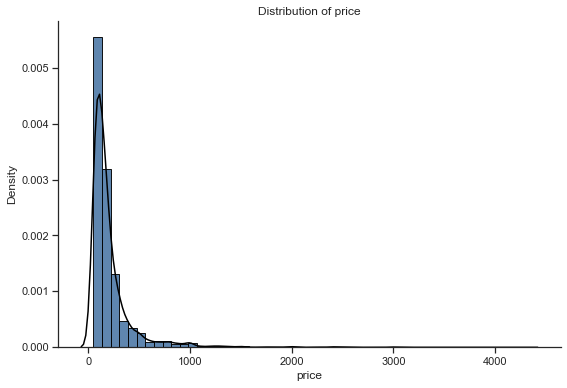
EDA

Response variable: Price

After the train-test split, the train set now contains 10635 observations. For the response variable, the maximum value is $4300.00 while the median price is $144.00 and the mean price is $212.00. It has a skewness of 5.7027. The distribution plots of price and the descriptive statistics suggest that the price is highly right skewed, with a large deviation from normal distribution. The EDA of price indicates that log-transformation may be needed.

|  |  |
| --- | --- |
| Price | |
| Count | 10635.00 |
| Mean | 212.3679 |
| STD | 255.6247 |
| Min | 51.00 |
| 25% | 89.00 |
| 50% | 144.00 |
| 75% | 231.00 |
| Max | 4300.00 |



Numerical variables:

For numerical variables, the correlation heatmap is created to explore the bivariate relationships between variables. The heatmap (see Appendix A) indicates that there are multiple highly correlated features need to be deleted for performance enhancement.

To decide the features that need to be deleted, we conduct further exploration with the help of correlation table and the mutual information with the response variable – price.

1. host\_listings\_count, host\_total\_listings\_count, cal\_host\_listings\_count and cal\_host\_count\_entire \_homes are highly correlated. Their correlations are higher than 0.9. However, the correlation with these four features and price is all below 0.2, indicating a weak linear relationship with price. Thus, only cal\_host\_count\_entire\_homes will be kept in the data set as it has the highest mutual information with price, 0.361.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | host\_listings\_count | host\_total\_listings\_count | cal\_host\_listings\_count | cal\_host\_count\_entire \_homes |
| host\_listings\_count | 1 |  |  |  |
| host\_total\_listings\_count | 1 | 1 |  |  |
| cal\_host\_listings\_count | 0.9371 | 0.9371 | 1 |  |
| cal\_host\_count\_entire \_homes | 0.9315 | 0.9315 | 0.9938 | 1 |
| Price CORR | 0.1641 | 0.1663 | 0.1641 | 0.3610 |
| Price MI | 0.1820 | 0.1820 | 0.1678 | 0.1808 |

1. There are eight features that are related to the minimum and maximum nights decided by each host. Among these eight variables, min\_average\_nights\_ntm and max\_average\_nights\_ntm will be kept for further data analysis based on the correlations with each other, the mutual information, and the correlation with price.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Min | Max | Min\_min | Max\_min | Min\_max | Max\_max | Min\_avg | Max\_avg |
| Min | 1 |  |  |  |  |  |  |  |
| Max | 0.003 | 1 |  |  |  |  |  |  |
| Min\_min | 0.863 | -0.012 | 1 |  |  |  |  |  |
| Max\_min | 0.868 | -0.002 | 0.957 | 1 |  |  |  |  |
| Min\_max | -0.002 | 0.008 | -0.002 | -0.001 | 1 | 1 |  |  |
| Max\_max | -0.002 | 0.008 | -0.002 | -0.001 | 1 | 1 |  |  |
| Min\_avg | 0.880 | -0.003 | 0.978 | 0.992 | -0.002 | -0.002 | 1 |  |
| Max\_avg | -0.002 | 0.008 | -0.002 | -0.001 | 1 | 1 | -0.002 | 1 |
| Price CORR | 0.012 | -0.015 | 0.015 | 0.024 | 0.010 | 0.010 | 0.017 | 0.010 |
| Price MI | 0.0885 | 0.108 | 0.0696 | 0.0971 | 0.0452 | 0.0289 | 0.114 | 0.0337 |

