Udacity Machine Learning Capstone Project

Price prediction in the AirBnB London Market

By Kjersti Mjøs Aase

Content:

Excecutive summary

The project shows how to develop a regressor to recommend prices in the airbnb market in London.

It starts with a data exploration phase where important variables are visualized. Then data are preprocessed and cleaned to fit the machine learning algorithms we will apply later. The feature engineering part is extensive, including a selection of which features to keep, transform and scale. Before a dimensionality reduction is tried by applying PCA to the dataset.

A benchmark model - Ridge linear model - is fitted both to the PCA dataset and the full feature dataset. And a pool of models were measured against the benchmark. The full feature dataset had much better results than the PCA dataset. And XGboost proved to be the best model and chosen for parameter tuning. The parameter tuning resulted in a R square of 0.78., which is a satisfying result, although several other measured could have been taken to improve the model given more time and computing power.

Project Overview

Airbnb started small in a living room in San Fransisco 2007, but has gained an enormous popularity since. May 2018 the company was valued at 38 billion usd [1]

The business model is an accomodation market place that allows people to list their available living spaces to be leased or rented by users looking for short term lodging. Airbnb has two main sources of revenue, commision from the hosts and transaction fee from the guests.

Airbnb allows bookings at listings in more than 81000 cities and across 191 countries.

Problem statement

Although Airbnb's income is dependent on the price, the setting of the price is completely up to the host. To support their decision they're advised to search for listings in their city or neighborhood to get an idea of market prices [2]. There is a huge potential in automating this process and optimizing the listing price. Especially if you are a host with multiple listings.

Beyond pricing is a provider of such a pricing software for vacation rentals. You are recommended a price based on the health of the listing, with factors such as neighborhood, seasonality, local demand and more. But it is not clear which models and methods are operating behind the scenes.

---

1 https://www.forbes.com/sites/greatspeculations/2018/05/11/as-a-rare-profitable-unicorn-airbnb-appears-to-be-worth-at-least-38-billion/#71347f262741

2 https://www.airbnb.com/help/article/52/how-should-i-choose-my-listing-s-price?locale=en

This project will discuss methods to and develop a price recommender using historical data in the London airbnb market.

First we get familiar with the dataset and applies data preprocessing and feature engineering to prepare the dataset for model testing. The second part discusses multiple regression models. A linear regression model is used as benchmark, and tested against several other regression models, such as Bagging, Random Forest and Xgboost. Then the best model is chosen and tuned.

The last part discusses the final results and suggest possible improvements that may be applied in further research

The dataset is publicly available on the http://insideairbnb.com/get-the-data.html site, and contains listings, calendars and reviews. I will only use the listings in the prediction model. However, in the concluding part I will discuss improvements that can be done in the modelling. Hereby also include calendar data to investigate seasonality.

Data exploration and data visualization

The airbnb listings dataset contains both categorical and numeric features. There are originally 74 153 listings in the dataset with 96 columns.

Analyzing the Target Variable

Price is the target variable that we are going to predict. And I will start by having a look at the distribution of price in the listings dataframe. The histogram shows that the price variable is skewed, and has its main weight on the observations below 400. I already now want to remove some of the outliers in the dataset. Thus I have chosen to eliminate 2% of the listings from the main dataframe to make a more uniform model.

Figure 1 . Price Plot before filtering

Further, I will have a look at the price once again, now in a distplot, to assess whether the variable is normally distributed.

The histogram below in figure 1 clearly shows that the price is skewed.

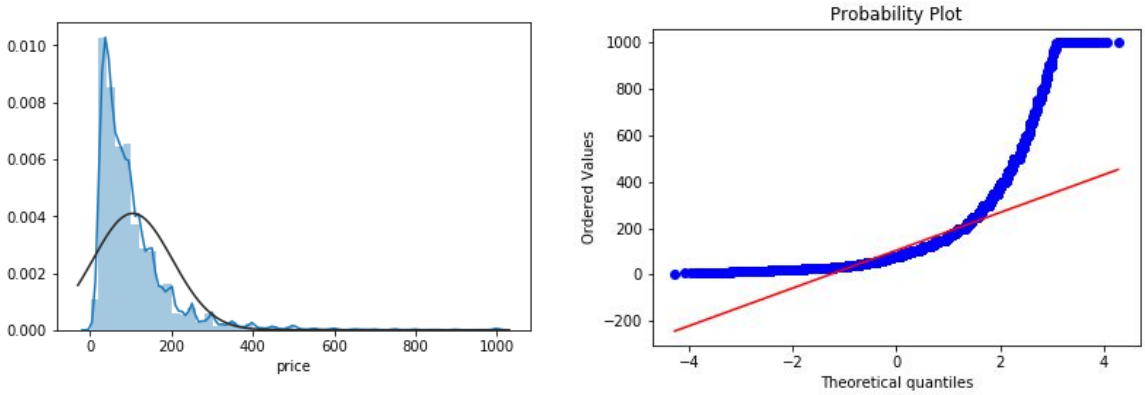


Figure 2. Price Plot

To reduce the skewness we use log transformation. This makes it easier to interpret patterns in the data and many statistical methods also require data to be normalized to function.

