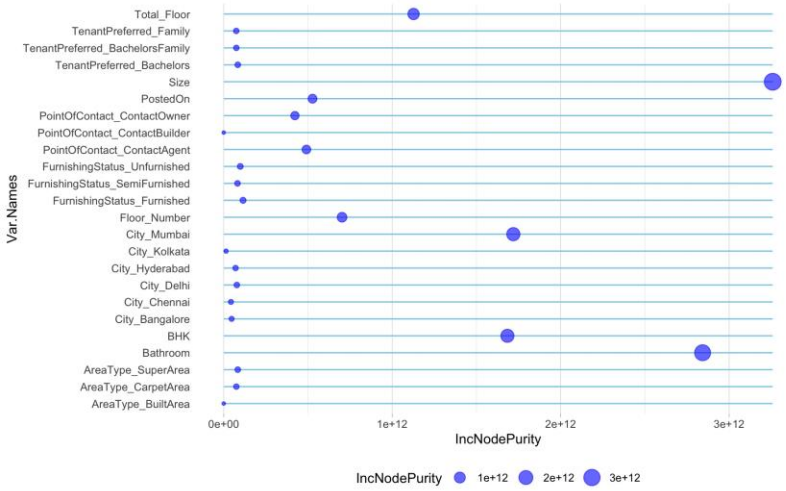


Figure 6: Visualization of the IncNodePurity of each variable

Our objective was to choose the best model to predict the fair market value of the rent and use this rent in our cost-benefit analysis framework to help the rental agents to 1) target underpriced real estate to seek arbitrage opportunities 2) improve future cash flows from existing rental housing stock by adding features that can bring in extra profits. The formula is as follows:

$$E(\mathrm{Pro}fit|X,a,b,c,d,O|) = Max\{(E(Rent|X,c,d) - E(Rent|X,a,b) - C(c-a,d-b)),0\} +$$

$$\{E(Rent|X,a,b) - C(O)\}$$

X = rental apartments feature except BHK and bathrooms | a = current number of BHK | b = current number of bathrooms | c = potential maximum number of BHKs the apartment could

have | d = potential maximum number of bathrooms the apartment could have | C (c-a, d-b) = incremental cost incurred to renovate the apartment to increase the number of BHK from a to c (e.g., from 4 bedrooms to 5 bedrooms) and to increase the number of bathrooms from b to d (e.g., from 1 bathroom to 2 bathrooms) | C(O) = price paid to the apartment owner (cost for obtaining the rental apartment）| E (Rent| X, a, b) - C(O)} = arbitrage profit agents can harvest for a bid-offer spread | Max {(E (Rent |X, c, d)- E (Rent| X, a, b)-C (c-a, d-b), 0} = maximum marginal profit we can earn by increasing the number of BHK and the number of bathrooms. If the marginal profit is less than 0, which means the marginal cost outweighs the marginal revenue, then agents should choose not to renovate, and the marginal profit could be 0.

Evaluation

We built K-fold structure to cross validate our models. We utilized 10 folds to train our linear regression, LASSO, post-LASSO, and random forest models. To find out which model gives us the best prediction, we looked at the Out-Of-Sample R-squared. To achieve this step, we set up a K-fold for loop. The following graph shows the performance of these six models in 10 folds.

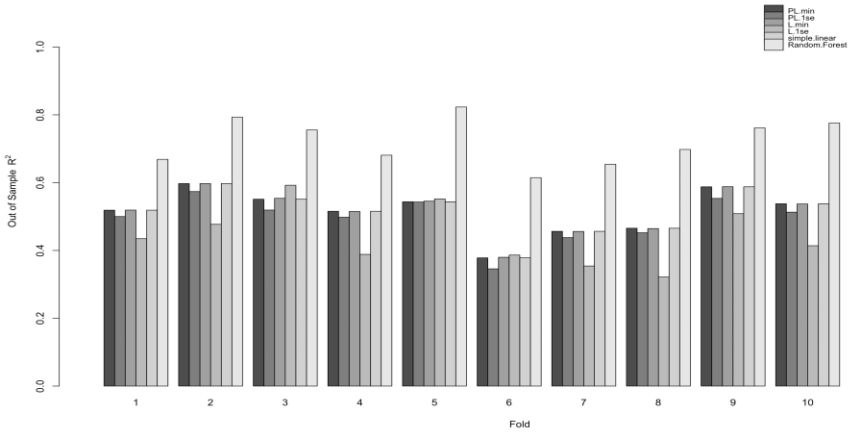


Figure 7: Performance of 6 different models based on 10 K-folds

Then, we calculated the average Out of Sample (OOS) R-squared of PostLassoMin, PostLasso1se, LassoMin, Lasso1se, simple linear regression and random forest. The average OOS R-squared values are 0.5152409, 0.4936628, 0.5157098, 0.4429373, 0.5152409, and 0.7226239 respectively. As we can see, random forest performs way better than other models on average. Thus, we are going to pick random forest as our final model.

