A MACHINE LEARNING PROJECT AND ANALYSIS ON AIRBNB - MONTREAL

Ruta Patel (20755706)
University of Waterloo

***Abstract*—** Airbnb is a multinational company that offers affordable short term accommodations [1] to people in around 100K cities with over 6 million plus listings in more than 191 countries.[2] It is an online market place through which guest can book their stay from millions of housing options during their vacation or short trip. It's a win-win situation for both, as guests can find a place to stay that falls under their budget in expensive cities such as New York, Toronto etc whereas hosts can make an extra income from their empty properties. The listings are posted on Airbnb website, with a description of the property, facilities it provides, neighborhood and location, any tourists' attraction nearby and much more. Based on all the above criteria a price is decided by a host. As there are so many features to decide from, predicting and analyzing the price distribution can be an interesting machine learning problem.

This report is an extended and improved version of the work that has been done by me [3] for the same. This project focuses on predicting prices for all the Airbnb listings hosted in Montreal city, Quebec, Canada. Further, the regression based analysis is turned into classification to get better accuracy. After evaluating all the models, results are drawn in the last section.

Key words: Machine learning, Regression, Classification, cross validation, Sentiment analysis, Overfitting, Data Modeling

I. INTRODUCTION

Airbnb continues to grow and is expanding in numerous countries as more and more people are switching to this affordable option rather than spending a sizeable chunk of money on over priced hotels in already expensive mega cities such as Chicago, Paris etc. One such city famous for its tourist's attractions is Montreal, Quebec in Canada with over 19,495 listings to choose from. When there are so many options and wide range of services being offered, it is a tedious task to predict an optimal price.

Given that users are increasing from both the ends, it is essential to understand how the prices vary for properties depending upon its housing style, seasonality, room type, location etc.

Additionally, with the increasing popularity of Airbnb and so many options to choose from, hosts would not want to undercharge or overcharge for their property. Hence, it is essential to answer how much should a host demand for their property and what is the correct price a guest should pay?

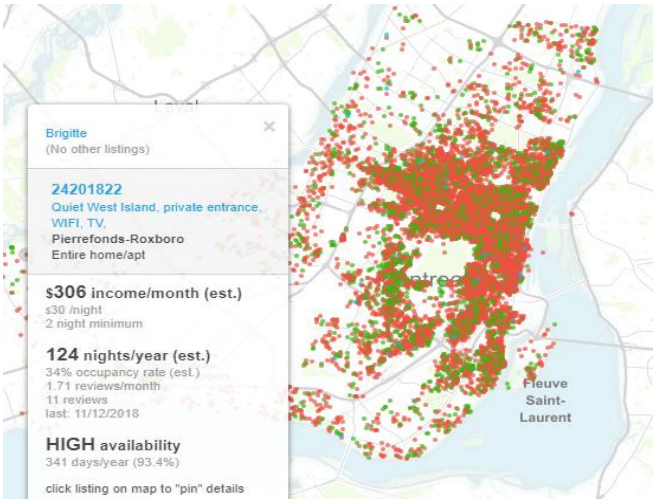
This is the problem, I seek to solve and analyze through Machine learning. This report includes a concise description of all the work I have done, starting with a brief introduction of my previous work that I had submitted as part of course requirement for another course [4] at University of Waterloo.

The flow of the report is as follows:

II. ABOUT THE DATASET

The dataset I am using is **Airbnb Montreal data**, collected from the official Airbnb website – **"Inside Airbnb"** [5], where it has datasets of listings from all the cities around the world.

The data is very fresh and was updated recently in July 2019 so I hope to cover all the recent trends and changers if any. The data present on this website is collected after web scrapping all the information from the official Airbnb websites.



[A]

- **A quick review on the dataset:**

1. The raw dataset consists of 106 explanatory or feature variables and 1 outcome variable- price per night.
2. The dataset is large with 21,104 data points that will allow the models to get trained and test well.
3. The price ranges from $10 to $12,960 a night with 14,332 unique hosts listings across 32 neighborhoods of Montreal.

III. LITERATURE REVIEW

There are few analyses done before on Airbnb data from cities such as New York, Toronto, Boston etc [6, 7, 8, 9, 10]. However, this is the only analysis done on Montreal dataset.
Therefore, I didn't know what to expect from the dataset.
Additionally, these analyses were only done through regression however this may be the first time that it is converted into a classification problem to better estimate the price.
For my project, I have included sentiment analysis as well because reviews given by guests might turn out negative or positive about a property. For example, if a listing has very good reviews than the property might turn out to be popular and in demand due to which there may be an increase in price and vice versa.

IV. FORMER DONE WORK AND RESULTS

As a part of my course requirement I had done price predictions on the same data using models such as Linear Regression, Random forest, decision tree and SVR.
In the raw dataset, there are 106 feature variables out of which 10 variables were selected for the predictions which are mentioned in section V.
For simplicity no variable of type text such as Amenities, house rules and reviews were included.
After exploratory data analysis, here are the observations:

1. There was no extreme correlation between any of the feature variables and outcome variable.
2. Highest accommodation provided was for 6 people.
3. The prices are equally distributed across all neighborhoods instead being higher and lower for some based on its vicinity to tourists spots.
4. The outcome variable price was heavily skewed towards the right, which was converted into log for a normalized distribution.