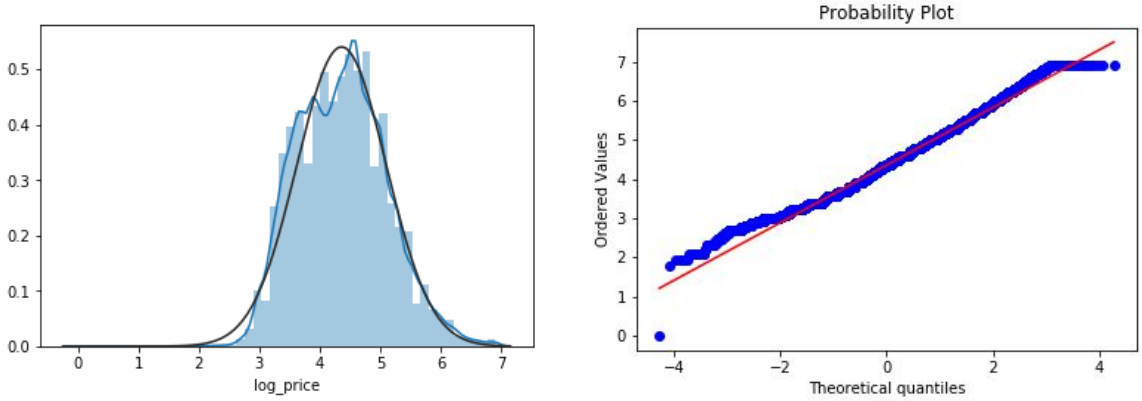


Figure 3: Log Price

Multivariable Analysis

The exploration phase is all about getting familiar and understand the data.

At the start off we now have 60 columns of categorical data and 36 columns of numerical data.

A heatmap is a good way to investigate the correlations between the features.

