Machine Learning Prediction of New York Airbnb Prices

Ang Zhu*
Computer Science
University of Waterloo
Waterloo, Canada
a7zhu@edu.uwaterloo.ca

Rong Li*
WeCloudData Inc
Toronto, Canada
rong.li@weclouddata.com

Zehao Xie*
Mathematics and Economics
University of British Columbia
Vancouver, Canada
zehao.xie@alumni.ubc.ca

***Abstract*—The sharing economy emerged just twenty years ago. It was hard to envision its importance for many industries back then, but today, it is closely incorporated into our daily lives. Airbnb is a prime example of this phenomenon; it brought a brand new business model to the hospitality industry. The Airbnb platform has created millions of micro-businesses with a distinctive pricing strategy. Understanding the pricing strategy will provide insights into this new business model.**

This paper analyzes a sample of 48 896 listings in New York City from Airbnb.com, and builds a price prediction model with natural language processing and machine learning techniques. Different methods ranging from linear regression, to a generalized additive model, to a deep neural network, to random forest, to XGBoost and bagging (merging the previous five methods) were explored for the creation of the final prediction model. Of these, bagging, XGBoost and random forest demonstrated the strongest performance with the test data collected from the Airbnb website on January 28th, 2020.

***Index Terms*—Machine Learning; Generalized Additive Model; Deep Neural Network; Random Forest; Airbnb;**

I. Introduction

Hotels long dominated the hospitality industry until sharing business, such as Airbnb, appeared. Founded in 2008, Airbnb introduced a new business model of sharing economy that revolutionized the hospitality industry. It connects owners of empty spaces with travelers seeking temporary accommodation on its digital marketplace [1]. As of 2019, there were over 6 million listings on Airbnb website in about 220 countries and regions, facilitating an average of 2 million stays per night [2].

Unlike the traditional hospitality industry, listings on Airbnb provide more diversified experiences and prices. While the star rating system is primarily used by hotels to develop their pricing strategies, no explicit pricing guideline is available for Airbnb hosts. Hence, pricing becomes more complicated in the context of the sharing economy. However, it is also critical to understand the pricing structure of Airbnb because it drives consumers' decision making as well as stakeholders' profitability [3]. Furthermore, doing so will also help researchers gain profound insights on the sharing economy.

This paper uses regression and other machine learning models trained on the listings in New York City and their attribute information from the Airbnb website, collected by Denis Gomonov and posted on Kaggle [4]. The models analyze the determinants of listing prices and forecast future prices on listings with publicly visible information. The models will provide important information for the pricing strategies of the hosts and other stakeholders, as well as insight into the overall accommodations industry in the context of the sharing economy.

* Equal contribution

II. Data Description

The New York Airbnb dataset includes 16 variables and 38 827 observations. Variables that are irrelevant to the analysis, such as "id" and "host name", were excluded. Additional adjustments were applied to variables such as "name" (listing title) with natural language processing techniques to accommodate model construction. The table below lists the final variables in the dataset:

- neighbourhood_group: neighbourhood of the listing in New York City
- latitude: latitude coordinates
- longitude: longitude coordinates
- room_type: listing space type
- price: price in dollars
- minimum_nights: minimum number of nights
- number_of_reviews: number of reviews of the listing
- reviews_per_month: number of reviews per month of the listing
- calculated_host_listings_count: number of listings per host
- availability_365: number of days per year that listing is available
- sentiment_score: this is calculated based on the name of the listing from the original dataset by the natural language processing (NLP) algorithm `sentimentr` [5]. In `sentimentr`, four kinds of valence shifters, negators, amplifiers (intensifiers), de-amplifiers (downtoners), and adversative conjunctions have very high weights. The score of each sentence is calculated by the frequency and position of these valence shifters.

A heat map of the average price (measured in dollars) of Airbnb listings in New York City is constructed and presented in figure 1, where the brightness of the map indicates different

978-1-7281-8701-3/20/$31.00 ©2020 IEEE
DOI 10.1109/AI4I49448.2020.00007

price ranges. Manhattan, the brightest region on the map, has the highest average price as expected.