RESULTS:
After considering 10 variables, the highest accuracy achieved was 49%(after cross validation) which is very bad hence, being the motivation for this project.
Linear regression and Random forest performed the best giving MSE around 20.
There seems to be no effect of neighborhood on price which is understandable given that all Montreal is expensive in general.

V. PRESENT WORK

Main objective was to increase the accuracy as much as possible. The following modifications/improvements are done for this course:

1. To increase the accuracy more variables where added to the model.
2. Reviews of the guests might affect the price distribution. Therefore, through sentiment analysis reviews were taken into consideration.
3. More models such as Ridge, Bayesian Ridge, and KNN are tested.
4. Cross validation for each model was done to test their accuracy on unseen data.
5. Further, regression based analysis was converted into Classification analysis to increase the accuracy.
6. Important features were identified to further improve the accuracy so that a classification model is built only using those variables that contribute the maximum.
7. All classification models are evaluated based on confusion matrix.
8. Again, through cross validation, best performing classifier is selected.
9. Final results and business insights are discussed in section XV and XVI.

To improve the accuracy, the following variables (in bold) are added to the analysis:

1. Id: identity of each listing (numeric)(non predictive)
2. neighbourhood_cleansed : names of neighborhood (categorical)
3. property_type: type of property (Apartment, condo)(categorical)
4. room_type: type of room (private room, shared etc)(categorical)
5. accommodates: number of people that can accommodate in a house (numeric)
6. bathrooms: number of bathrooms (numeric)
7. bedrooms: number of bedrooms (numeric)
8. beds: number of beds (numeric)
9. availability_365: availability of the rental throughout the year. (numeric)
10. review_scores_rating: average review score rating of the rental (numeric)
11. **number_of_reviews**: total count of the number of reviews for the rental (numeric)

12. **cleaning_fee:** fee charged for cleaning the house(numeric)
13. **security_deposit:** deposit to be paid before booking(numeric)
14. **reviews/comments:** reviews by guests (text)
15. **cancellation_policy:** strict, hard, flexible (categorical)
16. **reviews per month:** Number of reviews in a month(numeric)
17. **guests_included:** additional people allowed(numeric)
18. **host_is_superhost:** true or false (categorical)
19. Price: outcome variable(numeric)

VI. DATA CLEANING

Major data cleaning was already done before. However, the newly added variables are cleaned.

1. Cleaning fee and security fee, much like price was skewed towards the right. To avoid any bias, they were converted into log.
2. Outliers for all were detected and taken care of. The highest priced listing was $12,960 per night which can be either true or an outlier. However, for this analysis it was dropped. Also, highest security deposit asked was $6690 which was also removed.
3. Security fee had a huge number of 12,141 missing values. Deleting such large number reduced the dataset size so the missing values where replaced by the mean values.
4. Cleaning fee had 6170 missing values which were dropped.
5. Categorical variables are encoded into numeric using Pandas dummy variables.
6. Sentiment analysis required data cleaning of its own which is explained in section VIII.

VII. CORRELATION

Correlation matrix is generated with the newly added feature variables. Surprisingly, cleaning fee, which was missed in the prior data analysis had the highest correlation with the price followed by accommodates and room type. Also, guests included and security deposit showed some correlation.

Hence, it is expected that accuracy will increase after this addition.

price_log	**1.000000**
cleaning_log	**0.613102**
accommodates	**0.575621**
room_type_Entire home/apt	**0.507132**
beds	**0.500760**
bedrooms	**0.474382**
guests_included	**0.429111**
bathrooms	**0.311414**
security_log	**0.252149**

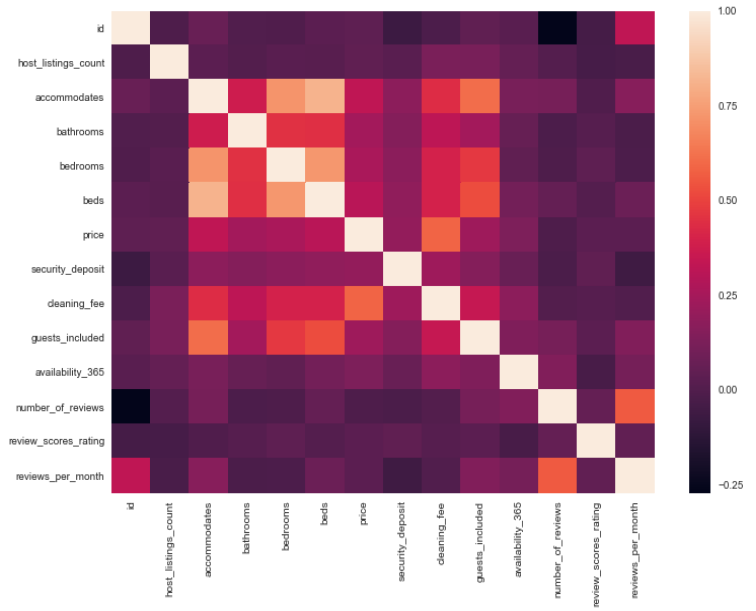


Fig.1. Correlation Matrix for numeric variables

VIII. SENTIMENT ANALYSIS

This part of the project consumed maximum amount of time. Sentiment analysis is done to understand human sentiments/feelings for a particular subject and use it in machine learning. From text, such as text in reviews , complaints etc, sentiment analysis is done to identify and categorize opinions and views expressed so as to better understand attitude or emotions of a person a writing them.