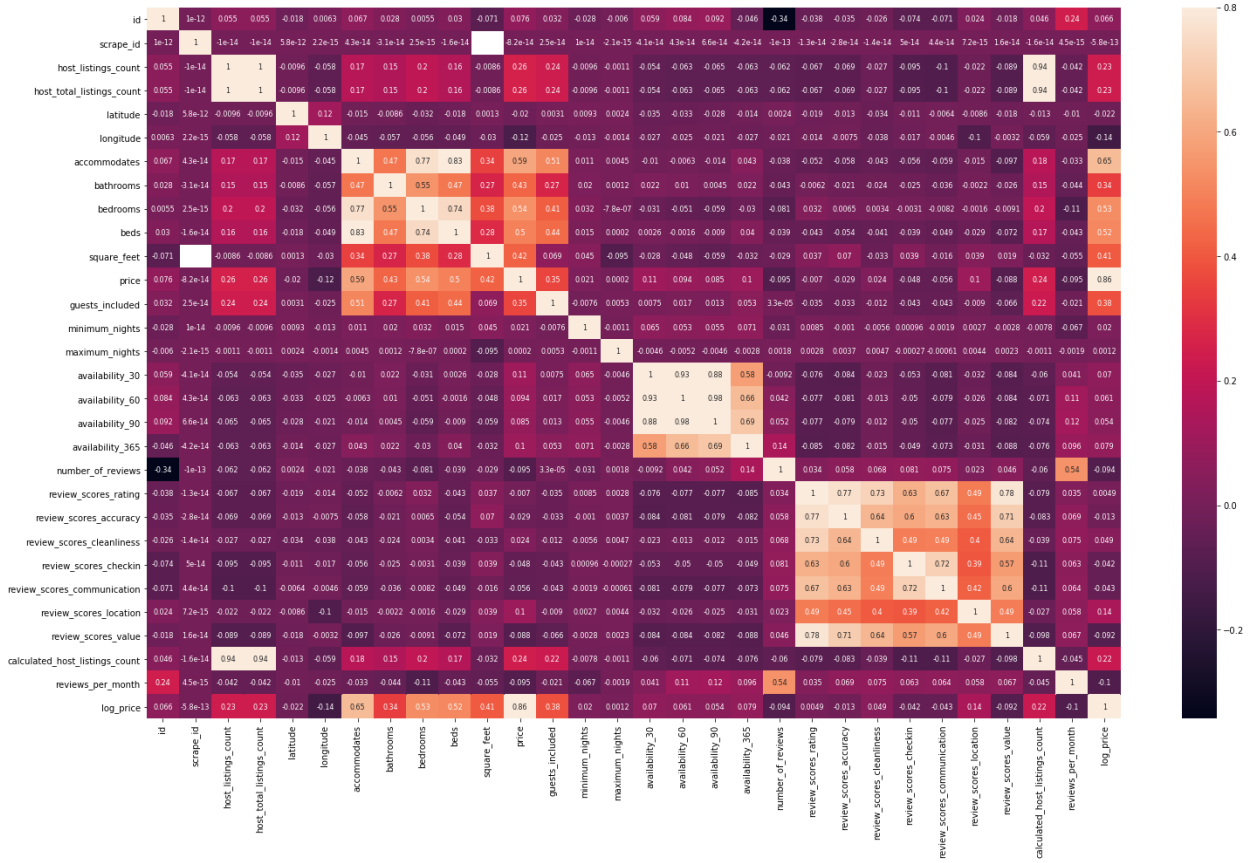


Figure 4. Heatmap Correlations

Not surprisingly beds, bedrooms, square feet, guests included have the highest correlations with price.

Data Preprocessing

Assess missing data

A central part of the data preprocessing is handling missing data. The frequency of missing data in each column is illustrated in a plot diagram. The features with the highest share of nans is removed from the dataset, e.g. license, square feet, weekly price, notes. Other are coded to binaries, f.ex. Access, either there is provided information about access or not.

