Airbnb Toronto Data science Project

Team 4

Deliverable 3: Modelling

Sheridan College

Date: Nov.22nd/2021

<u>Table of Contents</u>

Objective of Deliverable 3

The goal for Deliverable 3 was to create linear regression models to address the business case developed in Deliverable 1 with the clean dataset created in Deliverable 2. The first model would predict the price of an Airbnb rental listing or unit per day while the second would predict the review score. In addition, we needed to find the best model based on evaluation metrics that would predict price and review score. The two models would serve two primary groups, Stakeholders at Airbnb, and Airbnb Hosts. Stakeholders at Airbnb would use the models to develop business strategies that maximize features in the model that are predictive of higher prices and better reviews. For example, a finding that the location of listing positively affects price could be used to promote listings in areas where prices are higher. Additionally, a new Airbnb host could use these models to help set the price of their unit while giving them insights into how to maximize their review score.

The cleaned data (refer to Table 1 for description) generated from Deliverable 2 was used by the team to produce eight regression models using four types of Regression supervised methods, which allowed the prediction of price and review score for rental units in Toronto.

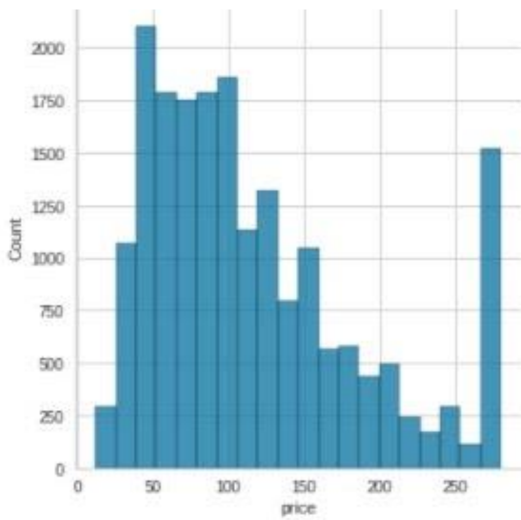
Steps in Development of Linear Regression Models of Price and Review Scores rating

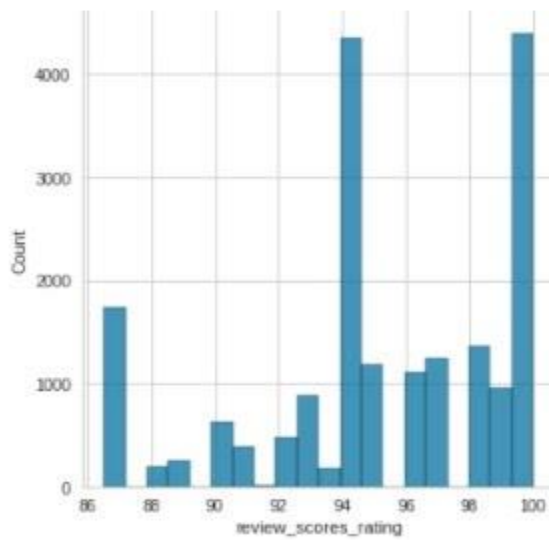
The modeling process followed a systematic set of steps. We first conducted a brief EDA of the cleaned dataset generated from Deliverable 2 to identify correlations between the features with target variables being price and review score ratings. Our intent was to determine if there were certain features predictive of price or review score ratings. After dummy coding categorical variables in our clean dataset, we implemented a standard regression to identify the contribution of features as predictors to price or review score ratings. We then implemented a machine learning based linear regression model with either price or review score ratings as being the target variables. Next, a random forest regression method was implemented, as we felt the complexity of our dataset would necessitate a more sophisticated machine learning method where decision trees were present. Finally, we implemented the pycaret package to identify which machine learning regression model would be best suited for prediction of our target variables. The Linear gradient boost regression (LGBM) and Gradient boost regression (GBM) were identified as the best model for price and review score ratings respectively, based on metrics of R-squared score and Root Mean Squared Error (RMSE). Thus, we implemented LGBM on price and GBM on review score ratings as targets. The results showed that there was a significant difference in model prediction for price compared to review score ratings.

Importing Packages and brief EDA

Relevant packages were imported for Linear regression from Statsmodel and Sklearn library to conduct standard linear regression and machine learning of the data. As presenting the description of the data types and correlations would take up space, we present the figures in the appendix. In the cleaned dataset as noted from *Figure 1* in appendix, there were 39 features, 14 of float data type, 16 of integer and 8 of the object type. We conducted a correlation analysis of price with other features in the dataset. Our correlation table shown in the *Figure 2* in Appendix, generated in Python showed 'accommodates' ($r = .56$), 'bedrooms' ($r = .54$), beds ($r = 0.49$) to be the three features that were more correlated to 'price' than other features; meaning the more bedrooms, beds and higher capacity in the listing the higher the rental price of the listing. What was also interesting was that 'review score ratings' had almost no relationship with 'price' ($r = .08$). We subsequently conducted a correlation analysis of review score ratings with other features in the dataset. Our correlation table shown in the *Figure 3* in Appendix, generated in Python showed 'review_scores_cleanliness' ($r = .65$), 'review_scores_accuracy' ($r = .64$), ‘review_scores_value’ (r = 0.64) to be the three features that were more correlated to ‘review scores ratings’ than other features. However, ‘review score ratings’ had almost no relationship with ‘price’ (r = .08).

Another aspect of the data which was relevant to the output of the regression models was the difference in distributions in data by ‘price’ and ‘review_ratings_score’. As seen from the figures below, Price has more of a normal distribution, while review score is skewed to the right.





