Melbourne Airbnb Price Prediction

Tiancheng Cai, Kevin Han, Han Wu

{caitch, kevinwh, hanwu71}@stanford.edu

Problem & Task

Motivation

- Ensuring fair pricing directly affects booking activities on Airbnb and the experience of hosts and customers.

Problem: Price Prediction in Original Scale

- Goal: predict price for Melbourne listings on Airbnb.
- Train both traditional ML models and deep learning models using continuous, categorical and text features, with R2 and MSE as evaluation metrics.
- Previous projects [1][2] only work on easier version of the task (transform to binary classification problem or evaluate on logarithmic scale of prices).

Results

- Gradient boosting with all features perform the best, while feature selection improves the performance of Random Forest.
- DL model using all features (continuous, categorical and text) achieves comparable accuracy, and DL model using text features alone also shows reasonable performance.

Dataset

DATASET: Public Dataset on Kaggle

- A CSV file with 84 columns containing detailed information of 22985 Airbnb listings in Melbourne on Dec 8th 2018.
- A CSV file containing all 469737 reviews for the corresponding Airbnb listings in Melbourne on Dec 8th 2018.

RESPONSE VARIABLE: Price for Each Listing

- Consider listings with price <= $1000.
- Use original listing price without changing the scale.

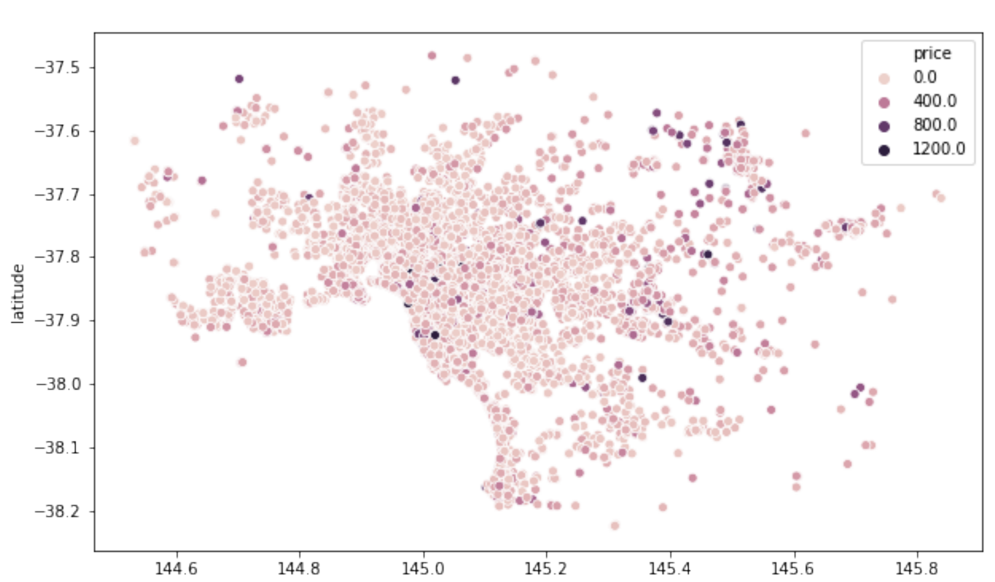


Figure 1: Geographical distribution of prices

Features

Continuous Features

- Include latitude, longitude, number of bedrooms, review scores, available days in future 30 days, etc.

Categorical Features

- Include amenities, types of listing, neighborhood, etc.
- Convert into one-hot encodings.
- Expand descriptive strings such as amenities (offered by a listing) into a number of categorical features.

Text Features

- Use description and reviews in the original dataset.
- Use GloVe50 Word Embedding for input layer.

Feature selection with LASSO

- Select 80 out of 155 features for selected models.

