AirBnB Price Prediction

Aniket K Singh

May 4, 2021

1. Introduction

Airbnb has become one of the essential elements of trips and vacation plans for over 150 million people. Since 2008, guests and hosts have used Airbnb for a unique and personalized experience of traveling with a wide range of travel possibilities. Before Airbnb, most consumers had to rely on hotels. As hotels aren't as widely available as Airbnb with their exceptional business model, they soon became the best vacation rental marketplace.
Because of the dramatic growth of Airbnb, price prediction becomes one of the essential elements for their platform. As hosts typically determine the price of the Airbnb. Both the host and Airbnb need to provide a fair price to the consumer, as it is an essential element of their model. Determining the price is crucial for the new and existing host/Airbnb because the price cannot be too high that they lose popularity or not get any guest. As for customers, they have options to check and compare prices depending on their needs.

2. Data Preparation

The dataset used in the project has been accessed from Kaggle's database. It is a public dataset of Airbnb that is accessible publicly on their original dataset website. This dataset describes the listing activity and metrics in NYC, NY for the year 2019. This data file includes all the needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions. The dataset is in Comma-Separated Values(.csv) file format. After importing the datset, we performed some analysis on the dataset. The dataset contained 48894 rows and 16 columns.

2.1. Handling Missing Values

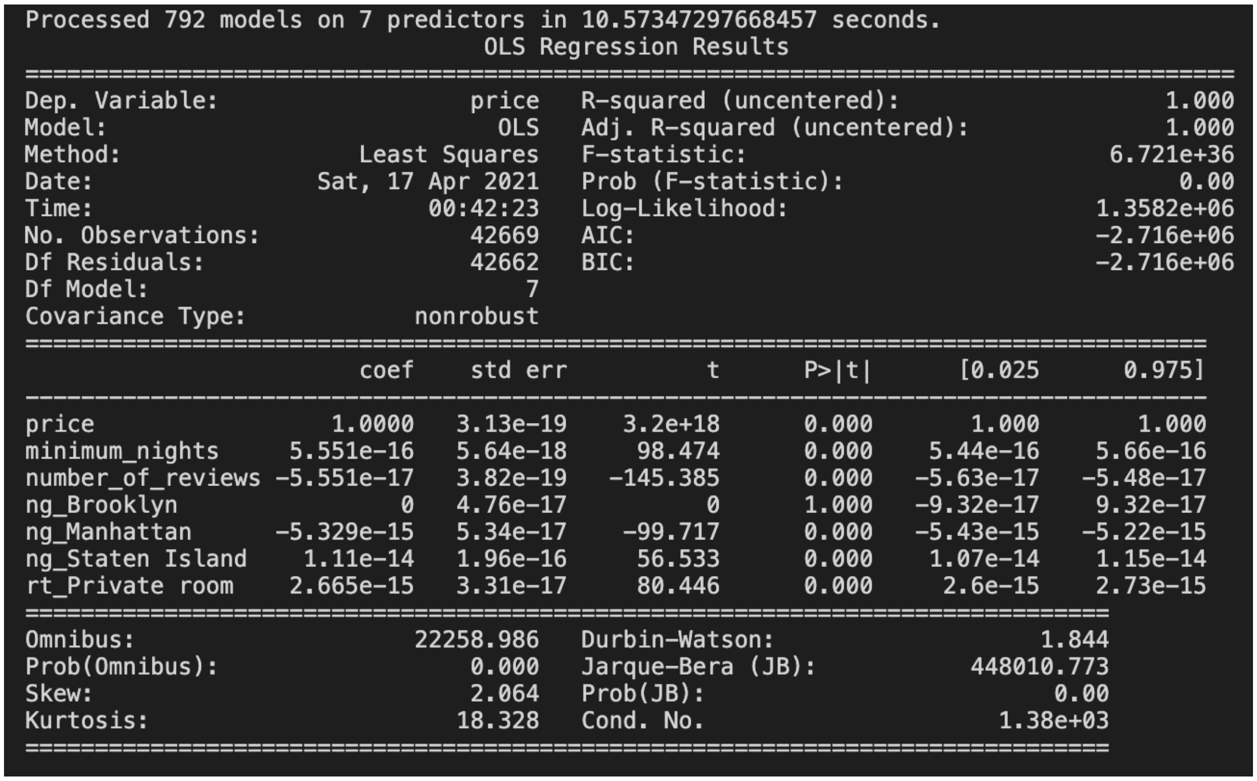
After importaing the dataset. we noticed that column "last_review", "reviews_per_month" had 10052 missing values. Also, few columns had some missing values that were either imputed or dropped. This dataset did not contain many missing values or the rows that contained majority of missing values wasn't used to train or build the model.