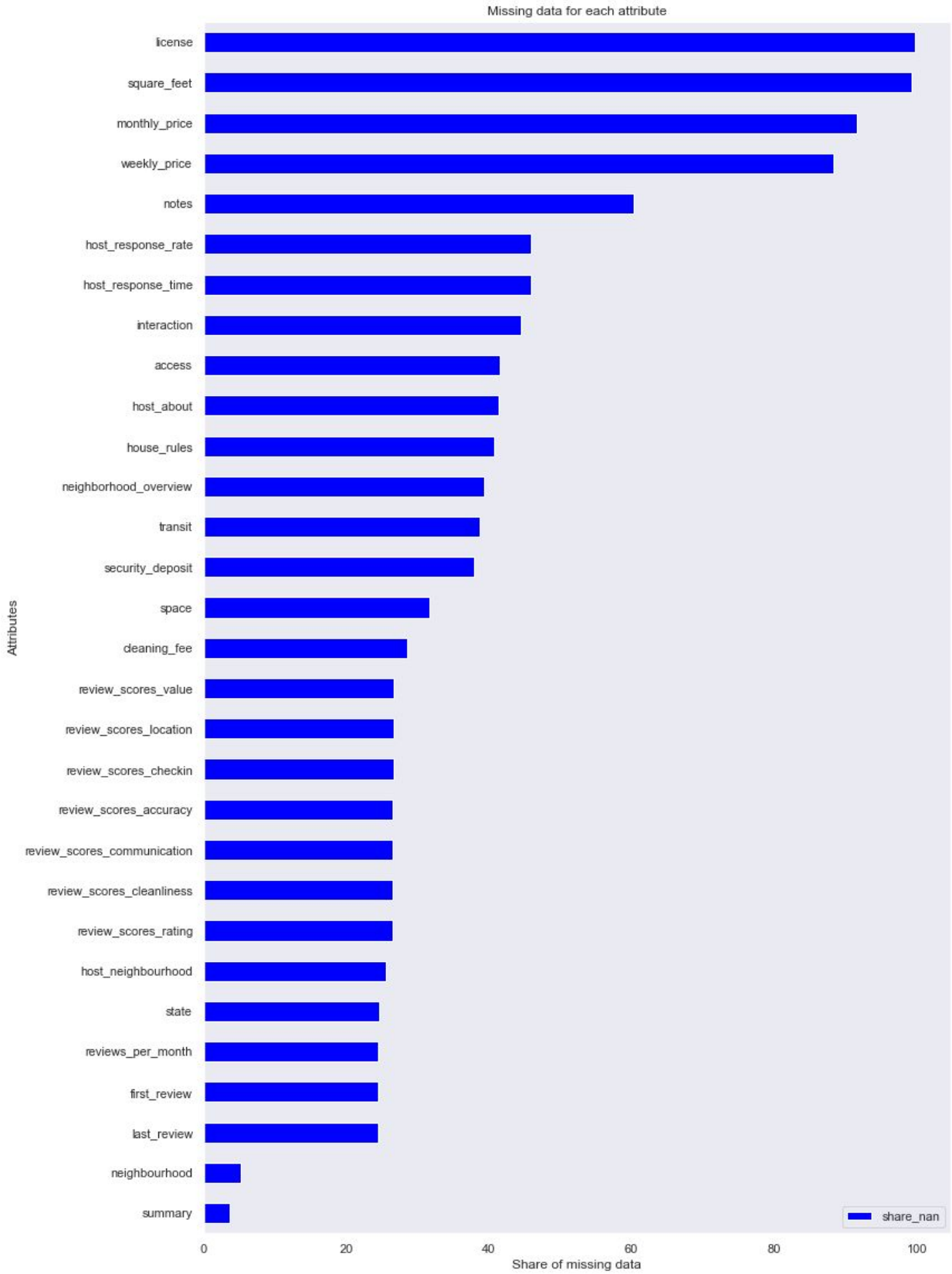


Figure 5 share of nan for each column

Detecting outliers

I have already made a filtering on the target variable to exclude outliers in price. A pairplot can be a good next step to search for patterns, relationships and anomalies in the dataset. But with the amount of data we have in this dataset it can be quite overwhelming as well.

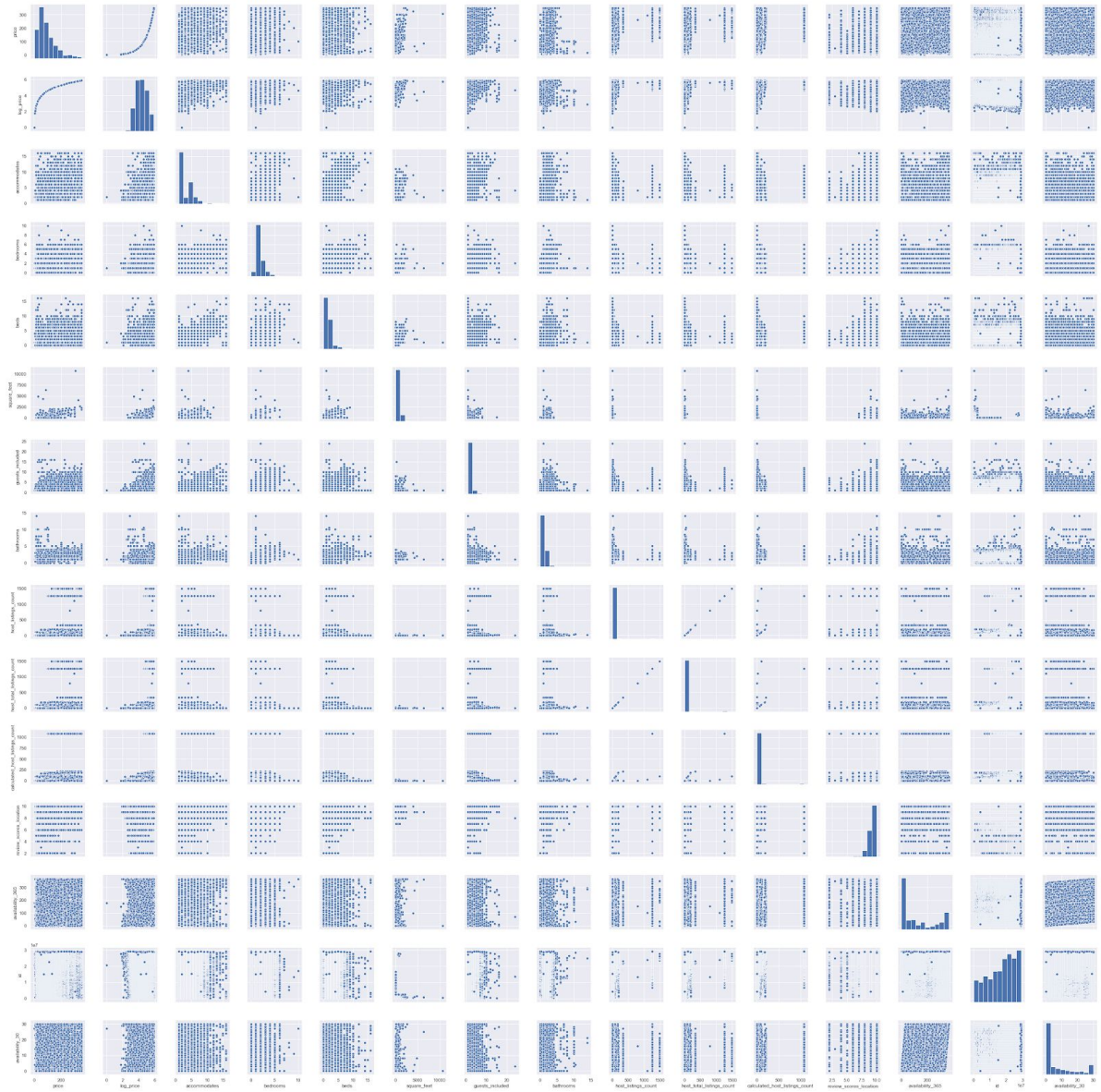


Figure 6 Pairplot most correlated columns

The pairplot show that we have a variety of data with different distributions. However, I have not done anything to delete outliers in this part. I will continue to go through the variables in the feature engineering part of the project.