Deployment

The result of the data mining can be deployed by the rental agents to better their profit. We would like to give the following example: Suppose an agent hears that there are two apartments in Mumbai available for rent and the basic information is provided. The agent uses our random forest model to forecast that the two apartments are worth 25,000 and 70,000 INR per month respectively in the market and plans to offer the apartment owner 21,000 and 62,000 INR. Luckily, the owner accepts the offer. Thus, the agent earns an arbitrage profit of 12,000 INR without putting in too much effort.

City	BHK	Bathroom	Size	Floor Num	Other Info	Forecast Rent	C(O)
Mumbai	1	1	320	2	….....	25,000	21,000
Mumbai	2	2	750	4	….....	70,000	62,000

To make it better, through our model, the rental agent is also able to know what extra profit he can earn if he/she divides the living room into more bedrooms or bathrooms before renting out these two apartments. Suppose a housing designer states that, if the rental agent wants to turn the living rooms into bathrooms and bedrooms, the two apartments could then become 2b2b and 3b3b as a result. This restructuring would incur a renovation cost of 49,000 and 51,000 INR respectively, and it would help to increase the forecast rent to 27,000 per month and 74,000 per month (Agent can get the newly forecasted rent by running the random forecast with the new bedroom numbers and bathroom numbers). Suppose the estimate is on a 2-year span.

City	BHK	Bathroom	Size	Floor Num	Other Info	Forecast Rent	C(O)	Reno Cost
Mumbai	2	2	320	2	….....	27,000	21,000	49,000
Mumbai	3	3	750	4	….....	74,000	62,000	51,000

For apartment 1, since *Max {(E (Rent |X, c, d)- E (Rent| X, a, b)-C (c-a, d-b), 0} = Max {((27,000-25,000) *24 month –49000),0} = Max {-1000,0} =0*, it would be more recommended not to renovate and keep the arbitrage profit of 4,000 (25,000-21,000).

For apartment 2, we can do the same calculation and get a marginal profit of *Max {(74,000-70,000) *24 month- 51,000),0} =45,000 INR.* Using our model, the rental agent can earn 57,000 INR in total with 12,000 INR as the arbitrage profit and 45,000 INR as the marginal profit for the decision to renovate apartment 2.

However, when using our model, agents should also be aware of the assumptions that we only consider the acquisition cost and renovation cost in our model. There might be other costs such as maintenance costs occurring in practice. Besides that, when deploying our model, the agents should be very careful about the time span the estimate is based on. Different time span would generate different decisions. In terms of risks, we think the riskiest part is that even though the random forest performs the best among those 6 models, its overall accuracy of 0.722623 in predicting the rent indicates we still have some uncertainty in our prediction. Our model is not well-trained for the high rent range since we didn't have enough data points for luxury apartments. Therefore, when agents try to forecast rent for expensive apartments, deviance might increase significantly. To mitigate those risks, we recommend agents to use our model as an important reference but not the only reference. Experience in the industry and business acumens also play a role here. As for the ethical considerations, we think our models might get accused of exploiting the tenants and apartment owners with algorithms. Since leasing market is a zero-sum game, if agents want to make a fortune from it, then someone else might feel exploited.

Appendix

Reference:

[1] *It's time to be a greedy landlord amid India's rental housing demand boom*. The Economic Times. (n.d.). Retrieved October 16, 2022, from https://economictimes.indiatimes.com/industry/services/property-/-cstruction/its-time-to-be-a-greedy-landlord-amid-indias-rental-housing-demand-boom/articleshow/93687475.cms

Contribution:

Erico Cuna: Data cleaning, Case write-up edits, Linear Regression Model.

Yixin Wang: Data Cleaning, Algorithm Selection, Modeling, Model Evaluation, Code Writing, Write-up Drafting, Write-up Editing, Visualization, Cost-Benefit Analysis.

Erik Wicks: PowerPoint, Proofreading, Cleaning, Code Review, Random Forest.

Rong Xiao: Data Cleaning, Linear Regression Models, LASSO, Major Write-up Editing, Proofreading.

Yue Zheng: Data Cleaning, Linear Regression Models, Write-up Drafting, Write-up Editing, Code Combination, Visualization on some models.