Data Modelling and Prediction for HDB resale prices

Name : Lee Yen Kai

Date: 2023-08-28

Introduction

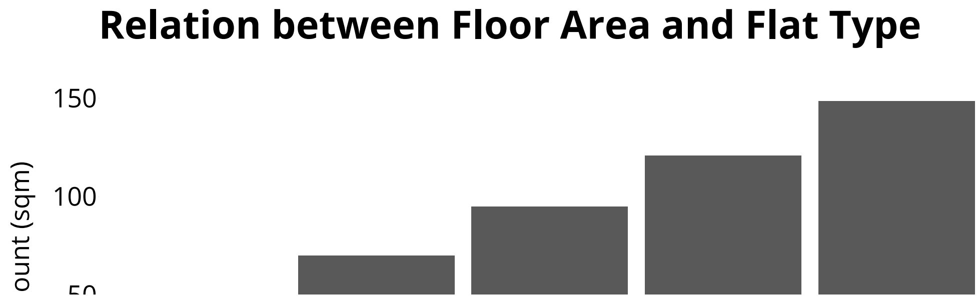
In a previous report, we have developed insights into the resale prices of HDBs. Now, in this report, I aim to develop a suitable model to predict the resale prices of HDBs given several key variables. The goal of this modelling is to accurately predict the price of HDBs given several parameters like the floor level or the area of the flat based on historical data. Since there are many variables driving the resale price of HDBs, we will look at creating a **Multiple Regression Model** to predict resale price.

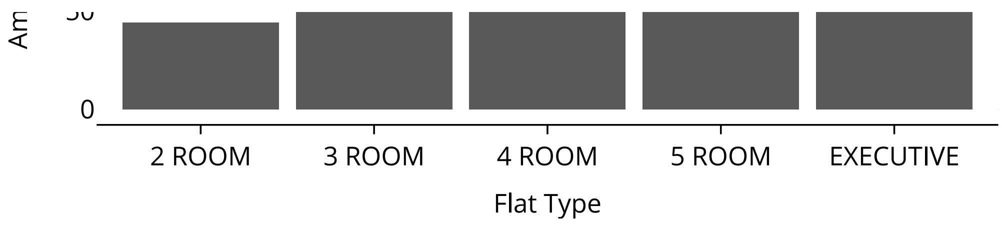
Multicollinearity

Firstly, before we develop our models, we have to first consider the prospect of multicollinearity. This occurs when there are two or more independent variables that are correlated with each other. In our model, our independent variables should correlate with our dependent y variable. Having dependent x regressors that are linearly correlated with each other would result in redundancy.

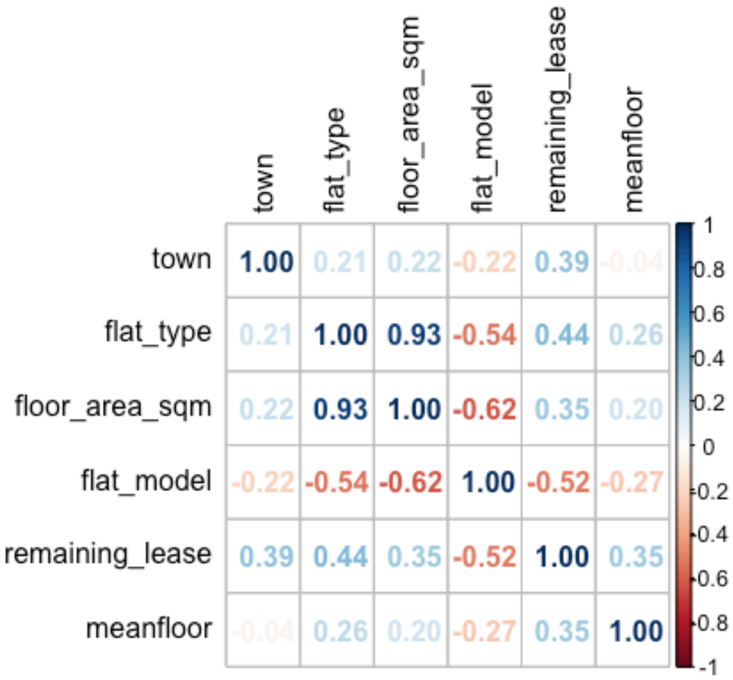
This is a problem because our independent variables should be sufficiently independent of each other. Where perfect multicollinearity is concerned, the effect of the affected x coefficients on our dependent variable would be indeterminate. Consider the equation $Y = aX + bZ + c$. If X and Z are linearly correlated, the effect of X (or Z) on Y would be difficult to determine since Z and X are linearly related, making causal inference impossible. Furthermore, it would be impossible to find the impact of increasing the coefficient a whilst holding bZ constant, since an increase in a would result in some change in bZ. This makes calculating the coefficients in a model impossible.

