Schätzung von Hauspreisen in Ames, Iowa mit Hilfe von Regressionsbäumen

David Weghorn

13. Juli 2018

Inhaltsverzeichnis

1. Einleitung

- Vorstellung einer beliebten Machine Learning Methode
- Simulation von Schwächen der Methode und Darstellung von Lösungen

2. Vorstellung des Datensatzes

Beschreibung von 1460 Hausverkäufen in Ames, Iowa

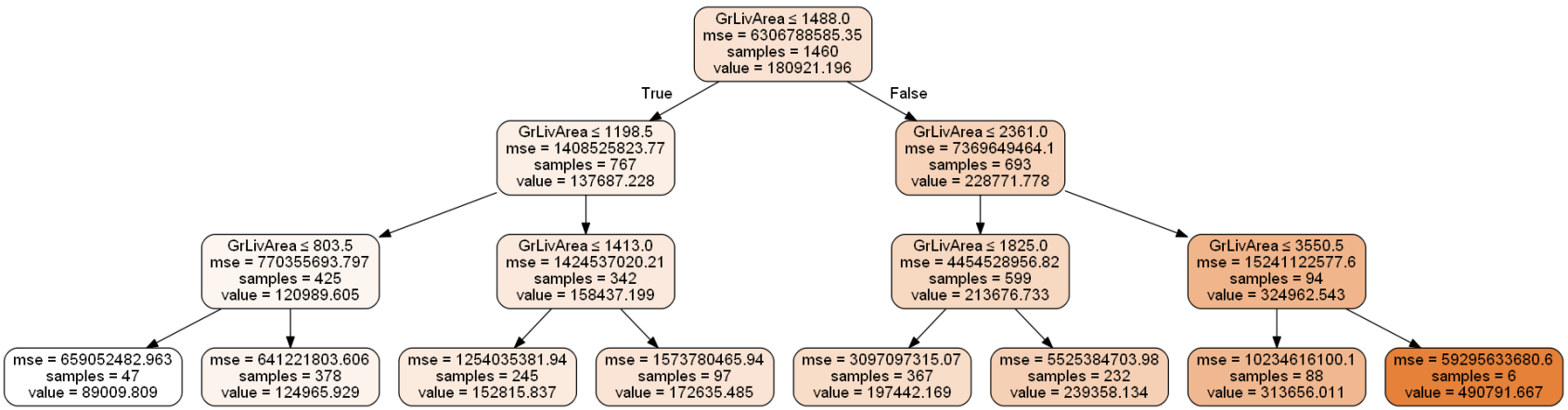
- Zielvariable: SalePrice in USD
- 79 erklärende Variablen:
  - Eigenschaften des Hauses
    (Garagenfläche, Qualität der Räume, Bauweise, Lage,...)
  - Umstände des Hausverkaufs
    (Zeitpunkt, Art der Bezahlung)

