MIS772 – Predictive Analytics

T1 2022

Assignment 2 – Individual
Student name: Mirna Arivalagan
Student number: 220142881

Executive summary (1 page)

AirBnB is an international property leasing company, specialising in homestays for tourists. This report covers the analysis of properties located in the Melbourne Central Business District (CBD). From the dataset provided, there was a total of 728 properties located in the CBD.

Using the reviews data, we were able to analyse the comments to gain a better understanding of the sentiments within the reviews and found that there is a weak positive correlation of 0.275 between the sentiment scores and the most common overall rating scores. This indicates that customer sentiment towards the listing does not really affect the rating the customer provides the property. However, as we are using the raw sentiment scores rather than the ratio of positive to negative words, further testing is required to truly understand the impact of sentiment towards the rating. As by using, the ratio we found that the correlation was much higher to be at 0.442, indicating that sentiment could perhaps positively impact the overall rating a customer leaves the listing.

When analysing the rating scores by property type, we found the middle ratings across the property types to be quite similar, where private rooms scored the highest with 97, followed by entire homes/apartments with 96 and the lowest was hotel rooms with a middle rating of 90.

Using the sentiment score, we were able to develop a predictor model to estimate the overall rating of a property. Alongside the sentiment, we found that the attributes of cleanliness, check-in and location review scores to play a statistically significant role in predicting the overall rating. When analysing the relationship between each review rating indicators (Accuracy, Cleanliness, Checking, Communication, Location and Value), we found that there is a positive linear relationship between them and the overall rating.

By using just 4 attributes (Sentiment Score, Cleanliness, check-in and Location), we were able to develop model with moderate predictive powers, where 63.4% of the variation in rating scores can be accounted for the variation in these attributes, and the remaining 36.6% is accounted by other variables that are not included in the model.

There are limitations to the predictor model developed to predict the rating, one of the key limitations is that the dataset used to train the model is quite small, with less than 700 listings used to develop the model, further testing is required to test the robustness of the model with a larger dataset.

We were able to develop a clustering model that segments the listings into 6 different clusters by using the review score ratings (Accuracy, Cleanliness, check-in, Communication, Location and Value). Although this model did not provide the best performance results, having 6 clusters was the most practical option, as the model with the best performance was one with 15 clusters.

The first cluster identified, is a set of listings that had higher accuracy, check-in and communication scores, but did not perform well in terms of cleanliness and value for money, indicating that these properties could be more expensive but not maintained well. The second cluster identified was one that rated well on accuracy and cleanliness but had low ratings with the check-in process and somewhat low ratings for value. The third cluster had listings with low accuracy, cleanliness and location ratings, this cluster also had the lowest check-in, communication and value ratings, indicating that these listings could not be reflective of what was being advertised. The fourth cluster, had listings with high ratings for cleanliness, value and accuracy, and this cluster also had the highest ratings across all indicators, thus are very likely to be popular listings. The fifth cluster had lower ratings for accuracy, cleanliness, and value, but had a somewhat high rating for communication, which means that the hosts on these listing probably charge more for their listings but are also very communicative and responsive to questions from travellers. Finally, the 6th cluster was listings that had really low communication and accuracy rating and sits in the middle in terms of ratings around cleanliness, check-in, location and value.

Data exploration, pattern discovery, and preparation (2 pages)

To prepare the data to analyse the sentiment scores for the listings in the dataset present – First upon reading the excel file for listing and reviews, we need to join both these datasets using a join operator joining on the id and listing id field across both datasets, once joined, I selected the comments, id and review scores rating attributes. I then, ran the process to check for any missing values within the comments and rating attributes and there wasn't any missing values.

Once the data set has been merged, I used a subprocess to process the negative and positive words separately in their own subprocesses. To process and obtain the negative and positive word count, the following steps was applied (Fig 1):

1) Calculate the term frequency using the TF-IDF method for each word, and words with a higher TF-IDF weight, the more important the word is. This process goes through tokenization, transform cases, filter stop words, stemming using snowball method and filter tokens using a minimum length value of 4 characters. (Fig 2)
2) Next, we use the process document from data operator to calculate the term occurrence using the binary term occurrences vector – pruning has been set to none and data management has been set to auto.
3) Once the binary term occurrence has been calculated, the generate aggregation operator was used to sum up the total number of positive/negative words. The selection to ignore missing was selected as well.
4) Next, select the attributes we want to use, and pass it through another round of aggregation to sum the positive count and group by the review scores rating using the mode rating value.
5) Once the above is completed, and the process has been completed for both negative and positive words we can use the join operator to join both datasets into 1 to calculate the sentiment score (positive – negative words) and generate the correlation matrix. (Fig 3)

