**PROJECT REPORT**

Airbnb Dataset

Submitted by:

|  |  |
| --- | --- |
| **Student Name** | **SIS ID** |
| **Pragadeesh R.C** | **O7HSA9OFQ8** |
| **ManojKumar B** | **BW9R5YUU6W** |
| **Jude Praveen Raj J** | **NEYO3435LA** |
| **Siva Vikram** | **X0MDVLOLL5** |
| **Shoba Palaksha** | **HK2QFT4WBK** |
| **Kannan V** | **EQX2GFLHRT** |

**Batch: DSE\_Chennai\_July2019**

**Mentor: Mr.Jatinder Bedi**



****

**Abstract**

Airbnb is an online-based marketing company that connects people looking for accommodation (Airbnb guests) to people looking to rent their properties (Airbnb hosts) on a short-term or long-term basis. The rentals properties include apartments (dominant), homes, boats, and whole lot more. Since its inception in 2008, Airbnb has steadily risen in terms of revenue growth and its range of service provisions. As of 2019, there 150 million users of Airbnb services in 191 countries, making it a major disruptor of the traditional hospitality industry (this is akin to how Uber and other emerging transportation services have disrupted the traditional intra-city transportation services).

Problem Statement: Price Prediction using Location and Reviews

**Chapter :1 - Introduction**

**Introduction:**

**Overview of Hospitality:**

Hospitality industry a multibillion business covering three are: Accommodation, food and beverage and travel and tourism industry. Accommodation cover hotel, motel and other lodging service. Food and beverage industry cover restaurant, pub, club, fast food chain etc. And last travel and tourism industry including airlines, train, bus and other means of transportation.

In USA, the accommodation segment is largest and accounts for 19% of the total hospitality industry. It is followed by the food services accounting for 16% of the travel and tourism market.

There was 3.4% of growth in restaurant industry in 2015 whereas the air travel sub sector accounts of 16% in travel industry. Among all the three sub-sectors the accommodation is largest which is catered by the hotels.[[1]](https://wikibizpedia.com/Hospitality_industry_in_the_US_estimates_83M_travellers_by_2020#cite_note-1)

According to the U.S. Travel Association, over 75 million international travellers visited the U.S. in 2016, and that number is expected to hit over 83 million by 2020. The U.S. is also the number one destination in the world for long-distance flights, with one of the largest groups of travellers flying out of Dubai International Airport.

**Need Of Hospitality:**

Historically, the concept of hospitality is about receiving guests in a spirit of goodwill—especially strangers from other lands. Hospitality implies warmth, respect and even protection; it builds understanding and appreciation among cultures. The Latin root hospes is formed from hostis, which means “stranger” or “enemy.” Related words are host, hospital, hostel and hotel.

Today, hospitality also refers to a segment of the service industry that includes hotels, restaurants, entertainment, sporting events, cruises and other tourism-related services. As such, the hospitality industry is important not only to societies—but to economies, customers and employees.

**Role of Analytics:**

In the hospitality industry, data analytics can be used in numerous ways in order to improve business operations, marketing strategies, occupancy rates and yield. For example, through analytics the concierge can know which local tours to recommend that fit a guest’s preference based on his past behaviour. It allows the restaurant department to predict which menu items are likely to be ordered, based for example on the local weather. It allows the reservation department to predict the optimal rate for a room. It enables the sales and marketing team to create and send tailored messages across different networks. Analytics can also help hoteliers cut down their energy costs without sacrificing guest comforts.

**Chapter :2 – Dataset Description**

**Airbnb Introduction:**

Airbnb is an online-based marketing company that connects people looking for accommodation (Airbnb guests) to people looking to rent their properties (Airbnb hosts) on a short-term or long-term basis. The rentals properties include apartments (dominant), homes, boats, and whole lot more. Since its inception in 2008, Airbnb has steadily risen in terms of revenue growth and its range of service provisions. As of 2019, there 150 million users of Airbnb services in 191 countries, making it a major disruptor of the traditional hospitality industry (this is akin to how Uber and other emerging transportation services have disrupted the traditional intra-city transportation services).

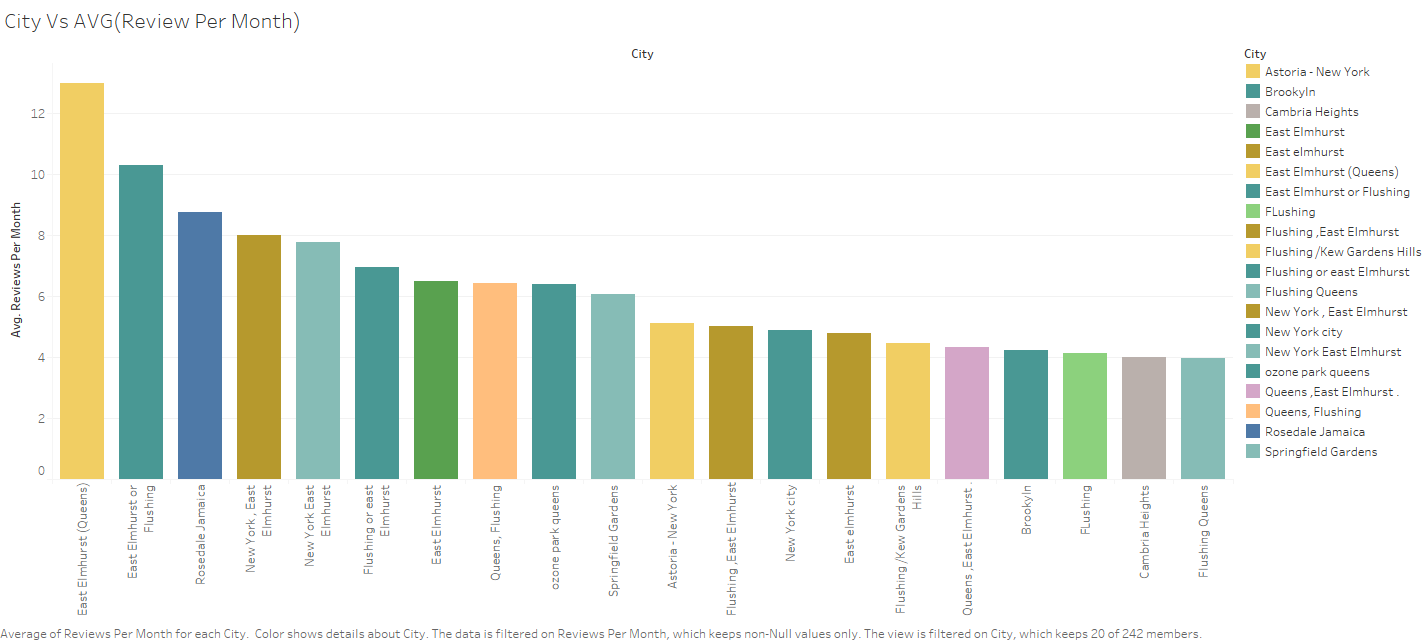
Airbnb generates revenue by charging its guests and hosts fees for arranging stays: hosts are charged 3% of the value of the booking, while guests are charged 6%-12% per the nature of the booking. As a rental ecosystem, Airbnb generates tons of data including but not limited to: density of rentals across regions (cities and neighborhoods), price variations across rentals, host-guest interactions in the form of reviews, and so forth.