Standard Linear regression

The purpose of conducting standard linear regression, which is a form of EDA leading into Machine Learning was to determine the contribution of predictors to target variable. Two metrics are relevant to standard linear regression, p-value <.05 as significance of contribution of predictor to variance in dependent variable and R-squared which is the proportion of variance in the dependant variable which is predicted from the independent variable. As shown from an excerpt of output from Standard Linear regression with target being price, the R- squared value was 0.53, meaning that the model explained 53% of the variance in price and with a F statistic of 509.3 indicating some features contributed significantly to variance in price. A description of the statistically significant features in the model are provided. Longitude and bedrooms had high coefficients showing that when longitude increased by 1, price would increase by $82 and when bedrooms increased by 1, price would increase by $29. Interestingly, for the dummy coded categorical variable ‘former city’, the districts North York and City of Toronto had large coefficients showing that with increase by one listing in North York and Old City of Toronto, price would increase by $23 and $20 respectively. However, certain features led to prediction of a decrease in price. For example, when listings were of the type private room and shared room, price would be predicted to decrease by $35 and $51 respectively.

Output from Standard Linear regression with target being price

OLS Regression Results

Dep. Variable:	price	**R-squared:**	0.532
Model:	OLS	**Adj. R-squared:**	0.531
Method:	Least Squares	**F-statistic:**	509.3
Date:	Wed, 17 Nov 2021	**Prob (F-statistic):**	0.00
Time:	20:48:00	**Log-Likelihood:**	-1.0255e+05
No. Observations:	19343	**AIC:**	2.052e+05
Df Residuals:	19299	**BIC:**	2.055e+05
Df Model:	43		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.663e+04	1059.930	15.688	0.000	1.46e+04	1.87e+04
longitude	82.3151	9.308	8.844	0.000	64.071	100.559
latitude	-233.2391	15.835	-14.729	0.000	-264.277	-202.201
bedrooms	29.1728	0.931	31.343	0.000	27.348	30.997
beds	-3.4195	0.779	-4.387	0.000	-4.947	-1.892
host_total_listings_count	1.8417	0.278	6.631	0.000	1.297	2.386
review_scores_rating	1.2362	0.145	8.551	0.000	0.953	1.520
review_scores_accuracy	1.1270	1.394	0.808	0.419	-1.606	3.860
review_scores_cleanliness	4.7269	0.744	6.356	0.000	3.269	6.185
review_scores_checkin	-12.3456	1.990	-6.204	0.000	-16.246	-8.445
review_scores_communication	-2.9191	2.289	-1.275	0.202	-7.407	1.568
review_scores_location	13.6364	1.522	8.960	0.000	10.653	16.620
review_scores_value	-5.7392	0.808	-7.100	0.000	-7.324	-4.155
reviews_per_month	2.2493	0.647	3.476	0.001	0.981	3.518
accommodates	11.4470	0.423	27.082	0.000	10.618	12.275

host_acceptance_rate	0.3453	0.076	4.542	0.000	0.196	0.494
host_length	0.4061	0.167	2.437	0.015	0.079	0.733
instant_bookable_t	-2.0346	0.797	-2.552	0.011	-3.598	-0.472
former_city_Etobicoke	8.5782	3.570	2.403	0.016	1.581	15.576
former_city_North York	23.4454	3.097	7.572	0.000	17.376	29.515
former_city_Old City of Toronto	20.4164	2.803	7.283	0.000	14.921	25.911
former_city_Scarborough	3.1213	3.197	0.976	0.329	-3.144	9.387
former_city_York	6.1394	3.529	1.740	0.082	-0.778	13.057
room_type_Hotel room	5.4936	6.379	0.861	0.389	-7.009	17.997
room_type_Private room	-35.5520	1.495	-23.776	0.000	-38.483	-32.621
room_type_Shared room	-51.9939	2.971	-17.500	0.000	-57.817	-46.170

Interestingly, as shown in excerpt from output in the figure below, for standard linear regression with target as ‘review score ratings’, the F-statistic was more significant (F = 811.1) with R-squared value being 0.64 which was higher than the price model, yet different predictors contributed to the variance in the model. Statistical significant features in the model can be seen in the figure below. For example, latitude, total host listings count, and all measures relevant to review score ratings such as review scores location were predictive of review score ratings, and there was contribution of district (i.e., former city).

Interestingly, when review scores of cleanliness and review scores location increased by 1, review score ratings would increase by 51 and 14 respectively. Yet with a decrease in host total listings count by one, there was a decrease in review scores rating. Why the major difference in statistical significant features in review score ratings model from the price model? As noted earlier, price has more of a normal distribution as most of the values are around the median, while review score is skewed to the right.

Standard linear regression with target as 'review score ratings'

OLS Regression Results

Dep. Variable:	review_scores_rating	R-squared:	0.644
Model:	OLS	Adj. R-squared:	0.643
Method:	Least Squares	F-statistic:	811.1
Date:	Sun, 21 Nov 2021	Prob (F-statistic):	0.00
Time:	21:39:57	Log-Likelihood:	-44489.
No. Observations:	19343	AIC:	8.907e+04
Df Residuals:	19299	BIC:	8.941e+04
Df Model:	43		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	106.4247	53.004	2.008	0.045	2.531	210.318
longitude	0.3017	0.463	0.651	0.515	-0.607	1.210
latitude	-2.1074	0.791	-2.664	0.008	-3.658	-0.557
bedrooms	-0.0135	0.047	-0.284	0.777	-0.106	0.079
beds	0.0036	0.039	0.094	0.925	-0.072	0.080
host_total_listings_count	-0.0386	0.014	-2.797	0.005	-0.066	-0.012
review_scores_accuracy	2.5273	0.067	37.797	0.000	2.396	2.658
review_scores_cleanliness	1.7785	0.035	51.237	0.000	1.710	1.847
review_scores_checkin	1.2208	0.099	12.381	0.000	1.028	1.414
review_scores_communication	2.6206	0.112	23.355	0.000	2.401	2.841
review_scores_location	1.0985	0.075	14.573	0.000	0.951	1.246
calculated_host_listings_count_entire_homes	0.0044	0.025	0.175	0.861	-0.045	0.054
calculated_host_listings_count_private_rooms	-0.0336	0.048	-0.701	0.483	-0.128	0.060
calculated_host_listings_count_shared_rooms	-2.407e-16	8.76e-17	-2.747	0.006	-4.12e-16	-6.89e-17
host_acceptance_rate	-0.0019	0.004	-0.495	0.620	-0.009	0.006
host_length	0.0086	0.008	1.041	0.298	-0.008	0.025
instant_bookable_t	0.0129	0.040	0.325	0.745	-0.065	0.091
former_city_Etobicoke	0.0568	0.177	0.320	0.749	-0.291	0.405
former_city_North York	0.2630	0.154	1.707	0.088	-0.039	0.565
former_city_Old City of Toronto	0.0384	0.140	0.275	0.783	-0.235	0.312
former_city_Scarborough	0.1517	0.159	0.955	0.340	-0.160	0.463
former_city_York	0.2510	0.175	1.431	0.152	-0.093	0.595