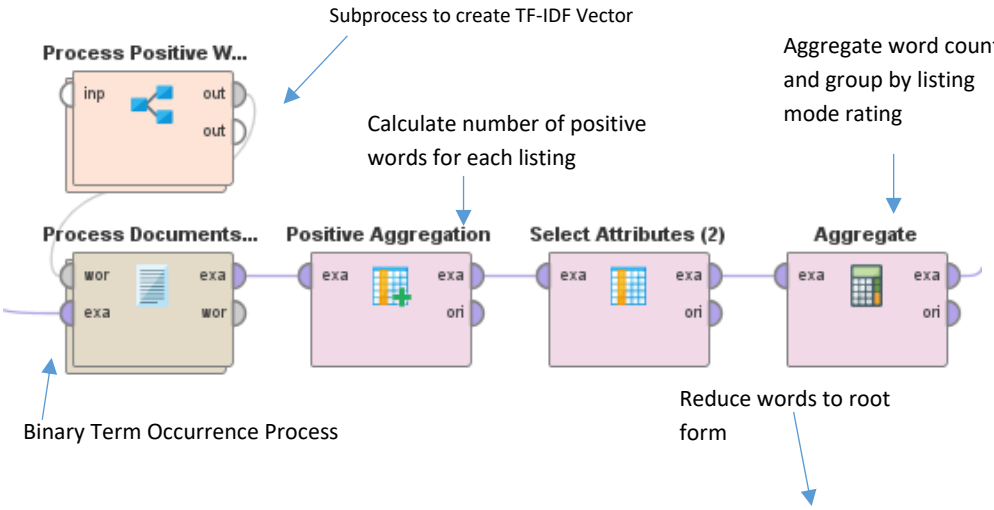


Fig 1

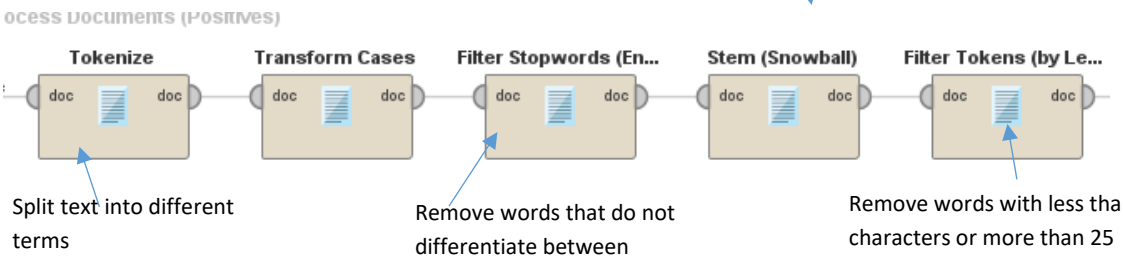
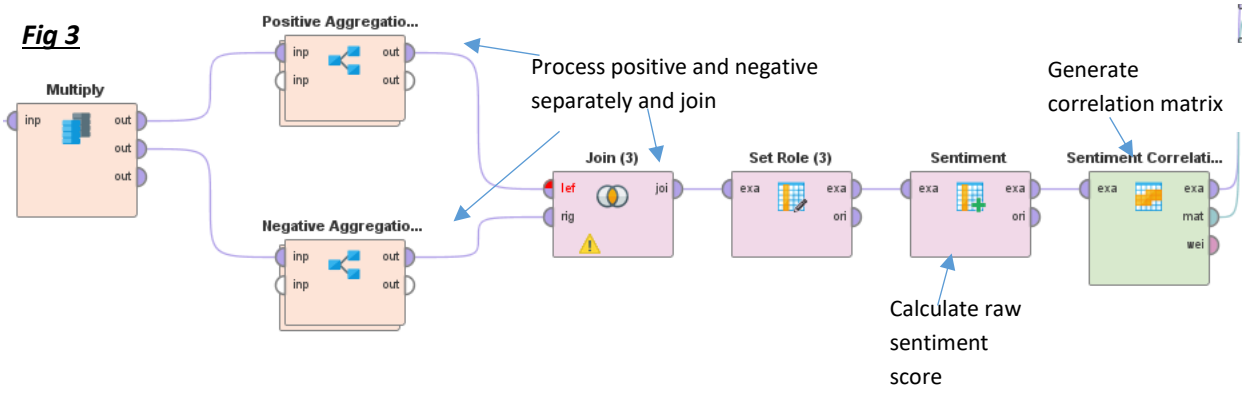


Fig 2

Fig 3



Following the process above, the correlation between the sentiment score and rating is a weak and positive one at 27.5%.