1. Among accommodates, bathrooms, bedrooms and beds, ‘accommodates’ will be deleted as it has a higher-than-0.8 correlation with both bedrooms and beds.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | accommodates | bathrooms | bedrooms | beds |
| accommodates | 1 |  |  |  |
| bathrooms | 0.5487 | 1 |  |  |
| bedrooms | 0.8422 | 0.5981 | 1 |  |
| beds | 0.8713 | 0.5196 | 0.8078 | 1 |
| Price CORR | 0.5738 | 0.4965 | 0.6004 | 0.5194 |
| Price MI | 0.3604 | 0.2265 | 0.3651 | 0.2632 |

1. Features about availability suggest the available days for a host for the next 30/60/90 days and one year. Among these four integer variables, availability\_30 is kept for further data analysis as it has the highest mutual information with price.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| availability | \_30 | \_60 | \_90 | \_365 |
| \_30 | 1 |  |  |  |
| \_60 | 0.9474 | 1 |  |  |
| \_90 | 0.8933 | 0.9774 | 1 |  |
| \_365 | 0.6315 | 0.7042 | 0.7471 | 1 |
| Price CORR | 0.0940 | 0.0894 | 0.0866 | 0.1240 |
| Price MI | 0.0435 | 0.0434 | 0.0401 | 0.0340 |

1. There are seven features related to the review score: rating, accuracy, cleanliness, checkin, communication, location and value. Among these features, review\_score\_rating will be deleted as it is an overall rating created by with other scores based on Airbnb’s algorithm.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | \_rating | \_accuracy | \_cleanliness | \_checkin | communication | Location | Value |
| Rating | 1 |  |  |  |  |  |  |
| Accuracy | 0.7523 | 1 |  |  |  |  |  |
| Clean | 0.7216 | 0.6595 | 1 |  |  |  |  |
| Checkin | 0.5810 | 0.5551 | 0.4570 | 1 |  |  |  |
| Commun | 0.6424 | 0.6186 | 0.4882 | 0.6934 | 1 |  |  |
| Location | 0.4862 | 0.4843 | 0.4123 | 0.4324 | 0.4664 | 1 |  |
| value | 0.7191 | 0.6885 | 0.6564 | 0.5208 | 0.5692 | 0.5155 | 1 |
| Price Corr | 0.0579 | 0.0394 | 0.0491 | 0.0395 | 0.0308 | 0.0555 | -0.0071 |
| Price MI | 0.0343 | 0.0728 | 0.0454 | 0.0851 | 0.1128 | 0.1128 | 0.0583 |

1. Among the features related to the number of reviews, number of reviews for the last twelve months is deleted as it has a higher-than-0.8 correlation with reviews\_per\_month.

|  |  |  |  |
| --- | --- | --- | --- |
|  | number\_of\_reviews | number\_of\_reviews\_ltm | reviews\_per\_month |
| number\_of\_reviews | 1 |  |  |
| number\_of\_reviews\_ltm | 0.7841 | 1 |  |
| reviews\_per\_month | 0.6653 | 0.8580 | 1 |
| Price Corr | -0.0718 | -0.0710 | -0.0664 |
| Price MI | 0.0340 | 0.0388 | 0.0583 |

Categorical variables:

There are 9 categorical variables in the train set. The boxplot (see Appendix B) shows that host\_is\_superhost, host\_identity\_verified, is\_location\_exact and instant\_bookable, have similar median and mean value of prices among different categories. These four variables will be deleted as they are not representative.

|  |  |  |
| --- | --- | --- |
| Median Value of Price ($) | False | True |
| host\_is\_superhost | 142.00 | 149.00 |
| host\_identity\_verified | 140.00 | 149.00 |
| is\_location\_exact | 140.00 | 144.00 |
| instant\_bookable | 150.00 | 138.00 |

|  |  |  |
| --- | --- | --- |
| Mean Value of Price ($) | False | True |
| host\_is\_superhost | 211.8419 | 215.3937 |
| host\_identity\_verified | 211.3579 | 214.1907 |
| is\_location\_exact | 204.9924 | 214.6989 |
| instant\_bookable | 230.5978 | 189.6095 |

Detailed EDA on remaining variables

**Host\_response\_rate & host\_acceptance\_rate:** Since the missing values in these two features are imputed with 0 to suggest a zero response and acceptance rate, these two features are concentrated at 0 and 1, but their relationship with price seems weak (see Appendix C).

**Host\_response\_time:** The count plot (see Appendix D) suggests that although this feature has five different categories, among which, ‘within a few hours’ and ‘within an hour’ have the lowest median value of price while ‘a few days or more’ has the lowest median value of price. And most of the values are concentrated in ‘None’ and ‘within an hour’.

