Melbourne Airbnb Price Prediction

Tiancheng Cai, Kevin Han, Han Wu
{caitch, kevinwh, hanwu71}@stanford.edu

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.

Models

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.

Traditional Machine Learning Models
• Linear Regression
• Ridge Regression ($\lambda = 100$)
• Random Forest (max feature=10, unlimited depth, 1000 estimators)
• Support Vector Regression

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

DL hyper-parameters tuning
• random search, early stopping

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.

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).

Future Work

Machine learning models
• Explore more ML models, and perform more careful 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