2.2. Dealing with categorical inputs:

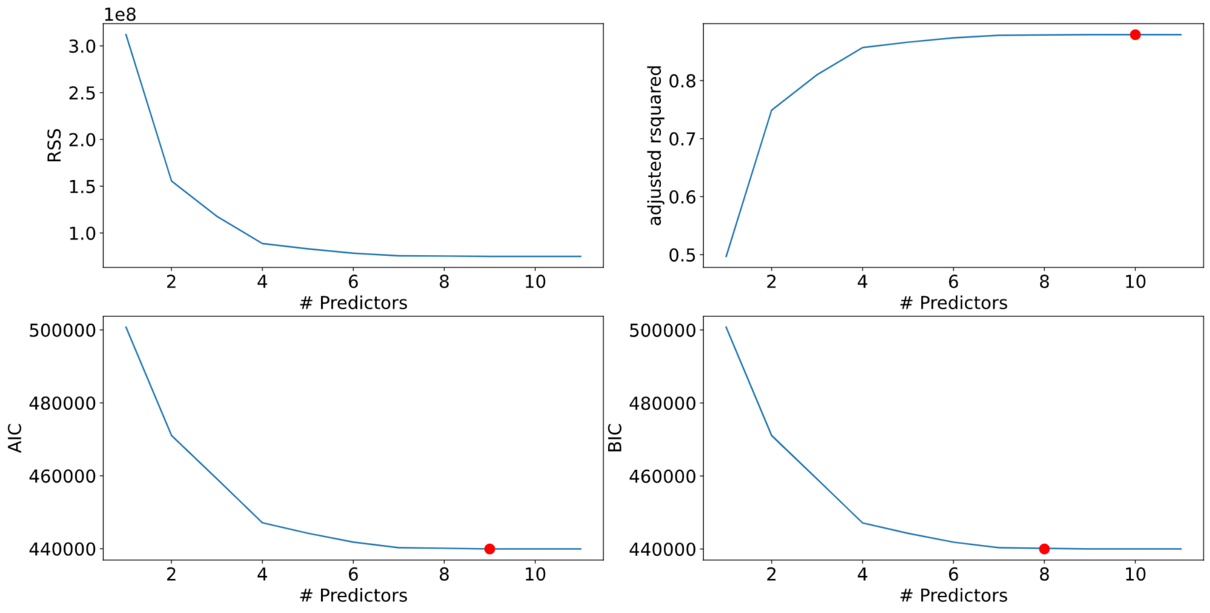
Too many levels of a categorical variable are one of the most frequently occurring problems in predictive modeling. As for our dataset, multiple level of categorical variables. Three columns categories of the categorical variables are neighborhood_group, neighborhood and room_type are. In case of neighborhood_group and room_type, they do not have multiple levels so it is possible to use dummy coding. But in case of neighborhood, we have 219 levels which makes dummy coding overly complex. The number of dummy variables for the neighborhood variable would be one less than the number of levels in neighborhood. It would lead to the problem of high dimensionality which will ultimately cause the over-fitting of the data. Thus, the model will not be able to generalize properly to a new dataset. Moreover, inclusion of categorical variables with too many inputs can lead to the problem of quasi-complete separation. The problem of quasi-complete separation occurs when a level of the categorical variable has a target response of either 0% or 100% and can cause the interpretation of the regression model. However, we did not include few variables such as neighborhood in our final model.

2.3. Variable screening:

In predictive modeling, carefully selected features can improve the accuary of the model while adding too many or too less variables can lead to overfitting or underfitting. The model can't generalize well and work on unseen dataset if the model is not "just right" or close to "just right". There are multiple methods for variable screening. We can perform many statistical tests such as AIC, BIC, F-test, Cross Validation to check what set of variables perform well on the model. There are many methods available such as Stepwise selection which uses either Forward or Backward selection method. But, we have a very small number of features, so we performed Best Subset selection method. Best subset selection method can be computationally expensive but in our case, it is possible to perform best subset.



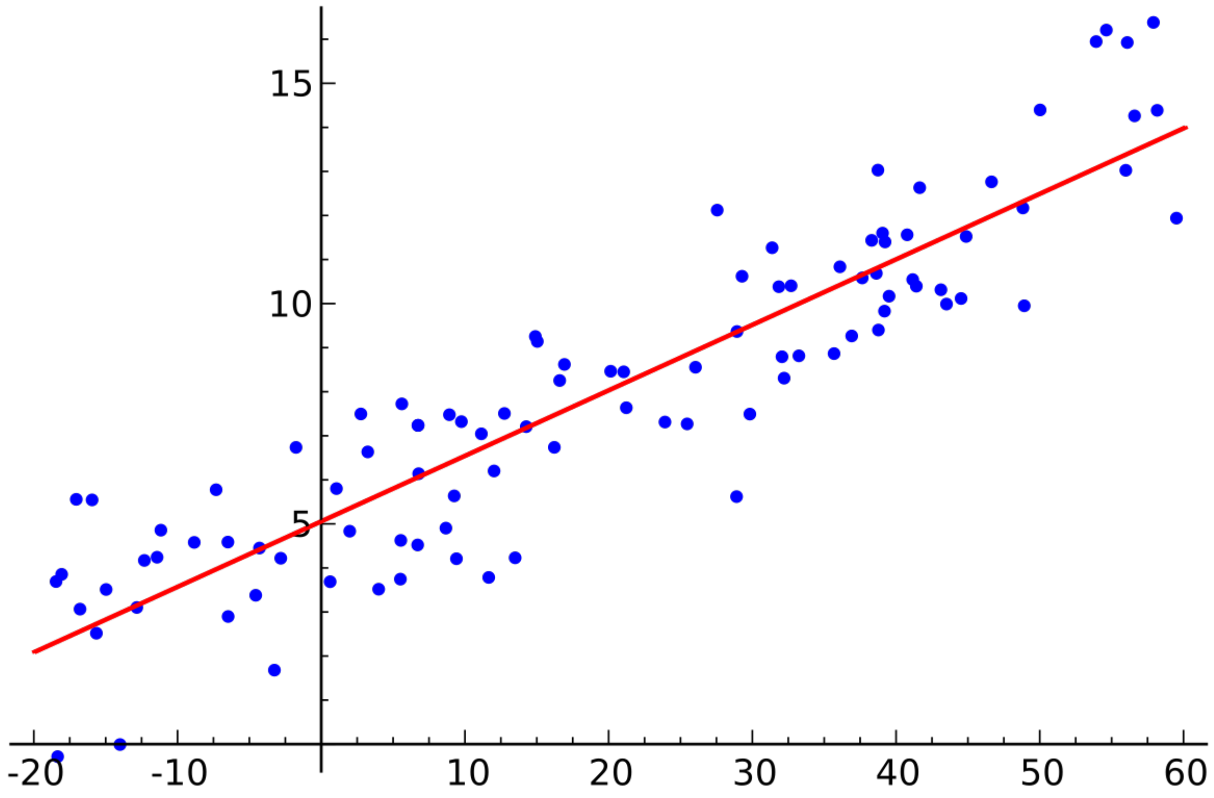
```
Processed 792 models on 7 predictors in 10.57347297668457 seconds.
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                  price   R-squared (uncentered):                   1.000
Model:                            OLS   Adj. R-squared (uncentered):              1.000
Method:                 Least Squares   F-statistic:                          6.721e+36
Date:                Sat, 17 Apr 2021   Prob (F-statistic):                        0.00
Time:                        00:42:23   Log-Likelihood:                      1.3582e+06
No. Observations:               42669   AIC:                                 -2.716e+06
Df Residuals:                   42662   BIC:                                 -2.716e+06
Df Model:                           7
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------
price                 1.0000   3.13e-19    3.2e+18      0.000       1.000       1.000
minimum_nights     5.551e-16   5.64e-18     98.474      0.000    5.44e-16    5.66e-16
number_of_reviews -5.551e-17   3.82e-19   -145.385      0.000   -5.63e-17   -5.48e-17
ng_Brooklyn                0   4.76e-17          0      1.000   -9.32e-17    9.32e-17
ng_Manhattan      -5.329e-15   5.34e-17    -99.717      0.000   -5.43e-15   -5.22e-15
ng_Staten Island    1.11e-14   1.96e-16     56.533      0.000    1.07e-14    1.15e-14
rt_Private room    2.665e-15   3.31e-17     80.446      0.000     2.6e-15    2.73e-15
==============================================================================
Omnibus:                    22258.986   Durbin-Watson:                   1.844
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           448010.773
Skew:                           2.064   Prob(JB):                         0.00
Kurtosis:                      18.328   Cond. No.                     1.38e+03
==============================================================================
```