**Neighborhood\_cleansed:** The count plot (see Appendix E) shows that Sydney (2890 observations), Waverley (1469) and Randwick (890) have the highest number of observations as Sydney is the most popular place for travelers in Australia and Waverley & Randwick have famous tourist attractions. Among all neighborhood, Pittwater has the highest median value of price, $350, followed by Manly ($195), Hunters Hill ($194.5), Mosman ($185) and Waverley ($165). These five neighborhoods are located near beaches and other famous tourist attractions. Thus, it is plausible for their price to be higher than other neighborhoods. In contrast, the neighborhoods with lowest prices, such as Campbelltown, are less popular among tourists.

|  |  |  |  |
| --- | --- | --- | --- |
| Price ($) | Highest 5 | Price ($) | Lowest 5 |
| Pittwater | 350.00 | **Campbelltown** | 89 |
| Manly | 195.00 | **City of Kogarah** | 88.5 |
| Hunters Hill | 194.50 | **The Hills Shire** | 85 |
| Mosman | 185.00 | **Blacktown** | 79 |
| Waverley | 165.00 | **Holroyd** | 74 |

**Property\_type** is mostly concentrated in ‘Apartment’ (6492 observations) and ‘House’ (2671 observations) (see Appendix F). Among different property types, boat and castle have the highest median value price, however, they only have 10 and 1 observations respectively.

**Room\_type:** (see Appendix F) there are four different types of room, among which, hotel room has the largest median price ($191) and shared room has the lowest (70$). The values are concentrated in ‘Entire home/apt’ (7202 observations) and ‘Private room’ (3307 observations).

**Cancellation\_policy** has seven categories, among which, two luxury cancellation policies, luxury\_moderate and luxury\_super\_strict\_125, have the highest median price and flexible cancellation policy has the lowest median price ($110) (see Appendix F). This is intuitive as the cancellation policy refers to the time required for free cancellation, the stricter the policy is, the more expensive the price will be to protect the interests of the hosts. The values are mostly concentrated in ‘flexible’ (3312 observations), ‘moderate’ (2586 observations), and ‘strict\_14\_with\_grace\_period’ (4651 observations).

**Bathrooms, Bedrooms & Beds:** The count plots (see Appendix G) show that these features all concentrated at certain values. As for bathrooms, the values are concentrated at 1.0 and 2.0. For bedrooms, the values are concentrated from 1.0 to 3.0. For the number of beds, the values are concentrated from 1.0 to 4.0. Besides that, the boxplots (see Appendix G) suggest that these three features are ordinal, i.e. the prices increase with the number of bathrooms, bedrooms and beds.

**Security\_deposit\_perc, cleaning\_fee\_perc & extra\_people\_perc:** The distribution plots show that these features are highly right skewed, indicating that most hosts keep the deposit, cleaning fee and the fee for extra people within a certain range unless the room or house provided requires certain considerations (see Appendix H). Transformations such as log, box-cox and Yeo-Johnson transformation, may be needed. There’re outliers in the features but we did not delete them because these three variables are decided by the hosts. Thus, all values should be plausible based on different hosts’ decisions.

**Minimum\_nights\_avg\_ntm & maximum\_nights\_avg\_ntm:** The distribution plots and regression plots (see Appendix I) show that these two variables are highly right skewed with a concentration from 0 to 100 for minimum\_nights\_avg\_ntm and 0 for maximum\_nights\_avg\_ntm. According to the regression plots, their relationships with price are weak and it is difficult to transform them into normal distributions. Therefore, these two features will be deleted.

**Guest\_included** ranges from 1 to 15 with a concentration on 1. And the price increases with the number of the guests included, indicating that the number of guest included should be considered as an ordinal variable (see Appendix I).

**Availability\_30** is the availability days for the next month set by the host. The values are concentrated at 0 (see Appendix I).

**Number of reviews** refers to the total number of reviews a host received and reviews\_per\_month refers to the customer review per month. Number\_of\_reviews\_ltm refers to the number of reviews for the last twelve months. Although these two features are highly correlated, we keep them both in the train set as they convey different messages. Transformations are required since they are highly right skewed (see Appendix J).

