Price prediction for Melbourne Airbnb listings

Abstract

Hosts on Airbnb often end up using a trial and error approach or have to perform an extensive market analysis to come up with a sensible price for their property. A helpful recommendation that Airbnb can provide is an approximate range of price based on similar properties hosted by others. Melbourne Airbnb dataset was used to perform predictive analysis of price of listings using Supervised Machine learning. Regression models were built and hyperparameters tuned using 5 fold cross validation. XGBoost Regressor gave a significantly higher performance with a Root Mean Squared error (RMSE) of 48.1 over a Naive baseline model (RMSE 75.8).

Introduction

Airbnb connects people with places to stay. The community is powered by hosts who can choose to share an extra room or an entire house with Airbnb customers. The website states that a host can charge as per their choice for a listing. This leaves the host with the responsibility to perform market analysis and come up with a sensible price, based on comparable listings in their city or neighbourhood. Perhaps a helpful feature that Airbnb can offer to their hosts is a suggested price range by utilizing all the information provided by a prospective host. The aim of this project was to predict the price of Melbourne Airbnb listings using Supervised Machine Learning. Several host and property related features were studied for their direct or indirect influence on price. Related features were utilized to build regression models like Polynomial Regression, Support Vector Regressor (SVR) and also more advanced algorithms like Random Forest and XGBoost were investigated. Polynomial regression[1] is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an nth degree polynomial in x. SVRs[2] are an extension

[1] "Optimal Experimental Design for Polynomial Regression ...." 5 Apr. 2012, https://amstat.tandfonline.com/doi/abs/10.1080/01621459.1971.10482260. Accessed 7 Jun. 2020.

[2] "Support Vector Regression Machines." http://dl.acm.org/ft_gateway.cfm?id=2999003&ftid=1788985&dwn=1&CFID=841807898&CFTOKEN=80744895. Accessed 7 Jun. 2020.

of the Support Vector Machine technique that was first developed for classification. By using the support vectors for local approximations, SVRs allow for non linear regression estimation. Random Forest[3] builds multiple decision trees in the form of an 'ensemble' usually trained with a 'bagging' method. XGBoost[4] is a scalable and improved version of the gradient boosting algorithm, designed for efficiency, computational speed and model performance. It is a tree learning algorithm that does parallel computations on a single machine.

Materials and Methods

1. Dataset and libraries

The dataset can be downloaded from Kaggle. The 'cleansed_listings_dec18.csv' file was processed using Python libraries - Pandas and Numpy libraries. Matplotlib and Seaborn were used for visualizations and all Machine Learning models were built using the Scikit-Learn library.

2. Cleaning, Feature Engineering and Selection

The original dataset consisted of 22895 rows and 84 columns. Features considered to be entirely irrelevant for the purpose of this study were dropped. The dataset was then cleaned by either imputation of missing values or dropping the corresponding rows as appropriate, changing datatypes of columns where necessary, removal of outliers and processing of individual categorical columns. With utilization of existing columns and data scraped from real estate website Domain, a feature called 'suburb_rank' was engineered. Similarly, column 'distance' which lists the distance in km of respective property from Melbourne CBD was created using geopy module. 3 more features related to facilities provided by hosts and considered relevant to price prediction were engineered. Finally, feature selection was performed using correlation analysis of numerical features with 'price'. With categorical columns, distribution of price with respect to each category was visualised using boxplots to determine if they were related to price. The final dataframe (21378 rows, 21 columns) with all highly correlated numerical

[3] "Classification and regression by randomForest." https://www.r-project.org/doc/Rnews/Rnews_2002-3.pdf. Accessed 7 Jun. 2020.

[4] "xgboost: eXtreme Gradient Boosting - CRAN." 25 Mar. 2020, https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf. Accessed 7 Jun. 2020.