To prepare for the regression modelling, first we needed to identify CBD properties for the modelling. I used the generate attributes filter in (Fig 4) and created the CBD attribute using the latitude and longitude points. There was a total of 728 properties in the CBD and 282 properties not in the CBD in the dataset provided. Upon filtering, there was a total of 13 records with missing attributes in the rating scores attributes that will be used for the modelling (Accuracy, Cleanliness, Checking, Communication, Location & Value), as the value of this is low, we can replace them with the average value using the missing values operator (Fig 4). There was also a total of 50 properties with missing value for the bedrooms, in this instance it does not make sense to use an average figure, so I opted to drop these records using the filter example operator, leaving a total of 678 records to use in the model.

Next, we assess the relationship between the rating and the number of listings a host has, from (Fig 5) it can be observed that there is a slight drop in rating or a negative linear relationship when the host has a higher number of listings – indicating that perhaps hosts with higher number of listings, more than 10 could be agencies or hotels managing the properties, thus less personal touch is seen in the accommodation leading to a lower rating. A positive linear relationship can be observed with all review scores attributes (accuracy, cleanliness, communication, check-in and location), where an increase in scores can potentially lead to a higher overall rating. (Fig 6)

Fig 4

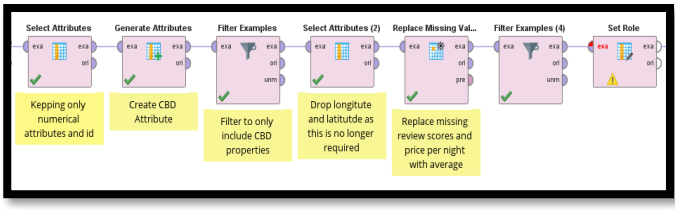
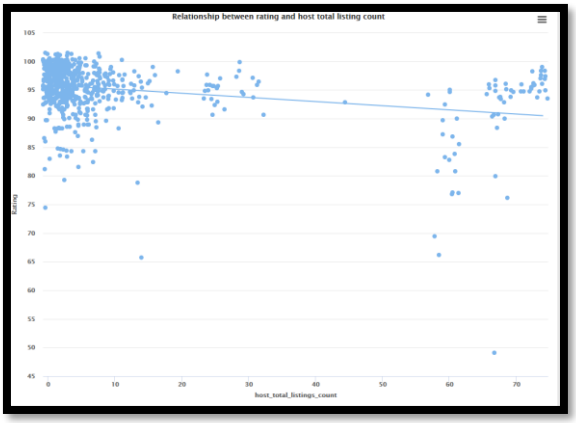


Fig 5



When analysing the rating medians, private rooms had the highest rating with a median rating of 97, followed by Entire homes/apt with a median of 96 and hotel rooms had the lowest median with a rating of 90. (Fig 7)

Fig 7

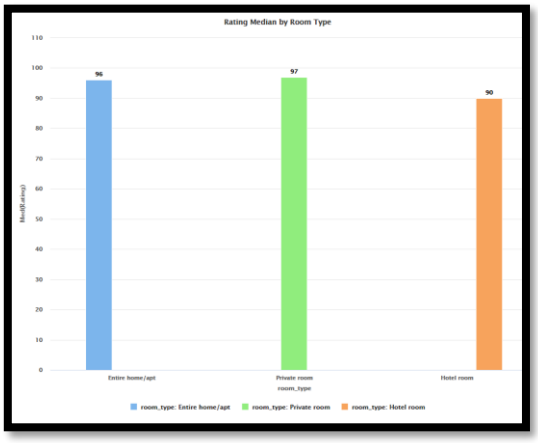
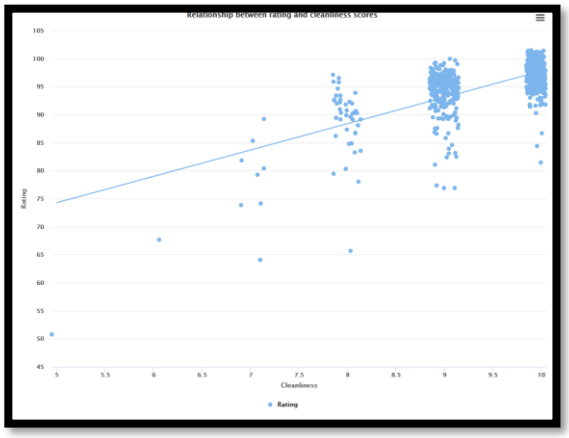
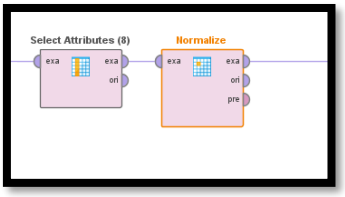


