Predicting Airbnb Listing Price

Yuanhang Luo, Xuanyu Zhou, and Yulian Zhou
royluo@stanford.edu, xuanyu98@stanford.edu, zhouyl@stanford.edu
Department of Computer Science

Abstract

Airbnb has become increasingly popular among travelers for accommodation across the world. Accordingly, there are large datasets being collected from the Airbnb listings with rich features. In this project, we aim to predict Airbnb listing price in two cities – New York City (NYC) and Paris with various machine learning approaches, including linear regression, k-nearest neighbor regression, random forest, XGBoost, as well as neural network. With XGBoost, we have achieved r-squared value 0.749 in train and **0.740** in test on NYC dataset, and 0.722 in train and **0.704** in test on Paris dataset.

Dataset

- Kaggle Airbnb open datasets in both NYC and Paris – 96 features
- The NYC dataset: 44317 listings, Oct, 2017-Oct, 2018
- The Paris dataset: 59881 listings, Dec, 2018-Dec, 2019
- Ground truth label: listing price
- Train:Dev:Test split is 7:2:1

Features

- **Continuous**: 30 features, 2-degree interaction terms are added.
- **Categorical**: 20 features, one-hot encoding is performed.
- **Text**: 12 features, tf-idf of unigrams and bigrams followed by truncated singular value decomposition.
- **Date**: 3 features, converted to continuous values.
- **Price**: 1 label, thresholded or transformed.