3. Models Used:

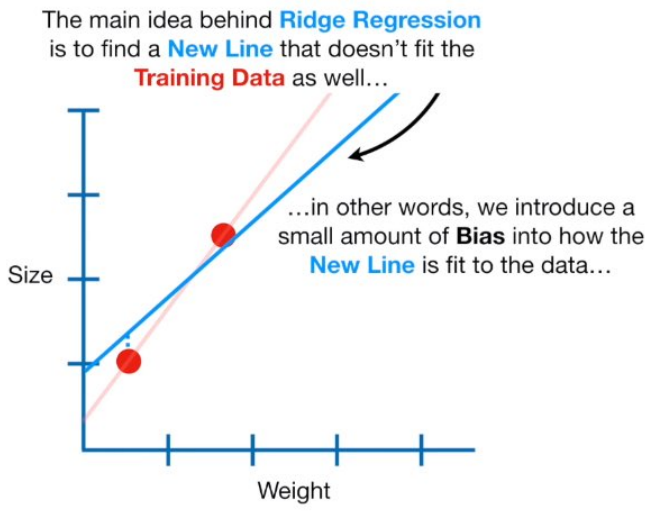
Linear Regression

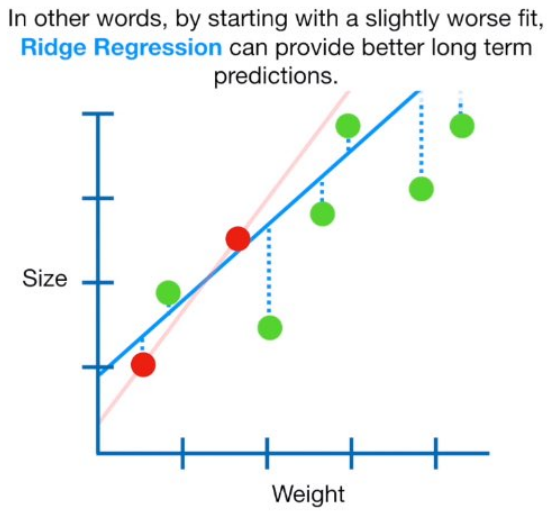
Linear regression is a technique used to model the relationships between observed variables. The idea behind simple linear regression is to "fit" the observations of two variables into a linear relationship between them. Graphically, the task is to draw the line that is "best-fitting" or "closest" to the points (x_i, y_i), where x_i and y_i are observations of the two variables which are expected to depend linearly on each other.



Ridge Regression

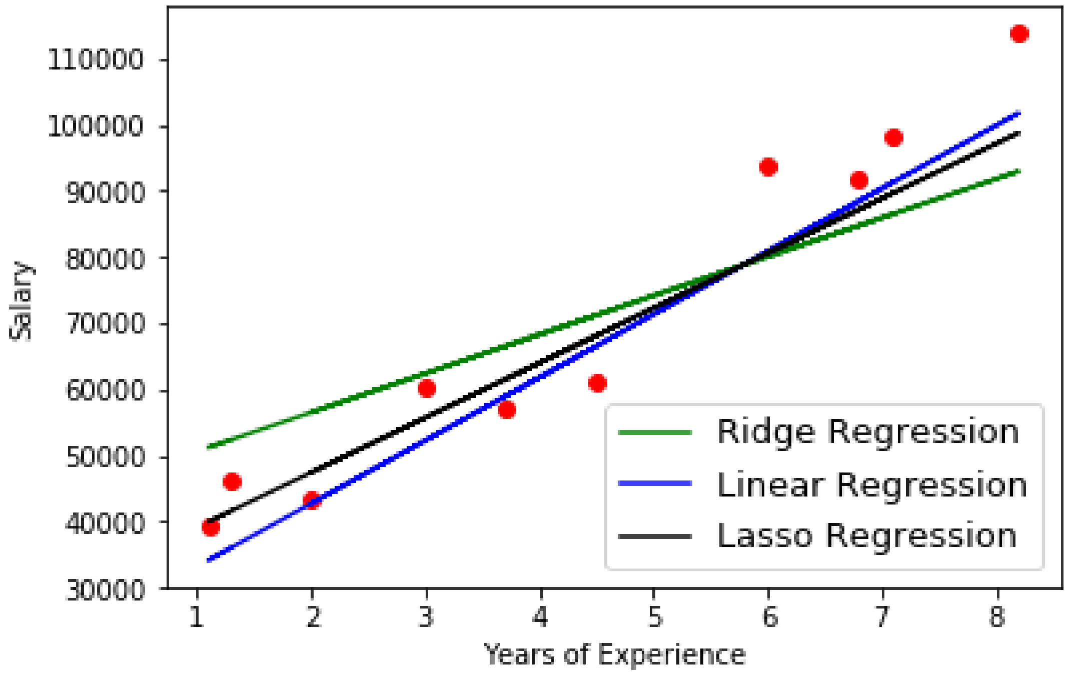
Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.


