Using Inside AirBnb Data for price Prediction with Deep Learning Methods

Juan Sebastián Aristizábal Ortiz, Tobias Rinnert

Statistical and Deep Learning WS 21-22
Institute of Statistics
University of Göttingen
Göttingen, Germany

22.02.2022

Table of Contents

Repeated, Absent, Irrelevant

Repeated and Absent

- "host listings count == "host total listings count"
- "bathrooms"
- "license"
- "calendar updated"

Irrelevant

- Latitude
- Longitude
- Scrape Id

NA Rate > 0.5 & Trustfulness variables

NA Rate > 0.5

- "neighborhood"
- "neighborhood overview",
- "host neighborhood"

Trustfulness variables correlate possible with reviews

- "host about",
- "host response rate",
- "host acceptance rate",
- "host response time"

NA Removal

Initial Dataset

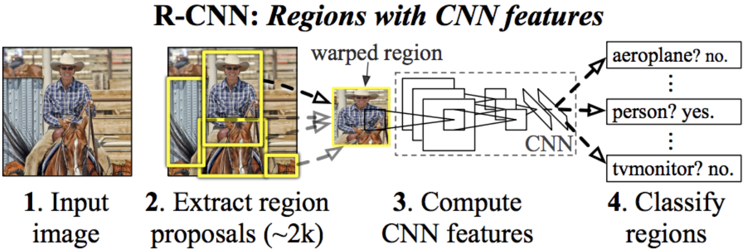
17290 observations

Cleaned Dataset

12175 i.e. 0.2958357% information loss.

Table of Contents

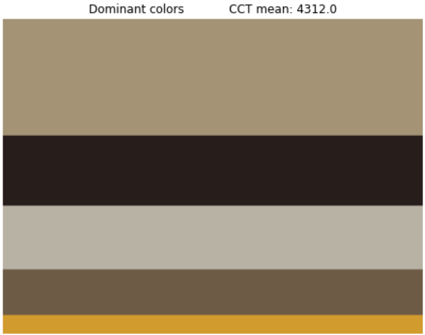
R-CNN Introduction



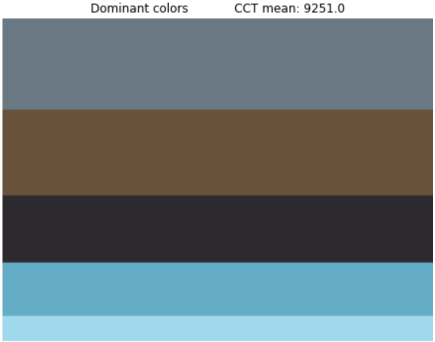
[Girshick et al.,]

Multi object detection: Example

Correlated color temperature Example 1



CCT Example 2



Brightness

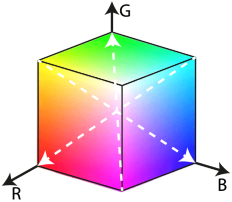


Figure: [Aguirre-Pablo et al., 2017]

3D colour space

$$brightness = \sqrt{R^2 + G^2 + B^2} \tag{1}$$

Perceived brightness formula

$$brightness = \sqrt{0.241 * R^2 + 0.691 G^2 + 0.068 B^2} \tag{2}$$

[Dobovizki, 2022]

Perceived brightness results

Figure: Brightness: 122.8

Figure: Brightness: 107.2

Resulting data

- Data set holding data for each picture per host.
  - huge number of columns/variables
- Data set summarizing the results:
  - sums per detected object per host
  - means of brightness and cct per host

Table of Contents

Plan

- Way to proceed was dynamic

Given this resources constrain and aiming to fulfill interpretability requirement

- Plan: Use DNN for image scrapping and then employ a regularization model (Lasso) for variable selection.
- Expanded to include further "competitive" models (GBM, Random Forests)

Partitioning and Data Analysis

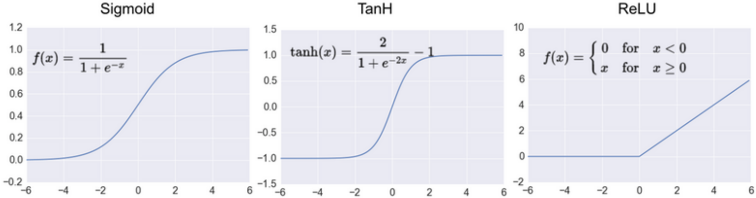
- We initially worked for Berlin working with a partition of 80:10:10 for train validation and test
- Then, Munich came $\rightarrow$ Berlin $90:10$ and Munich fully used as test set
- Munich demanded an analysis of the data by the same criteria as Berlin i.e Absence $>$ Na Rates $>$ Irrelevance

Munich data set before 4995
After cleaning: 3222
Lost rate: 0.354955%

Table of Contents

Hyperparameter Tuning

- k fold cross-validation
- Method: adaptive cv
- Results:
  - activation function: TanH
  - dropout: 0.17