The most obvious example of multicollinearity in our dataset is the relation between the floor area of a flat and the flat type. We can very clearly see this linear relationship in our model.





We should also attempt to find other sources of multicollinearity in our dataset, as there might be less conspicuous relations between our independent variables. We can see in the matrix below that, aside from the multicollinearity mentioned earlier, there is no other strong correlation between our independent x variables.



We can see that if we were to keep the linear relation between flat type and floor area, we will encounter an error whilst developing our models. The error shown below indicates a condition known as **perfect multicollinearity**. The relationship between floor area and flat type is so evidently strong that VIF values cannot be generated.

```
fitexample <- lm(resale_price~.,data=testdata)
vif(fitexample)
```

```
## Error in vif.default(fitexample): there are aliased coefficients in the model
```

If we were to remove flat type from our data, we can see that there is no multicollinearity within our independent variables. When evaluating GVIF values, we can see that none of our $GVIF^{1/(2*Df)}$ values are less than √5, indicating no other multicollinearity in our data.

```
##                       GVIF Df GVIF^(1/(2*Df))
## town              1.650473  1        1.284708
## floor_area_sqm    2.700122  1        1.643205
## flat_model        8.868194  9        1.128905
## remaining_lease 2.500979  1        1.581448
## meanfloor         1.473381  1        1.213829
```

Data Modelling

Here, we will look at two multiple regression models, one of which disregards the town HDBs are sold in and the other takes that into consideration. Here, I aim to look at the impact of a HDB’s location on our modelling and if including it makes a difference.

Model 1

In our first model, we will create a model that takes the town, floor area, flat model, remaining lease, and the floor level as independent variables to predict resale price. The model created below gives us a rather high R^2 value at 0.8545 and an extremely infinitesimally small P value. The small P value indicates to us that it is very unlikely that the regression occurred by chance, and therefore, we should accept the model. The R^2 value (otherwise known as the coefficient of determination) shows us that approximately 85.5% of the variation in resale price can be explained by our model.

```
fit <- lm(resale_price~.,data=traindata)
summary(fit)$adj.r.squared
```

```
## [1] 0.8545377
```

```
pvalue(fit)
```

```
## [1] 0
```

We can then evaluate this model by fitting it onto our test dataset. Since our model was created using the training dataset, there is a possibility that our model may not actually reflect real-world data and is only representative of the training dataset. Since the test dataset has not been touched on by our model, we would like to see how our model would perform on data that it has not been trained on. This shows us how accurate our model is when it comes to real-world data and ensures that our model has not been overfitted to the training dataset.

```
fittest <- lm(resale_price~.,data=testdata)
summary(fittest)$adj.r.squared
```

```
## [1] 0.9335449
```

```
pvalue(fittest)
```

```
## [1] 1.055959e-152
```

We can see that our R^2 value has in fact increased with our P values remaining negligible. This shows that the model is reliable and that the predictions this model generates can be extrapolated without fear that the prediction might lose the accuracy seen in the model summary. In short, the model created here is acceptable.