Fig 6



I repeated the data prep process above in a new process to prepare for the clustering models. In addition to the steps in Fig 4, I then select attributes operator to select the review scores attributes and then selected the Normalize operator to normalize the review scores to a range of 0 and 1 (Fig 8).



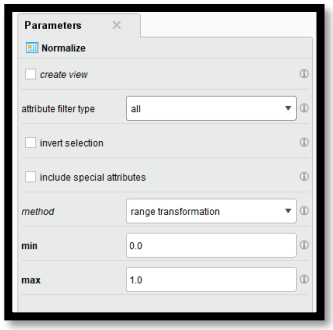


Fig 8

Predictive modelling (2 pages)

Linear regression was used to predict the overall rating of the property. This method was selected because we have a range of numerical review scores attributes (Accuracy, Cleanliness, Location, Check-in, Communication and Value) which is perfect to use in a linear regression model as this model works by using the strength of relationship between the numerical independent variable and the predictor variable (overall rating), the model will also provide insights around the strength of influence these independent variables has towards the predictor variable.

Upon completing the pre-processing steps in (Fig 4 above), I used the weight by correlation and correlation matrix operator (Fig 9) to understand correlation and relationship between the predictor variable and independent variable. Accuracy had the highest weight (0.769) and Reviews Per Month had the lowest weight (0.004). I opted to use all review scores and the sentiment score in the base model.

For the base model, I used the cross-validation operator, to cross validate across 3 splits as the dataset that we are dealing with is quite small and having more folds may overfit the models a little. The sampling method was set to automatic and local random seed was set to 2022 to ensure consistent reproducible results (Fig 10). The parameter used for the base linear regression model was set to M5 Prime feature selection, use bias is set to true, and eliminate colinear features was set to false (Fig 13). M5 Prime was used as the feature selection as this method uses regression trees to identify the best attributes to use in the model. The apply model operator and performance operator was used to apply the test data to the model and output the root mean squared error (RMSE), mean absolute error (MAE), correlation and squared correlation (R^2) values (Fig 11). RMSE measures the standard deviation of the residuals from the model and provides insights on how concentrated the data is around the line of best fit. MAE is a measure of errors between the actual values and predicted values and can be used as an indicator if outliers are present in the dataset. R^2 measures the proportion of variance in the predictor variable that is explained by the independent variables.

The performance of the base model was good with a R^2 value of 0.709 indicating that 70.9% of variation in the overall rating scores can be explained by the variation in accuracy, cleanliness, value, communication, location, check-in and sentiment scores. The remaining 28.9% of variation of rating is explained by other statistically significant independent variables that are not included in the model.

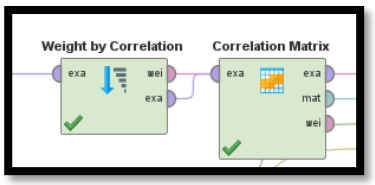
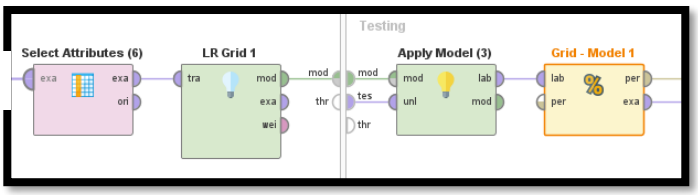


Fig 11



attribute	weight ↓
Accuracy	0.769
Value	0.701
Cleanliness	0.692
Communication	0.615
Checkin	0.614
Location	0.495
host_total_listings...	0.296
Sentiment Score	0.286
accommodates	0.162
bedrooms	0.094
price_per_night	0.042
Reviews Per Month	0.004