Metrics for our evaluation of Machine Learning Regression Models

Standard Linear regression enables a general understanding of whether there are features predictive of the target variable. To evaluate machine learning models which would enable us to determine the best machine learning model applied to our clean dataset, six metrics were collected. Mean Absolute error (MAE)-how far away predicted values are from observed values; Mean Squared Error (MSE)-The quality of a predictor based on the average square difference between the observed and predicted values; Root Mean Squared Error (RMSE)-Value difference between the true and predicted values; R-squared (R)-Proportion of variance in the dependant variable, which is predicted from the independent variable, Root Mean Squared Log Error and Mean Absolute Percentage Error. A detailed definition of Metrics is provided in *Figure 4*.

Modeling-Linear Regression

To test our a linear regression model on our cleaned dataset, we split the dataset into a training and test dataset. We decided on a larger test/train split (70/30) due to the skew of data points to get a more accurate result. We assigned 70% of our dataset to training and 30% to testing. We imported the LinearRegression from sklearn.linear_model and created an object of the Linear Regression class, after which we fitted our x and y to the machine learning regression model to predict price or review score.

Price as target

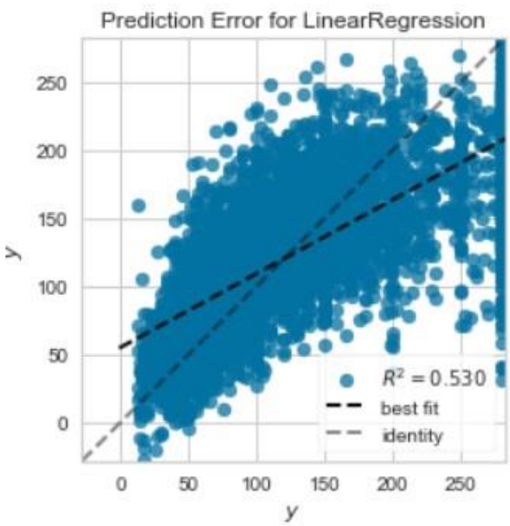
The excerpt of code below shown along with output indicates the R-squared, MAPE, MAE and RMSE values for Linear regression model with price being the target. As shown from the output, our R-squared value was 0.53 and the linear equation was y = 16770.51 +81.53x; MAPE being 0.39, MAE = 36 and RMSE = 48. The graph below shows prediction error with trend line for Linear Regression.

```
#In this case, 70% of data allocated to training set
xtrain, xtest, ytrain, ytest = train_test_split(X,y,train_size=0.70,random_state=42)

xtrain.shape, xtest.shape, ytrain.shape, ytest.shape

#Create the regressor
lm2 = LinearRegression()
lm2.fit(xtrain, ytrain)

predicted = lm2.predict(xtest)
```




```
R2 (explained variance): 0.53
Mean Absolute Prediction Error (Σ(|y-pred|/y)/n): 0.39
Mean Absolute Error (Σ|y-pred|/n): 36
Root Mean Squared Error (sqrt(Σ(y-pred)^2/n)): 48
 y = 16770.51990337229 + x * 81.53415644397592
```

A split of training set into 80% of the dataset yielded a similar output.

Review score ratings as target

Interestingly, the linear regression model for review score ratings was more significant in comparison. As shown in the output below, our R-squared was 0.64 and the equation was y = 183.67-3.59x; MAPE being 0.02, MAE = 2 and RMSE = 2. When comparing the RMSE (value difference average in values) from linear models with Price and Review score ratings as target; we can note for price minimum is 12 and maximum is ~280, with RMSE being ~48. For review score minimum is ~5.5 and maximum is ~10, with RMSE being 2. A spread of 40 in a range of 270 is better than a spread of 2 in a range of ~5. Thus, the RMSE for linear model on price is better than that of review score ratings.

```
R2 (explained variance): 0.64
Mean Absolute Prediction Error (Σ(|y-pred|/y)/n): 0.02
Mean Absolute Error (Σ|y-pred|/n): 2
Root Mean Squared Error (sqrt(Σ(y-pred)^2/n)): 2
 y = 183.66782964660536 + x * -3.588079172700013e-14
```

Modeling-Random Forest regression

We felt that a different regression model such as the case with Random Forest regression would yield a better prediction of price or review score ratings based on training dataset. The rationale being that random forest regression is very useful for complex datasets by splitting the data as a tree-like structure, into smaller and smaller subsets and then make predictions based on what subset a new example would fall into. An advantage of implementation of decision trees is that the sum of squared residuals is minimized by splitting the training examples because of learning by decision trees. The output value of a decision tree is predicted by taking the average of all the examples that fall into a certain leaf on the decision tree and the leaf is used as an output prediction [1].

Price as target

Indeed, with the implementation of a random forest decision tree on our dataset, R-squared increased from 0.53 found in Linear Regression model to 0.6 and RMSE decreased from 48 found in Linear Regression Model to 44. All indicators that the RF model was working better than Linear regression model.

```
R2 (explained variance): 0.6
RandomForest Regressor MSE is: 1957.5054863394244.
RandomForest Regressor MAE is: 31.661670549715666.
RandomForest Regressor RMSE is: 44.24370561265663.
```

Review score ratings as target

As was found with linear regression, there was higher R-squared for random forest regression model with review score ratings being the target variable.

```
R2 (explained variance): 0.65
RandomForest Regressor MSE is: 5.941974852378349.
RandomForest Regressor MAE is: 1.4869528739468227.
RandomForest Regressor RMSE is: 2.437616633594862.
```

Modeling-Light Gradient Boosting Model selected as best Linear Regression model

As it would not be efficient to test out every model one at a time on our cleaned dataset to determine the best model for prediction of our target variables, we opted to use a well-designed package that allowed a quick comparison of several regression models applied to our dataset. The supervised modeling package called PyCaret was implemented to select the best model for each feature. The documentation on PyCaret is found here [2].