The main idea behind Ridge Regression is to find a New Line that doesn't fit the Training Data as well…
…in other words, we introduce a small amount of Bias into how the New Line is fit to the data…
Size
Weight


In other words, by starting with a slightly worse fit, Ridge Regression can provide better long term predictions.
Size
Weight

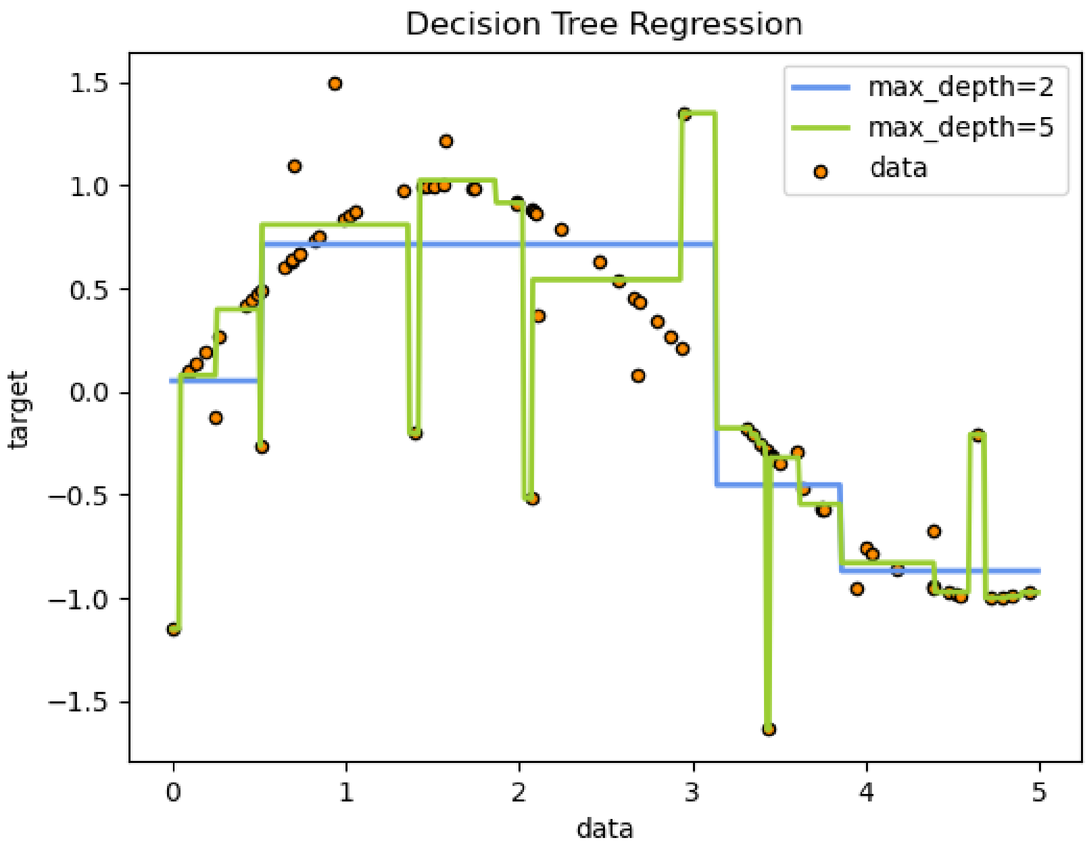
Lasso Regression

The "LASSO" stands for Least Absolute Shrinkage and Selection Operator. Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.



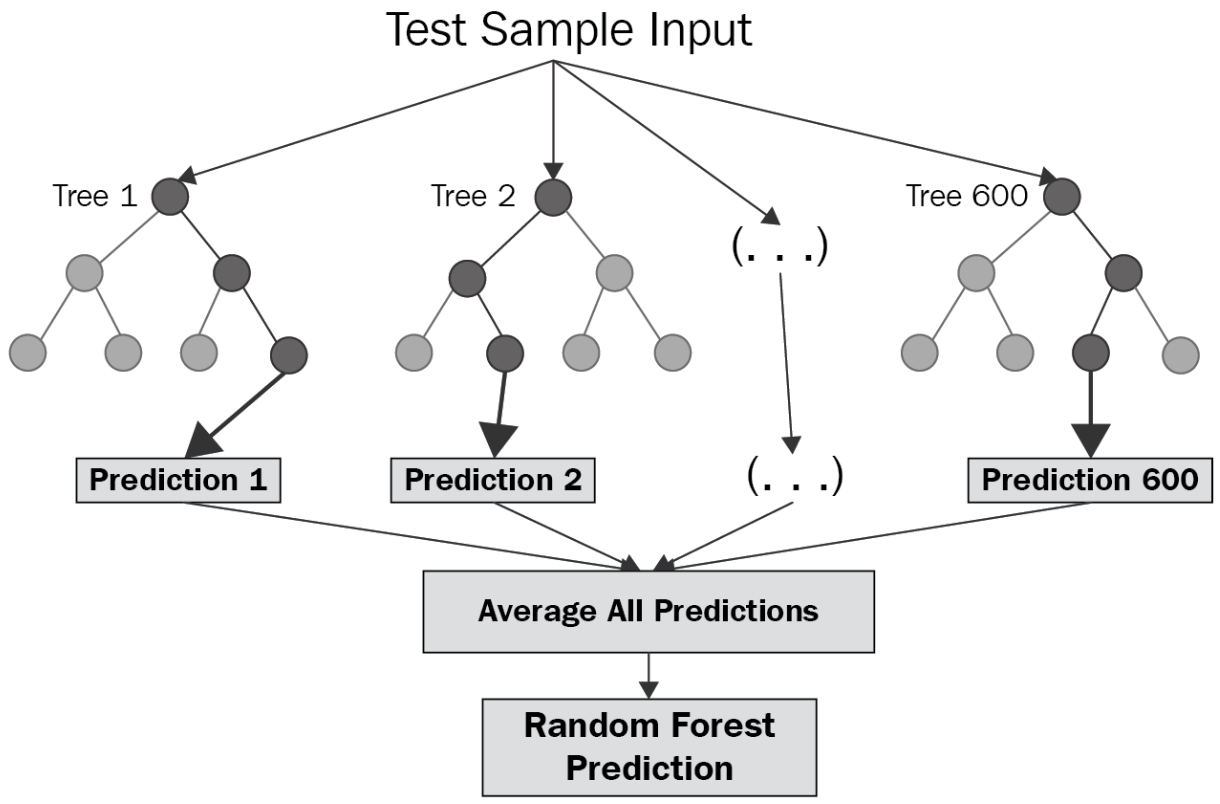
Decision Tree Regression

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. [?]