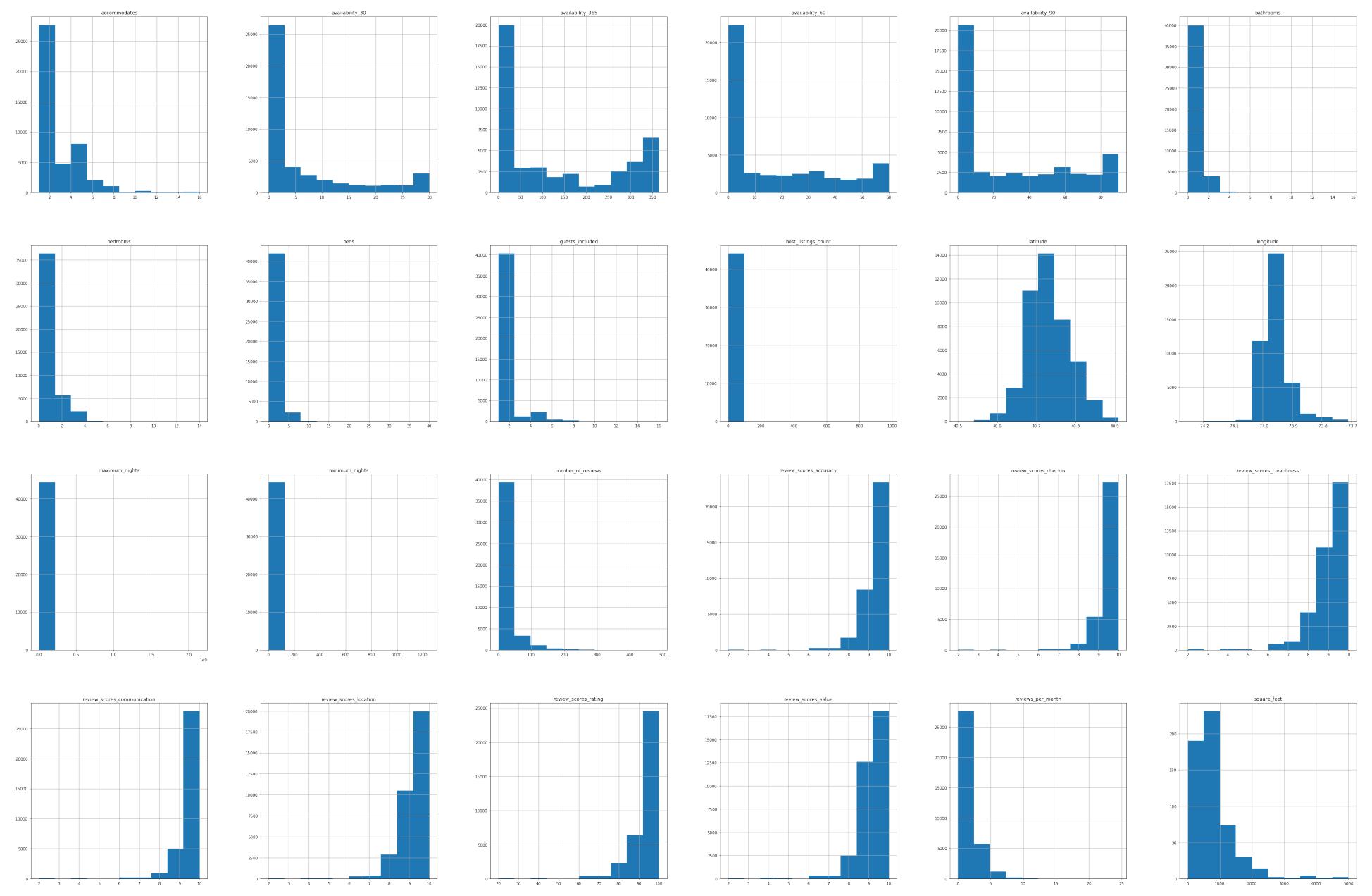

Figure: Continuous feature visualization

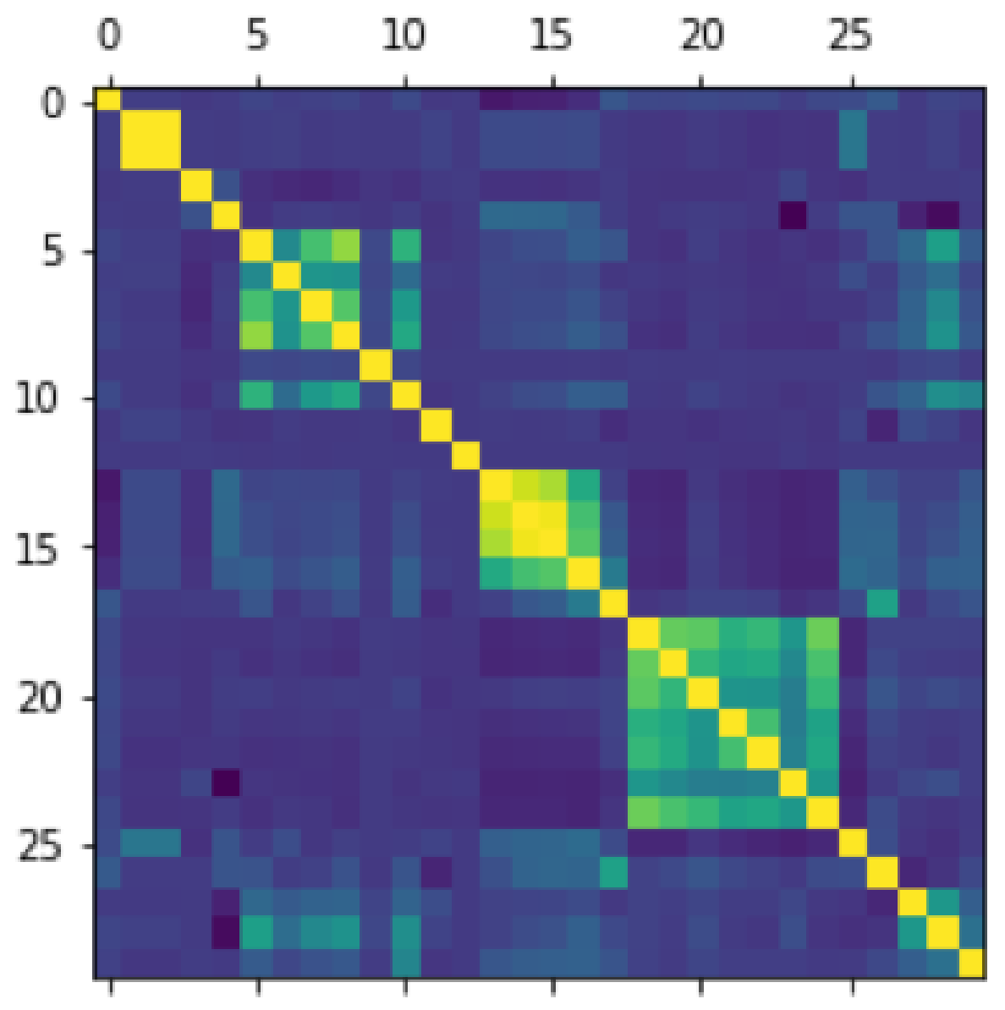

Figure: Continuous feature correlation

Models

- **Baseline**

K-nearest neighbor and linear regression w/wo regularization.

- **Random Forest**