**Review scores:** there are six different review scores including accuracy, cleanliness, check-in, communication, location, and value. According to the boxplots, these features are nominal variables that play as a good indicator of the level of customer satisfaction. The distribution plots suggest that most customers’ scores are higher than 0.8. The regression plots suggest that the they are positively correlated with the price. (see Appendix K)

**Calculated\_host\_listings\_count\_entire\_homes** is the number of listings a host have for entire homes, while **calculated\_host\_listings\_count\_private\_rooms** is the number of a listings a host have for private rooms. These two features are highly right skewed with a range from 0 to 2 for entire homes and 0-2.5 for private rooms. (see Appendix L)

Feature engineering:

1. For variables concentrated at certain categories or values, we combine the categories or values with fewer observations into new category to avoid unrepresentativeness.
   1. Host Response Time: combine ‘within a few hours’, ‘within a day’, and ‘a few days or more’ into a new category: ‘after an hour’.
   2. Host\_response\_rate & host\_acceptance\_rate: combine values smaller or equal to 0.5 into a new category: ‘0-0.5’, combine other values into another category ‘0.5-1’.
   3. Property type: combine categories except ‘apartment’ and ‘house’ into a new category: ‘other’.
   4. Room type: combine ‘Hotel room’ and ‘Shared room’ into a new category: ‘Other’.
   5. Cancellation policy: combine categories except ‘flexible’, ‘moderate’ and ‘strict\_14\_with\_grace\_period’ into a new category: ‘Other’.
   6. Bathrooms: combine 0.0 and 0.5 to ‘<1’; combine 1.0 and 1.5 to ‘1-2’; combine 2.0 and 2.5 to ‘2-3’; combine 3.0 and values larger than 3 to ‘>=3’.
   7. Bedrooms: Since the values are concentrated from 0 to 3, we combine values larger than 3 to ‘>3’.
   8. Beds: Since the values are concentrated from 0 to 5, we combine values larger than 5 to ‘>5’.
   9. Guest\_included: combine values larger than 4 into ‘>4’.
   10. Calculated\_host\_listings\_count\_entire\_homes & calculated\_host\_listings\_count\_private\_rooms: combine all values larger or equal to 2 into a new category: ‘>=2’.
2. For numerical variables with a skewness larger than 0.75, we applied log transformation since machine learning usually prefer normal distribution.
3. Based on the exploratory data analysis and domain knowledge, we applied label encoder for ‘host\_response\_time’, 'bathrooms','bedrooms','beds','guests\_included', 'calculated\_host\_listings\_count\_entire\_homes','calculated\_host\_listings\_count\_private\_rooms' to encode these variables into ordinal variables.
4. Create dummy variables for categorical variables.

Methodology

Lasso

Lasso refers to the Least Absolute Shrinkage and Selection Operator, which performs regularization to enhance model performance with two main properties: variable selection and shrinkage. By shrinkage and variable selection, Lasso can successfully reduce the variance and minimize the bias.

The most important part in the formula is the hyperparameter . The value of the hyperparameter is essential in enhancing the prediction accuracy and interpretability as it controls the strength of variable selection and shrinkage (Wikipedia, n.d.).

* When the hyperparameter equals to 0, the Lasso will be simply reduced to an OLS problem.
* When , all of the coefficients in the model will be shrined to zero.
* Increasing will increase the bias and decrease the variance, leading to an overfitting problem, and vice versa.

Compared with Ridge, which performs a regularization, Lasso can shrink the coefficient of variables to zero, leading to the second property of Lasso – variable selection. Therefore, an important advantage of Lasso over Ridge regression is its interpretability.

Compared with subset selection, which chooses the best subset of variables based on certain criterion, Lasso is much more computationally efficient. The subset selection method is much more time-consuming because of the combinatorial explosion of the parameters. It also leads to a nonconvex problem that is trickier to fit.

The main disadvantage of the Lasso is it does not work well when there are highly correlated variables in the dataset. We avoided this problem by manually deleted highly correlated variables after the exploratory data analysis part.

Since Lasso puts constraints on the size of the coefficients based on the magnitude of each variable, we first standardized the predictors by Robust Scaler to let the predictors contribute equally to the data analysis before fitting the model. The reason for standardization before Lasso is that Lasso