Models

Traditional Machine Learning Models

- Linear Regression
- Ridge Regression ($\lambda = 100$)

$$\arg\min_{\beta} \sum_{i=1}^{N} (y_i - x_i^T \beta)^2 + \lambda ||\beta||_2^2$$

- Random Forest (max feature=10, unlimited depth, 1000 estimators)
- Support Vector Regression

$$\begin{aligned} \arg\min_{w,b,\xi_i} \quad & \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i \\ \text{subject to} \quad & y_i - w^T \phi(x_i) - b \leq \epsilon + \xi_i \quad \forall i = 1, N \\ & w^T \phi(x_i) + b - y_i \leq \epsilon + \xi_i \quad \forall i = 1, N \\ & \xi_i \geq 0 \quad \forall i = 1, N \end{aligned}$$

- Gradient Boosting (max depth=7, max features=6, 200 estimators)
- Model Averaging (Random Forest, Gradient Boosting)

ML hyper-parameters tuning

- random search with 5-fold cross validation

Deep Learning Models

- Using continuous/categorical features only
- Using text features (comment/ description) only
- Using Combined features (Figure 1, lr = 0.001, with Adam Optimizer)

DL hyper-parameters tuning

- random search, early stopping

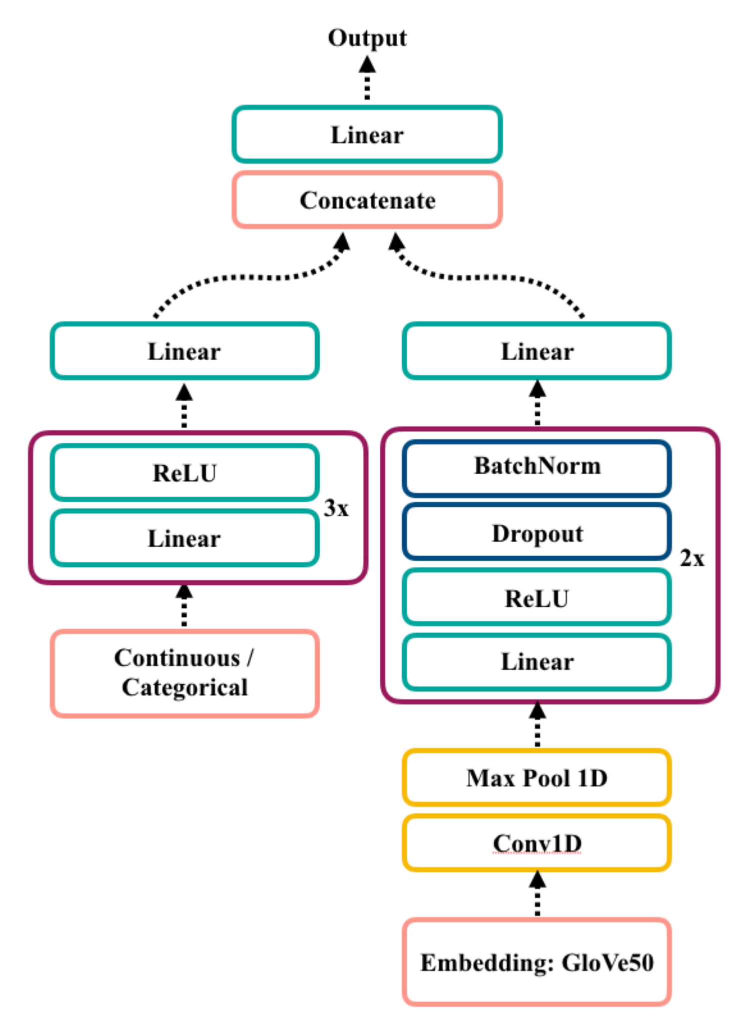


Figure 2: Combined model pipeline. Continuous and categorical features on the left, text features on the right