Random Forest Regression

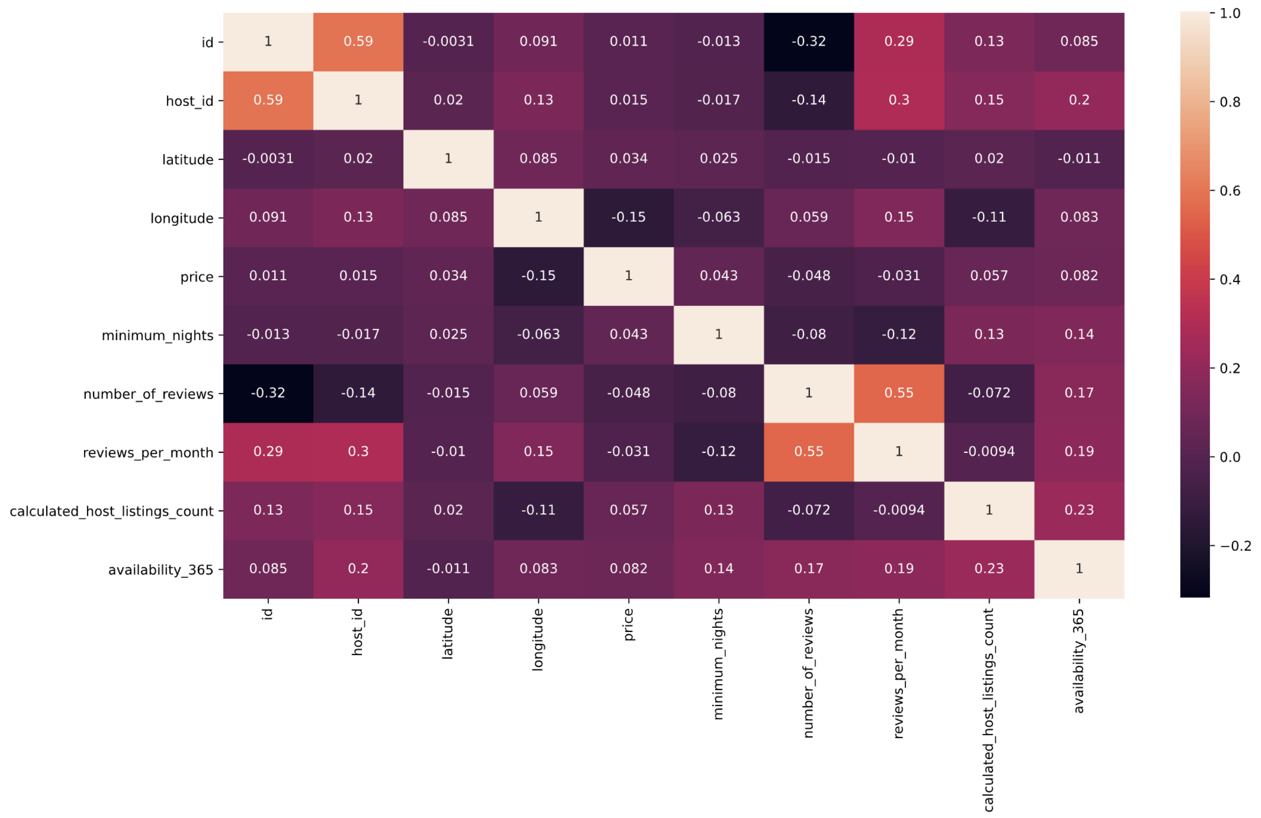
A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.



3. Analysis

3.1 Exploratory Data Analysis

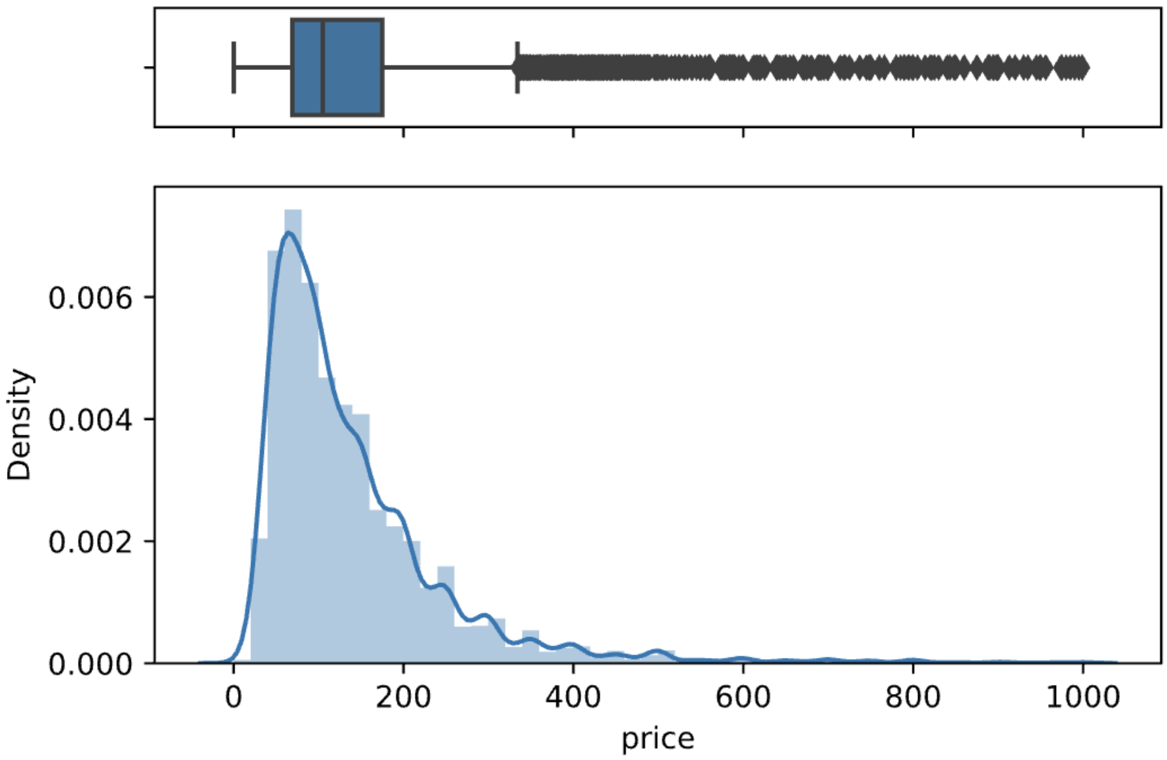
Correlation Analysis
A correlation coefficient is a way to put a value to the relationship. Correlation coefficients have a value of between -1 and 1. A "0" means there is no relationship between the variables at all, while -1 or 1 means that there is a perfect negative or positive correlation (negative or positive correlation here refers to the type of graph the relationship will produce).