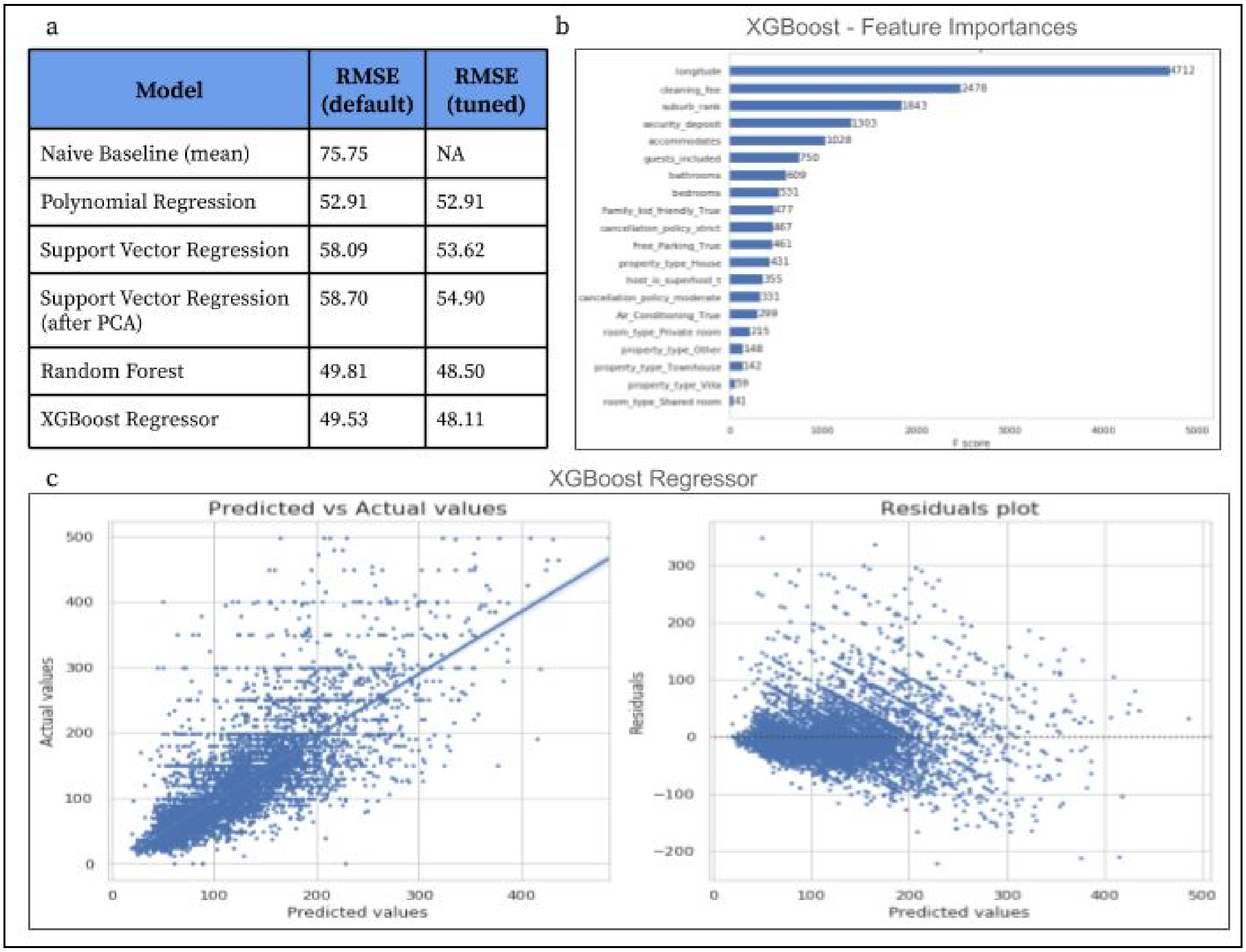
features and one hot encoded categorical features was saved as a csv file to be used for predictive analysis.

3. Regression Models

After performing a train test split, a ‘Naive baseline’ model with mean price was established and an RMSE score on the test set was evaluated. Regression models using Polynomial Regression, Support Vector Regression, Random Forests and XGBoost algorithms were built. An extensive hyperparameter search for each model was performed using grid search with 5 fold cross validation. The performance of each regression model - before and after hyperparameter tuning was evaluated by comparing respective RMSE scores with that of the naive model. The best performing model was saved for future use.

Results and Discussion

In this study, four prediction algorithms were investigated with each of them being subject to extensive hyperparameter tuning using grid search with a 5 fold cross validation. The best models of each algorithm were evaluated on a test set. Fig.1 a shows the RMSE scores obtained with default and tuned models.



Model	RMSE (default)	RMSE (tuned)
Naive Baseline (mean)	75.75	NA
Polynomial Regression	52.91	52.91
Support Vector Regression	58.09	53.62
Support Vector Regression (after PCA)	58.70	54.90
Random Forest	49.81	48.50
XGBoost Regressor	49.53	48.11

Fig.1: Scores of models, Feature Importances and Plots with XGBoost Regressor

Support Vector Regression model was run with and without reducing dimensions with PCA. The RMSE score after reducing dimensions was slightly higher, thus PCA seems to worsen the performance of SVR model. Lowest RMSE score was obtained with XGBoost Regressor. 'Longitude', 'cleaning_fee', 'suburb_rank' (engineered feature) 'security_deposit' and 'accommodates' are the 5 most important features used by the XGBoost model (Fig.1b). For prices below 250 AUD, majority of the datapoints/actual values lie close to the fitted model (Fig.1c). Predictions become worse as the price increases as also shown by the heterodecstaticity in the residual plot. A similar trend was seen with all the other prediction models. Overall the error with the best model is ~48 AUD. This error is much lower than the Naive Baseline model (~76 AUD), thus the model is certainly performing better. However, a range of ∓ 48 AUD is too high and not a very useful recommendation feature for new hosts on

Airbnb. There are some inherent limitations of the data collected. A very important feature that should be highly correlated with price would be size or area of property. This information was lacking in the dataset. Also, images and descriptions of property listings greatly influence the price.

Conclusions

A Supervised Machine learning approach was used to build regression models for predictive analysis of Melbourne Airbnb listings. Feature selections were performed using Pearson's correlation and examining boxplots. Of the 3 features engineered, suburb_rank was found to be most correlated with price. XGBoost Regressor model gave a lowest error of ∓ 48.1 AUD and a significantly higher performance than the Naive Baseline model (∓ 75.8 AUD). The performance of the model may increase further if we get information on property size, perform text & image analysis and engineer some new features.