Price LightGBM Model

Setting up the Price Model

After validating our cleaned data, it was sent through a preprocessing pipeline included in the package. The package allows imputations, code dummy variables among other preprocessing steps. Dummy columns were created automatically for the categorical data in preparation for model testing. We decided on a larger test/train split (70/30) due to the skew of data points to get a more accurate result.

```
In [7]: #Using pycaret module to determine which model would be best.
        #Hit enter when prompted to continue set up, asking about column variable types.
        from pycaret.regression import *
        s = setup(data, target = 'price',
                  session_id = 123, train_size = 0.7)
```

	Description	Value
0	session_id	123
1	Target	price
2	Original Data	(19343, 40)
3	Missing Values	False

Determining metrics of importance and PyCaret model evaluation

Using the compare_models() function, the data went through each of the 17 models and returned a table listing metrics for comparison and an output report was generated showing the standard six metrics mentioned earlier for each as well as the time taken to generate the model.

In terms of why we selected certain metrics over other metrics for evaluation of regression models, we provide our reasoning as the following. Firstly, when reporting the predicted price, we were less focused on reporting precise values and more on a range of values.

There were several factors that are not included in our preprocessed and cleaned dataset such as the aesthetic value, relative location to popular attractions and venues and crime levels. As such we decided to use RMSE rather than the R-squared value to determine the viability of our tested models. This would allow us to tell potential renters a range of potential earnings they can expect for their rental unit. For review score ratings, we were most concerned with how accurate our predictions were. As most of the review ratings are received during or after listing transactions, we wanted to find which features had the greatest impact on a rental listing's review rating. We decided to go with the R-squared score for our determinable metric as it signifies the variation of the review and we can find which features have the most impact on it.

The comparison below for target variable being price, lists Light Gradient Boosting Machine as the model of choice for RMSE, as well as each of the other metrics followed closely by random forest. LightGBM is an open-source framework based on random forest modeling but paths vertically by leaves rather than horizontally by branches. Trees are added one at a time and fit to correct predictions errors made by previous generations, thereby ‘boosting’ the results. The model itself has an RMSE of around 40, which is acceptable when predicting a range of values for renting.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
lightgbm	Light Gradient Boosting Machine	31.4060	1929.9485	43.9074	0.6199	0.3567	0.3129	0.0690
rf	Random Forest Regressor	32.2055	2027.4467	45.0038	0.6006	0.3645	0.3223	1.2620
gbr	Gradient Boosting Regressor	33.3540	2108.1634	45.8943	0.5848	0.3776	0.3377	0.4250
et	Extra Trees Regressor	32.5954	2165.4108	46.5085	0.5735	0.3744	0.3250	1.4540
lr	Linear Regression	36.0115	2366.6580	48.6258	0.5340	0.4371	0.3769	0.2890
ridge	Ridge Regression	35.9928	2367.1384	48.6306	0.5339	0.4337	0.3761	0.0210
br	Bayesian Ridge	36.0759	2382.4473	48.7876	0.5309	0.4300	0.3757	0.1350
omp	Orthogonal Matching Pursuit	36.8443	2497.4200	49.9512	0.5083	0.4290	0.3869	0.2380
lasso	Lasso Regression	37.5344	2563.1525	50.6071	0.4954	0.4301	0.3998	0.2840
en	Elastic Net	41.0248	2878.7650	53.6444	0.4332	0.4714	0.4715	0.0910
huber	Huber Regressor	39.2292	2924.5352	54.0645	0.4242	0.4641	0.3890	0.6670
ada	AdaBoost Regressor	51.5616	3523.7673	59.3484	0.3061	0.5769	0.6938	0.3750
dt	Decision Tree Regressor	42.7730	4046.5708	63.5813	0.2023	0.4991	0.4187	0.0520
knn	K Neighbors Regressor	51.7494	4584.2339	67.6944	0.0975	0.5855	0.5938	0.0800
llar	Lasso Least Angle Regression	57.0030	5082.7976	71.2876	-0.0004	0.6439	0.7275	0.2960
dummy	Dummy Regressor	57.0030	5082.7975	71.2876	-0.0004	0.6439	0.7275	0.0220
par	Passive Aggressive Regressor	61.4767	6126.6620	76.5601	-0.1999	0.7654	0.7576	0.0400

Creating the Price Model

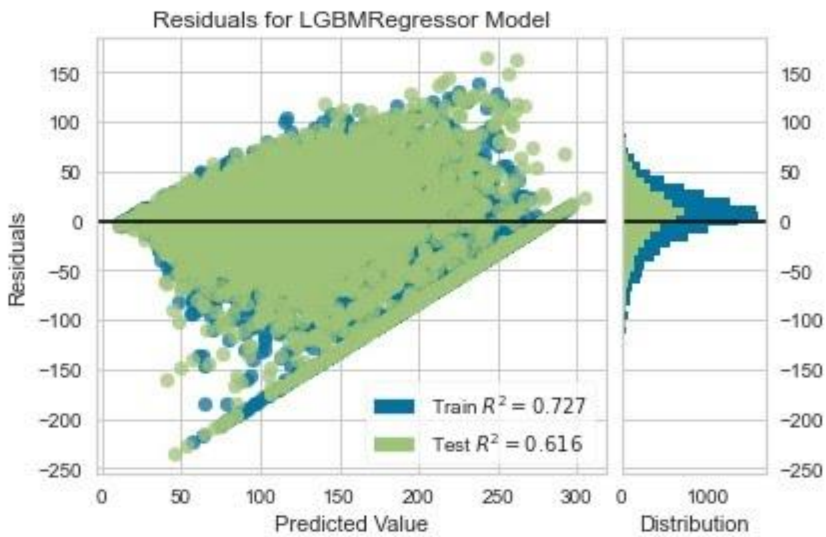
The next step was to create the price model by passing the abbreviation for our chosen model ‘lightgbm’ into the create_model function. We chose to do 10 folds for validation during creation. This means that the data was split into 10 random subsets and 10 iterations were completed with each subset being chosen as the test set once, with the unchosen remainder being the train set. The scores were then viewed, and if there were significant differences between iterations there is likely an issue with the data (outliers, improper scaling, etc).