New York city (NYC) has an extremely active Airbnb market with more than 48,000 listings as of August in the 2019 calendar year (this corresponds to a rental density of 48000 rentals per 468 square miles, which equates to 102 rentals per square mile). This project focuses on the gleaning patterns and other relevant information about Airbnb listings in NYC. To be more specifically, the goals of this projects are to answer questions such as: (i) how are rental properties distributed across the neighborhoods of NYC (there are 221 neighborhoods); (ii) how do prices vary with respect neighborhoods, rental property types and rental amenities; (iii) more importantly, how well can machine learning models be trained to predict Airbnb rental prices using features such as rental property type, the  number of people a rental can accommodate, the number of available beds, and so forth.

|  |
| --- |
| **Feature Name** |
| id |
| listing\_url |
| scrape\_id |
| last\_scraped |
| name |
| summary |
| space |
| description |
| experiences\_offered |
| neighborhood\_overview |
| notes |
| transit |
| access |
| interaction |
| house\_rules |
| thumbnail\_url |
| medium\_url |
| picture\_url |
| xl\_picture\_url |
| host\_id |
| host\_url |
| host\_name |
| host\_since |
| host\_location |
| host\_about |
| host\_response\_time |
| host\_response\_rate |
| host\_acceptance\_rate |
| host\_is\_superhost |
| host\_thumbnail\_url |
| host\_picture\_url |
| host\_neighbourhood |
| host\_listings\_count |
| host\_total\_listings\_count |
| host\_verifications |
| host\_has\_profile\_pic |
| host\_identity\_verified |
| street |
| neighbourhood |
| neighbourhood\_cleansed |
| neighbourhood\_group\_cleansed |
| city |
| state |
| zipcode |
| market |
| smart\_location |
| country\_code |
| country |
| latitude |
| longitude |
| is\_location\_exact |
| property\_type |
| room\_type |
| accommodates |
| bathrooms |
| bedrooms |
| beds |
| bed\_type |
| amenities |
| square\_feet |
| price |
| weekly\_price |
| monthly\_price |
| security\_deposit |
| cleaning\_fee |
| guests\_included |
| extra\_people |
| minimum\_nights |
| maximum\_nights |
| minimum\_minimum\_nights |
| maximum\_minimum\_nights |
| minimum\_maximum\_nights |
| maximum\_maximum\_nights |
| minimum\_nights\_avg\_ntm |
| maximum\_nights\_avg\_ntm |
| calendar\_updated |
| has\_availability |
| availability\_30 |
| availability\_60 |
| availability\_90 |
| availability\_365 |
| calendar\_last\_scraped |
| number\_of\_reviews |
| number\_of\_reviews\_ltm |
| first\_review |
| last\_review |
| review\_scores\_rating |
| review\_scores\_accuracy |
| review\_scores\_cleanliness |
| review\_scores\_checkin |
| review\_scores\_communication |
| review\_scores\_location |
| review\_scores\_value |
| requires\_license |
| license |
| jurisdiction\_names |
| instant\_bookable |
| is\_business\_travel\_ready |
| cancellation\_policy |
| require\_guest\_profile\_picture |
| require\_guest\_phone\_verification |
| calculated\_host\_listings\_count |
| calculated\_host\_listings\_count\_entire\_homes |
| calculated\_host\_listings\_count\_private\_rooms |
| calculated\_host\_listings\_count\_shared\_rooms |
| reviews\_per\_month |

EDA & Inferences:

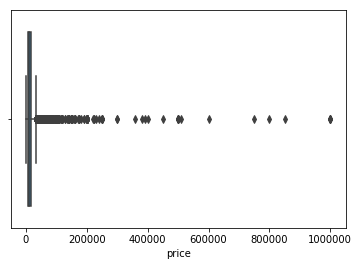
City Vs Avg



City Vs Price Per person

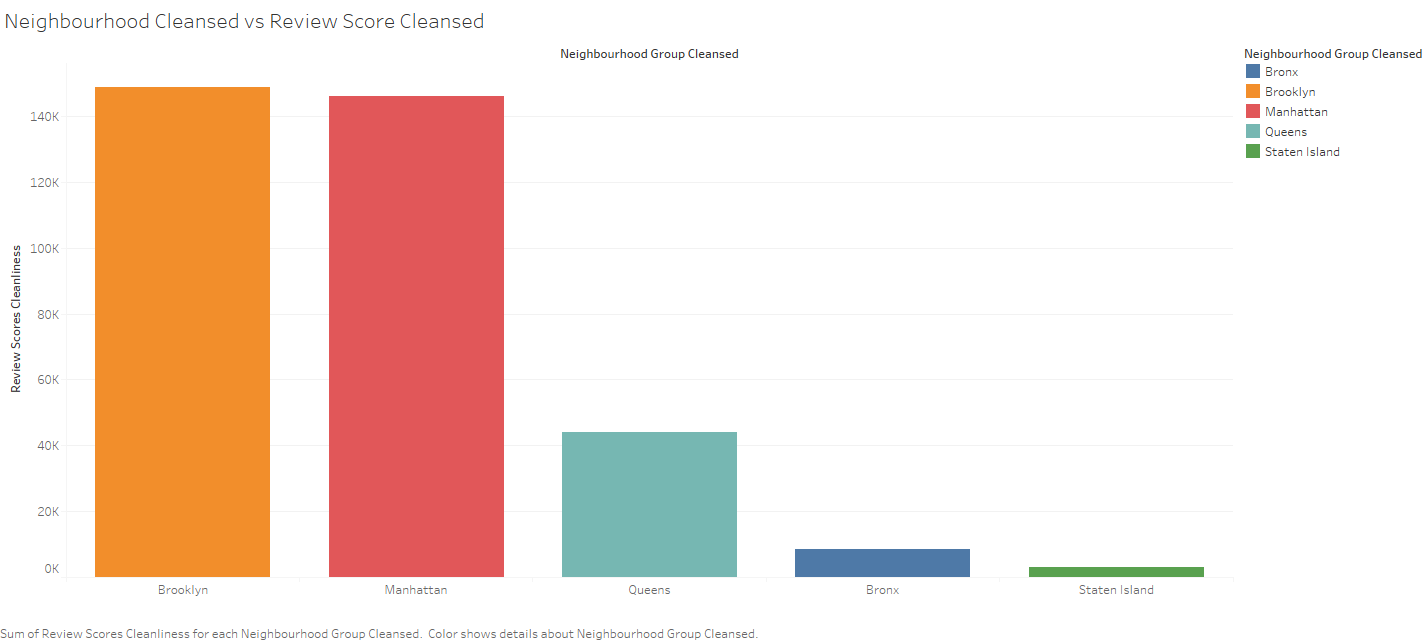


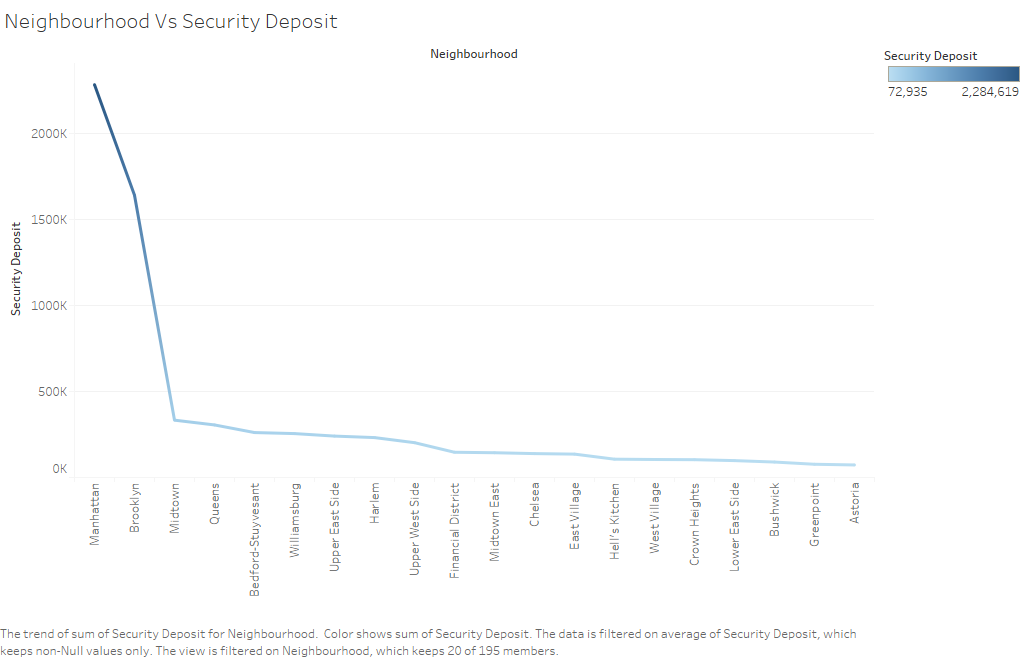
Boxplot:



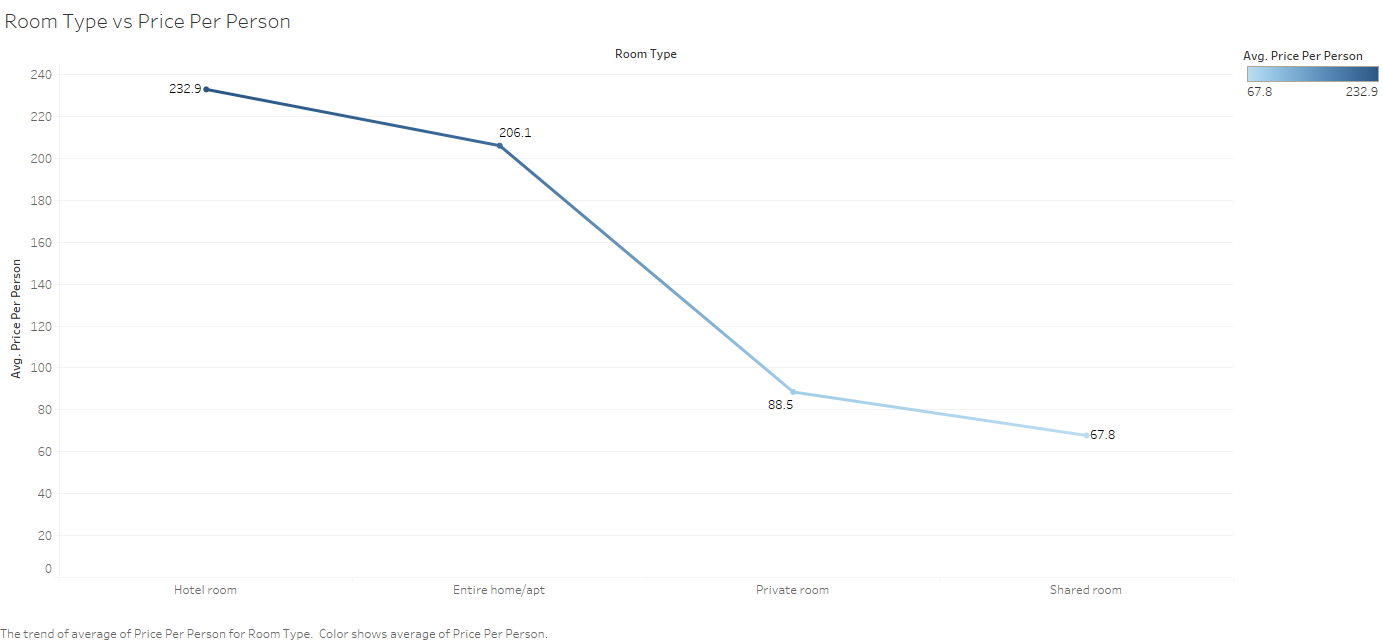
As from the above boxplot we can see that there are lot of big hotels which are having high pricing which are seen as outlier in this.

