Airbnb Price Estimation

Liubov Nikolenko
SUNet ID: liubov

Hoormazd Rezaei
SUNet ID: hoormazd

Pouya Rezazadeh
SUNet ID: pouyar

Project Category: General Machine Learning
gitlab.com/hoorir/cs229-project.git

Final Report

1 Introduction

Pricing a rental property on Airbnb is a challenging task for the owner as it determines the number of customers for the place. On the other hand, customers have to evaluate an offered price with minimal knowledge of an optimal value for the property. This project aims to develop a price prediction model using a range of methods from linear regression to tree-based models, support-vector regression (SVR), K-means Clustering (KMC), and neural networks (NNs) to tackle this challenge. Features of the rentals, owner characteristics, and the customer reviews will be used to predict the price of the listing.

2 Related Work

Existing literature shows that some studies focus on non-shared property purchase or rental price predictions. In a CS229 project, Yu and Wu [1] tried to implement a real estate price prediction using feature importance analysis along with linear regression, SVR, and Random Forest regression. They also attempted to classify the prices into 7 classes using Naive Bayes, Logistic Regression, SVC and Random Forest. They declared a best RMSE of 0.53 for their SVR model and a classification accuracy of 69% for their SVC model with PCA. In another paper, Ma et al. [2] have applied Linear Regression, Regression Tree, Random Forest Regression and Gradient Boosting Regression Trees to analyzing warehouse rental prices in Beijing. They concluded that the tree regression model was the best-performing model with an RMSE of 1.05 CNY/m^2-day

Another class of studies which are more pertinent to our project, inspect the hotels and sharing economy rental prices. In a recent work, Wang and Nicolau [3] have studied price determinants of sharing economy by analyzing Airbnb listings using ordinary least squares and quantile regression analysis. In a similar study, Masiero et al. [4] use quantile regression model to analyze the relation between travel traits and holiday homes as well as hotel prices. In a simpler work, Yang et al. [5] applied linear regression to study the relationship between market accessibility and hotel prices in Caribbean. They also include the user ratings and hotel classes as contributing factors in their study. Li et al. [6] also study a clustering method called Multi-Scale Affinity Propagation and apply Linear Regression to the obtained clusters in an effort to create a price prediction model for Airbnb in different cities. They take the distance of the property to the city landmarks as the clustering feature.

This project has tried to further the experimented methods from the literature by focusing on a variety of feature selection techniques, implementing Neural Networks, and leveraging the cus-

tomer reviews through sentiment analysis. The authors were unable to find the last two mentioned undertakings in the existing literature.

3 Dataset

The main data source for this study is the public Airbnb dataset for New York City[1]. The dataset includes 50,221 entries, each with 96 features. See figure 1 for the geographic distribution of listing prices.

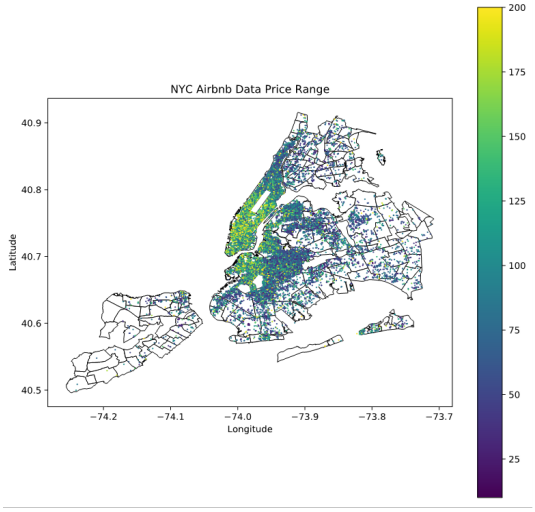

Figure 1: Geographic spread of price labels (with filtered outliers)

For the initial prepossessing, our team has inspected each feature of the dataset to (i) remove features with frequent and irreparable missing fields or set the missing values to zero where appropriate, (ii) convert some features into floats (e.g. by getting rid of the dollar sign in prices), (iii) change boolean features to binaries, (iv) remove irrelevant or uninformative features, e.g. host picture url, constant-valued fields or duplicate features, and (v) convert the 10 categorical features in the final set, e.g. 'neighborhood name' and 'cancellation policy,' into one-hot vectors. In addition, the features were normalized and the labels were converted into $\log$ of the price to mitigate the impact of the outliers in the dataset. The team has split the data into train (90%), validation (5%), and test (5%) sets. Since the dataset is relatively large, 10% of the data was deemed sufficient for testing and validation sets.