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	31.8590	1985.0409	44.5538	0.6140	0.3517	0.3073
1	30.6679	1822.9177	42.6956	0.6439	0.3446	0.3026
2	29.3134	1688.4623	41.0909	0.6750	0.3398	0.2954
3	31.9212	2003.7859	44.7637	0.6146	0.3567	0.3101
4	31.3446	1950.6029	44.1656	0.6235	0.3548	0.3058
5	32.6895	2121.5910	46.0607	0.5937	0.3751	0.3287
6	32.5593	2088.8907	45.7044	0.5822	0.3631	0.3232
7	31.7084	1948.3915	44.1406	0.6022	0.3690	0.3219
8	31.0981	1889.2625	43.4656	0.6071	0.3625	0.3229
9	30.8985	1800.5397	42.4328	0.6429	0.3502	0.3108
Mean	31.4060	1929.9485	43.9074	0.6199	0.3567	0.3129
SD	0.9391	126.2602	1.4460	0.0261	0.0103	0.0102

As seen from the above output, most of the scored values did not vary by a large degree. As a result, we accepted the LGBM as an accurate model for predicting rental price per day.

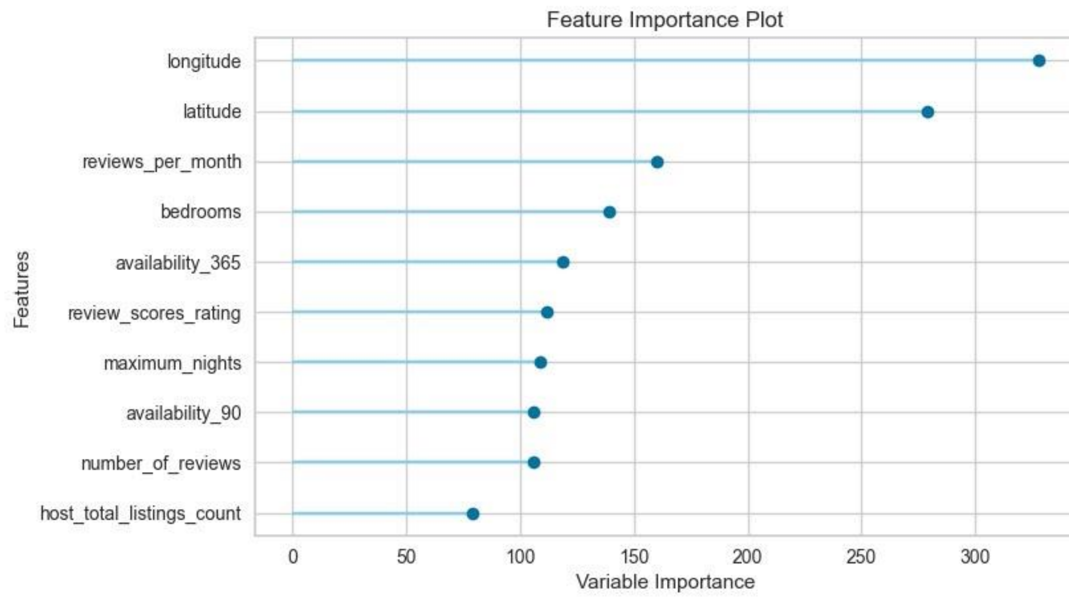
Plotting the Price Model

The plot_model function generated a plot which shows a comparison for the residuals and relative distribution for the test and training sets. As can be seen from the figure below, for this comparison the overlap on residuals was consistent and the value distributions were similar (i.e., lining up) showing that there is no significant overfitting issue in the training set. The test set had a lower R-squared but not to an extreme. This was likely due to differences between the data contained in the test and train sets.



Evaluating the Price Model

We implemented the evaluate_model function to generate explanatory plots to assist in explaining the model. From the generated figure below, we could see which features are most important in predicting price. The feature importance plot indicated that the physical location (longitude and latitude) are the biggest contributing factor to determining price. This was most likely due to how close physical location is to major attractions. Amount of reviews and review rating are also important, indicating that the length and score of reviews for a rental directly impacts how much can be charged. Finally, the total amount of listings per client and the maximum nights stay are likely due to closeness of corporate entities such as hotels.



Finalizing and Saving the Model

The model was exported for deployment, finalize_model fit the estimator onto the dataset. The output below shows the arguments for the model and allows the user to tune the model (more branches, give weights, max depth, etc.). The model was then exported for deployment, finalize_model would fit the estimator onto the dataset.

```
#Finalizing the model for deployment
finalize_model(price_model)
```

```
LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
              importance_type='split', learning_rate=0.1, max_depth=-1,
              min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
              n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
              random_state=123, reg_alpha=0.0, reg_lambda=0.0, silent=True,
              subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
```

Review Score GBR Model

Setting up the Review Score Model

Similar to the setup of the price model, we are instead chose review score ratings as the target variable while keeping the test train split at 70/30.

```
In [17]: #Continuing, we will do the same for review scores.
         s = setup(data, target = 'review_scores_rating',
                   session_id = 123, train_size = 0.7)
```

	Description	Value
0	session_id	123
1	Target	review_scores_rating
2	Original Data	(19343, 40)
3	Missing Values	False