Model 2

In our second model, we will remove town as an independent variable. In the model created below, we see an adjusted R^2 of 0.801 and, like the above model, a negligible P value. Given that the P values of both models are <0.05, both models can be accepted. The P value statistic from both models shows us that the two generated models are significant and are extremely unlikely to be the result of pure chance. The R^2 figure shows us that approximately 80% of the variation in resale prices (our dependent variable) can be explained by our model and the dependent regressors within, a decrease of 5% from our previous model.

```
fit2 <- lm(resale_price~.,data=traindata[,-c(1)])
summary(fit2)$adj.r.squared
```

```
## [1] 0.8010705
```

```
pvalue(fit2)
```

```
## [1] 0
```

Similarly, to the above model, we can also apply the model to our test dataset to ensure accuracy. When applied, we can see that our model holds up with an increased R^2 value of 0.877. In both models created, the R^2 increased substantially when applied to the test dataset.

```
fittest2 <- lm(resale_price~.,data=testdata[,-c(1)])
summary(fittest2)$adj.r.squared
```

```
## [1] 0.8770027
```

```
pvalue(fittest2)
```

```
## [1] 8.64017e-118
```

Model Selection

Now that we have generated our models, we would need to evaluate both of them in order to select our final model. Here, we will be looking at two statistics: Akaike Information Criterion (AIC) and Root Mean Square Error (RMSE)

Using Akaike Information Criterion (AIC)

Examining the Akaike Information Criterion (AIC) statistic provides us with a method of comparing the quality of models relative to each other. It estimates the amount of information lost by a given model when generating predicted data. By itself, the AIC value of a single model cannot provide much insight. However, the AIC values of two different models can be used to allow us to find the better model.

In a nutshell, AIC tests how well a model fits the data whilst at the same time ensuring that overfitting is penalised. Thus, AIC can be described as a test of the “Goodness of fit” of a model and how representative a model is of the data. Here, the lowest AIC value is desirable as the model with a lower AIC value would have the least amount of information lost, in addition to having a better fit for the data.

In our first model, we can see a generated AIC figure of 411230.4. This, by itself, cannot tell us much, so we have to examine the AIC value of the second model.

```
step <- stepAIC(fit, direction="both", trace = FALSE)
extractAIC(step)[2]
```

```
## [1] 411230.4
```

Here we see a larger AIC figure of 417186.4 for our second model, a difference of 5956 or 1.5%.

```
step2 <- stepAIC(fit2, direction="both", trace = FALSE)
extractAIC(step2)[2]
```

```
## [1] 417186.4
```

Given that the AIC value for the first model is lower, we can say that the first model is more desirable than the second. Perhaps the inclusion of the town parameter creates a better fit for the model and does not cause excessive overfitting. Likewise, we can accept that the second model, which did not include the town parameter, did not explain the data as well and might have been underfitted.

Using Root Mean Square Error (RMSE)

The second method of comparing models would be by using the Root Mean Square Error (RMSE) statistic. This statistic measures the difference between predicted values and actual values, thereby acting as a measure of errors present in a given model. In an ideal model where there is no difference between predicted values and the actual values, our RMSE values would be 0. Hence, we are looking for as small a number as possible.

RMSE can be calculated using $RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (P_i - O_i)^2}$ where n is the number of records, P_i is the predicted value, and O_i is the original known value. In our first model, we can see that our first RMSE value is 49220.09 when using the training dataset.

```
rmse.train<-sqrt(sum(residuals(fit)^2) / length(traindata$flat_model))
rmse.train
```

```
## [1] 49220.09
```

Likewise, for our AIC analysis, we would also have to find out the RMSE values for our second model in order to evaluate both models. In the second model, we see an increased RMSE value at 57560.97 for our training data, an increase of 8,340.11 or 16.9%! Therefore, the first model is more desirable due to its lower RMSE value, indicating that it is a more accurate model with fewer errors.

```
rmse.train2<-sqrt(sum(residuals(fit2)^2) / length(traindata$flat_model))
rmse.train2
```

```
## [1] 57560.97
```

Selected Model

After looking at our AIC and RMSE statistics, our final choice of model should be abundantly clear. Our first model has the smallest AIC statistic, as well as the smallest RMSE value, indicating that it is the best model to use. It also has a more desirable R^2 value, as well as a small P value. Therefore, all statistics considered, the first model (with the town variable) is our final selected model.

```
##   Models R.2.Train  R.2.Test      AIC RMSE.Train RMSE.Test
## 1      1 0.8545377 0.9335449 411230.4   49220.09  37499.62
## 2      2 0.8010705 0.8770027 417186.4   57560.97  51111.21
```