  - A Decision Tree is a tree-structured plan of a set of attributes to test in order to predict the output.
  - Random forest is improved from bagged decision tree, and uses modified tree learning algorithm with feature bagging.

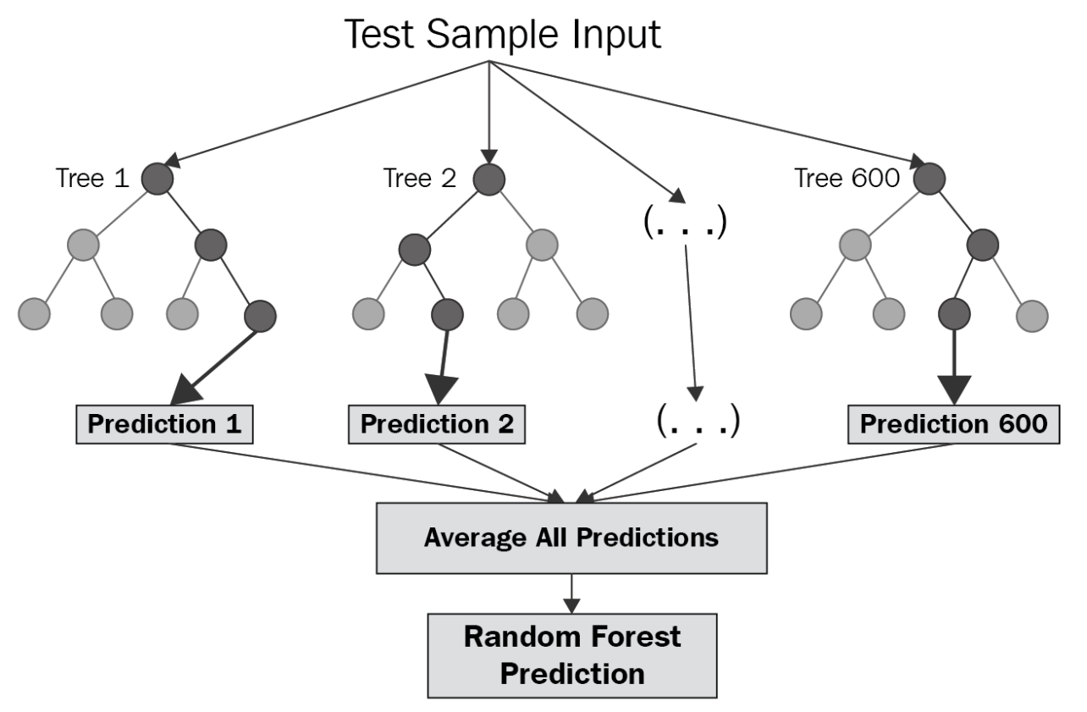

Figure: Illustration of random forest.