For review score, we have already decided to select R-squared as our metric of determination. We implemented a similar pipeline as before, replacing ‘R-squared’ as the sort. As seen from the figure below we saw that Gradient Boosting Regressor (GBR) had the highest R-squared as well as most of the other metrics. LightGBM was extremely close and if processing time was a factor would be chosen. However, as we were focused on maximizing accuracy, we selected GBR. GBR is another decision tree model with a stronger emphasis on reducing bias oversimplification. While it isn’t a factor in this case, it also helps improve models by reducing the effect that overfitting would have.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	1.4718	5.5544	2.3551	0.6627	0.0248	0.0156	0.4440
lightgbm	Light Gradient Boosting Machine	1.4543	5.6000	2.3648	0.6599	0.0249	0.0154	0.0530
br	Bayesian Ridge	1.6357	5.8661	2.4207	0.6437	0.0255	0.0174	0.1230
ridge	Ridge Regression	1.6354	5.8700	2.4215	0.6435	0.0256	0.0174	0.0170
lr	Linear Regression	1.6356	5.8706	2.4216	0.6434	0.0256	0.0174	0.0500
omp	Orthogonal Matching Pursuit	1.6158	5.8721	2.4220	0.6433	0.0256	0.0172	0.2360
rf	Random Forest Regressor	1.4686	5.8923	2.4258	0.6420	0.0256	0.0156	1.2090
et	Extra Trees Regressor	1.5153	6.6061	2.5680	0.5986	0.0271	0.0161	1.3480
ada	AdaBoost Regressor	2.0287	7.7024	2.7744	0.5319	0.0291	0.0214	0.2250
en	Elastic Net	2.5643	10.4956	3.2389	0.3625	0.0342	0.0272	0.0240
lasso	Lasso Regression	2.6037	10.7556	3.2789	0.3467	0.0346	0.0277	0.0220
dt	Decision Tree Regressor	1.8296	11.3538	3.3676	0.3100	0.0355	0.0194	0.0490
huber	Huber Regressor	2.6800	11.7639	3.4292	0.2853	0.0362	0.0285	0.7370
llar	Lasso Least Angle Regression	3.2357	16.4627	4.0569	-0.0002	0.0428	0.0345	0.2700
dummy	Dummy Regressor	3.2357	16.4627	4.0569	-0.0002	0.0428	0.0345	0.0180
knn	K Neighbors Regressor	3.2473	17.5998	4.1945	-0.0693	0.0442	0.0346	0.0730
par	Passive Aggressive Regressor	3.5355	20.9040	4.3938	-0.2633	0.0459	0.0372	0.0580

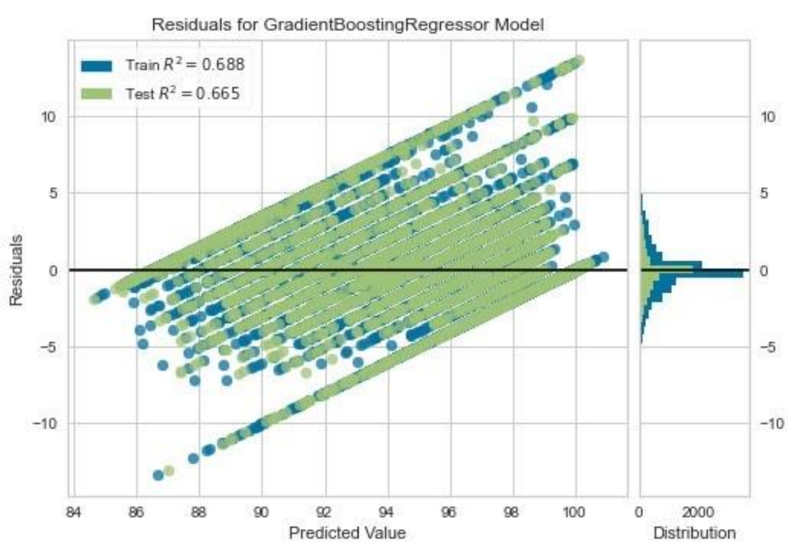
Creating the Review Score Model

As before, we created the model using the gbr argument with 10 folds for validation. The output is displayed in the figure below.

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	1.4320	5.0629	2.2501	0.6914	0.0237	0.0152
1	1.4260	5.1476	2.2688	0.6817	0.0239	0.0151
2	1.4616	5.6728	2.3818	0.6729	0.0251	0.0155
3	1.5416	6.0749	2.4647	0.6238	0.0260	0.0164
4	1.5368	6.1902	2.4880	0.6323	0.0263	0.0164
5	1.4053	5.0455	2.2462	0.6838	0.0236	0.0149
6	1.4861	5.6047	2.3674	0.6567	0.0250	0.0158
7	1.5466	5.8707	2.4229	0.6534	0.0256	0.0165
8	1.4489	5.7951	2.4073	0.6569	0.0254	0.0154
9	1.4334	5.0800	2.2539	0.6739	0.0238	0.0152
Mean	1.4718	5.5544	2.3551	0.6627	0.0248	0.0156
SD	0.0501	0.4172	0.0886	0.0211	0.0010	0.0006