Sentiment analysis can be used in product reviews, comment on social media etc.

In this project, it is used to analyze the sentiments of guests' reviews for the listings. Reviews matter a lot and is an effective to way to check if a property is good or not. Reviews can affect price in a sense that if a property has got many positive reviews it maybe because that particular property is very good and is popular. As the demand increases, the owner might increase the price. Similarly, host might decrease the price if he/she is not getting enough bookings due to bad reviews.

For sentiment analysis, there was a totally separate dataset available for Montreal listings. The dataset was much larger than the main dataset with **4, 35,002 data points**. Sentiment analysis works by giving tokens to each text /reviews and then adding polarity and subjectivity. A positive polarity indicates a positive comment whereas negative indicates a negative comment. It is done by using **Textblob,** a python library. However, to tokenize such large dataset will take lots of time and computational energy. Initially, when I tried doing on just 40,000 reviews it took approximately 45 minutes for just one step of tokenizing and it never worked for the next step of adding polarity. Hence, after lots of trials, I randomly dropped some rows from the dataset and reduced the size to 10,438 reviews, which is the size of our original dataset used for predictions.

I was able to do the analysis for the reduced dataset and generated sentiment for each review given for various listings. Later, the sentiments were merged with the original dataset according to the **listing_id.**

	21	88	106
id	1608777	12338946	19732164
comments	Super friendly and flexable!!\r\nI loved the c...	Nelia is wonderful host. She greeted me very w...	Nelia was a great host and welcoming us to sta...
tok_reviews	(0.42265624999999996, 0.5708333333333333)	(0.3957692307692308, 0.73)	(0.4993589743589744, 0.5988461538461538)
polarity	0.422656	0.395769	0.499359
subjectivity	0.570833	0.73	0.598846

IX. TRAINING AND TESTING

For training and testing, the dataset is divided into **[70:30]** ratio. It means that all models will be trained on 70% of the data and tested on the rest 30%. As the dataset size is large, the models have enough data to get trained so as to generalize better on the unseen data. All the models are than evaluated based on evaluation metrics and than the best performing models are tested again through cross validation.

The training and testing sets are as follows:

X has the feature variables and Y has the outcome variable price.

X_train = (10438, 83)
Y_train = (10438)
X_test = (4474, 83)
Y_train = (4474)

X. BUILDING REGRESSION MODELS

Models are built using 18 variables mentioned in section V.

"**Yellowbrick regression visualizer**" is used to visualize the model performance and results clearly. Also, evaluation metrics are generated for each model using evaluation metrics library from **Sklearn.**

A brief introduction to the plots is as follows:

1) Prediction error plot: It plots the actual outcome variable against values predicted by the model. A comparison between the 45 degree line and the regression line shows the performance of the model.

2) Residual plots: It is simply a plot of prediction errors. A good regression model will have points that are randomly distributed around the horizontal line and the histogram should also be normalized around zero.

1) LINEAR REGRESSION: In supervised learning, it is the most popular algorithm and works very well with numeric data. An outcome variable is predicted using multiple or single independent explanatory variables.

The results for linear regression are as follows:

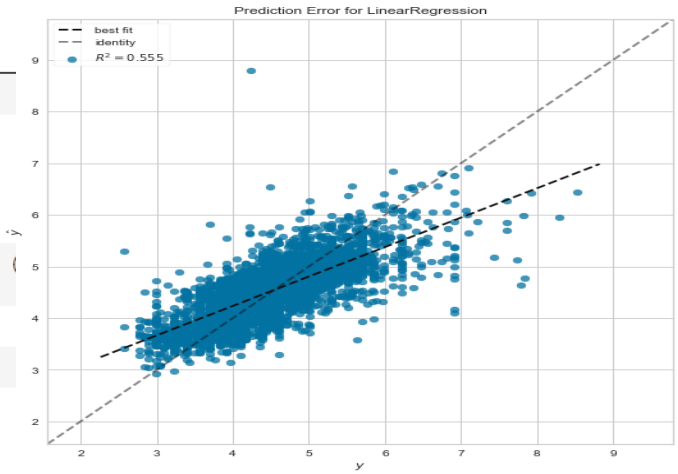


Fig.2. Prediction Error for Linear Regression

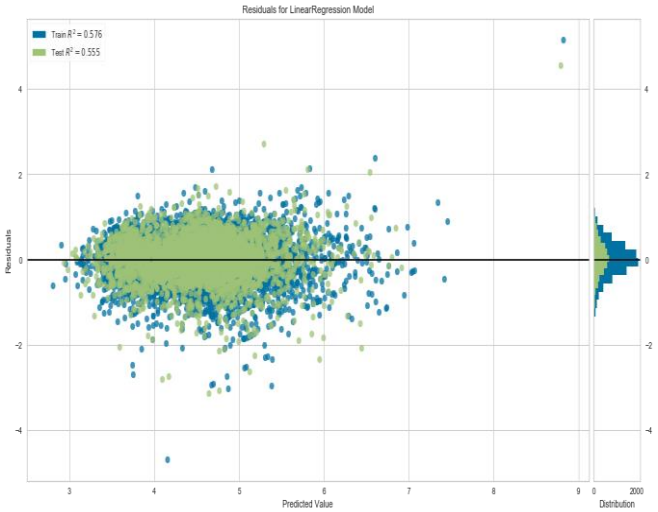


Fig.3. Residuals for Linear Regression Model

2) RANDOM FOREST: In this algorithm, a forest is created with number of decision trees. It performs better in terms of prediction when there are more number of trees.

The results of Random forest are as follows:

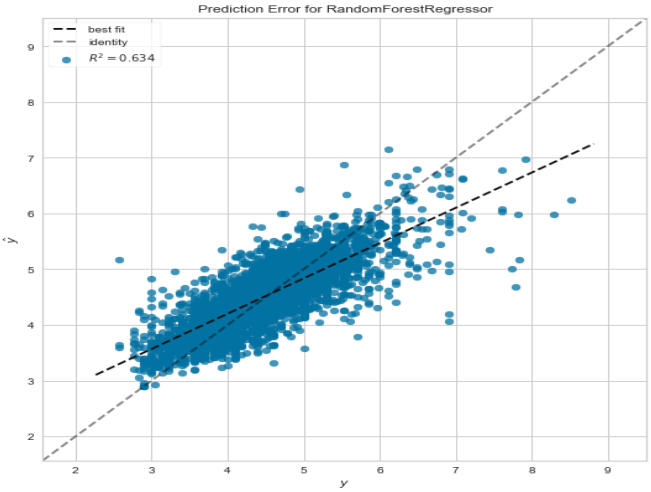


Fig.4. Prediction Error for Random Forest Regression

Fig.5. Residuals for Random Forest Regression

3) DECISION TREE: In decision tree algorithm the tree is generated by selecting the best feature to predict the outcome. Using that node or feature, the tree continuously grows unless we get the least error or there are no more feature variables left. Due to its greedy heuristic nature of continuously growing the tree, it tends to over fit the data.
The results are as follows:

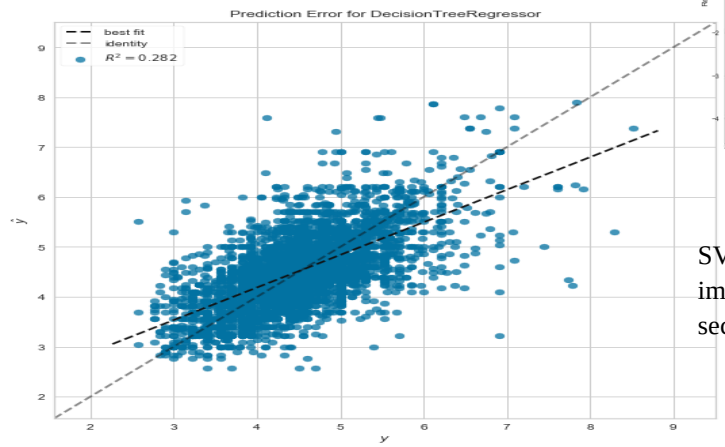


Fig.6. Prediction Error for Decision Tree Regression

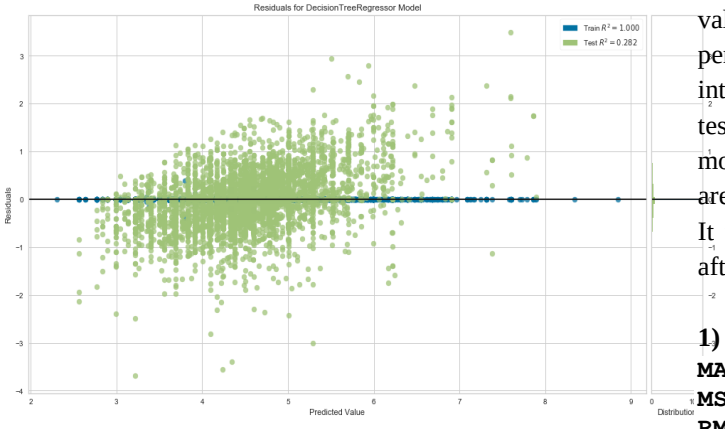


Fig.7. Residuals for Decision Tree Regression

4) KNN: K nearest neighbor makes prediction in a way that it predicts the outcome variable depending upon its K nearest neighbors. For example, is K = 3, than the outcome variable will be assigned a value considering the 3 of its nearest neighbors.
The results of KNN are as follows:

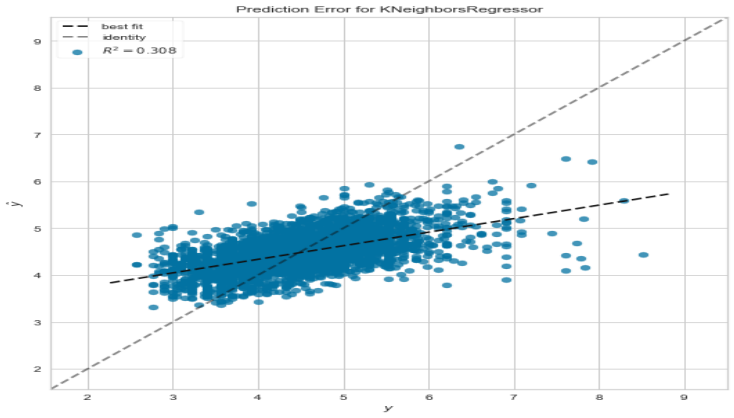


Fig.8. Prediction Error for KNN

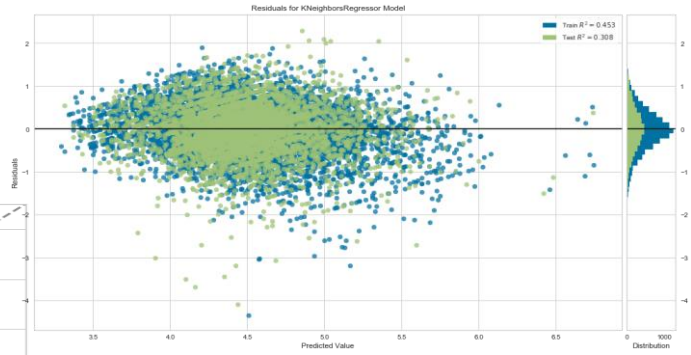


Fig.9. Residuals for KNN

SVR, Ridge, Ridge Bayesian and Lasso were also implemented. Their results are discussed and showed in section XI.

XI. RESULTS

All models are evaluated and validated through cross validation. Cross validation is done to check which model performed the best for the given data. The dataset is divided into k folds where the models are trained on k-1 folds and tested on k^{th} fold. This process is iterated for k times so that model is tested on each fold once. In this project, as there are more variables, cross validation is done with 8 iterations. It was observed that as the number of iterations increased after 8, the accuracy decreased.

1) LINEAR REGRESSION:

```
MAE: 0.3447155041031951
MSE: 0.22143522395514387
RMSE: 0.47056904270802163
Coefficient of Determination: 0.555317791329527
```

CROSS VALIDATION SCORE:

```
Accuracy of every fold in cross validation :[0.57486271 0.59361205 0.5388506  0.51439
```

184 0.60239875 0.53332458 0.56746384 0.497
03521]
**Mean of the validation score: 0.5527424475
785188**
MSE of every fold in cross validation: [0.
15741595 0.19285576 0.22343683 0.22916046
0.21831153 0.23273965 0.2533794 0.232904
Mean of MSE: 0.21752551112166219

2) RANDOM FOREST

MAE is 0.3113802298704169
**Coefficient of Determination: 0.6341835486
002143**
Root Mean Squared Error: 0.426805583364520
Mean Squared Error: 0.18216300599112864

CROSS VALIDATION SCORE:

**Accuracy of every fold in cross validation
:** [0.57486271 0.59361205 0.5388506 0.51439
184 0.60239875 0.53332458
0.56746384 0.49703521]
Mean of the validation score: 0.5527424475
MSE of every fold in cross validation: [0.
15741595 0.19285576 0.22343683 0.22916046
0.21831153 0.23273965 0.2533794 0.232904
Mean of MSE: 0.21752551112166219

3) DECISION TREE

Mean Absolute Error: 0.43564331008562657
Mean Squared Error: 0.35773478700869155
Root Mean Squared Error: 0.598109343689506
**Coefficient of Determination: 0.2816034758
8823353**

CROSS VALIDATION SCORE:

**Accuracy of every fold in cross validation
:** [0.052507 0.27510441 0.25737279 0.2145
9258 0.30617688 0.21380927 0.29134416 0.02
952259]
Mean of the validation score: 0.205053710
MSE of every fold in cross validation: [0.
35082903 0.34400697 0.3598189 0.37063694 0
.38095853 0.39208784 0.41513013 0.4493924
Mean of MSE: 0.38285759748565373

4) KNN

Mean Absolute Error: 0.4388522563403552
Mean Squared Error: 0.3443976556743703
Root Mean Squared Error: 0.586854032681356
**Coefficient of Determination: 0.3083868616
2579587**

CROSS VALIDATION SCORE:

**Accuracy of every fold in cross validation
:** [0.20764569 0.34630828 0.31449581 0.3069
4417 0.34155514 0.2555207 0.2537903 0.19
Mean of the validation score: 0.2780471157
MSE of every fold in cross validation: [0.
2933857 0.31021641 0.33214157 0.32705586 0
.36153333 0.37128558 0.43712915 0.371322
Mean of MSE: 0.3505087790924355

It is clearly seen from the results that Random forest performed the best and gave highest accuracy and lowest MSE followed by Ridge, Bayesian ridge, linear regression and SVR.