Feature Engineering

Feature Selection

Feature selection is the process of selecting relevant features to use in a prediction model. Having irrelevant features in the dataset can decrease the accuracy of many models.

- Irrelevant or redundant features are removed from the dataset. Typical features are url, notes and other description fields.
- Some features are coded from category to binary variables, e.g. access, transit. This is because I want to distinguish between listings that have the features filled out or those who don't.
- The amenities description is really a collection of several amenities and is therefor splitted in several columns, such as tv, wifi, parking etc. I also count all the amenities that seem to be listed in the column.
- Features with true (t) or false (f) values are coded to 1/0.

Some algorithms, f.ex. Random Forest already have a built-in feature selection of features. Which is more explained under the model selection chapter.

Feature Transformation

To be able to use certain machine learning algorithms it is necessary to convert categorical data to numeric. I have applied one-hot-encoding, which means that for each unique value a new binary variable is added[3]. An example is for for the neighborhood_cleansed where Camden, City of London, Croydon etc. is represented by one variable each and with binary representation.

Features with a high share of nans has been removed from the dataset. However, there are still some nan values left. To handle this I have considered two different methods, median or filling with 0. I have run the benchmark model with both methods, and experienced best results filling with 0.

[3] https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

Feature Scaling

The data contain features with a mixture of scales. Many machine learning models works more effective if the features have the same scale.

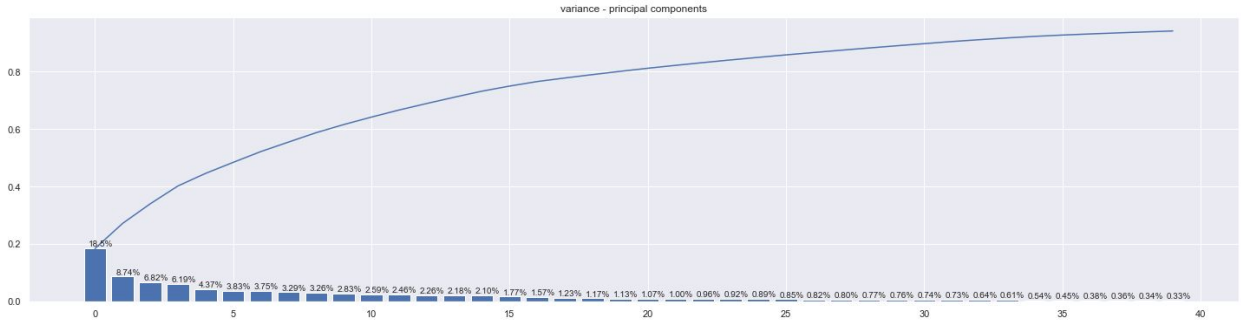
To transform the features to have the same scale relative to one another I have chosen to use the sklearn scaler. The standard scaler works best when the data is normally distributed within each feature. Because the data seem to be somewhat skewed I will use the MinMaxScaler.

Feature extraction (dimensionality reduction)

The intention with feature extraction is to create a new set of fewer features that still capture most of the variance.

PCA is one method for unsupervised feature extraction. It creates a linear combination of the original features. [4] The advantages with PCA is that it has proven to work well in practice and is fast and easy to implement. The weaknesses is that the components is not that easily interpretable and to manually set a threshold for the number of components.

With 20 components about 80% of the variance is explained, and with about 40 components 90% of the variance is explained.



Further, I have tested both the pca dataset and full dataset with the benchmark model and in the comparison of the other models. After that I chose to run a gridsearch on the best alternattive.

Model Testing and Selection

Benchmark Model

For a benchmark model I have used a linear model, which is one of the simplest way to predict an output.