Fig 9

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
Accuracy	2.498	0.260	0.291	0.409	9.600	0	****
Cleanliness	1.857	0.170	0.274	0.601	10.919	0	****
Checkin	1.197	0.249	0.127	0.598	4.809	0.000	****
Communication	0.768	0.310	0.069	0.549	2.480	0.013	**
Location	1.985	0.379	0.116	0.777	5.242	0.000	****
Value	1.613	0.213	0.203	0.513	7.574	0.000	****
Sentiment Score	0.001	0.000	0.073	0.936	3.654	0.000	****
(Intercept)	-1.369	3.418	?	?	-0.401	0.689	

Fig 12

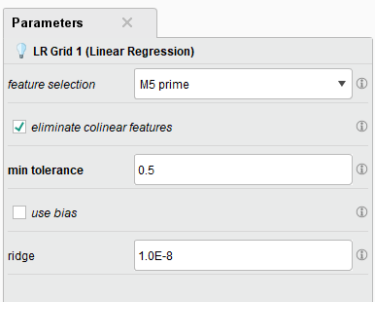


Fig 13

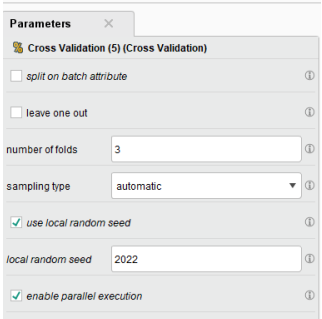


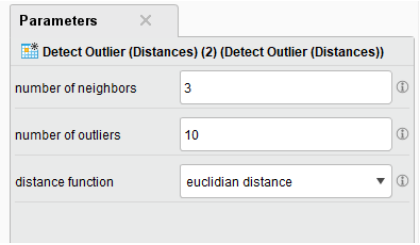
Fig 10

However, when we assess the output (Fig 12) of the regression model, it can be observed that accuracy and value had low tolerance values indicating that these attributes are multicollinear and problematic that should be removed (accuracy: 0.409, value: 0.513). Sentiment score even though had a weaker correlation weighting has the highest tolerance value at 0.936 thus we should keep this attribute in the model.

To prepare for the K means clustering model, similar pre-processing that was applied to the linear regression models was applied to the clustering models with the addition of using a normalizer and as K means is sensitive to outliers, I used the

detect outlier (distances) operator, where the parameters was set to number of neighbours = 3, number of outliers = 10 and distance function is Euclidian distance (Fig 14).Once the outliers was removed, there was a total of 656 listings that can be used in the modelling

Fig 14



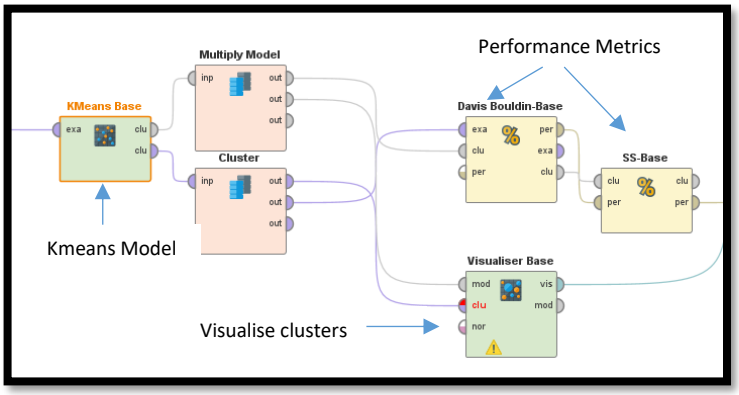
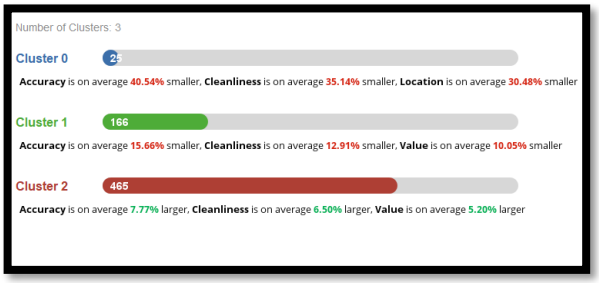