Results

Machine Learning Models		Test MSE	Test R2	Training MSE
No Feature Selection	Linear Regression	6460.7976	0.5046	7087.0750
	Ridge Regression	6460.7368	0.5046	7087.4613
	Random Forest	4513.5144	0.6539	761.5576
	Gradient Boosting	4024.7052	0.6914	2468.5471
	Support Vector Regression	5246.2777	0.5977	4878.6639
Feature Selection with LASSO	Linear Regression	6800.5157	0.4786	7484.7884
	Ridge Regression	6800.8531	0.4786	7485.6137
	Random Forest	4422.7124	0.6609	789.7775
	Gradient Boosting	4156.5544	0.6813	2392.1191
	Support Vector Regression	5459.5413	0.5814	5296.0410
Deep Learning Models		Test MSE	Test R2	Training MSE
Original Features	Four-Layer Feedfowrad NN	4632.5926	0.6448	4197.0896
Text Data Only	Using Description	8523.1239	0.3465	1599.9195
	Using Description + Review	5467.3576	0.4717	2301.7509
Original Features + Text	Combined model	4526.6191	0.6529	1023.2743

Table 1: Results from traditional ML and DL Models

Data split

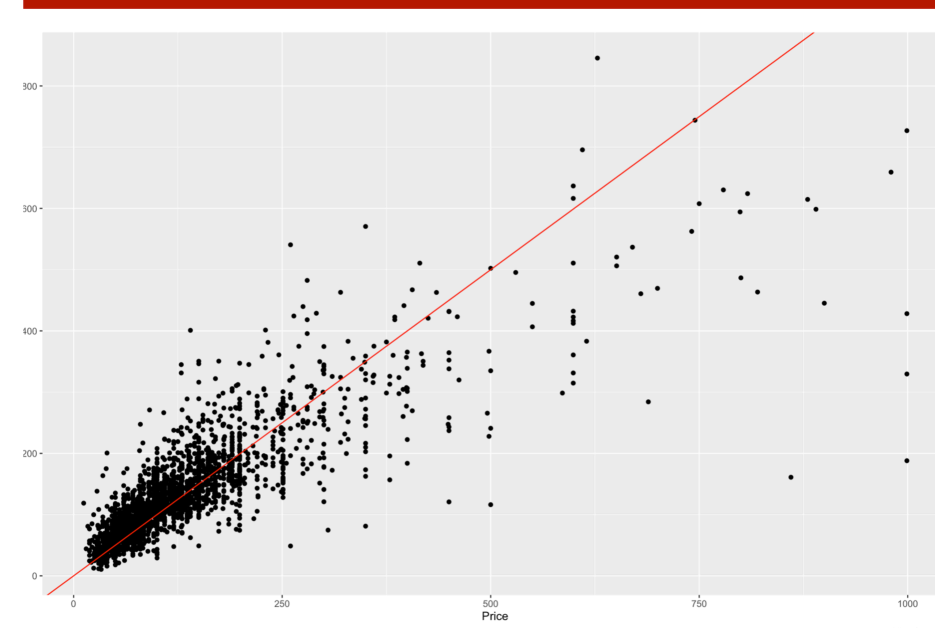
Continuous/Categorical:

- Train (18179, , 70%)
- Dev (2273, 15%)
- Test (2273, 15%)

Text (description, comments):

- Train (388175)
- Dev (51444)
- Test (45797)

Discussion



- Our best model tends to underestimate the price of listings with higher prices.
- If we instead consider listings with price <= $500 only, we would achieve significant improvements. (e.g. MSE on test set with Deep Learning model using only descriptions and reviews drops to 2894.2764).

Figure 3: Actual prices against best predictions by Gradient Boosting on test set

Future Work

Machine learning models

- Explore more ML models, and perform more careful feature feature selection and hyper-parameter tuning.
- Try out two-step modeling. Divide training sets into K groups based on price range and build separate models for each group. Classify group label and then run price regression.

Deep learning models

- Use more complex NLP models.
- Perform more hyper-parameter tunings.

Reference

[1] P. R. Kalehbasti, L. Nikolenko, and H. Rezaei. Airbnb price prediction using machine learning and sentiment analysis, 2019.

[2] E. Tang and K. Sangani. Neighborhood and price prediction for san francisco airbnb listings. CS 229 Final Project Report, 2015.