Neighbourhood cleansed vs Review Score Cleansed:





Room Type vs Price Per Person:



**Chapter :3 – Regression Models**

**Linear Regression:**

**Linear regression** is a [linear](https://en.wikipedia.org/wiki/Linearity) approach to modeling the relationship between a scalar response (or [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable)) and one or more [explanatory variables](https://en.wikipedia.org/wiki/Explanatory_variable) (or [independent variables](https://en.wikipedia.org/wiki/Independent_variable)). The case of one explanatory variable is called [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression). For more than one explanatory variable, the process is called **multiple linear regression**.[[1]](https://en.wikipedia.org/wiki/Linear_regression#cite_note-Freedman09-1) This term is distinct from [multivariate linear regression](https://en.wikipedia.org/wiki/Multivariate_linear_regression), where multiple correlated dependent variables are predicted, rather than a single scalar variable.[[2]](https://en.wikipedia.org/wiki/Linear_regression#cite_note-2)

In linear regression, the relationships are modeled using [linear predictor functions](https://en.wikipedia.org/wiki/Linear_predictor_function) whose unknown model [parameters](https://en.wikipedia.org/wiki/Parameters) are [estimated](https://en.wikipedia.org/wiki/Estimation_theory) from the [data](https://en.wikipedia.org/wiki/Data). Such models are called [linear models](https://en.wikipedia.org/wiki/Linear_model).[[3]](https://en.wikipedia.org/wiki/Linear_regression#cite_note-3) Most commonly, the [conditional mean](https://en.wikipedia.org/wiki/Conditional_expectation) of the response given the values of the explanatory variables (or predictors) is assumed to be an [affine function](https://en.wikipedia.org/wiki/Affine_transformation) of those values; less commonly, the conditional [median](https://en.wikipedia.org/wiki/Median) or some other [quantile](https://en.wikipedia.org/wiki/Quantile) is used. Like all forms of [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), linear regression focuses on the [conditional probability distribution](https://en.wikipedia.org/wiki/Conditional_probability_distribution) of the response given the values of the predictors, rather than on the [joint probability distribution](https://en.wikipedia.org/wiki/Joint_probability_distribution) of all of these variables, which is the domain of [multivariate analysis](https://en.wikipedia.org/wiki/Multivariate_analysis).

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.[[4]](https://en.wikipedia.org/wiki/Linear_regression#cite_note-4) This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed [data set](https://en.wikipedia.org/wiki/Data_set) of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information about the response.

Linear regression models are often fitted using the [least squares](https://en.wikipedia.org/wiki/Least_squares) approach, but they may also be fitted in other ways, such as by minimizing the "lack of fit" in some other [norm](https://en.wikipedia.org/wiki/Norm_(mathematics)) (as with [least absolute deviations](https://en.wikipedia.org/wiki/Least_absolute_deviations) regression), or by minimizing a penalized version of the least squares [cost function](https://en.wikipedia.org/wiki/Loss_function) as in [ridge regression](https://en.wikipedia.org/wiki/Ridge_regression) (*L*2-norm penalty) and [lasso](https://en.wikipedia.org/wiki/Lasso_(statistics)) (*L*1-norm penalty). Conversely, the least squares approach can be used to fit models that are not linear models. Thus, although the terms "least squares" and "linear model" are closely linked, they are not synonymous.

Base Model:

|  |
| --- |
|  |
| **Dep. Variable:** | price | **R-squared:** | 0.732 |
| **Model:** | OLS | **Adj. R-squared:** | 0.732 |
| **Method:** | Least Squares | **F-statistic:** | 3213. |
| **Date:** | Tue, 19 Nov 2019 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 11:14:50 | **Log-Likelihood:** | -4.0117e+05 |
| **No. Observations:** | 37659 | **AIC:** | 8.024e+05 |
| **Df Residuals:** | 37626 | **BIC:** | 8.027e+05 |
| **Df Model:** | 32 |  |  |
| **Covariance Type:** | nonrobust |  |  |
| **Omnibus:** | 73157.883 | Durbin-Watson: | 1.872 |
| **Prob(Omnibus):** | 0.000 | Jarque-Bera (JB): | 2862431836.224 |
| **Skew:** | 14.192 | Prob(JB): | 0.00 |
| **Kurtosis:** | 1353.339 | Cond. No. | 7.33e+11 |

**Assumptions:**

Jacque Bera:

In [statistics](https://en.wikipedia.org/wiki/Statistics), the **Jarque–Bera test** is a [goodness-of-fit](https://en.wikipedia.org/wiki/Goodness-of-fit) test of whether sample data have the [skewness](https://en.wikipedia.org/wiki/Skewness) and [kurtosis](https://en.wikipedia.org/wiki/Kurtosis) matching a [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution). The test is named after [Carlos Jarque](https://en.wikipedia.org/wiki/Carlos_Jarque) and [Anil K. Bera](https://en.wikipedia.org/wiki/Anil_K._Bera). The test statistic is always nonnegative. If it is far from zero, it signals the data do not have a normal distribution.