---

[4] https://elitedatascience.com/dimensionality-reduction-algorithms

Due to a large number of features and relatively high collinearity, I will use the Ridge regression model[5]. This is a regularized regression model, balancing bias and variance better than the Ordinary Least Squared method. Ridge regression adds a small bias factor to the variables[6], but can greatly reduce the variance, resulting in a better mean-squared error

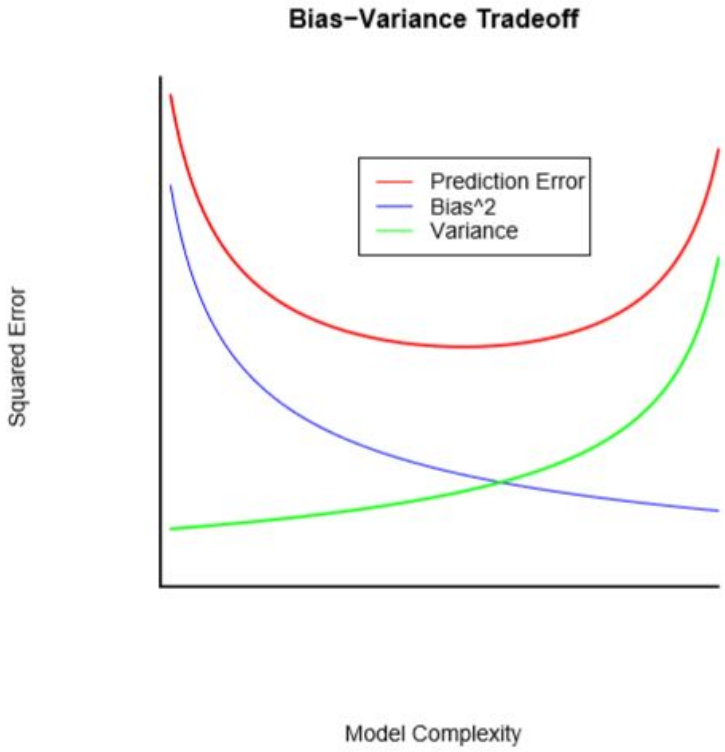


Figure Bias-Variance Tradeoff[7]

Evaluation Metric

R-Square is often used for explanatory purposes and is a metric of how well the independent variables explain the variability in the dependent variables .

Mathematically, it can be written as:

$$R - Square = 1 - \frac{\sum(Y_{actual} - Y_{predicted})\text{^}2}{\sum(Y_{actual} - Y_{mean})\text{^}2}$$

[5] https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

[6] https://stats.stackexchange.com/questions/52653/what-is-ridge-regression

[7] http://statweb.stanford.edu/~tibs/sta305files/Rudyregularization.pdf

“The numerator is MSE (average of the squares of the residuals) and the denominator is the variance in Y values. Higher the MSE, smaller the R_squared and poorer is the model.”[8]

Best score is 1.0. This means that 100% of the variation is explained by the model. The value can be negative for equations that do not contain a constant term. Then the fit is actually worse than just fitting a horizontal line.

These are the results from the Ridge regression model with PCA:

```
MSE train: 0.191, test: 0.187
R^2 train: 0.605, test: 0.609
```

And these are the results with the full features set: :

```
MSE train: 0.132, test: 0.128
R^2 train: 0.728, test: 0.732
```

This is a good starting point for evaluating different regression models. Although the full features set is getting a substantially better score I will fit the regressor model to the PCA dataset as well to measure the performance.

Model Selection

Bagging regressor

The bagging regressor builds multiple decision trees on subsets of the dataset
It improves variance by averaging/majority selection of outcome from multiple fully grown trees on variants of training set. It uses Bootstrap with replacement to generate multiple training sets.

Random Forest Regressor

The Random forest algorithm also builds multiple decision trees, but each tree only considers a random subset of features.

Because only a subset of the features are chosen, variance are improved. and correlation between the trees are reduced. By pooling them together the idea is to get a more accurate and stable prediction