3.1 Sentiment Analysis on the Reviews

The reviews for each listing were analyzed using TextBlob[2] sentiment analysis module. This method assigns a score between -1 and 1 to each review and the scores are averaged across each listing. The final scores for each listing was included as a new feature in the model.

3.2 Feature Selection

After data preprocessing, the feature vector contained 764 elements which was deemed excessive and, when fed to models, resulted in a high variance of error. Consequently, several feature selection techniques were used to find the features with the most predictive values to both reduce the model variances and reduce the computation time. Based on prior experience with housing price estimation, the first effort was manual selection of features to create a baseline for the selection process.

[1] `http://insideairbnb.com/get-the-data.html`

[2] `https://textblob.readthedocs.io/en/dev/index.html`

The second method was tuning the coefficient of linear regression model with Lasso regularization trained on the train split, from which the model with the best performance over validation split was selected. Second set of features consisted of 78 features with non-zero values based on this method.

Finally, lowest p-values of regular linear regression model trained on train split were used to choose the third set of features. Selection was bound by the total number of features to remain less than 100. The final set of features were those for which linear regression model performed best on validation split.

The performance of manually selected features as well as p-value and Lasso feature selection schemes were compared using the R^2 score of the linear regression models trained on the validation set. All models outperformed the baseline model, which used the whole feature set, and the second method, Lasso regularization, yielded the highest R^2 score.

4 Methods

Linear Regression was set as a baseline model on the dataset using all of the features as model inputs. After selecting a set of features using Lasso feature selection, several machine learning models were considered in order to find the optimal one. All of the models except neural networks were implemented using scikit-learn library [7]. The neural network model was implemented with the help of Keras module [8].

4.1 Ridge Regression

Linear Regression with L_2 regularization adds a penalizing term to the squared error cost function in order to help the algorithm converge for linearly separable data and reduce overfitting. Therefore, Ridge Regression minimizes $J(\theta) = ||y - X\theta||_2^2 + \alpha||\theta||_2^2$ with respect to θ, where X is a design matrix and α is a hyperparameter. Since the baseline models were observed to have high variance Ridge Regression seemed to be an appropriate choice to solve the issue.

4.2 K-means Clustering with Ridge Regression

In order to capture the non-linearity of the data, the training examples were split into different clusters using k-means clustering on the features and the Ridge Regression was run on each of the individual clusters. The data clusters were identified using the following algorithm:

Algorithm 1 K-means Clustering

Initialize cluster centroids $\mu_i, ..., \mu_k$ randomly
repeat
  Assgin each point to a cluster: $c^{(i)} = \arg\min_j ||x^{(i)} - \mu_j||_2^2$
  For each centroid: $\mu_j = \frac{\sum_{i=1}^m 1\{c^{(i)}=j\}x^{(i)}}{\sum_{i=1}^m 1\{c^{(i)}=j\}}$
  Calculate the loss function for the assignments and check for convergence: $J(c, \mu) = \sum_{i=1}^m ||x^{(i)} - \mu_{c^{(i)}}||_2^2$
until convergence

4.3 Support Vector Regression

In order to model the non-linear relationship between the covariates, the team employed support vector regression with RBF kernel to identify a linear boundary in a high-dimensional feature space. Using the implementation based on LIBSVM paper [9] the algorithm provides a solution for the following optimization problem:

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^m \xi_i + C\sum_{i=1}^m \xi_i^*, \text{subject to}$$

$$w^T\phi(x^{(i)}) + b - y^{(i)} \leq \epsilon + \xi_i,$$

$$y^{(i)} - w^T\phi(x^{(i)}) - b \leq \epsilon + \xi_i^*,$$
$$\xi_i, \xi_i^* \geq 0, i = 1, ..., m,$$

where $C > 0, \epsilon > 0$ are given parameters. This problem can be converted into a dual problem that does not involve $\phi(x)$, but involves $K(x, z) = \phi(x)\phi(z)$ instead. Since we are using RBF kernel, $K(x, z) = \exp\left(\frac{||x-z||^2}{2\sigma^2}\right)$.