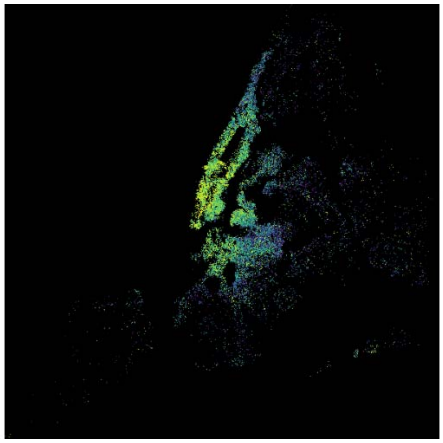

Fig. 1. New York Airbnb heat map

Figure 2, another map of New York City, illustrates the price level with colored dots. The orange dots correspond to economical ($0 to $55), purple dots correspond to mediocre ($55 to $148), dark green dots correspond to expensive ($148 to $403), and blue dots indicate luxury (≥$403). Price levels around the landmarks of New York City, indicated by magenta stars, are either luxury (blue dots) or expensive (dark green dots). Moreover, the geological patterns in the neighbourhoods of Manhattan, south-western Bronx, west Queen, and north-western Brooklyn neighborhoods are much denser than the rest.

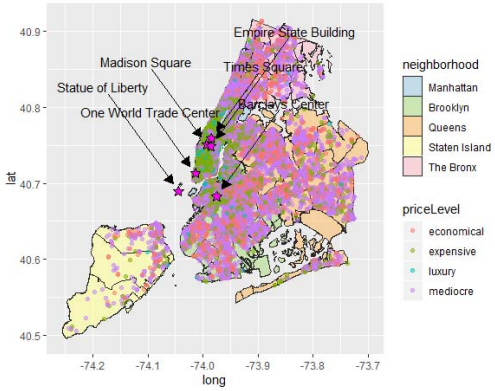

Fig. 2. New York City map

Figure 3 demonstrates the boxplots of log price and the log of number of reviews. Both boxplots are ordered by Bronx, Brooklyn, Manhattan, Queens, and Staten Island. As the graphs indicates, the plots have clear but opposite patterns, where regions with higher prices gain less reviews, and vice versa.

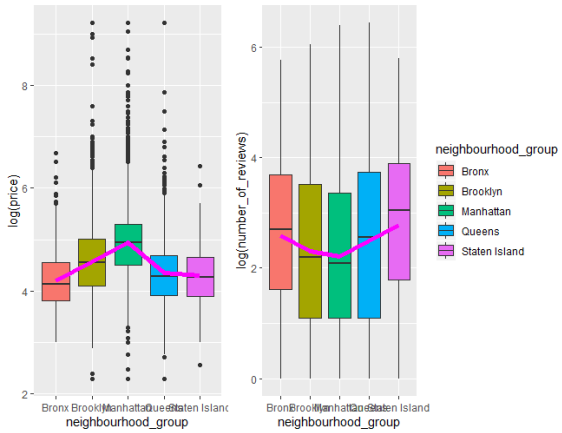

Fig. 3. Boxplots of log(price) and log(number_of_reviews) with different neighbourhoods. Airbnbs in Manhattan have higher prices but obtaining the least reviews.

III. Methodology

The baseline model is a linear regression model (LM) with selected features as explanatory variables. A generalized additive model and several machine learning methods were considered in comparison to identify the optimal prediction model. The implemented models were introduced in the following sections.

A. Regression

The most straightforward approach of this analysis is to use a linear regression model with the response variable Y ('log(price)') and mean structure μ:

$$Y = \mu + r$$

where $\mu = [\mu_i], i = 1, ..., n$ and n is the number of observations. To reduce possible risk of multicollinearity, the "step wise BIC" method is implemented to select variables [6] [7]:

$$BIC = log(n)k - 2log(\hat{l})$$

where k is the estimated parameters and $\hat{l}$ is the maximized likelihood function. Exercising the variable selection process filtered out explanatory variables, "population", "GDP", and "Region". The final model is presented as follows:

$$\begin{aligned} Y \quad = \quad & longitude + availability365 + roomType + \\ & neighbourhoodGroup + latitude + \\ & minimumNights + numberOfReviews + \\ & sentimentScore + r \qquad (0) \end{aligned}$$

where $r_i \overset{iid}{\sim} N(0, \sigma^2)$, and the residual sum of squares is

$$RSS_1 = (Y - X\beta)^T (Y - X\beta)$$

where $\mathbf{X}$ is the column space of independent variables.