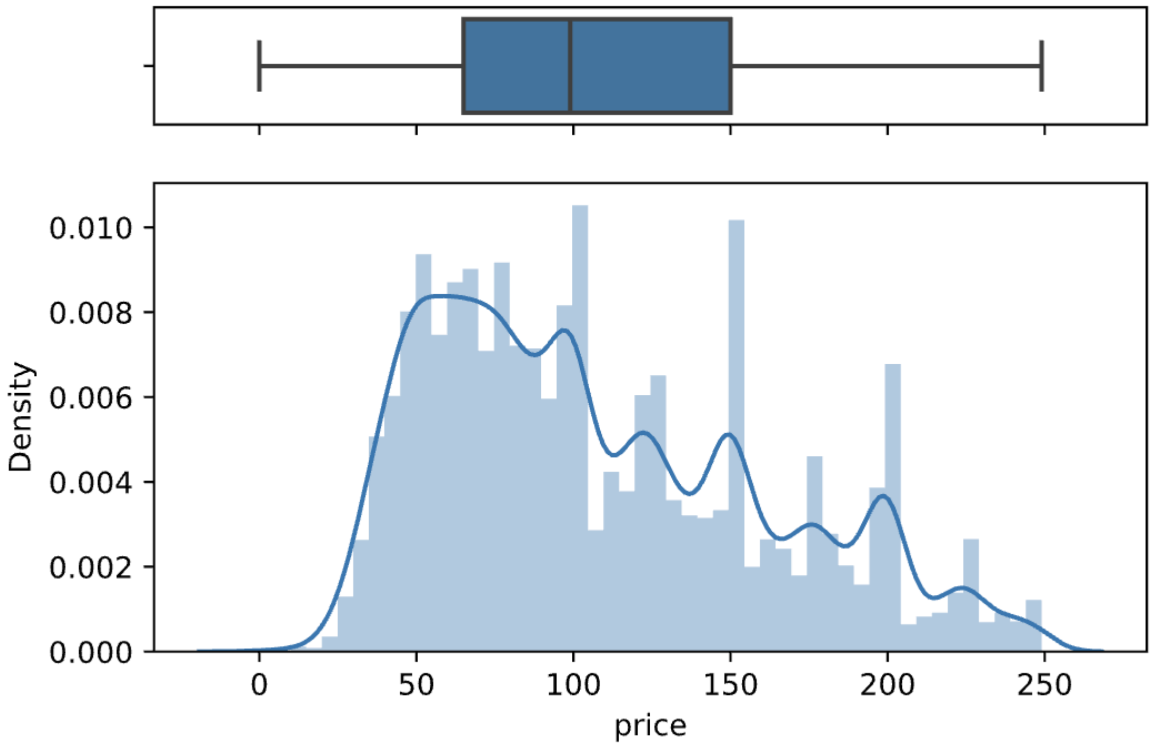
As, we can see from the Correlation Plot above, our features are correlated to each other. We can observe strong correlation between features. This can sometimes be an indicator for multicollinearity.

Price Analysis

Machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results.
In order to build a good model, it is important to have a dataset that doesn't have too many outliers. So we try to detect outliers in our prices.