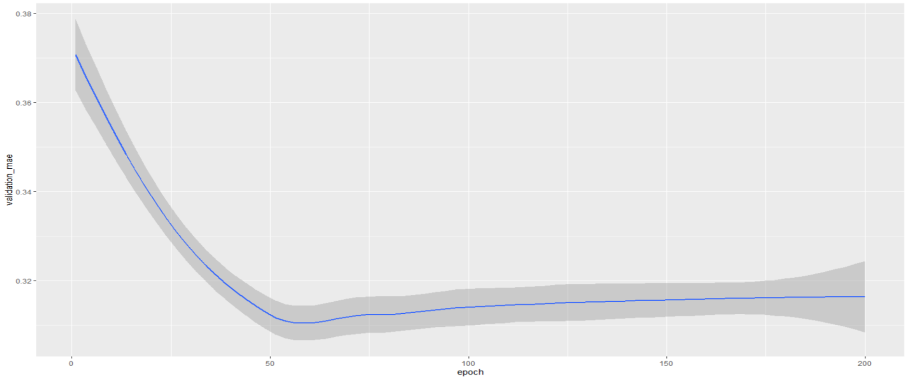
[Develop Paper, 2020]

- adaptive learning rate: rmsprop (Root Mean Square Propogation)

DNN Summary

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_184 (Dense)            (None, 189)               12096