- **XGBoost**
  - Model (k trees): $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in \mathscr{F}$
  - Objective (train loss + tree complexity): $\text{Obj} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$

- **Neural Network**
  - 4 fully connected layers + ReLU activation
  - Input: 30 continuous features.
  - Optimizer: Adam optimizer $lr = 1e^{-4}$, weight_decay $= 1e^{-5}$
  - Loss: Mean squared loss

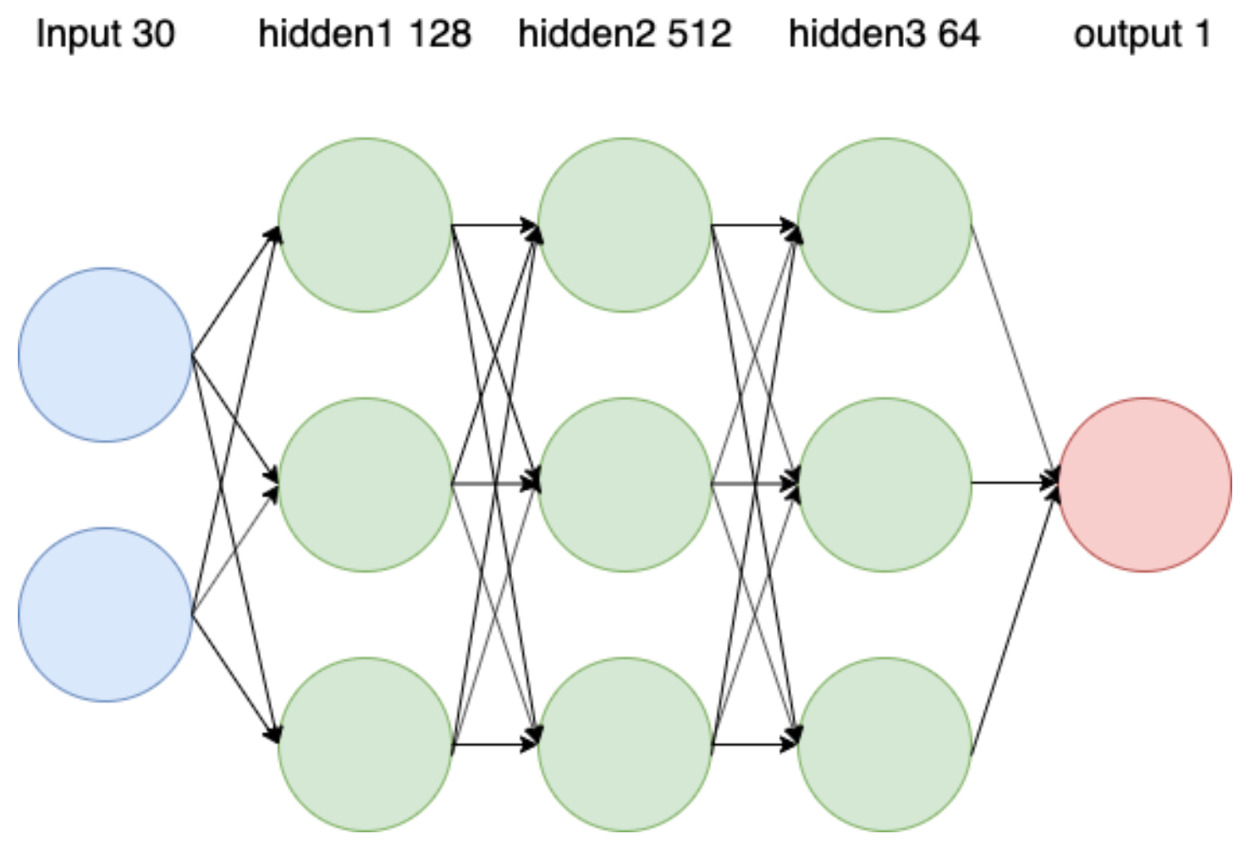

Figure: Illustration of neural network.