[8] https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

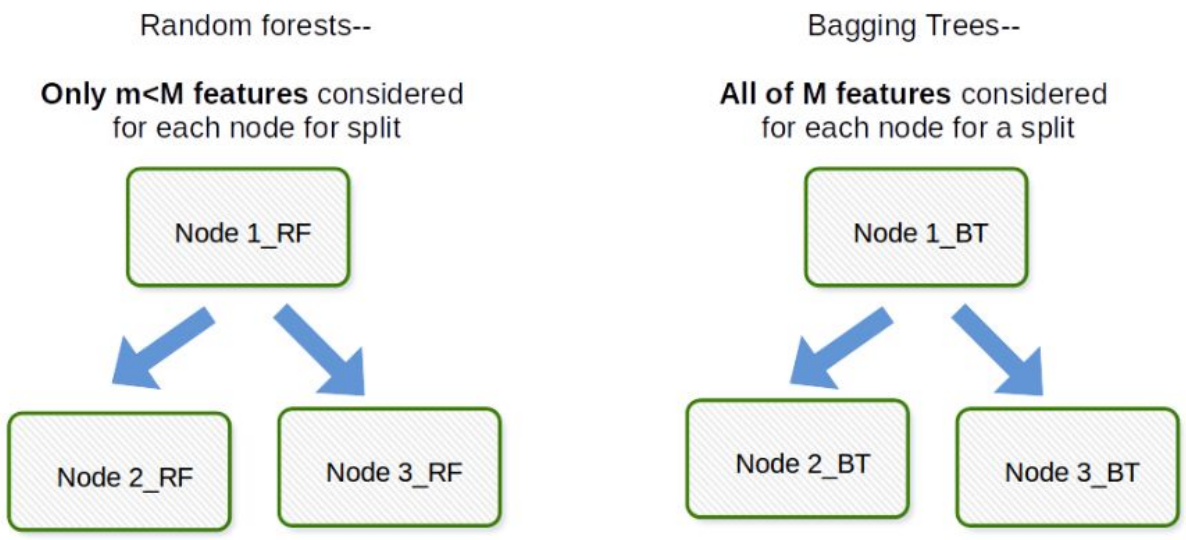


Figure 7: Difference between bagging and random forests[9]

AdaBoost regressor

I don't expect a high score on the adaboost regressor, but I have included it in the testing pool. The main principle of the adaboost algorithm is to fit a regressor on the original dataset and then fit additional copies of the same regressor, but now the weights of the instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases[10].

Gradient Boosting regressor

Gradient boosting involves week learners to make predictions. A loss function to be optimized. And to minimize the loss function, an additive model to add weak learners [11]

[9] https://stats.stackexchange.com/questions/264129/what-is-the-difference-between-bagging-and-random-forest-if-only-one-explanatory

[10] https://scikit-learn.org/stable/modules/ensemble.html#adaboost

[11] https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

XG-Boost regressor

XGboost stands for extreme gradient boosting, and has become one of the most popular machine learning algorithms. A good summary to the advantages of the algorithm, is given by Jain Aarshay in an article at AnalyticsVidhya[12].

- The algorithm is fast and scalable.
- It can penalize complex models through regularization and thus prevent overfitting
- It can handle missing data

Boosting refers the ensemble learning technique of building many models sequentially.
XG-boost provides parallel tree boosting. The parallelisation happens during the construction of each trees, at a very low level. Each independent branches of the tree are trained separately[13]

Huber regressor

The Huber Regressor is similarily to ridge a linear model, but is different because it applies a linear loss to samples that are classified as outliers. So the loss function is not heavily influenced by the outliers while still not completely ignoring their effect. [14]

Cross validation

Cross validation is a statistical method to compare and select a model for a given predictive problem. The advantages is that it is easy to understand, implement and generally have a lower bias than other methods.

Kfold cross validation is a procedure to estimate the skill of the model on new data. The k refers to the number of groups that a given sample is divided into.

The procedure described in “a gentle introduction to k-fold cross validation” Jason Brownlee, MachinLearningMastery 2018: [15]

---

[12] https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/

[13] https://medium.com/blablacar-tech/thinking-before-building-xgboost-parallelization-f1a3f37b6e68

[14] https://scikit-learn.org/stable/modules/linear_model.html#huber-regression

[15] https://machinelearningmastery.com/k-fold-cross-validation/

1. *Shuffle the dataset randomly.*
2. *Split the dataset into k groups*
3. *For each unique group:*
    a. *Take the group as a hold out or test data set*
    b. *Take the remaining groups as a training data set*
    c. *Fit a model on the training set and evaluate it on the test set*
    d. *Retain the evaluation score and discard the model*

. *4. Summarize the skill of the model using the sample of model evaluation scores*

Model scores

The models are run with 5 splits. First on the PCA dataset, then on the dataset with all features.