dropout_138 (Dropout)        (None, 189)               0

dense_183 (Dense)            (None, 126)               23940

dropout_137 (Dropout)        (None, 126)               0

dense_182 (Dense)            (None, 63)                8001

dropout_136 (Dropout)        (None, 63)                0

dense_181 (Dense)            (None, 1)                 64

=================================================================
Total params: 44,101
Trainable params: 44,101
Non-trainable params: 0
_________________________________________________________________
```

Training Curves



Lasso

Models to be trained:

1. OLS for reference
2. No data preprocessing
3. Normalized i.e location and scale
4. Normalized i.e location and scale with log(price)

Parameters to tune

- Parameter to tune: λ i.e shrinkage parameter.
- Grid: from 10^{10} to 0.01
- best λ: 1.14

[James et al., 2021]

Training times on Berlin data set: circa 3 Minutes

GBM

Models to be trained:

1. No data preprocessing
2. Normalized i.e location and scale
3. Normalized i.e location and scale with log(price)

Tuning was attempted for every parameter.
Computationally prohibitive i.e. failed after 24 hours CPU time

Parameters to tune

- interaction.depth $= 1$
- shrinkage seq(0.001, 0.202, 0.04). Best: 0.001
- n.trees $= 5000$
- n.minobsinnode 10

[James et al., 2021]

Training times on Berlin Data Set: circa 16 hours.

Random Forests

1. No data preprocessing
2. Normalized i.e location and scale
3. Normalized i.e location and scale with log(price)

Parameters to tune

- mtry $= -7 + p/3$, $p/3$, $7 + p/3$ with $p/3 = 21$. Best: 21
- min.node.size $= 5$

[Hastie et al., 2009]

Training times on Berlin Data Set: circa 3 hours.

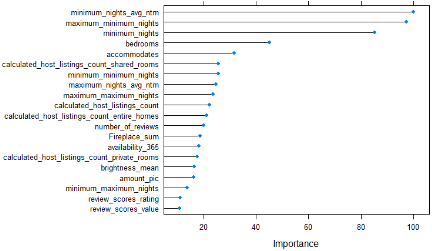
Table of Contents

Results Berlin

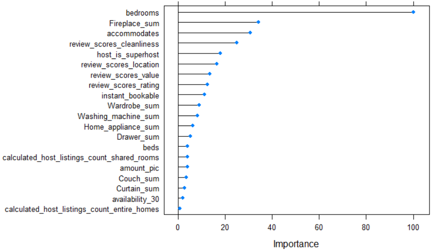
	RMSE <dbl>	Rsquared <dbl>	MAE <dbl>
OLS	0.5547540	0.3818374	0.4107917
Lasso	0.4935600	0.4465568	0.3757873
Lasso Standard	0.4935600	0.4465568	0.3757873
Boost	0.5088589	0.4297071	0.3963260
RF	0.4085395	0.6140226	0.3083759
RF Centered	0.4087117	0.6139556	0.3084462
Lasso S + log(price)	0.4436540	0.5181270	0.3442663
DNN	0.3807523	0.6436791	0.2860852

Variable Importance Berlin

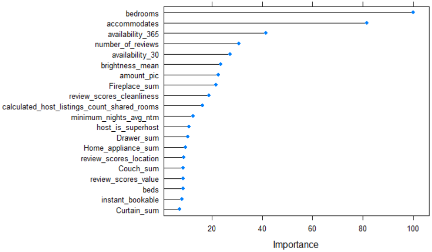
OLS



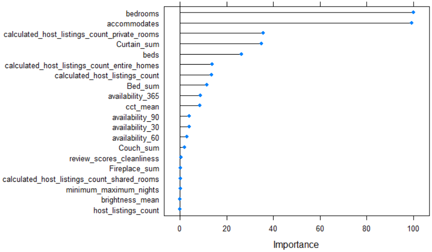
Lasso



Lasso Centered



Boost

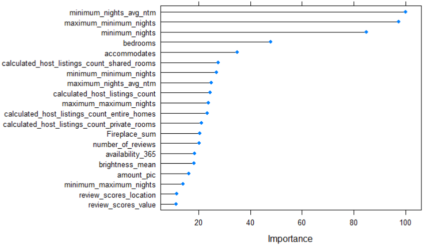


Results Munich

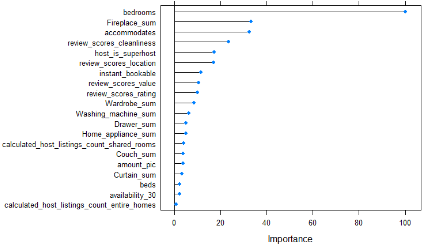
	RMSE <dbl>	Rsquared <dbl>	MAE <dbl>
OLS	0.6330120	0.2460199	0.4687229
Lasso	0.5868852	0.2998020	0.4312133
Lasso N	0.5868852	0.2998020	0.4312133
Boost	0.6560061	0.2191735	0.4775103
Boost N	0.6575854	0.2427723	0.4705026
RF	0.5270313	0.3860209	0.3767017
RF N	0.5304562	0.3782364	0.3798950
Lasso N-log(price)	0.6404690	0.2869786	0.4614091
Boost N-log(price)	0.5896727	0.3410397	0.4201564
RF N-log(price)	0.5456279	0.4245355	0.3867661
DNN	0.8698481	-0.7196968	0.4800993

Variable Importance Munich Lasso

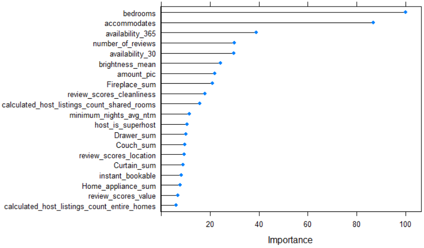
OLS



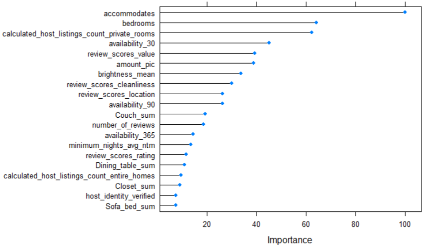
Lasso



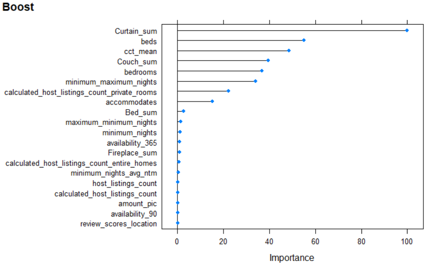
Lasso N

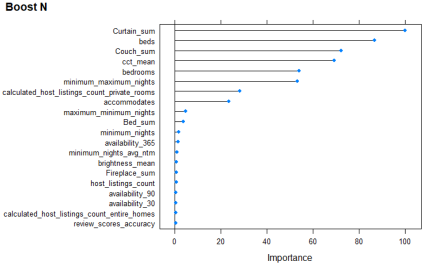


Lasso N-log(price)



Variable Importance Munich Boost





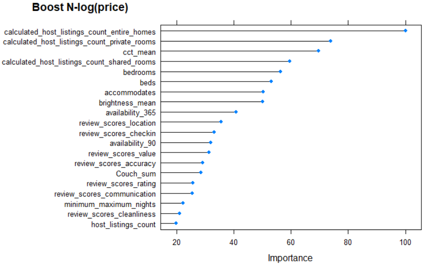


Table of Contents

Conclusions

Takeaways

- Lacking computational power
- further theoretical analysis: Outliers, Variable Exclusion pre-training
- vary threshold for multi detection model
- increase cluster amount for CCT analysis
- scene identification to exclude pictures not showing the flat

Table of Contents

Literature and further References I

Aguirre-Pablo, A. A., Alarfaj, M. K., Li, E. Q., Hernández-Sánchez, J. F., and Thoroddsen, S. T. (2017).
Tomographic particle image velocimetry using smartphones and colored shadows.
Scientific reports, 7(1):3714.

Develop Paper (2020).
Activation function of attention mechanism: adaptive parameterized relu activation function - develop paper.

Dobovizki, N. (18.01.2022).
Calculating the perceived brightness of a color.

Girshick, R., Donahue, J., Darrell, T., and Malik, J.
Rich feature hierarchies for accurate object detection and semantic segmentation.

Literature and further References II

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009).
The elements of statistical learning: Data mining, inference, and prediction / Trevor Hastie, Robert Tibshirani, Jerome Friedman.
Springer series in statistics. Springer, New York, 2nd ed. edition.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021).
An Introduction to Statistical Learning: With Applications in R.
Springer eBook Collection. Springer US and Imprint: Springer, New York, NY, 2nd ed. 2021 edition.