Fig 15

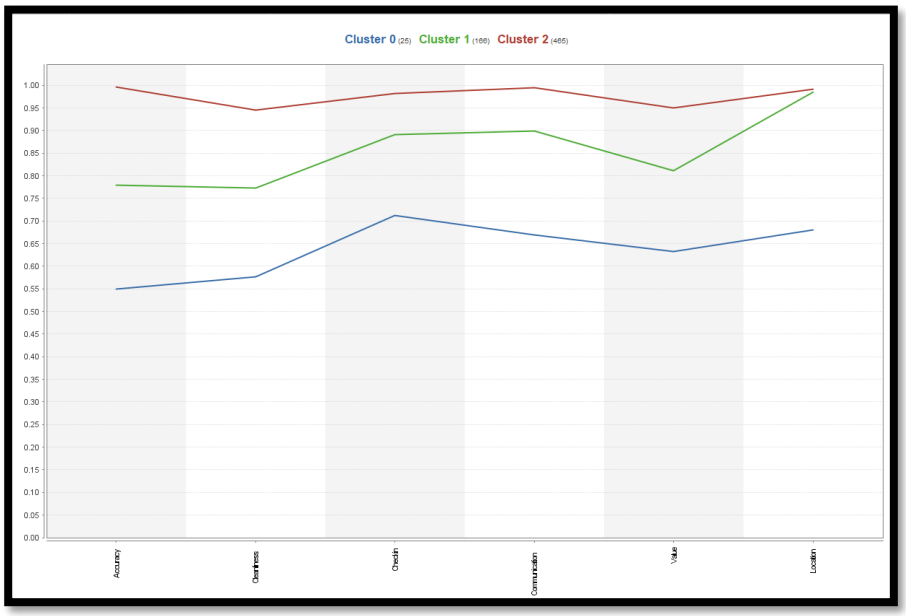
Next a subprocess is utilized to run a base K means model, where the clustering operator was used and the parameters was set to K = 3, max runs = 10, measure types = NumericalMeasures, Numerical measure = Euclidean Distance, Random Seed = 2022 and determine good start values was set to true. The model is then multiplied where the clusters is fed into the cluster model visualizer, and the clusters was also applied to the cluster distance performance operator to output the Davies Bouldin and Sum of Squares Error (SSE) values to measure the fit of the clusters. Davies Bouldin (DB) is the average similarity measure of each cluster with it most similar cluster, thus clusters that are further apart and more dispersed will provide a lower score indicating a better fit, as a score closer to 0 indicates better clustering. SSE measures the difference between each data point and the average within the cluster. (Fig 15)

Fig 16



The performance of the base model with K = 3, did not produce the best results, the DB score was 1.182, and the SSE score is 0.568. As per image (Fig 16) it can be observed that cluster 0, are properties with low accuracy, cleanliness and location ratings. Accuracy on average was rated 40.54% lower, compared to the other 2 clusters, Cleanliness on average was rated 35.14% lower and location was on average 30.48% lower than the other 2 clusters.

Fig 17



Cluster 2 are listing with high accuracy, cleanliness, and value ratings, where accuracy on average is 7.77% higher than the other 2 clusters, Cleanliness is on average 6.50% higher and value is on average 5.20% higher. Cluster 2 was also the clusters where listings had the highest rating scores across all rating attributes (Fig 17)

There was a total of 25 listings in cluster 0, 166 in cluster 1 and 465 in cluster 2 which is the biggest cluster.

Model evaluation and improvement (2 pages)

To improve on the regression modelling to predict the rating scores, I used the loop parameters operator to use it as a grid search to find the best combination of parameters to use for the model. The combination of parameters used is as follow:

1) Feature Selection: M5 Prime or Greedy
    - Greedy feature selection uses a stepwise selection method and can potentially lead to model overfitting if not used cautiously, and M5 Prime uses regression trees to select the best independent variables for the modelling.
2) Use Bias: True or False
    - Determines if an intercept value should be calculated or not
3) Min Tolerance: Set between the range of 0.5 to 0.7 in 5 steps using a linear scale. I've opted for a slightly higher min tolerance starting at 0.5 to reduce the problem of multicollinearity.

Alongside the loop parameter, I used the log to data operator to log the performance of the models in each iteration. (Fig 18).

Fig 18

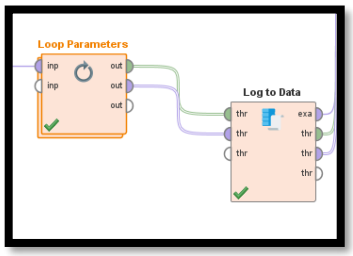
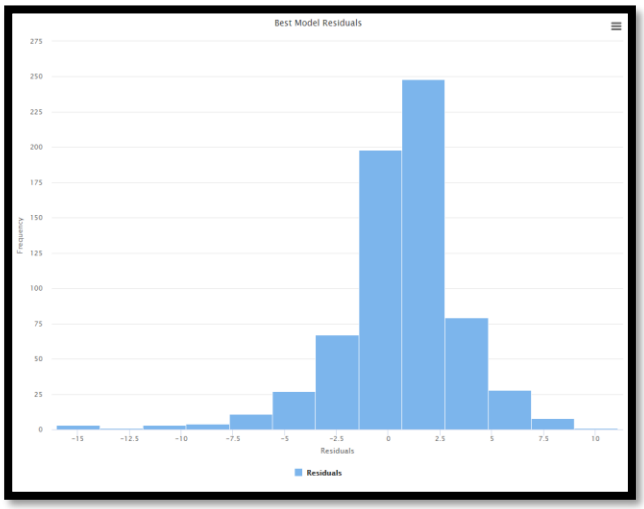