The [test statistic](https://en.wikipedia.org/wiki/Test_statistic) *JB* is defined as

{\displaystyle {\mathit {JB}}={\frac {n}{6}}\left(S^{2}+{\frac {1}{4}}(K-3)^{2}\right)}JB=n/6(S2+1/4(K-3)2)

If the data comes from a normal distribution, the *JB* statistic [asymptotically](https://en.wikipedia.org/wiki/Asymptotic_analysis) has a [chi-squared distribution](https://en.wikipedia.org/wiki/Chi-squared_distribution) with two [degrees of freedom](https://en.wikipedia.org/wiki/Degrees_of_freedom_(statistics)), so the statistic can be used to [test](https://en.wikipedia.org/wiki/Statistical_hypothesis_testing) the hypothesis that the data are from a [normal distribution](https://en.wikipedia.org/wiki/Normal_distribution). The [null hypothesis](https://en.wikipedia.org/wiki/Null_hypothesis) is a joint hypothesis of the skewness being zero and the [excess kurtosis](https://en.wikipedia.org/wiki/Excess_kurtosis) being zero. Samples from a normal distribution have an expected skewness of 0 and an expected excess kurtosis of 0 (which is the same as a kurtosis of 3). As the definition of *JB* shows, any deviation from this increases the JB statistic.

For small samples the chi-squared approximation is overly sensitive, often rejecting the null hypothesis when it is true. Furthermore, the distribution of [*p*-values](https://en.wikipedia.org/wiki/P-value) departs from a [uniform distribution](https://en.wikipedia.org/wiki/Uniform_distribution_(continuous)) and becomes a [right-skewed](https://en.wikipedia.org/wiki/Right-skewed) [unimodal distribution](https://en.wikipedia.org/wiki/Unimodal_distribution), especially for small *p*-values. This leads to a large [Type I error](https://en.wikipedia.org/wiki/Type_I_error) rate. The table below shows some *p*-values approximated by a chi-squared distribution that differ from their true alpha levels for small samples.

Inference :

After performing Jarque Bera Test we got a value of 2862431836.224

**Rainbow Test:**

Linear Rainbow Test

•The basic idea of the Rainbow-Test is that even if the true relationship is nonlinear, over a subsample of data given a good linear fit can be achieved.

•The null hypothesis is rejected whenever the overall fit is significantly inferiousto the fit of the subsample.

•The test statistic under H0 follows a F distribution with df1 and df2 degree of freedom.

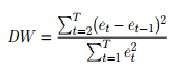
•This particular procedure compares a subsample consisting of all data points without the upper and lower quartile (of time index), thus 50 data points.

•If the true relationship is concave or convex, the null hypothesis should be rejected.

**Durbin Watson Test:**

The Durbin Watson Test is a measure of [autocorrelation](https://www.statisticshowto.datasciencecentral.com/serial-correlation-autocorrelation/)(also called [serial correlation](https://www.statisticshowto.datasciencecentral.com/serial-correlation-autocorrelation/)) in [residuals](https://www.statisticshowto.datasciencecentral.com/residual/)from [regression analysis](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/regression-analysis/). Autocorrelation is the similarity of a [time series](https://www.statisticshowto.datasciencecentral.com/timeplot/) over successive time intervals. It can lead to underestimates of the [standard error](https://www.statisticshowto.datasciencecentral.com/find-standard-error-regression-slope/) and can cause you to think predictors are [significant](https://www.statisticshowto.datasciencecentral.com/what-is-statistical-significance/)when they are not. **The Durbin Watson test looks for a specific type of serial correlation, the**[**AR(1) process**](https://www.statisticshowto.datasciencecentral.com/autoregressive-model/)**.**  
The Hypotheses for the Durbin Watson test are:  
H0 = no first order autocorrelation.  
H1 = first order correlation exists.  
(For a first order correlation, the lag is one time unit).  
Assumptions are:

* That the errors are [normally distributed](https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/normal-distributions/) with a [mean](https://www.statisticshowto.datasciencecentral.com/mean/)of 0.
* The errors are [stationary](https://www.statisticshowto.datasciencecentral.com/stationarity/).

The test statistic is calculated with the following formula:  
[](https://www.statisticshowto.datasciencecentral.com/wp-content/uploads/2016/06/durbin-watson.png)  
  
  
Where Et are [residuals](https://www.statisticshowto.datasciencecentral.com/residual/)from an [ordinary least squares regression](https://www.statisticshowto.datasciencecentral.com/least-squares-regression-line/).

The Durbin Watson test reports a test statistic, with a value from 0 to 4, where:

* 2 is no autocorrelation.
* 0 to <2 is positive autocorrelation (common in time series data).
* >2 to 4 is negative autocorrelation (less common in time series data).

A **rule of thumb**is that test statistic values in the range of 1.5 to 2.5 are relatively normal. Values outside of this range could be cause for concern. Field(2009) suggests that values under 1 or more than 3 are a definite cause for concern.

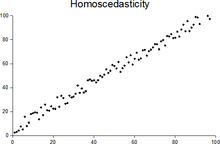
|  |  |
| --- | --- |
| **Output of Durbin – Watson test is 1.869** |  |

**Homoscedasticity test:**

In [statistics](https://en.wikipedia.org/wiki/Statistics), a [sequence](https://en.wikipedia.org/wiki/Sequence) (or a vector) of [random variables](https://en.wikipedia.org/wiki/Random_variable) is **homoscedastic**  if all its random variables have the same finite [variance](https://en.wikipedia.org/wiki/Variance). This is also known as **homogeneity of variance**. The complementary notion is called [heteroscedasticity](https://en.wikipedia.org/wiki/Heteroscedasticity). The spellings *homos****k****edasticity* and *heteros****k****edasticity* are also frequently used.[[1]](https://en.wikipedia.org/wiki/Homoscedasticity#cite_note-1)

The assumption of homoscedasticity simplifies mathematical and computational treatment. Serious violations in homoscedasticity (assuming a distribution of data is homoscedastic when in reality it is heteroscedastic [/ˌhɛtəroʊskəˈdæstɪk/](https://en.wikipedia.org/wiki/Help:IPA/English)) may result in overestimating the [goodness of fit](https://en.wikipedia.org/wiki/Goodness_of_fit) as measured by the [Pearson coefficient](https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient).