Results

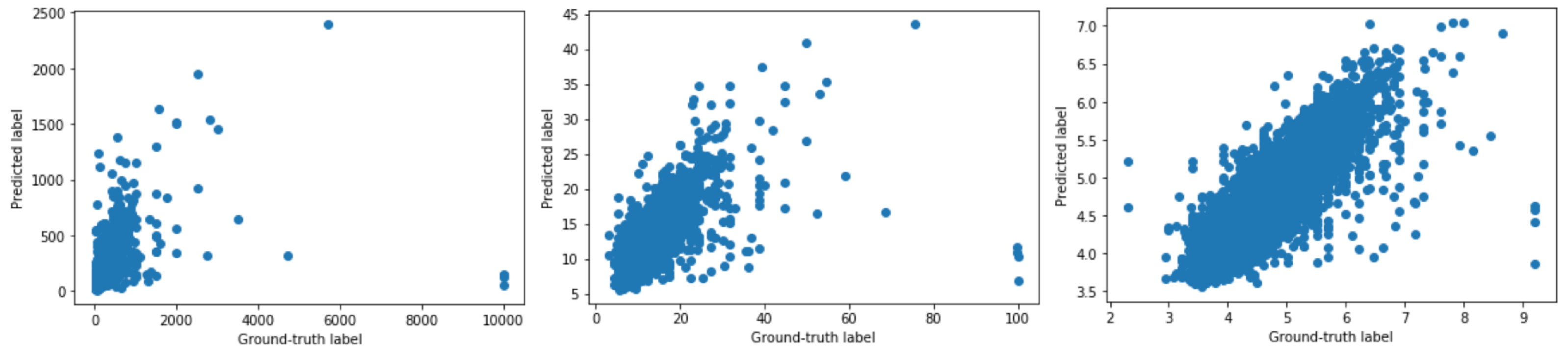

Figure: Label transformation (None, Sqrt, Log)

Table: Performance comparison between different models

			Train		Test	
Model	**Feature**	**Label**	**MSE**	**r-squared**	**MSE**	**r-squared**
Linear regression	C[a]	TH[b]	3514.64	0.489	3541.99	0.497
K-nearest neighbor	C	TH	4466.30	0.351	4760.61	0.323
Random forest	C	TH	1706.86	0.752	2427.05	0.655
Neural network	C	TH	2059.43	0.702	2367.44	**0.665**
XGBoost	C	TH	2273.98	0.669	2357.83	**0.665**
XGBoost	C	N/A	16748.6	0.558	58003.1	0.195
XGBoost	C	Sqrt[c]	7.17806	0.652	10.4435	0.537
XGBoost	C	Log[d]	0.13905	0.695	0.14765	0.669
XGBoost	C+I[e]	Log	0.13061	0.708	0.14704	**0.676**
XGBoost	O[f]	TH	3134.42	0.546	3219.57	0.532
XGBoost	T[g]	TH	3771.55	0.452	4231.74	0.399
XGBoost	D[h]	TH	6654.41	0.033	6927.05	0.016
XGBoost	C+O+T	TH	1885.25	0.724	2078.78	0.706
XGBoost	C+O+T	Log	0.11383	0.749	0.11731	**0.740**

[a] Continuous feature with normalization
[b] Threshold $y <= 500$
[c] $y' = \sqrt{y}$
[d] $y' = \log y$
[e] Add 2-degree interaction
[f] Categorical feature with one-hot encoding
[g] Text feature with latent semantic analysis
[h] Date feature transformed to continuous value

Discussion & Future Work

- Data cut-off or label transformation alleviate the problem caused by abnormally high price listings.
- Continuous features work best for price prediction; categorical and text features are added to boost model performance; date features are less useful in this application.
- Neural network works well with only continuous features, yet it is subjected to overfit problem with all C + O + T features.
- We observed similar model performance in both NYC and Paris.

In future work:

- Additional feature engineering, error analysis and hyperparameter tuning of neural nets.
- Transfer learning with neural network to other similar tasks.

Acknowledgement

The authors would like to thank all CS229 course staff for instructions and guidance on this project.