Fig 19

iteration	LR Loop...	LR Loop...	LR Loop.mi...	root_m...	absolut...	square...
3	M5 prime	false	0.500	2.366	1.703	0.735
7	M5 prime	false	0.540	2.366	1.703	0.735
4	greedy	false	0.500	2.366	1.703	0.735
11	M5 prime	false	0.580	2.366	1.703	0.735
8	greedy	false	0.540	2.366	1.703	0.735
15	M5 prime	false	0.620	2.366	1.703	0.735
19	M5 prime	false	0.660	2.366	1.703	0.735
16	greedy	false	0.620	2.366	1.703	0.735
23	M5 prime	false	0.700	2.366	1.703	0.735

Fig 21



The results from the loop that had 24 iterations in total only seen a slight improvement compared to the base model, and the R^2 values for each iteration was 0.735, thus I sorted the MAE values from lowest to biggest to select the best iteration with the lowest MAE values meaning less errors with the prediction. (Fig 19)

Iteration 3 was the best iteration with the MAE being 1.703, and RMSE being 2.366, and the best parameters was M5 Prime for feature selection, use bias is set to false, and the min tolerance is 0.5. (Image XX)

Although iteration 3 had the best performance values, the tolerance for some attributes such as accuracy (0.409) and value (0.513) was quite poor. So, we will retest again using the same parameters, but with accuracy, value removed as independent variables.

I then created another sub process to test the grid search parameters with Accuracy and Value excluded, and the R^2 value dropped to 0.666, and the tolerance has seen a slight improvement, however communication still had a low tolerance 0.602.

So, I then proceeded to run another experiment using the exact same parameters, but with communication removed as an attribute. This reduced the R^2 value to 0.643, but all attributes now has a tolerance of greater than 0.7 (Image Fig 20).

Overall, even though the grid search/loop parameters reduced the overall model's performance in terms of its predictive capabilities, the grid search enabled us to deal with the problem of multicollinearity with the attributes used, thus is a more robust model.

This final model, has a R^2 of 0.634, meaning that 63.4% of the variation in overall rating, can be explained by the variation in accuracy, check-in, location and sentiment scores. All variables used in the final model is also statistically significant as all p-values are 0

Fig 20

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
Cleanliness	3.236	0.174	0.477	0.786	18.620	0	****
Checkin	2.840	0.249	0.301	0.737	11.396	0	****
Location	3.754	0.424	0.219	0.842	8.848	0	****
Sentiment Score	0.001	0.000	0.109	0.947	4.650	0.000	****
(Intercept)	-1.022	3.983	?	?	-0.257	0.797	

The regression equation is: overall rating = -1.022 + (3.236 * cleanliness) + (2.840 * check-in) + (3.754* Location) + (0.001 * Sentiment Score).

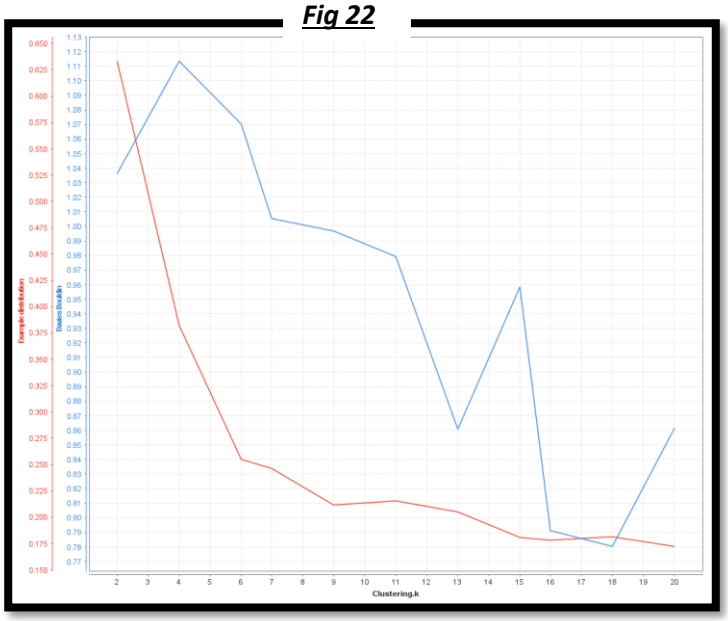
The residual for the model is slightly skewed, and this is due to outliers present in the dataset that was not removed for the regression model. (Fig 21). All 3 models (Base, Grid Search & Final) used a cross validation with 3 folds and sampling type is set to automatic.