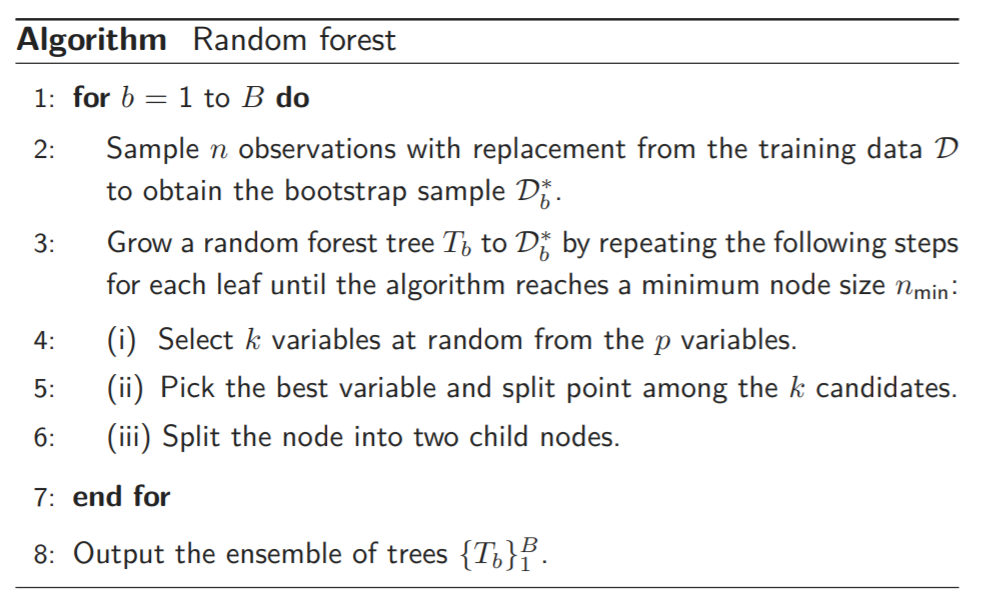
Commonly, the value of is chosen by cross-validation. The hyperparameter for Lasso in this project is set to be …

The result for this model is …

### Random Forest model

Random forest is based on the idea of bagging, which means “to average many noisy but approximately unbiased models, therefore reducing variance” (Lecture 9). In bagging, we first generate bootstrap samples by randomly sampling from the training data with replacement, after which a decision tree will be fitted to each sample and the prediction will be the average prediction across all trees (Lecture 9). In this way, the variance will be largely reduced with an improvement on the model accuracy (James et al. 2013).

Compared with bagging, random forest furtherly improve the model performance by only selecting a subset of the features for each decision node.



The hyperparameter in the random forest model are the number of selected variables *k* and the minimum node size *nmin*. By default, the number of selected variables is equal to for classification problems, and for regression problems. And the minimum node size is set to be 1 for classification problems, and 5 for regression problems. However, we will select these hyperparameters by cross-validation in practice.

The main advantage of random forest is its ability to decorrelation. To achieve decorrelation, random forest model avoids the domination of subsets of strong features that will lead to highly correlated decision trees and gives a chance to other predictors, largely reducing the variance (James et al. 2013). Also, random forest does not require any standardization before fitting and it suits well for both categorical and numerical variables. But it still has some drawbacks in computational efficiency and interpretability.

In this data analysis project, we choose the hyperparameter …

The result for this model is …

XGBoost

We use XGBoost in this project as there are more than 10000 observations with more than 100 features after feature engineering and the dataset contains both categorical and numeric variables. The execution speed and the performance of XGBoost will largely enhance our result.

XGBoost refers to extreme gradient boosting with three main features: gradient boosting, stochastic gradient boosting and regularized gradient boosting. The main advantages of XGBoost compared to other implementations of gradient boosting is its computational efficiency and better model performance on structured dataset (Jason, 2016).

In boosting, we use an additive model with basis functions that are essentially weak learners to add up to a strong learner. By increasing the number of basis functions, the capacity of the model will also increase but will basically adapt to the size of the training data. However, in XGBoost, it uses a more regularized form of gradient boosting (Neetika, 2020). It solves the regularized risk minimization problem with a penalty factor on complexity .

We specified the penalty term by:

The penalty term in XGBoost can be considered as a regularization and the hyperparameters in is and . In the algorithm, there are multiple important hyperparameters that need to be considered such as the learning rate, the penalty, the maximum depth, the number of trees etc.

In this project, …

Reference

James, G, Witten, D, Hastie, T, Tibshirani, R, Casella, G, Fienberg, S, & Olkin, I 2013, *An Introduction to Statistical Learning: with Applications in R*, vol. 103, Springer New York, New York, NY, doi: 10.1007/978-1-4614-7138-7.

Wikipedia n.d., *Lasso (Statistics),* viewed 22 May 2021, https://en.wikipedia.org/wiki/Lasso\_(statistics)

Jason, B. 2016. A gentle introduction to XGBoost for Applied Machine Learning, viewed 22 May 2021, <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>

Neetika, K. 2020. A brief introduction to XGBoost. Viewd 22 May 2021, https://towardsdatascience.com/a-brief-introduction-to-xgboost-3eaee2e3e5d6