As expected decision tree gave the worst accuracy. From the residual plot it is seen that it gave good accuracy on training data but failed to perform on unseen data during testing.

The below table summarizes the R2 score and MSE for all the models.

The left table shows the results from my previous analysis and right table shows the results from analysis done after including 8 more variables.

Lasso is not considered because unacceptable results.

	MSE
Linear Regression	0.202012
Decision Tree	0.343124
KNN	0.292306
SVM	0.244387
Random Forest Regressor	0.195555

	MSE
Linear Regression	0.221435
SVR	0.290851
Decision Tree	0.362825
KNN	0.344398
Random Forest	0.182163
Ridge	0.221049
Bayesian Ridge	0.221053
Lasso	0.479412

Table 1: MSE results of previous (left) and present analysis(right)

	R2 score
Linear Regression	0.537620
Decision Tree	0.223208
KNN	0.330948
SVM	0.440627
Random Forest Regressor	0.552399

	R2 score
Linear Regression	0.555318
SVR	0.415919
Decision Tree	0.275548
KNN	0.308387
Random Forest	0.634184
Ridge	0.556093
Bayesian Ridge	0.556086
Lasso	0.037253

Table 2: R2 scores of previous (left) and present analysis (right)

XII. FEATURE IMPORTANCE USING XGBOOST

XGBoost stands for extreme gradient boosting and it has been recently popular open source software for accurate implementation of gradient boosting machines [11]. It is used for model performance and high computational speed. It offers several features for algorithm enhancement and model tuning. [11]. It is known to give accurate results very quickly and without any computational load on the computer. To try something new, this model has been implemented in this project to get a baseline for accuracy. Additionally, it gives important features that explain our outcome variable more accurately. Not all variables are important to make predictions. Some variables are useless, make no contribution towards the prediction and only increases computational load. Hence, to get better results in

classifications only the important features will be selected and used. Out of 83(after encoding categorical variables) features that were used in regression, only 14 selected important features are used in classification.

Results for XGBoost:

Training MSE: 0.1772
Validation MSE: 0.1964
Training r2: 0.6389
Validation r2: 0.6055

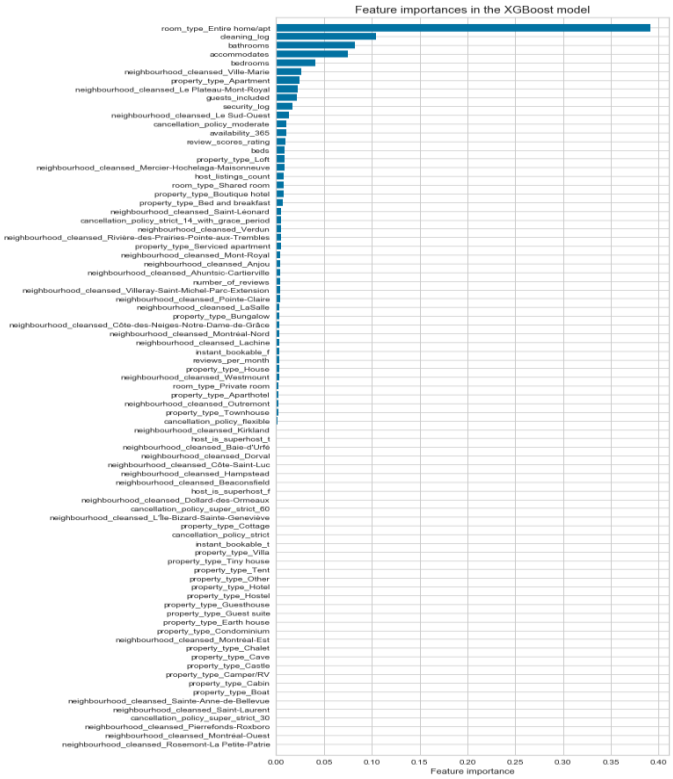


Fig.10. Important Features

XIII. CLASSIFICATION

Just by regression, we cannot estimate if a model is performing good or not. Just on the basis of R2 score and MSE score, we cannot be confident that a model will give the same accurate results when tested on totally unseen data. Hence, regression models are turned into classification so as to better measure their performance. For classification, evaluation metrics will be accuracy, recall, precision and F-score. Also, confusion matrix is generated to check if the model is perfect fit for the data or not.

XIII. I. EVALUATION METRICS FOR CLASSIFICATION

1) Confusion Matrix: It is an NxN matrix where N is the number of classes a model has to predict from. The correct and incorrect predictions made by a model are shown against the target variable in the data.

Model/ Target	Positive	Negative
Position	a	b
Negative	c	d

A confusion matrix has 4 parts:

a) Accuracy: it is the ratio between correct predictions and the total predictions made by a model.

It is calculated as: **Accuracy = (a+d)/(a+b+c+d)**

b) Precision: it is the ratio of true positives and the total positive classes that were predicted.

Precision: = (d/a+c)

c) Sensitivity/Recall: it is the ratio of actual true positives that are identified correctly.