Model	RMSE	MAE	Correlation	R^2
LR Base	2.379 +/- 0.418	1.706 +/- 0.257	0.856 +/- 0.047	0.745 +/- 0.079
LR Grid 1	2.721 +/- 0.319	1.914 +/- 0.206	0.816 +/- 0.016	0.666 +/- 0.026
LR Best	2.833 +/- 0.324	2.0204 +/- 0.221	0.802 +/- 0.015	0.643 +/- 0.024

To improve the performance of the models, I used the Optimize parameter operator, where the K value was set between 2 to 20 clusters in 20 steps, meaning that there will be 20 iterations with different K values. The modelling process is the same as our base K means model that we touched on earlier, except that the models are fitted into the optimize parameter operator.

Again here, we used Davies Bouldin and Sum of Squares Error to evaluate the iteration and find the best number of clusters for the final model. Here we will aim for the lowest possible DB value as closer to 0 is optimum and the lowest SSE value.

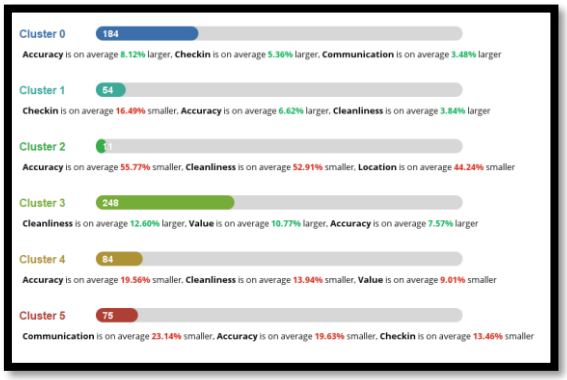
Fig 22



We need to check true to the log all criteria option in the optimize parameter operator to log the results. From the optimize parameter results, K = 15 provided with the lowest DB value of 0.823 (Fig 22), however it is not practical to have 15 clusters to use for marketing segmentation purposes, thus I plotted the result into a line chart to use the elbow method to find the optimum number of clusters. As per (Fig 22), the elbow bend can be observed to be at K = 6.

6 clusters provide a DB value of 1.233 and SSE value of 0.258. So even though the model optimisation process increased the DB value slightly, it managed to reduce the SSE value from 0.568 to 0.258, indicating that the data points within the clusters are closer to the average within the cluster, indicating closer similarities in property characteristics. Lower SSE and DB is achievable but will not be practical in a business setting to have many segments, especially considering the dataset the modelling was done on is quite small.

There is 184 listings in cluster 0, and these listings tend to rate accuracy on average 8.12% higher than other clusters, check-in is on average 5.36% higher and communication is 3.48% higher, these listings also see a lower cleanliness and value rating compared to other clusters, but not the lowest. Cluster 1 are listings with accuracy being rated 6.62% higher and cleanliness rated 3.84% higher, however these listings received a much lower check-in rating, on average 16.49% lower, there was a total of 54 listings in this cluster. Cluster 2 was the smallest cluster with only 11 listings, and this cluster on average 55.77% lower accuracy ratings, and is the cluster with the lowest rating scores across all attributes. Cluster 3 was the largest cluster with a total of 248 listings, and this cluster has seen on average 12.60% higher ratings for cleanliness, 10.77% higher ratings for value and accuracy ratings is 7.57% higher than other clusters. Cluster 3 also had the highest ratings across all attributes. Cluster 4 had a total of 84 listings, and this cluster rated accuracy (19.56%), cleanliness (13.94%) and value (9.01) lower compared to other clusters. Cluster 4 also had good ratings for cleanliness, communication, and location. The final cluster, cluster 5 had 75 listings, and this cluster has communication ratings 23.14% lower, accuracy 19.63% lower and check-in 13.46% lower on average compared to other clusters.



Optimized cluster model clusters summary