Since there are two factor variables, "roomType" and "neighbourhoodGroup", more complex models should be considered, such as the following:

$$\begin{aligned} Y &= roomType \times (longitude + ...) + \\ & \quad r \qquad (1) \\ Y &= neighbourhoodGroup \times \\ & \quad (longitude + ...) + r \qquad (2) \\ Y &= (neighbourhoodGroup + roomType) \times \\ & \quad (longitude + ...) + r \qquad (3) \\ Y &= (neighbourhoodGroup \times roomType) \times \\ & \quad (longitude + ...) + r \qquad (4) \end{aligned}$$

The ANOVA test [8] results indicate that model (4) is the most appropriate. More details could be added to the model, such as involving interaction terms, for numerical variables. However, it appears that the majority of the signal is captured by the interactions of categorical variables. Hence, adding further interactions with the risk of overfitting and sacrificing the residual degree of freedom is deprecated.

The R^2 of the model (4) is 0.5581, which indicates 55.81% of the variability of $\mathbf{Y}$ that is predictable from the selected independent variables. The mean squared error is 0.1948.

This linear regression model is under a strong constraint such that each numerical predictor is assumed to be linear with the predictand. A more generalized form of the model can be adopted to release this assumption, where:

$$\begin{aligned} Y &= (neighbourhoodGroup \times roomType) \times \\ & \quad (te(longitude) + te(availability365) + \\ & \quad te(latitude) + te(minimumNights) + \\ & \quad te(numberOfReviews) + \\ & \quad te(sentimentScore)) + r \qquad (5) \end{aligned}$$

This model is widely known as the generalized additive model (GAM) [9] where $te()$ are the tensor product smooths [10]. Tensor product smooths are used instead of conventional product smooths to rescales the X and minimize possible scaling issues. Its penalized residual sum of squares is defined as follows:

$$RSS_2 = (Y - N\gamma)^T(Y - N\gamma) + \lambda\gamma^T\Omega_N\gamma$$

where $N = [N_{ij}]$ is an $n \times n$ matrix whose (i, j) element, $N_{ij} = N_j(x_i)$, is the jth natural cubic spline basis function [11] evaluated by X column space and Ω_N is the penalized term. The solution of $\hat{\gamma}$ is

$$\hat{\gamma} = (N^TN + \lambda\Omega_N)^{-1}N^TY$$

Smoothing parameter λ is related to degree of freedom chosen so that the target function can be optimized to pursuit minimal generalized cross validation (GCV):

$$\text{GCV} = \frac{nD}{(n - df_2)^2}$$

where D is the deviance. After fitting the "mgcv" with package [12] [13], $R^2 = 0.591$ is obtained with an increase of 4 percent and GCV $= 0.1816$, which indicates that the model is appropriate.

B. Machine Learning Methods

Three other machine learning techniques, deep neural network, random forest, and XGBoost, are applied to achieve the best results:

1) Deep Neural Network (DNN): In a deep neural network, the input layer of the network has the same dimension as the column space of the independent variables, *X*, which is 11. The output layer is a single node, *Y*, being the predicted price given an observation of *X* [14]. Considering the limited feature space, only 2 middle layers with 128 and 32 nodes respectively, with a drop-out rate [15] of 0.2 for both layers were used to reduce the possibility of overfitting. The "relu" activation function is applied to every middle-layer node to introduce non-linearity. The network was trained using a back-propagation algorithm to minimize the mean absolute error between the predicted price and the corresponding ground truth. In our fit, the GCV score is 0.1792.

2) Random Forest (RF): Random forest algorithm is one of the ensemble learning methods that combines predictions from decision trees [16]. It is a bagging technique that makes each tree run independently and then aggregates the output at the end without preference to any tree. Compared to decision trees, random forests are less exposed to over-fitting the training set. The GCV score is 0.1745 for random forest, lower than in regression models.

3) XGBoost: XGBoost algorithm is a scalable end-to-end tree boosting system that introduces several new applications for approximate tree learning and a weighted quantile sketch for efficient proposal calculation [17]. The algorithm adds a regularization term to the target function of the model that serves as a penalty. It controls the complexity of the model and helps to determine the stopping point when training at a specific step of tree learning. The final GCV score is 0.1702, the best of all the models tested. The following table illustrates some summary statistics of each model, From this table, we can tell that the XGBoost model performs the best on both R^2 and GCV; the performance of linear regression, on the other hand, is the worst of the models tested.