Figure 8: Model scores with PCA component dataset

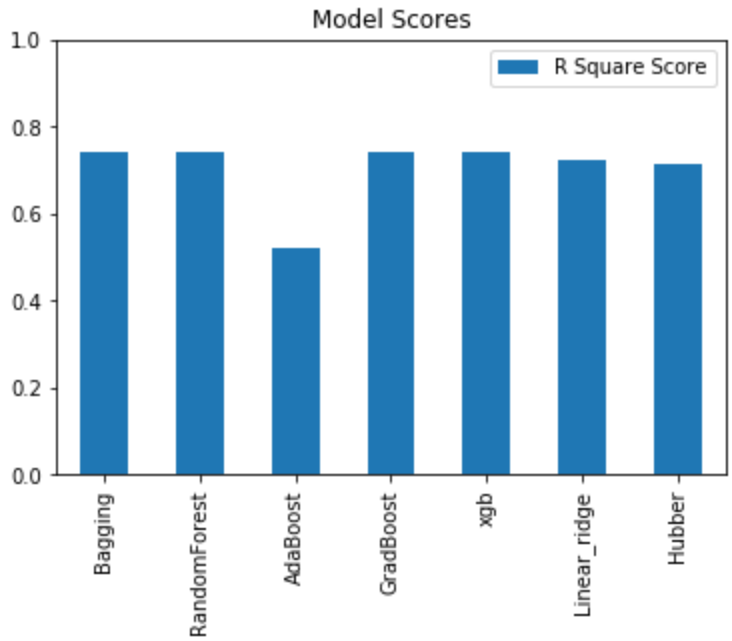


Figure 9: Model scores from dataset with all features

Model Tuning

XGBoost had the best R square score and there is a large number of advanced parameters which we can try to tune to improve the model.

Due to limitations in computing power I can't test all the parameters to optimize the model.
I have chosen to try parameters for max depth, number of estimators, learning rate and gamma.

Max_depth is quite explicit, it is the depth of the tree.
N_estimators is the number of trees,
Learning_rate (or eta) is set to control the weights of the new trees that are added to the model.
Gamma specifies the minimum loss reduction required to make a split.

The parameter tuning gets us from 0.73 to 0.784.

Results and Conclusion

So how good is really R square score of 0.784? The score represents how good the model is compared to the baseline model. 0.784 means 78.4% of the variations in the target variable are explained by the independent variables present in our model.But there are pitfalls. R square score never decreases when more independent variables are included. [16]

The model includes a feature of importance, which shows that the amenities count, reviews per month and overall, yearly availability, number of listings to the host etc.

Features of importance

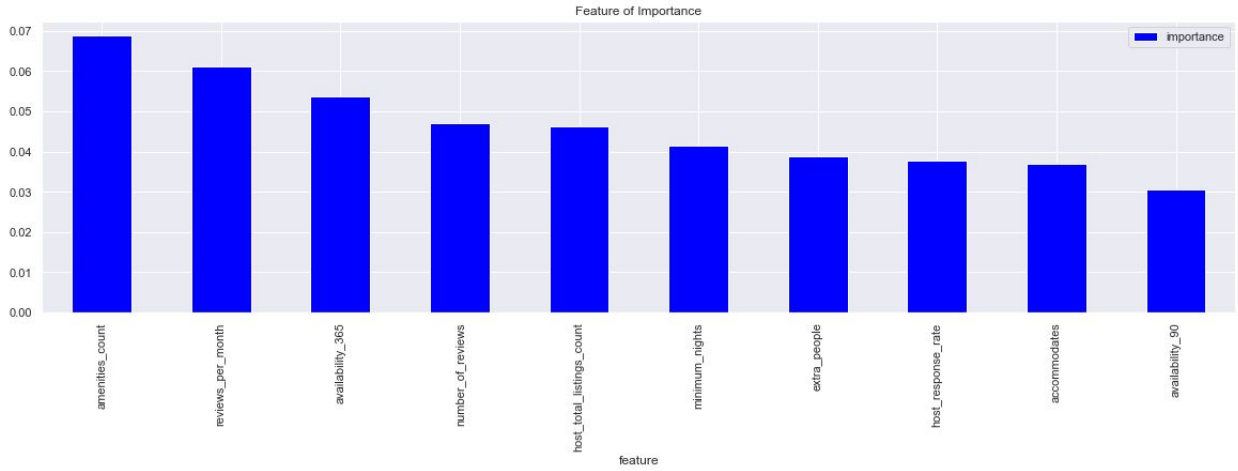


Figure 10 Features of importance from tuned model

The model also have implemented a predict where we can test the listings.

Improvements

I have done much preprocessing of the data and a fair amount of feature engineering. There are however of course several more steps that could be taken an more features to be investigated thoroughly.

The gridsearch can also be extended to more parameters. But my computing power is limited.

[16] https://towardsdatascience.com/coefficient-of-determination-r-squared-explained-db32700d924e

I have not used the calendar dataset. Seasonality in pricing should also be investigated to have a more robust price recommender. The calendar dataset shows bookings one year ahead. And i have started to se if I can find some seasonality patterns and discuss methods this could be investigated more thoroughly.
The calendar file contains the prices from october 6th and a year ahead in time.

- Booking the next few days is substantially more expensive than booking ahead.
- holiday season (christmas and bank holidays)
- weekends vs working days

However, to make any conclusions we should have more continous data, not just one year ahead. Booking the next few days will be more expensive. But it will also depend on which season and the demand in that period. And booking long ahead are probably less expensive.

We could have added some features on weekend vs working days in the dataset. On Fridays and Saturdays it is more expensive to find a room than the rest of the weeks. Also we could have a holiday parameter.