4.4 Neural Network

Neural network was used to build a model that combined the input features into high level predictors. The architecture of the optimized network had 3 fully-connected layers: 20 neurons in the first hidden layer with relu activation function, 5 neurons in the second hidden layer with relu activation function, and 1 output neuron with a linear activation function.

4.5 Gradient Boost Tree Ensemble

Since the relationship between the feature vector and price is non-linear, regression tree seemed like a proper model for this problem. Regression trees split the data points into regions according to the following formula

$$\max_{j,t} L(R_p) - (L(R_1) - L(R_2)),$$

where j is the feature the dataset is split on, t is the threshold of the split, R_p is the parent region and R_1 and R_2 are the child regions. Squared error is used as the loss function.

Since standalone regression trees have low predictive accuracies individually, gradient boost tree ensemble was used to increase the models' performance. The idea behind a gradient boost is to improve on a previous iteration of the model by correcting its predictions using another model based on the negative gradient of the loss. The algorithm for the gradient boosting is the following [10]:

Algorithm 2 Gradient Boosting

Initialize F_0 to be a constant model
for m = 1, ..., number of iterations **do**
  for all training examples $(x^{(i)}, y^{(i)})$ **do**
    For squared error $R(y^{(i)}, F_{m-1}(x^{(i)})) = -\frac{\partial \text{Loss}}{\partial F_{m-1}(x^{(i)})} = y^{(i)} - F_{m-1}(x^{(i)})$
  end for
  Train regression model h_m on $(x^{(i)}, R(y^{(i)}, F_{m-1}(x^{(i)})))$, for all training examples
  $F_m(x) = F_{m-1}(x) + \alpha h_m(x)$, where α is the learning rate
end for
return F_m

5 Experiments and Discussion

Mean absolute error (MAE), mean squared error (MSE) and R^2 score were used to evaluate the trained models. Training (39,980 examples) and validation (4,998 examples) splits were used to choose the best-performing models within each category. The test set, containing 4,998 examples, was used to provide an unbiased estimate of error, with the final models trained on both train and validation splits. Results for the final models[3] are provided below.

Model Name	train split			test split		
	MAE	MSE	R^2 Score	MAE	MSE	R^2 Score
Linear Reg. (Baseline)	0.2744	0.1480	0.690	96895.82	2.4E13	-5.1E13
Ridge Reg.	0.2813	0.15461	0.6765	0.2936	0.1613	0.6601
Gradient Boost	0.2492	0.1376	0.7121	0.3282	0.1963	0.5864
K-means + Ridge Reg.	0.2717	0.1438	0.6992	0.2850	0.1543	0.6748
SVR	0.2132	0.1067	0.7768	0.2761	0.1471	0.6901
Neural Net	0.2602	0.1316	0.7246	0.2881	0.1570	0.6692

[3]Optimized models to be found at `gitlab.com/hoorir/cs229-project.git`

The outlined models had relatively similar R^2 scores which implicates that Lasso feature importance analysis had made the most impact on improving the performance of the models by reducing the variance. Even after the feature selection, the resulting input vector was relatively large leaving room for model overfitting. This explains why Gradient Boost - a tree-based model prone to high variance - performed worse than the rest of the models despite it not performing the worst on the training set.

Despite expanding the number of features in the feature vector, SVR with RBF kernel turned out to be the best performing model with the least MAE and MSE and the highest R^2 score on both train and test sets (figure 2). RBF feature mapping was able to better model the prices of the apartments which have a non-linear relationship with the apartment features. Since regularization is taken into account in the SVR optimization problem, parameter tuning ensured that the model was not overfitting.