As used in describing [simple linear regression](https://en.wikipedia.org/wiki/Simple_linear_regression) analysis, one assumption of the fitted model (to ensure that the least-squares estimators are each a [best linear unbiased estimator](https://en.wikipedia.org/wiki/Best_linear_unbiased_estimator) of the respective population parameters, by the [Gauss–Markov theorem](https://en.wikipedia.org/wiki/Gauss%E2%80%93Markov_theorem)) is that the standard deviations of the error terms are constant and do not depend on the *x*-value. Consequently, each probability distribution for *y* (response variable) has the same standard deviation regardless of the *x*-value (predictor). In short, this assumption is homoscedasticity. Homoscedasticity is not required for the estimates to be unbiased, consistent, and asymptotically normal.[[2]](https://en.wikipedia.org/wiki/Homoscedasticity#cite_note-2)



Multicollinearity:

In [statistics](https://en.wikipedia.org/wiki/Statistics), **multicollinearity** (also **collinearity**) is a phenomenon in which one predictor [variable](https://en.wikipedia.org/wiki/Variable_(mathematics)) in a [multiple regression](https://en.wikipedia.org/wiki/Multiple_regression) model can be linearly predicted from the others with a substantial degree of accuracy. In this situation the [coefficient estimates](https://en.wikipedia.org/wiki/Regression_coefficient) of the multiple regression may change erratically in response to small changes in the model or the data. Multicollinearity does not reduce the predictive power or [reliability](https://en.wikipedia.org/wiki/Reliability_(statistics)) of the model as a whole, at least within the sample data set; it only affects calculations regarding [individual predictors](https://en.wikipedia.org/wiki/Dependent_and_independent_variables#Use_in_statistics). That is, a multivariate regression model with collinear predictors can indicate how well the entire bundle of predictors predicts the [outcome variable](https://en.wikipedia.org/wiki/Dependent_variable#Use_in_statistics), but it may not give valid results about any individual predictor, or about which predictors are redundant with respect to others.

Note that in statements of the assumptions underlying regression analyses such as [ordinary least squares](https://en.wikipedia.org/wiki/Ordinary_least_squares), the phrase "no multicollinearity" usually refers to the absence of perfect multicollinearity, which is an exact (non-stochastic) linear relation among the predictors. In such case, the [data matrix](https://en.wikipedia.org/wiki/Data_matrix_(multivariate_statistics)) {\displaystyle X} has less than full [rank](https://en.wikipedia.org/wiki/Rank_(linear_algebra)), and therefore the [moment matrix](https://en.wikipedia.org/wiki/Moment_matrix) {\displaystyle X^{\mathsf {T}}X} cannot be [inverted](https://en.wikipedia.org/wiki/Matrix_inversion). Under these circumstances, for a general linear model {\displaystyle y=X\beta +\epsilon }, the ordinary least squares estimator {\displaystyle {\hat {\beta }}\_{OLS}=(X^{\mathsf {T}}X)^{-1}X^{\mathsf {T}}y} does not exist.

In any case, multicollinearity is a characteristic of the data matrix, not the underlying [statistical model](https://en.wikipedia.org/wiki/Statistical_model). Since it is generally more severe in small samples, [Arthur Goldberger](https://en.wikipedia.org/wiki/Arthur_Goldberger) went so far as to call it "micronumerosity."

**Lasso :**

It is “Least Absolute Shrinkage and Selection Operator”

•It is the summation of the absolute value of the coefficients.

•It shrinks the regression coefficients toward zero by penalizing the regression model with a penalty term called **L1-norm**, which is the sum of the absolute coefficients.

•LASSO is used when you have more variables and when you want to remove unwanted variables to the model, as it can bring the value to 0

•Lasso can be also seen as an alternative to the subset selection methods for performing variable selection in order to reduce the complexity of the model.

* λ is the hypermeter, whose value is equal to the alpha in the Lasso function
* LASSO is generally used when we have more number of features, because it automatically does feature selection.
* One obvious advantage of lasso regression over ridge regression,
* It produces simpler and more interpretable models that incorporate only a reduced set of the predictors.
* However, neither ridge regression nor the lasso will universally dominate the other.



lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region.

The lasso has a major advantage over ridge regression, in that it produces simpler and more interpretable models that involved only a subset of predictors.

•The lasso leads to qualitatively similar behavior to ridge regression, in that as λincreases, the variance decreases and the bias increases.

•The lasso can generate more accurate predictions compared to ridge regression.

•Cross-validation can be used in order to determine which approach is better on a particular data set.

**After performing Lasso we are getting an output of 0.722**

**Ridge Regression:**

It is the summation of the squared value of coefficients.

•Ridge helps to reduce or shrink the variance and making prediction less sensitive to the unwanted variable but not removes it

•The shrinkage of the coefficients is achieved by penalizing the regression model with a penalty term calledL2-norm, which is the sum of the squared coefficients.

•The amount of the penalty can be fine-tuned using a constant called lambda (λ). Selecting a good value for λ is critical.

•When λ=0, the penalty term has no effect, and ridge regression will produce the classical least square coefficients. However, as λ increases to infinite, the impact of the shrinkage penalty grows, and the ridge regression coefficients will get close zero.

•λ given here is actually denoted by alpha parameter in the ridge function. So by changing the value of alpha we are controlling the penalty term

It shrinks the parameters, therefore it is mostly used to prevent multicollinearity.

•One important advantage of the ridge regression,

•It still performs well, compared to the ordinary least square method in a situation where you have a large multivariate data with the number of predictors (p) larger than the number of observations (n).

•One disadvantage of the ridge regression,

•It will include all the predictors in the final model, unlike the stepwise regression methods which will generally select models that involve a reduced set of variables.

•Ridge regression shrinks the coefficients towards zero, but it will not set any of them exactly to zero. The lasso regression is an alternative that overcomes this drawback.

**Output:**

**After performing Ridge regression we were able to get an accuracy score of 0.66**

**Chapter :4 – Feature Engineering**

Introduction:

Basically, all machine learning algorithms use some input data to create outputs. This input data comprise features, which are usually in the form of structured columns. Algorithms require features with some specific characteristic to work properly. Here, the need for **feature engineering** arises. I think feature engineering efforts mainly have two goals:

* Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
* Improving the performance of machine learning models.