TABLE I
SUMMARY STATISTICS OF EACH MODEL

	LM	GAM	DNN	RF	XGBoost
R^2	55.8%	59.1%	60.4%	61.2%	61.8%
GCV	0.195	0.182	0.179	0.175	0.17

IV. Cross-validation

The mean squared error obtained above is underestimated, as the training model constructed is based on training dataset. Cross-validation is one of the most widely used methods to estimate the true error [18]. It helps to test the robustness of a model's prediction on an new dataset that is not involved in model training. Hence, reducing the possibility of overfitting and bias. A 5-fold cross-validation was performed on each of the five models (linear regression, generalized additive models, deep neural network, random forest, and XGBoost), the boxplots of the mean squared errors are displayed in figure 4. It appears that the XGBoost model is the most

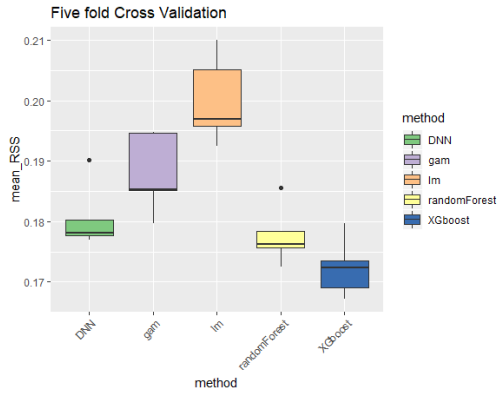

Fig. 4. Cross validation on five models

effective of the models with the lowest mean of residual sum of squares, followed by random forest and DNN, GAM and linear regression, which has the worst performance. Additionally, compared with model GAM and LM, XGBoost model has a smaller variance; compared with model DNN and random forest, XGBoost model also does not have any outliers that indicate the performance of XGBoost in each trail is very robust.

Bootstrap aggregating, also known as bagging, is believed to be the best practice to reach the best results.

> "Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor." - Leo Breiman [19]

There are many benefits to bagging models. Of these, the most valuable advantage is that there is not a dominant predictor to outperform other models, which yields more robust results. Figure 5 displays the predictions on a validation set. The black lines represent $y = x$ and the red lines represent the fitted lines of the true value versus the predicted value. Almost every model has an identical fit.

V. Prediction

To further assess the model performance, 1 545 new records were obtained from the Airbnb web pages dated on January 28, 2020 with the following attributes: neighbourhood group, room type, location coordinates, number of reviews and titles (transformed to sentiment score). The information of available

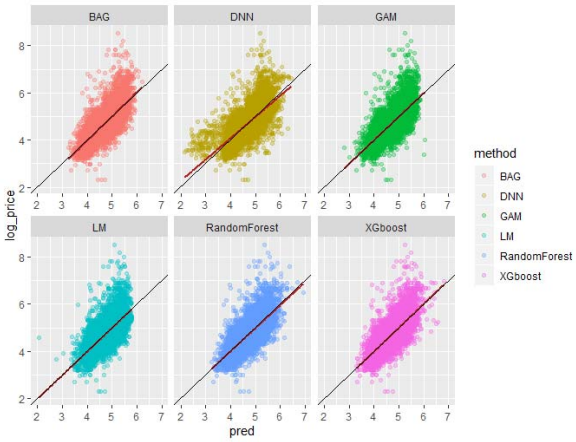

Fig. 5. Prediction on validation set

days out of 365 and minimum required nights were unavailable for the collection. Hence, the medians of corresponding variables from the training set were used instead. The model predictions are presented in figure 6, indicating that