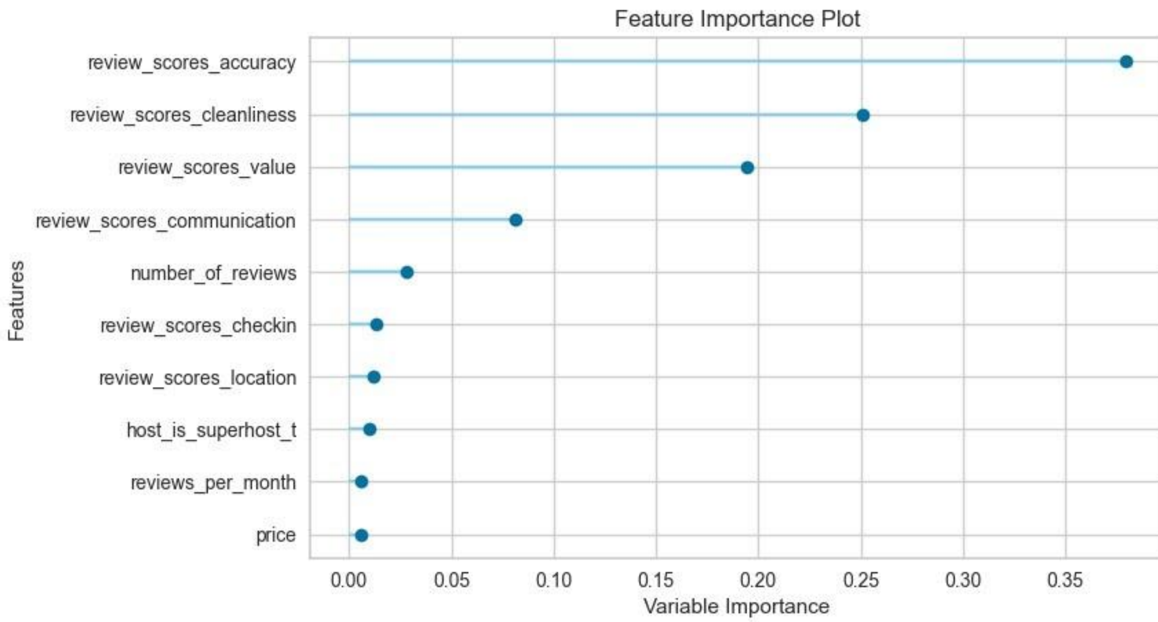
Plotting the Review Score Model

The plot for score showed that the test and train sets are quite close in their r2 and value distribution. The model seemed acceptable for use.



Evaluating the Review Score Model

Reviewing the important features it seemed that the accuracy of the rental listing was the most important feature for determining score in this model. Other important features were perceived value, cleanliness, and communication. It may be prudent to recommend to new renters that these factors play an important role in maximizing their review scores.



Finalizing and Saving the Model

The model was exported for deployment, finalize_model fit the estimator onto the dataset.

Conclusion and Recommendation

After implementing standard modeling techniques (Linear Regression and Random Forest), the team used the pycaret Machine learning package to find which model would perform best with our data. Based on evaluation of metrics, we selected models based on LightGBM and GBR to support our business objectives. Feature importance plots from LightGBM and GBR were useful to infer the best two predictors for price and review score ratings. The best two predictors of price per day of listing is precise location (i.e., latitude and longitude) and reviews per month.

To maximize profit, Airbnb company and Host Airbnb should consider increasing the price of listings in Old City Toronto and listings in districts close to Old City Toronto, conversely, they should decrease the price in districts far away from Old City Toronto. Airbnb hosts should consider requesting more reviews per month from their clients as increases the price per day. In contrast, the best two predictors of review scores rating was review scores accuracy and review scores cleanliness. To maximize profit related to review score ratings, Airbnb company should consider enforcing a cleanliness of listing bench score to ensure all listings score high on a score of cleanliness. Accuracy of review scores should also be ensured. A recommendation for producing a more accurate and reliable Machine learning model that incorporates prediction of both price and review score ratings would be to find Airbnb Toronto datasets that include additional relevant features such as number of parking lots, postal code, distance from subway and bus, aesthetic value, relative location to popular attractions and hotels, venues and crime levels that could be reliable and valid predictors.

References

[1] https://mlcorner.com/linear-regression-vs-decision-trees/

[2] *PyCaret Regression Library Documentation*. PyCaret. (2020, July 31). Retrieved November 9, 2021, from https://pycaret.org/regression1/

[3]https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbm-ensemble/

Appendix

Figures

Table 1. Fields or features in the AirBnb Data Set with description and type of feature stated

Column Name	**Description**	**Feature type(e.g., Numeric, String)**
host_since	Date an individual became a host	Numeric
Former_city	District in Toronto	Categorical
host_response_time	Time to respond to customer booking inquiry; ranges from 1hour to few days	Numeric
host_response_rate	Ranges from 0% to 100% for reply to booking inquiries	Numeric
host_acceptance_rate	Ranges from 0% to 100% response for acceptance of booking	Numeric
host_is_superhost	Is either t(true) or f(false)	Boolean
host_total_listings_count	Total number of listings made by host	Numeric
latitude	The angular distance of a location or object north or south of the Earth's celestial equator	String
longitude	The angular distance of a location or object east or west of the meridian	String
property_type	Type of property (e.g., Entire house)	String
room_type	Specific type of room (e.g., Entire home/apt)	String
accommodates	How many people can stay (e.g., 6)	Numeric
bedrooms	Number of bedrooms in property	Numeric
beds	Number of beds in property	Numeric
price	Price of stay per night	Numeric
minimum_nights	Minimum nights can be booked by same individual	Numeric

maximum_nights	Maximum nights can be booked by same individual	Numeric
availability_30	Availability of Property in 30 days	Numeric
availability_60	Availability of Property in 60 days	Numeric
availability_90	Availability of Property in 90 days	Numeric
availability_365	Availability of Property in 365 days	Numeric
number_of_reviews	Total number of reviews that a listing has received from customers	Numeric
number_of_reviews_ltm	The number of reviews that a listing has received last twelve month	Numeric
number_of_reviews_l30d	The number of reviews that a listing has received per 130 days	Numeric
review_scores_rating	***Customer-provided score rating (0% to 100%); A customer-provided review score attributed to a listing based on overall experience and satisfaction***	Numeric
review_scores_accuracy	Accuracy of review scores (0 to 10)	Numeric
review_scores_cleanliness	Cleanliness score (0 to 10)	Numeric
review_scores_checkin	Over-all check in score (0 to 10)	Numeric
review_scores_communication	Score on communication with host (0 to 10)	Numeric
review_scores_location	***Score on location based on factors such as nearby transportation, noise level (0 to 10)***	Numeric
review_scores_value	Over- all value/quality/experience (0 to 10)	Numeric
instant_bookable	True or False if customer can instantly book	BOolean
calculated_host_listings_count	Calculated number of listings by host	Numeric
calculated_host_listings_count_entire_homes	Number of host listings which are entire homes	Numeric
calculated_host_listings_count_private_rooms	Number of host listings which are private rooms	Numeric
calculated_host_listings_count_shared_rooms	Number of host listings which are shared homes	Numeric

reviews_per_month	Number of Customer reviews of the host accommodation or accommodations per month	Numeric