## List of Techniques

[1.Imputation](https://medium.com/p/3a5e293a5114#3abe)

[2.Handling Outliers](https://medium.com/p/3a5e293a5114#1c08)

[3.Binning](https://medium.com/p/3a5e293a5114#7559)

[4.Log Transform](https://medium.com/p/3a5e293a5114#199b)

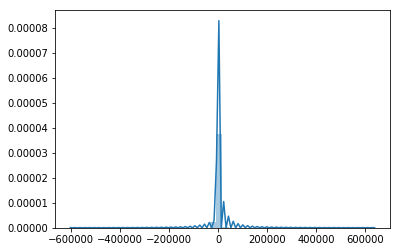
[5.One-Hot Encoding](https://medium.com/p/3a5e293a5114#7c18)

[6.Grouping Operations](https://medium.com/p/3a5e293a5114#ad97)

[7.Feature Split](https://medium.com/p/3a5e293a5114#3149)

[8.Scaling](https://medium.com/p/3a5e293a5114#83e6)

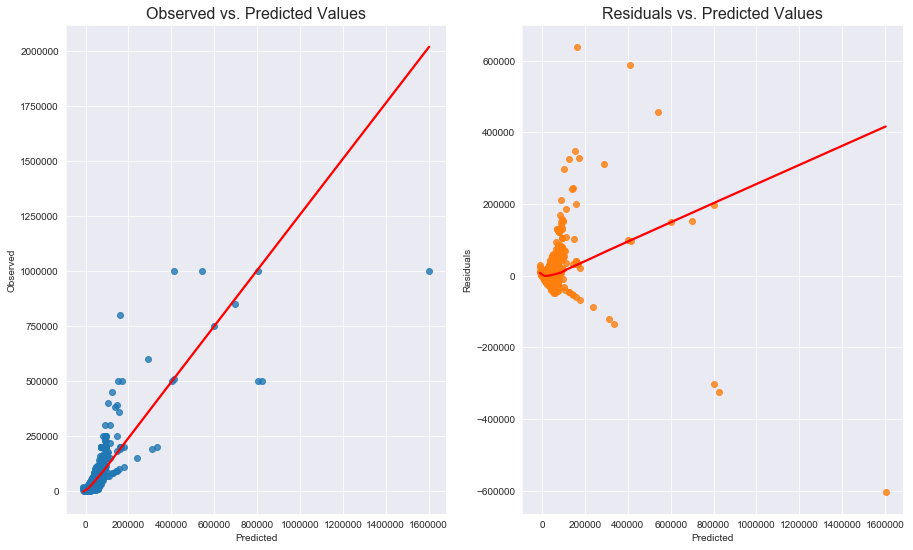
[9.Extracting Date](https://medium.com/p/3a5e293a5114#8068)



p-value is less than alpha value so it is normal.

we reject the null hypothesis that the error terms are normally distributed.

After performing linearity test we have got the below charts:



Chapter :5 – Ensemble Techniques

**Bagging:**

Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method.

An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.

Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithm that have high variance. An algorithm that has high variance are decision trees, like classification and regression trees (CART).

Decision trees are sensitive to the specific data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree can be quite different and in turn the predictions can be quite different.

Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.

Let’s assume we have a sample dataset of 1000 instances (x) and we are using the CART algorithm. Bagging of the CART algorithm would work as follows.

1. Create many (e.g. 100) random sub-samples of our dataset with replacement.
2. Train a CART model on each sample.
3. Given a new dataset, calculate the average prediction from each model.

For example, if we had 5 bagged decision trees that made the following class predictions for a in input sample: blue, blue, red, blue and red, we would take the most frequent class and predict blue.

When bagging with decision trees, we are less concerned about individual trees overfitting the training data. For this reason and for efficiency, the individual decision trees are grown deep (e.g. few training samples at each leaf-node of the tree) and the trees are not pruned. These trees will have both high variance and low bias. These are important characterize of sub-models when combining predictions using bagging.

The only parameters when bagging decision trees is the number of samples and hence the number of trees to include. This can be chosen by increasing the number of trees on run after run until the accuracy begins to stop showing improvement (e.g. on a cross validation test harness). Very large numbers of models may take a long time to prepare, but will not overfit the training data.

Just like the decision trees themselves, Bagging can be used for classification and regression problems.

After performing Bagging the output is as below:

Accuracy: 0.24 (+/- 0.15) [Bagging Tree]

**Boosting:**

Boosting is a [machine learning ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) [meta-algorithm](https://en.wikipedia.org/wiki/Meta-algorithm) for primarily reducing [bias](https://en.wikipedia.org/wiki/Supervised_learning#Bias-variance_tradeoff), and also variance[[1]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-1) in [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), and a family of machine learning algorithms that convert weak learners to strong ones.[[2]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-2) Boosting is based on the question posed by [Kearns](https://en.wikipedia.org/wiki/Michael_Kearns_(computer_scientist)) and [Valiant](https://en.wikipedia.org/wiki/Leslie_Valiant) (1988, 1989):[[3]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Kearns88-3)[[4]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-4) "Can a set of **weak learners** create a single **strong learner**?" A weak learner is defined to be a [classifier](https://en.wikipedia.org/wiki/Classification_(machine_learning)) that is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

[Robert Schapire](https://en.wikipedia.org/wiki/Robert_Schapire)'s affirmative answer in a 1990 paper[[5]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Schapire90-5) to the question of Kearns and Valiant has had significant ramifications in [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [statistics](https://en.wikipedia.org/wiki/Statistics), most notably leading to the development of boosting.[[6]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-6)

When first introduced, the *hypothesis boosting problem* simply referred to the process of turning a weak learner into a strong learner. "Informally, [the hypothesis boosting] problem asks whether an efficient learning algorithm […] that outputs a hypothesis whose performance is only slightly better than random guessing [i.e. a weak learner] implies the existence of an efficient algorithm that outputs a hypothesis of arbitrary accuracy [i.e. a strong learner]."[[3]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Kearns88-3) Algorithms that achieve hypothesis boosting quickly became simply known as "boosting". [Freund](https://en.wikipedia.org/wiki/Yoav_Freund) and Schapire's arcing (Adapt[at]ive Resampling and Combining),[[7]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-7) as a general technique, is more or less synonymous with boosting.

While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learners' accuracy. After a weak learner is added, the data weights are readjusted, known as "re-[weighting](https://en.wikipedia.org/wiki/Weighting)". Misclassified input data gain a higher weight and examples that are classified correctly lose weight.[[note 1]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-9) Thus, future weak learners focus more on the examples that previous weak learners misclassified.