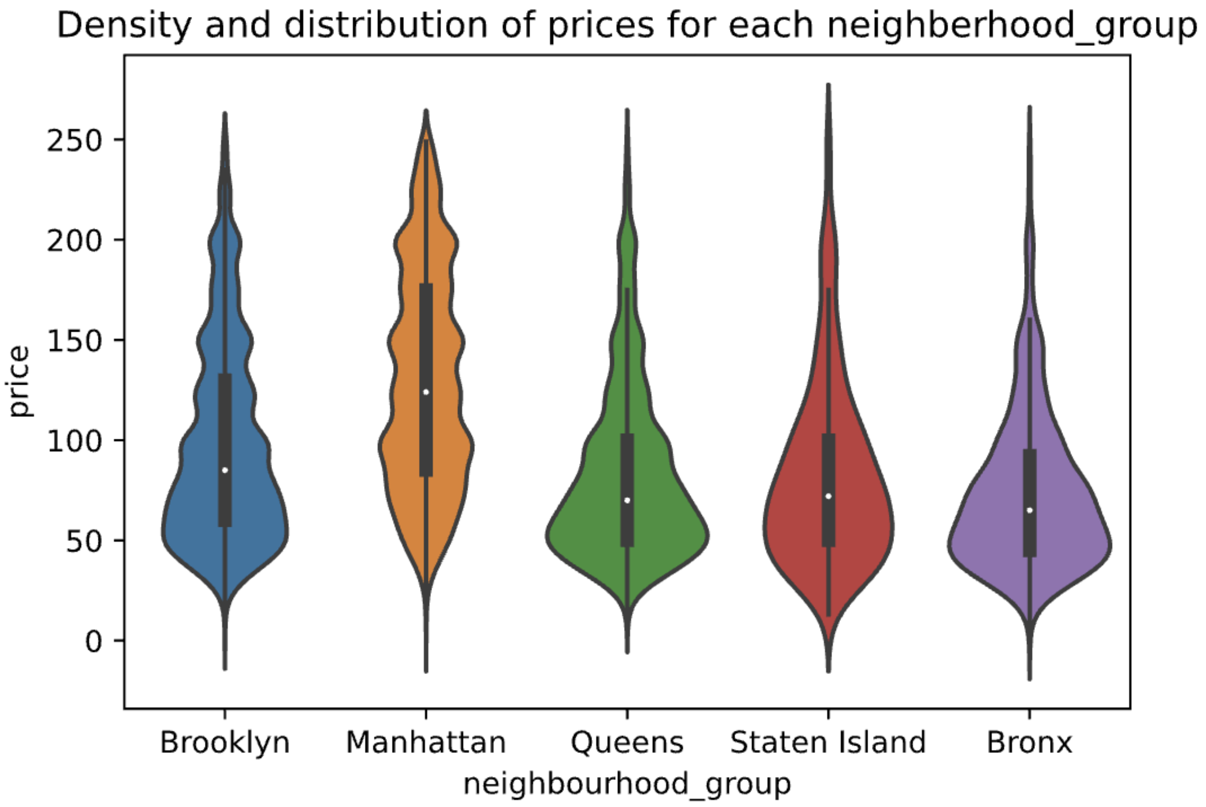


As we can notice from the plot above, we have many outliers in our price distribution. So, we drop all the outliers from our dataset.



Neighbourhood Group vs Price Analysis

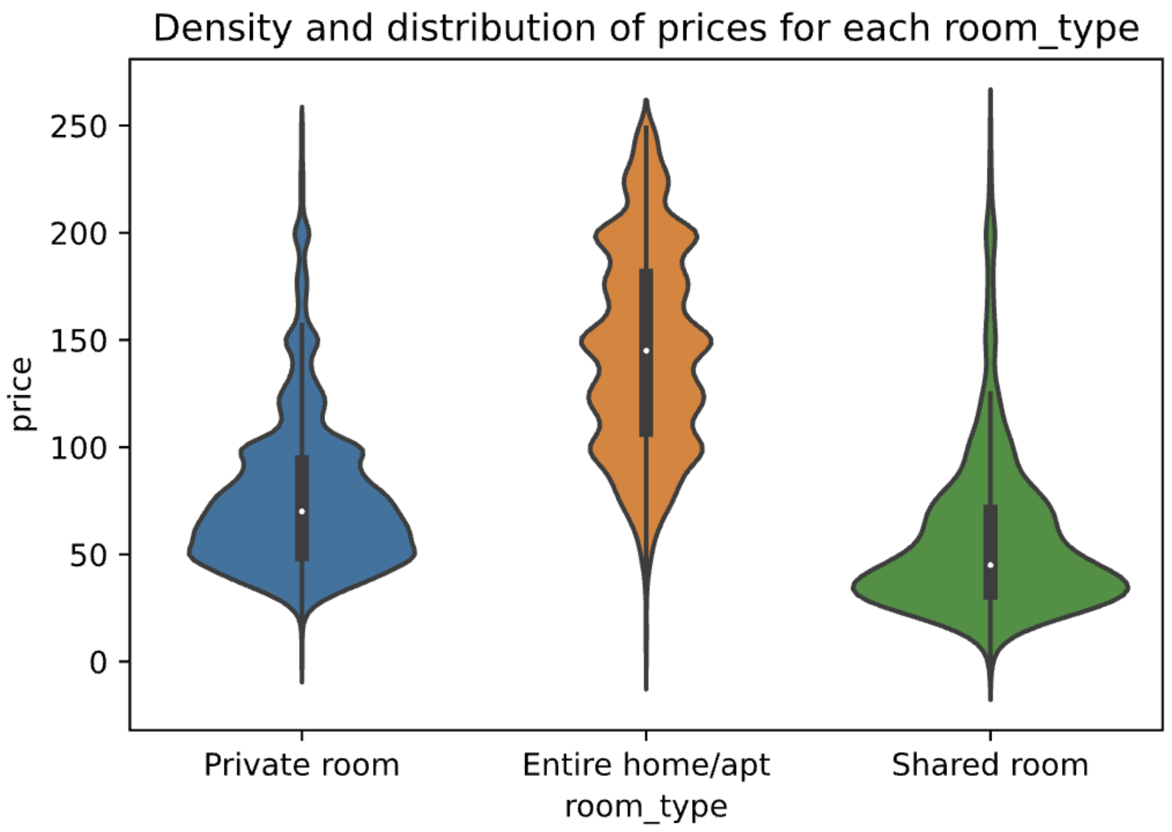
Let's now look into the relationship between neighbourhood group and price. This would help us understand if the price variable is impacted by neighbourhood groups.



We can notice from the violin plot above, that Manhattan and Brooklyn are more expensive and distributed around the upper price ranges. Whereas Queens, Staten Island and Bronx is more distributed in the bottom price ranges. This indicates that the prices are being impacted by the loction of Airbnb.

Room Type vs Price Analysis

Now we look into the relationship between room type and price. This would help us understand if the price variable is impacted by room type.