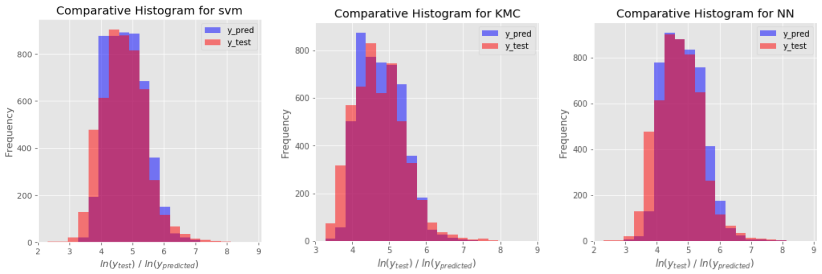

Figure 2: Comparative histograms of predicted and actual prices for the top 3 models: SVR, KMC, and NN

Ridge regression, neural network, K-means + Ridge regression models had similar R^2 scores even though the last two models are more complex than Ridge regression. The architecture complexity of neural network was limited by the insufficient number of training examples for having too many unknown weights. K-means clustering model faced a similar issue: since the frequency of some prices was greatly exceeding the frequency of others, some clusters received too few training examples and drove down the overall model performance.

6 Conclusions and Future Work

This project attempts to come up with the best model for predicting the Airbnb prices based on a set of features including property specifications, owner information, and customer reviews on the listings. Machine learning techniques including Linear Regression, Tree-based models, SVR, and neural networks along with feature importance analyses are employed to achieve the best results in terms of Mean Squared Error, Mean Absolute Error, and R^2 score. The initial experimentation with the baseline model proved that the abundance of features leads to high variance and weak performance of the model on the validation set compared to the training set. Lasso Cross-validation feature importance analysis reduced the variance and using advanced models such as SVR and neural networks resulted in higher R^2 score for both the validation and test sets. Among the models tested, Support Vector Regression (SVR) performed the best and produced an R^2 score of 69% and a MSE of 0.147 (defined on ln(price)) on the test set. This level of accuracy is a promising outcome given the heterogeneity of the dataset and the involved hidden factors, including the personal characteristics of the owners, which were impossible to consider.

The future works on this project can include (i) studying other feature selection schemes such as Random Forest feature importance, (ii) further experimentation with neural net architectures, and (iii) getting more training examples from other hospitality services such as VRBO to boost the performance of K-means clustering with Ridge Regression model in particular.

7 Contributions

- Liubov Nikolenko: data cleaning, splitting categorical features, implementing sentiment analysis of the reviews, initial neural network implementation, SVR implementation and tuning, K-means + Ridge tuning.
- Hoormazd Rezaei: implementation of linear regression and tree ensembles, data-preprocessing, implementation of the evaluation metrics, feature selection methods implementation, tuning of the neural network.
- Pouya Rezazadeh: data cleaning and auxiliary visualization, splitting categorical features, result visualizations, tree ensembles tuning, K-means + Ridge implementation.

References

[1] H. Yu and J. Wu, "Real estate price prediction with regression and classification," *CS229 (Machine Learning) Final Project Reports*, 2016.

[2] Y. Ma, Z. Zhang, A. Ihler, and B. Pan, "Estimating warehouse rental price using machine learning techniques.," *International Journal of Computers, Communications & Control*, vol. 13, no. 2, 2018.

[3] D. Wang and J. L. Nicolau, "Price determinants of sharing economy based accommodation rental: A study of listings from 33 cities on airbnb. com," *International Journal of Hospitality Management*, vol. 62, pp. 120–131, 2017.

[4] L. Masiero, J. L. Nicolau, and R. Law, "A demand-driven analysis of tourist accommodation price: A quantile regression of room bookings," *International Journal of Hospitality Management*, vol. 50, pp. 1–8, 2015.

[5] Y. Yang, N. J. Mueller, and R. R. Croes, "Market accessibility and hotel prices in the caribbean: The moderating effect of quality-signaling factors," *Tourism Management*, vol. 56, pp. 40–51, 2016.

[6] Y. Li, Q. Pan, T. Yang, and L. Guo, "Reasonable price recommendation on airbnb using multi-scale clustering," in *Control Conference (CCC), 2016 35th Chinese*, pp. 7038–7041, IEEE, 2016.

[7] https://scikit-learn.org/stable/, *scikit-learn Machine Learning in Python*.

[8] https://keras.io/, *Keras: The Python Deep Learning Library*.

[9] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[10] R. Johansson, *An intuitive explanation of gradient boosting*. http://www.cse.chalmers.se/ richajo/dit865/files/gb_explainer.pdf.