EDA

Figure 1.

```
#Check data is intact
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19343 entries, 0 to 19342
Data columns (total 40 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   longitude                                     19343 non-null  float64
 1   latitude                                      19343 non-null  float64
 2   bedrooms                                      19343 non-null  float64
 3   beds                                          19343 non-null  float64
 4   host_total_listings_count                     19343 non-null  float64
 5   review_scores_rating                          19343 non-null  float64
 6   review_scores_accuracy                        19343 non-null  float64
 7   review_scores_cleanliness                     19343 non-null  float64
 8   review_scores_checkin                         19343 non-null  float64
 9   review_scores_communication                   19343 non-null  float64
 10  review_scores_location                        19343 non-null  float64
 11  review_scores_value                           19343 non-null  float64
 12  reviews_per_month                             19343 non-null  float64
 13  accommodates                                  19343 non-null  int64
 14  price                                         19343 non-null  float64
 15  minimum_nights                                19343 non-null  int64
 16  maximum_nights                                19343 non-null  int64
 17  availability_30                               19343 non-null  int64
 18  availability_60                               19343 non-null  int64
 19  availability_90                               19343 non-null  int64
 20  availability_365                              19343 non-null  int64
 21  number_of_reviews                             19343 non-null  int64
 22  number_of_reviews_ltm                         19343 non-null  int64
 23  number_of_reviews_l30d                        19343 non-null  int64
 24  instant_bookable                              19343 non-null  object
 25  calculated_host_listings_count                19343 non-null  int64
 26  calculated_host_listings_count_entire_homes   19343 non-null  int64
 27  calculated_host_listings_count_private_rooms  19343 non-null  int64
 28  calculated_host_listings_count_shared_rooms   19343 non-null  int64
 29  former_city                                   19343 non-null  object
 30  room_type                                     19343 non-null  object
 31  host_since                                    19343 non-null  object
 32  host_response_time                            19343 non-null  object
 33  host_is_superhost                             19343 non-null  object
 34  host_acceptance_rate                          19343 non-null  int64
 35  host_response_rate                            19343 non-null  int64
 36  labels_host_acceptance_rate                   19343 non-null  object
 37  labels_host_response_rate                     19343 non-null  object
 38  host_since_year                               19343 non-null  int64
 39  host_length                                   19343 non-null  int64
dtypes: float64(14), int64(18), object(8)
memory usage: 5.9+ MB
```

Figure 2.

```
price                                           1.000000
accommodates                                    0.595983
bedrooms                                        0.540802
beds                                            0.493700
calculated_host_listings_count_private_rooms    0.412672
calculated_host_listings_count_entire_homes     0.339368
latitude                                        0.275433
review_scores_location                          0.147562
availability_30                                 0.092657
host_since_year                                 0.078497
host_length                                     0.078497
review_scores_rating                            0.076440
review_scores_cleanliness                       0.072589
availability_60                                 0.068199
maximum_nights                                  0.059920
availability_90                                 0.055047
calculated_host_listings_count                  0.035810
availability_365                                0.033946
review_scores_checkin                           0.033755
reviews_per_month                               0.032303
review_scores_accuracy                          0.028995
host_acceptance_rate                            0.024670
minimum_nights                                  0.020220
number_of_reviews                               0.013625
review_scores_communication                     0.009418
review_scores_value                             0.008806
longitude                                       0.005433
number_of_reviews_ltm                           0.001903
host_total_listings_count                       0.000012
number_of_reviews_l30d                               NaN
calculated_host_listings_count_shared_rooms          NaN
host_response_rate                                   NaN
Name: price, dtype: float64
```

Figure 3.

```
review_scores_rating                            1.000000
review_scores_cleanliness                       0.650398
review_scores_accuracy                          0.645803
review_scores_value                             0.643939
review_scores_communication                     0.559467
review_scores_checkin                           0.480229
review_scores_location                          0.380516
host_total_listings_count                       0.179073
calculated_host_listings_count                  0.170597
calculated_host_listings_count_private_rooms    0.103404
price                                           0.076440
calculated_host_listings_count_entire_homes     0.072359
availability_30                                 0.072297
availability_60                                 0.068570
availability_90                                 0.060145
number_of_reviews_ltm                           0.051054
number_of_reviews                               0.050883
host_length                                     0.049167
host_since_year                                 0.049167
latitude                                        0.047717
availability_365                                0.043711
minimum_nights                                  0.043324
maximum_nights                                  0.038906
host_acceptance_rate                            0.019183
bedrooms                                        0.010735
reviews_per_month                               0.010130
longitude                                       0.004461
accommodates                                    0.003989
beds                                            0.000978
number_of_reviews_l30d                               NaN
calculated_host_listings_count_shared_rooms          NaN
host_response_rate                                   NaN
Name: review_scores_rating, dtype: float64
```

MACHINE LEARNING

Figure 4. Description of Metrics

Name of Metric	Definition
Mean Absolute Error	How far away predicted values are from observed values
Mean Squared Error	The quality of a predictor based on the average square difference between the observed and predicted values
Root Mean Squared Error	Value difference between the true and predicted values
R2	Proportion of variance in the dependant variable which is predicted from the independent variable
Root Mean Squared Log Error	Ratio difference between the observed and predicted values
Mean Absolute Percentage Error	Prediction accuracy shown as a percentage value

Figure 5.

70% training set	R-squared	RMSE
Price_Linear Regression	0.53	48
Review score ratings_Linear Regression	0.64	2
Price_Random Forest Regression	0.6	44.24
Review score ratings_Random Forest Regression	0.65	2.43
Price_LGBM	0.61	43.9
Review score ratings_GBM	0.66	2.36