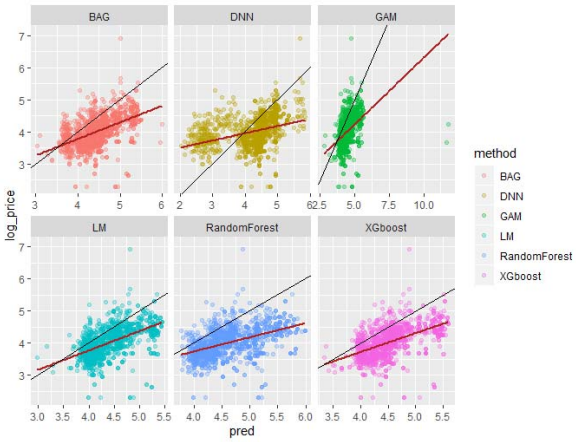

Fig. 6. Cross-validation on five models

all prediction methods tend to overestimate the actual price. The difference between the black lines and the red lines can be seen as a proxy metric for model assessment, where the smaller the angle is, the better the prediction. Bagging, linear regression, random forest and XGBoost appear to out-perform the other models. In contrast, the predictions of GAM seem to be distorted by several outliers. DNN performs the worst of all. The main possible reasons for the observed over-prediction are as follows: 1. Overfitting seems to be present in the models of DNN and GAM. 2. Missing values of two main variables in new data collection could reduce the models' predictive power. 3. January is in a season with relatively low accommodation demand, resulting in unexpected price adjustments to attract more business; such behavior is beyond the modeling scope.

VI. Conclusion

The GAM model, which relieves the linear combination of independent variables, may increase the model's flexibility and enhance its performance. The deep neural network uses a back propagation algorithm [20] to train models until they convergence. However, a model with high complexity also means a decrease in degrees of freedom, which may cause over-fitting and impact the robustness of the model.

For this topic, the bagging model is strongly recommended. A combination of different models brings the benefits of multiple models into one, while decreasing the dominating power of one specific model.

References

[1] R. Botsman and R. Rogers, "Product service systems," in *What's Mine is Yours*. New York, NY, USA: HarperCollins, 2010, pp. 106–108.

[2] "Fast facts," *Airbnb Newsroom,* Dec. 2019. [Online]. Available: https://news.airbnb.com/about-us/. [Accessed Dec. 2, 2019]

[3] C. Gibbs, D. Guttentag, U. Gretzel, J. Morton, and A. Goodwill, "Pricing in the sharing economy: a hedonic pricing model applied to Airbnb listings," J. Travel Tour. Mark., pp. 1–11, Apr. 2010.

[4] D. Gomonov, "New York City Airbnb Open Data," *Kaggle,* 2019. [Online]. Available: https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data. [Accessed Dec. 15, 2019]

[5] T. W. Rinker, "Sentimentr: calculate text polarity sentiment", version 2.7.1. (2019). [Online]. Available: http://github.com/trinker/sentimentr. [Accessed Dec. 15, 2019]

[6] N. Draper and H. Smith, *Applied Regression Analysis*, 2nd ed., Hoboken, NJ, USA: Wiley, 1981.

[7] G. E. Schwarz, "Estimating the Dimension of a Model," Ann. Statist., vol. 6, pp. 461–464, Mar. 1978.

[8] E. R. Girden, *Anova: Repeated Measures*, Thousand Oaks, CA, USA: Sage, 1992.

[9] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*, Boca Raton, FL, USA: CRC, Jun. 1990.

[10] S. N. Wood, "Low-rank scale-invariant tensor product smooths for generalized additive mixed models," Biometrics, vol. 62, pp. 1025–1036, Dec. 2006.

[11] C. H. Reinsch, "Smoothing by spline functions," Numer. Math., vol. 10, pp. 177–183, Oct. 1967.

[12] S. Wood, "mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation," 2019. [Online]. Available: https://CRAN.R-project.org/package=mgcv. [Accessed Dec. 17, 2019]

[13] S. Wood, "mgcv: GAMs in R," 2019. [Online]. Available: https://people.maths.bris.ac.uk/ sw15190/mgcv/tampere/mgcv.pdf. [Accessed Dec. 17, 2019]

[14] D. F. Specht, "A general regression neural network," IEEE Trans. Neural Netw., vol. 2, pp. 568–576, Nov. 1991.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, pp. 1929–1958, Jun. 2014.

[16] A. Liaw and M. Wiener, "Classification and regression by randomForest," R news, vol. 2, no. 3, pp. 18–22, 2002.

[17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. on Knowl. Discov. and Dat. Mining, New York, NY, USA: ACM, 2016, pp. 785–794.

[18] M. W. Browne and R. Cudeck, "Single sample cross-validation indices for covariance structures," Multivar. Behav. Res., vol. 24, no. 4, pp. 445–455, 1989.

[19] L. Breiman, "Bagging predictors," Mach. Learn., vol. 24, pp. 123–140, Aug. 1989.

[20] S. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," in IEEE Trans. Neural Netw., vol. 3, pp. 801–806, Sep. 1992.