Recall = a/(a+c)

d) Specificity: it is the ratio of actual negatives that are correctly identified.

Specificity = d/(b+d)

e) ROC plot: it shows how much variance a classifier has covered of the data. The wider the curve, better the classifier has performed.

XIII.II. CONVERSION OF OUTCOME VARIABLE

In classification, instead of predicting a numeric outcome variable, one single class is predicted from given number of classes to choose from. For this, the outcome variable price is converted into 2 classes i.e. a binary classification 0 or 1.

All the prices are split into two sets based on the **median value.** One class can be labeled as **"affordable"** and the other as **"luxurious**". In simple words, all the listings that have a lower or affordable price range can be classified as 'affordable' and those with higher prices is classified as 'luxurious'.

The median calculated for the prices is **4.49,** therefore all the values below or equal to 4.49 will be class affordable and values greater than that will be class luxurious.

XIV. TRAINING AND TESTING

As done in regression analysis, for classification, **data is split in to 75: 25 ratios.**

XIV.I. CLASSIFICATION MODELS

Many models were tested but the best performing ones are as below:

1) LOGISTIC REGRESSION

When there is more than one independent variable in a dataset, a logistic regression is used. The outcome variable in this case is a discrete value rather than a continuous value.

The results and ROC plot of Logistic regression is as follows:

The accuracy of Logistic Regression is 0.8684 21052631579
Precision: 0.8932767624020888
Recall: 0.9437931034482758
F-score: 0.9178403755868544

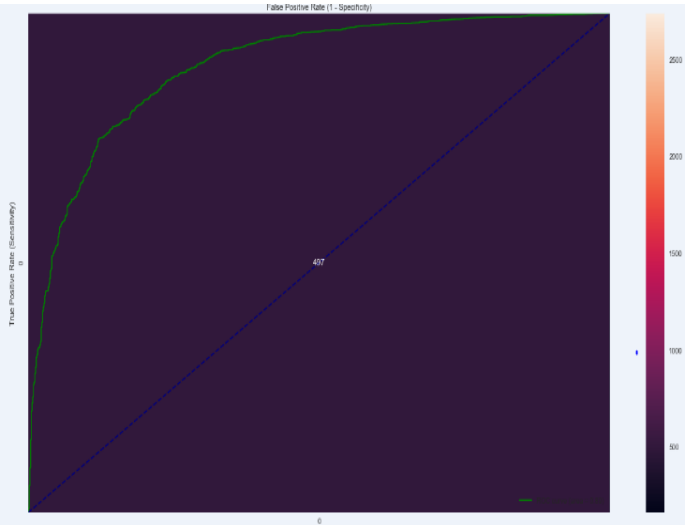


Fig.11. ROC Plot for Logistic Regression

2) RANDOM FOREST

In classification, random forest as the name suggests builds decision trees by randomly selecting features and observation and averaging the results. As mentioned, decision trees over fits the data as they constantly builds the rules by going deeper, however random forests avoids overfitting by building smaller trees from subsets of features. Later, all the sub trees are combined.

The results of Random forest classifier are as follows:

The accuracy of Random Forest Classifier is 0.8748657357679914
Precision: 0.9011206328279499
Recall: 0.9427586206896552
F-score: 0.9214694978092349

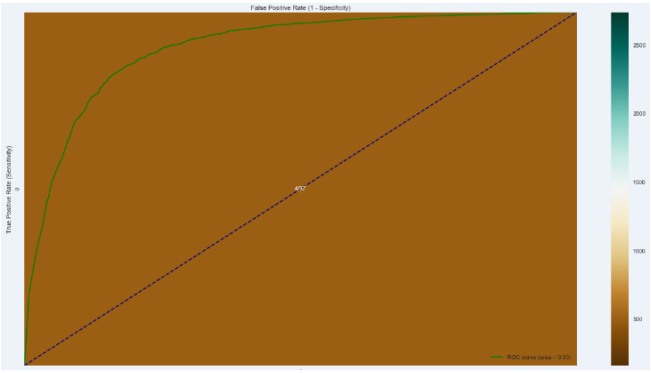


Fig.12. ROC Plot Random forest classifier

3) SVM CLASSIFIER

Support vector machine or SVM is an algorithm that can be used for both regression and classification. Each data point is plotted in N-dimensional space where the coordinate values denote the value of feature variable. Classification is performed by finding a hyper plane that perfectly differentiates the 2 classes.