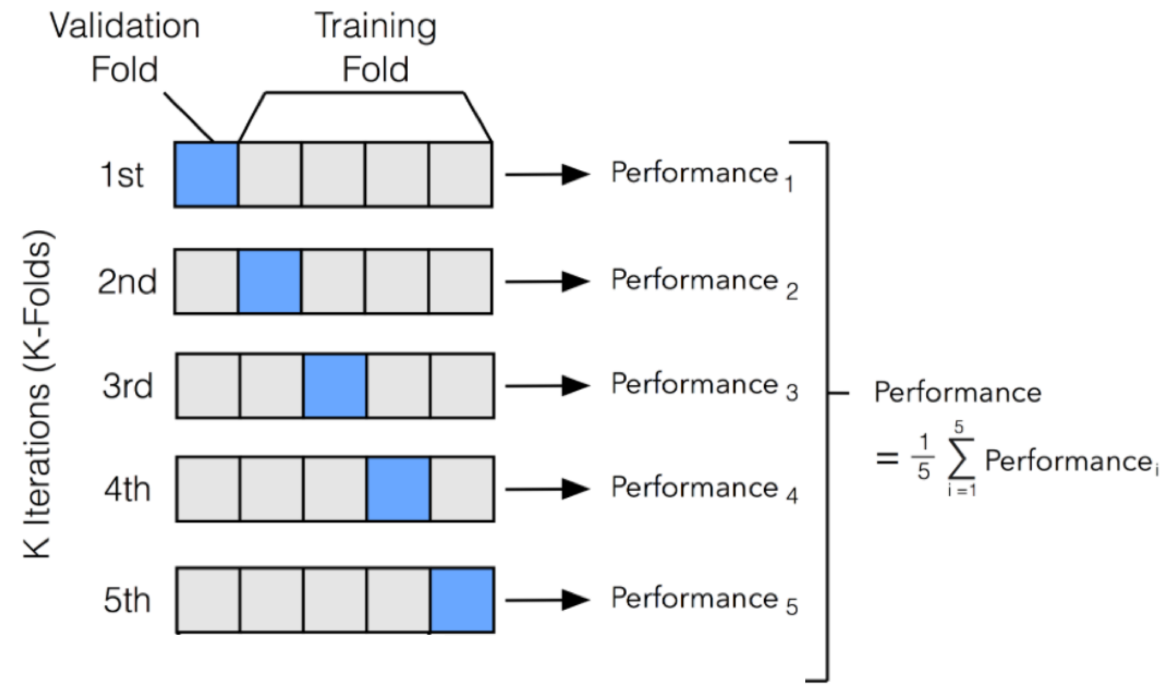
As we can notice from our violin plot above, that getting an entire apartment/home is significantly more expensive whereas shared room is much cheaper. While Private rooms are bit more expensive than shared rooms but the major price ditribution is between shared rooms and entire apartment.

3.2 Model Optimization and Tuning Techniques

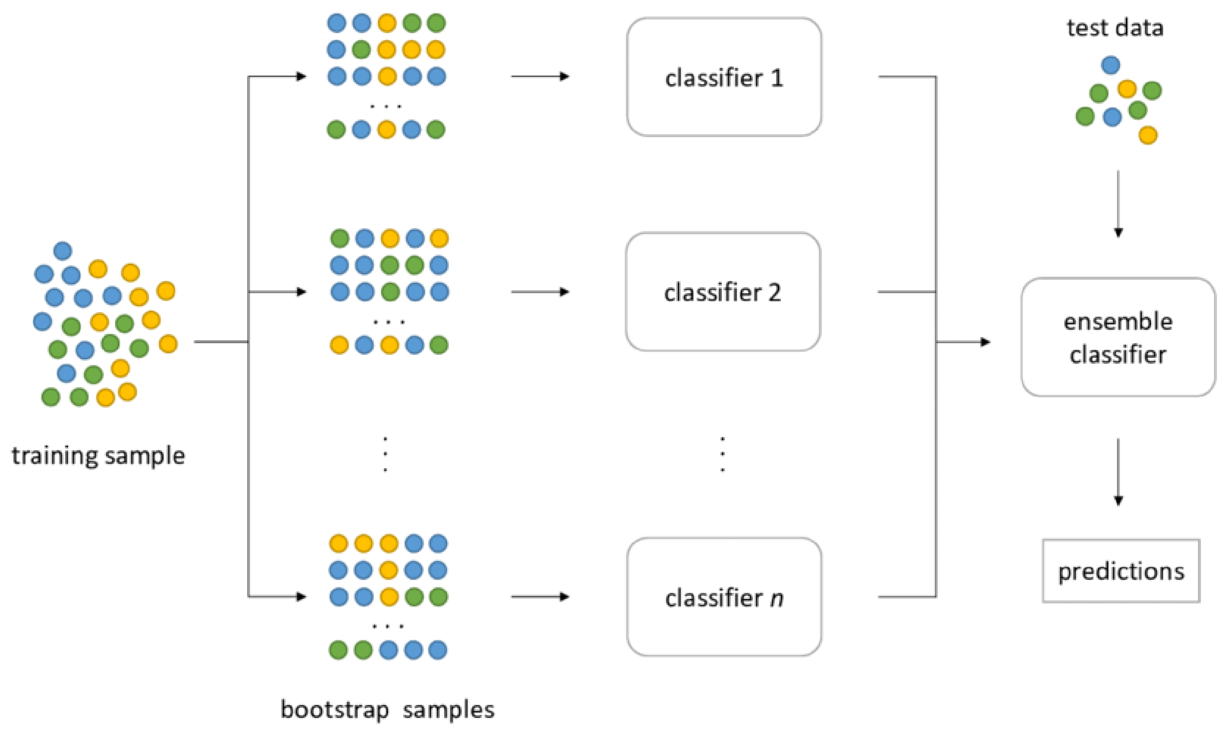
In our models, we have used multiple optimization and tuning techniques for Random Forest Regression. Before we analyze our models and results, it is important to understand these techniques. Here are the techniques that we have used to tune models using Python.
Cross Validation
Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data. This is the basic idea for a whole class of model evaluation methods called cross validation.
There are many **Cross Validation** techniques, but we have used **K-fold Cross Valida-**

tion.

K-fold Cross Validation is one way to improve over the holdout method. The data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased. The disadvantage of this method is that the training algorithm has to be rerun from scratch k times, which means it takes k times as much computation to make an evaluation. A variant of this method is to randomly divide the data into a test and training set k different times. The advantage of doing this is that you can independently choose how large each test set is and how many trials you average over.



Bootstrap
Bootstrap aggregating, also called bagging (from bootstrap aggregating), is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.


training sample
bootstrap samples
classifier 1
classifier 2
classifier n
test data
ensemble classifier
predictions

Decision Tree Regression: https://www.saedsayad.com/decision_tree_reg.htm#:~:text=Decision%20tree%

Ridge Regression: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Ridge_Regression.pdf

Lasso Regression: https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/

Linear Regression: https://brilliant.org/wiki/linear-regression/

Outlier https://datascience.foundation/sciencewhitepaper/knowing-all-about-outliers-in-machine-learning#:~:text=Machine%20learning%20algorithms%20are%20sensitive,models%20and%20ulti
Random Forest:

Cross validation: https://www.cs.cmu.edu/~schneide/tut5/node42.html