There are many boosting algorithms. The original ones, proposed by [Robert Schapire](https://en.wikipedia.org/wiki/Robert_Schapire) (a [recursive](https://en.wikipedia.org/wiki/Recursion_(computer_science)) majority gate formulation)[[5]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Schapire90-5) and [Yoav Freund](https://en.wikipedia.org/wiki/Yoav_Freund) (boost by majority)[[9]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Mason00-10), were not [adaptive](https://en.wikipedia.org/wiki/Adaptive_behavior) and could not take full advantage of the weak learners. Schapire and Freund then developed [AdaBoost](https://en.wikipedia.org/wiki/AdaBoost), an adaptive boosting algorithm that won the prestigious [Gödel Prize](https://en.wikipedia.org/wiki/G%C3%B6del_Prize).

Only algorithms that are provable boosting algorithms in the [probably approximately correct learning](https://en.wikipedia.org/wiki/Probably_approximately_correct_learning) formulation can accurately be called *boosting algorithms*. Other algorithms that are similar in spirit to boosting algorithms are sometimes called "leveraging algorithms", although they are also sometimes incorrectly called boosting algorithms.

The main variation between many boosting algorithms is their method of [weighting](https://en.wikipedia.org/wiki/Weighting) [training data](https://en.wikipedia.org/wiki/Training_data) points and [hypotheses](https://en.wikipedia.org/wiki/Hypothesis). [AdaBoost](https://en.wikipedia.org/wiki/AdaBoost) is very popular and the most significant historically as it was the first algorithm that could adapt to the weak learners. It is often the basis of introductory coverage of boosting in university machine learning courses.[[10]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-11) There are many more recent algorithms such as [LPBoost](https://en.wikipedia.org/wiki/LPBoost" \o "LPBoost), TotalBoost, [BrownBoost](https://en.wikipedia.org/wiki/BrownBoost" \o "BrownBoost), [xgboost](https://en.wikipedia.org/wiki/Xgboost" \o "Xgboost), MadaBoost, [LogitBoost](https://en.wikipedia.org/wiki/LogitBoost" \o "LogitBoost), and others. Many boosting algorithms fit into the AnyBoost framework,[[9]](https://en.wikipedia.org/wiki/Boosting_(machine_learning)#cite_note-Mason00-10) which shows that boosting performs [gradient descent](https://en.wikipedia.org/wiki/Gradient_descent) in a [function space](https://en.wikipedia.org/wiki/Function_space) using a [convex](https://en.wikipedia.org/wiki/Convex_function) [cost function](https://en.wikipedia.org/wiki/Loss_functions_for_classification).

XGBoost:

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

In this post you will discover XGBoost and get a gentle introduction to what is, where it came from and how you can learn more.

After reading this post you will know:

* What XGBoost is and the goals of the project.
* Why XGBoost must be apart of your machine learning toolkit.
* Where you can learn more to start using XGBoost on your next machine learning project.

Discover how to configure, fit, tune and evaluation gradient boosting models with XGBoost [in my new book](https://machinelearningmastery.com/xgboost-with-python/), with 15 step-by-step tutorial lessons, and full python code.

Let’s get started.

## What is XGBoost?

XGBoost stands for e**X**treme **G**radient **B**oosting.

*The name xgboost, though, actually refers to the engineering goal to push the limit of computations resources for boosted tree algorithms. Which is the reason why many people use xgboost.*

## XGBoost Features

The library is laser focused on computational speed and model performance, as such there are few frills. Nevertheless, it does offer a number of advanced features.

### Model Features

The implementation of the model supports the features of the scikit-learn and R implementations, with new additions like regularization. Three main forms of gradient boosting are supported:

* **Gradient Boosting** algorithm also called gradient boosting machine including the learning rate.
* **Stochastic Gradient Boosting** with sub-sampling at the row, column and column per split levels.
* **Regularized Gradient Boosting** with both L1 and L2 regularization.

### System Features

The library provides a system for use in a range of computing environments, not least:

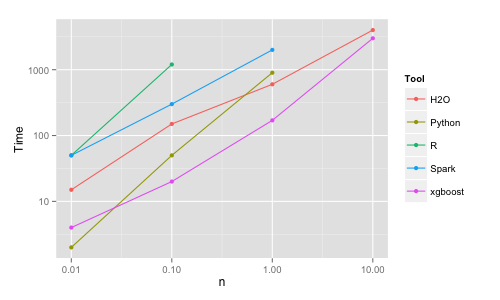
* **Parallelization** of tree construction using all of your CPU cores during training.
* **Distributed Computing** for training very large models using a cluster of machines.
* **Out-of-Core Computing** for very large datasets that don’t fit into memory.
* **Cache Optimization** of data structures and algorithm to make best use of hardware.

### Algorithm Features

The implementation of the algorithm was engineered for efficiency of compute time and memory resources. A design goal was to make the best use of available resources to train the model. Some key algorithm implementation features include:

* **Sparse Aware** implementation with automatic handling of missing data values.
* **Block Structure** to support the parallelization of tree construction.
* **Continued Training** so that you can further boost an already fitted model on new data.

XGBoost is free open source software available for use under the permissive Apache-2 license.



**After performing XG Boost the output of the predicted value is 92.44%**

**Chapter :6 – Conclusion**

XG Boost performs better in both the models with Feature Engineering and without Feature Engineering

Business recommendations:

With the help of the predicted price it will be easy for Airbnb to list out the actual prices

**Chapter :7 – References**

**References:**

1. <https://wikibizpedia.com/Hospitality_industry_in_the_US_estimates_83M_travellers_by_2020>
2. <https://www.kendall.edu/blog/why-is-the-hospitality-industry-important/>
3. <https://analyticsindiamag.com/how-analytics-can-help-the-hospitality-industry/>
4. <https://en.wikipedia.org/wiki/Jarque%E2%80%93Bera_test>
5. <https://en.wikipedia.org/wiki/Multicollinearity>
6. <https://en.wikipedia.org/wiki/Lasso_(statistics)>
7. <https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/>
8. <https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/>
9. <https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114>