The results of SVM classifier as follows:

The accuracy of SVM Classifier is 0.778732545 6498388
Precision: 0.7787325456498388
Recall: 1.0
F-score: 0.8756038647342995

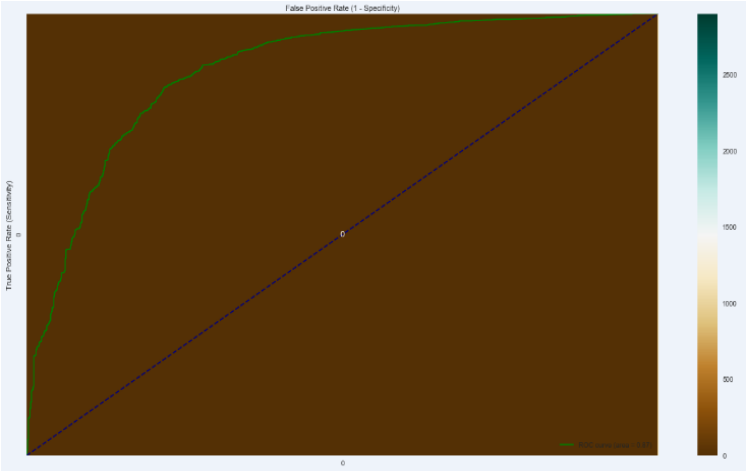


Fig.13. ROC Plot for SVM

XV. CONCLUSION

This project was mainly conceived to increase the poor accuracy I obtained while doing an early data analysis project on this dataset. By just taking 10 variables into consideration, the models performed poorly and the highest accuracy received was 49% by Random forest. This means that only 49% of the variability in the data was justified whereas a huge 51% percent remained unexplained.

This became a motivation for further analysis and improvements.

1. A total of 8 algorithms were implemented using 18 variables. It is evident from the tables 1 and 2 that the accuracy has significantly increased to 63%. Again, Random forest performed the best.
2. These 18 variables caused 63% of the variability in the price of the given listings.
3. Sentiment analysis was done on the reviews provided by the guests and it was found that almost each review was positive however it did not influenced greatly on the price.
4. It was unexpected that cleaning fee would have such high correlation with the price. (More than accommodation, property type, and number of beds) This was an important feature that was missed before and might be the most significant contributor towards variation in price.
5. A unique part of this project was transforming the regression analysis into classification and it increased the accuracy to a large extend. Predicting a particular price can be difficult but by converting prices into ranges it gives a more practical approach to a host to identify in which category his/her property falls.
6. Random forest classifier gave **accuracy of 87%** and **precision of 90%** whereas Logistic regression followed closely with **86% accuracy and 89% of precision**. In classification, **precision is more important than accuracy.** It is more important that a classifier predicts the correct class, so even if accuracy is low but precision is high, the classifier is considered good and perfect fit for the data.
7. Still, there seems to be lots of scope of improvement in this project and much more variance in the dataset can be explained further by taking **seasonality and calendar data** in to consideration. Prices may increase or decrease depending upon the time of the year. For Example, during Christmas and other holidays, it is obvious that there will be higher prices.
8. Also, the listings come with **house rules** such as "No pets allowed" and "no smoking" and various amenities provided which can be further examined. This data is purely text with lots of categories. To avoid complexity, these texts were not considered in this analysis.

XVI. SOME BUSINESS INSIGHTS

Overall, through this analysis**, 63% of variation** in the prices of Montreal Airbnb listings was explained. **MAE of $31.13 and MSE of $18.21** were successfully predicted using random forest. Through this analysis, both hosts and guests can get an insight about what is the optimal price that should be paid and charged for a given property. New hosts' can get an idea about which category (affordable or luxurious) their house belongs to based on their property features. They can be confident in demanding the price their property deserves. Similarly, sometimes few properties are falsely overcharged by the hosts in order to make more profit. Guests can identify such listings and avoid obliging to unfair prices.

REFERENCES

[A] Leaflet, @openstreetmap, date: November 12, 2018 <http://insideairbnb.com/montreal/?neighbourhood=&filterEntireHomes=false&filterHighlyAvailable=false&filterRecentReviews=false&filterMultiListings=false>

1) 1. Sraders, Anne (September 24, 2018). "How Does Airbnb Work for Hosts and Travelers?" TheStreet, Inc. Retrieved April 12, 2019.

2) 2 Airbnb Fast facts, < https://press.airbnb.com/fast-facts/>

3) Airbnb Montreal price predictions, Ruta Patel, Nitheesha Reddy, Electrical and computer department, University of Waterloo, Canada.

4) MSCI 623, Spring 2019, University of Waterloo.

5) Inside Airbnb data, < http://insideairbnb.com/get-the-data.html>

6) Philip Mohun, Making models – Airbnb price predictions, date : 27th February, 2018 <https://medium.com/datadriveninvestor/making-models-airbnb-price-prediction-data-analysis-15b9af87c9d8>

7) Laura Lewis, Predicting Airbnb prices with machine learning and deep learning, date: 22nd May, 2019 < https://towardsdatascience.com/predicting-airbnb-prices-with-machine-learning-and-deep-learning-f46d44afb8a6>

8) Dmytro Lakubovskyi, Digging into Airbnb data : reviews, sentiments, superhosts, price predictions, date: 1st October, 2018 < https://towardsdatascience.com/digging-into-airbnb-data-reviews-sentiments-superhosts-and-prices-prediction-part1-6c80ccb26c6a>

9) FaisalAl-Tameemi, Airbnb listings data- Toronto date: 7th November, 2018 < https://medium.com/datadriveninvestor/airbnb-listings-analysis-in-toronto-october-2018-2a5358bae007>

10) Samuel klam, Airbnb pricing prediction, date: 22nd January, 2018, <https://airbnb-pricing-prediction.herokuapp.com/ >

11)<https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html