3.1 Theoretische Grundlage Regressionsbäume

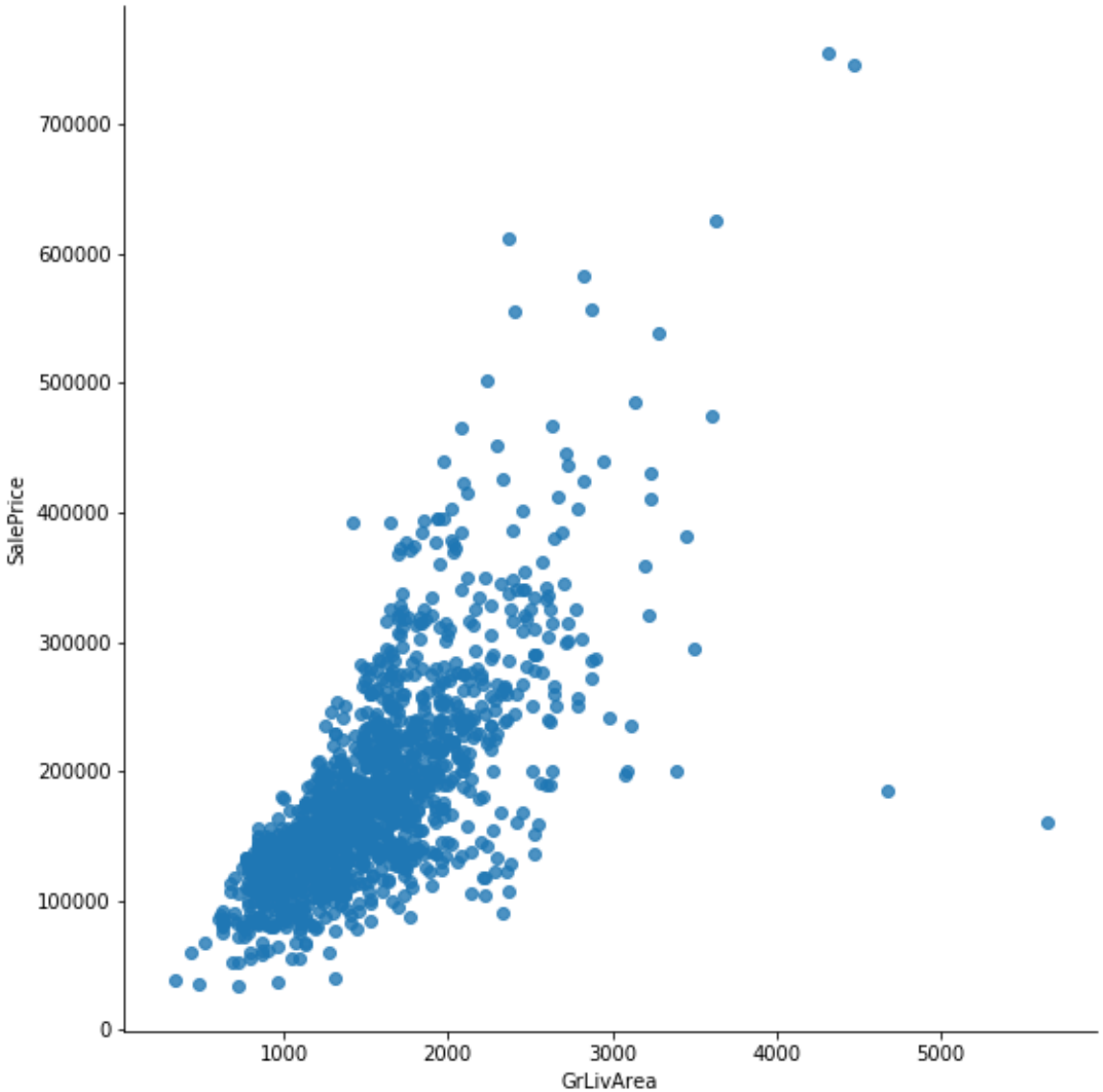
Konstruktion von Regressionsbäumen

- Unterteilung des Datensatzes:
  - $\min_{j,s}[\sum_{x_i \in R_m(j,s)}(y_i - c_m)^2 + \sum_{x_i \in R_{m+1}(j,s)}(y_i - c_{m+1})^2]$
    mit j als Split-Variable und s als Split-Punkt
  - $\hat{c}_m = \frac{1}{n}\sum_{i=1}^{N}(y_i | x_i \in R_m(j,s))$

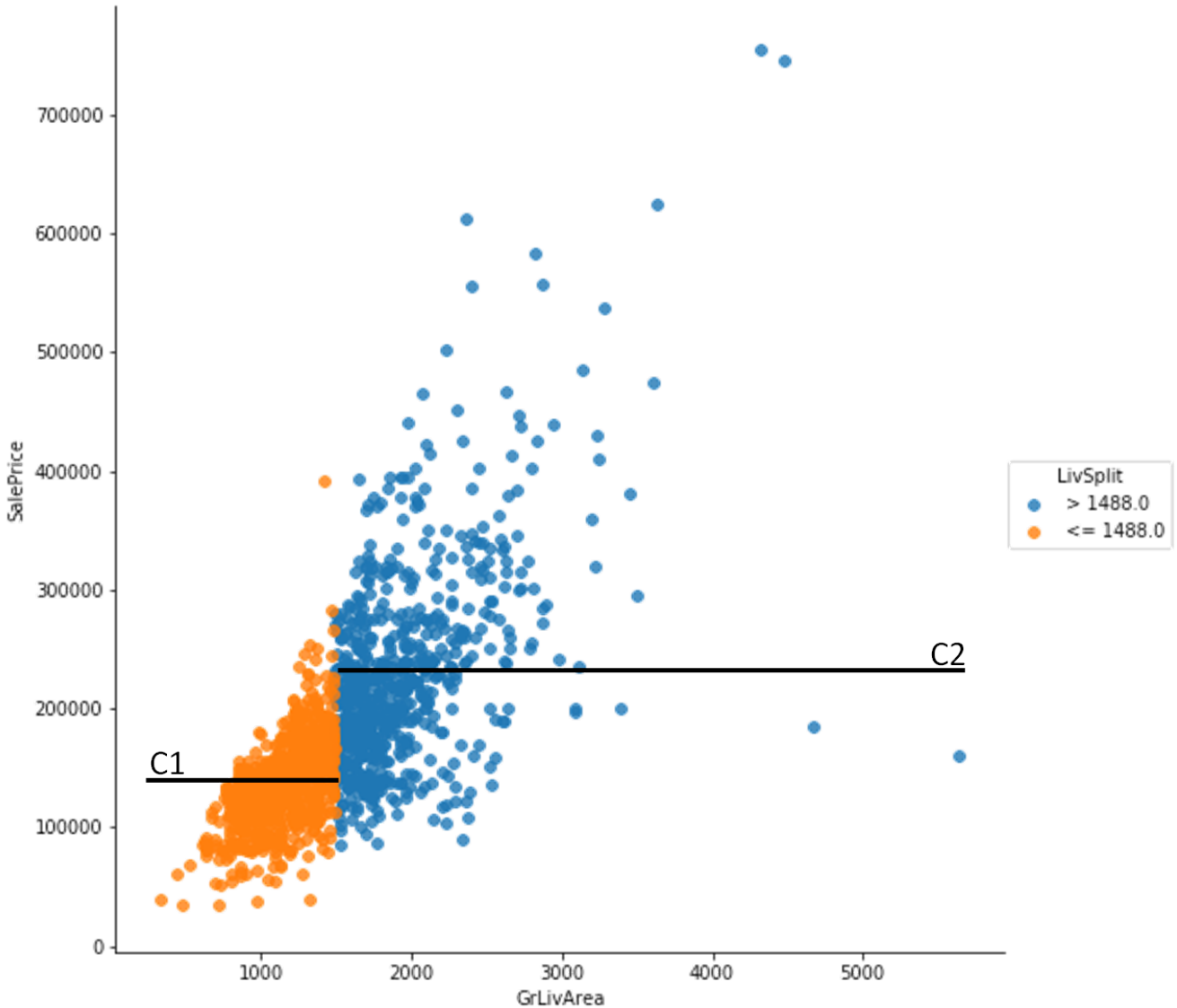
3.2 Grundlage Regressionsbäume 2D Fall



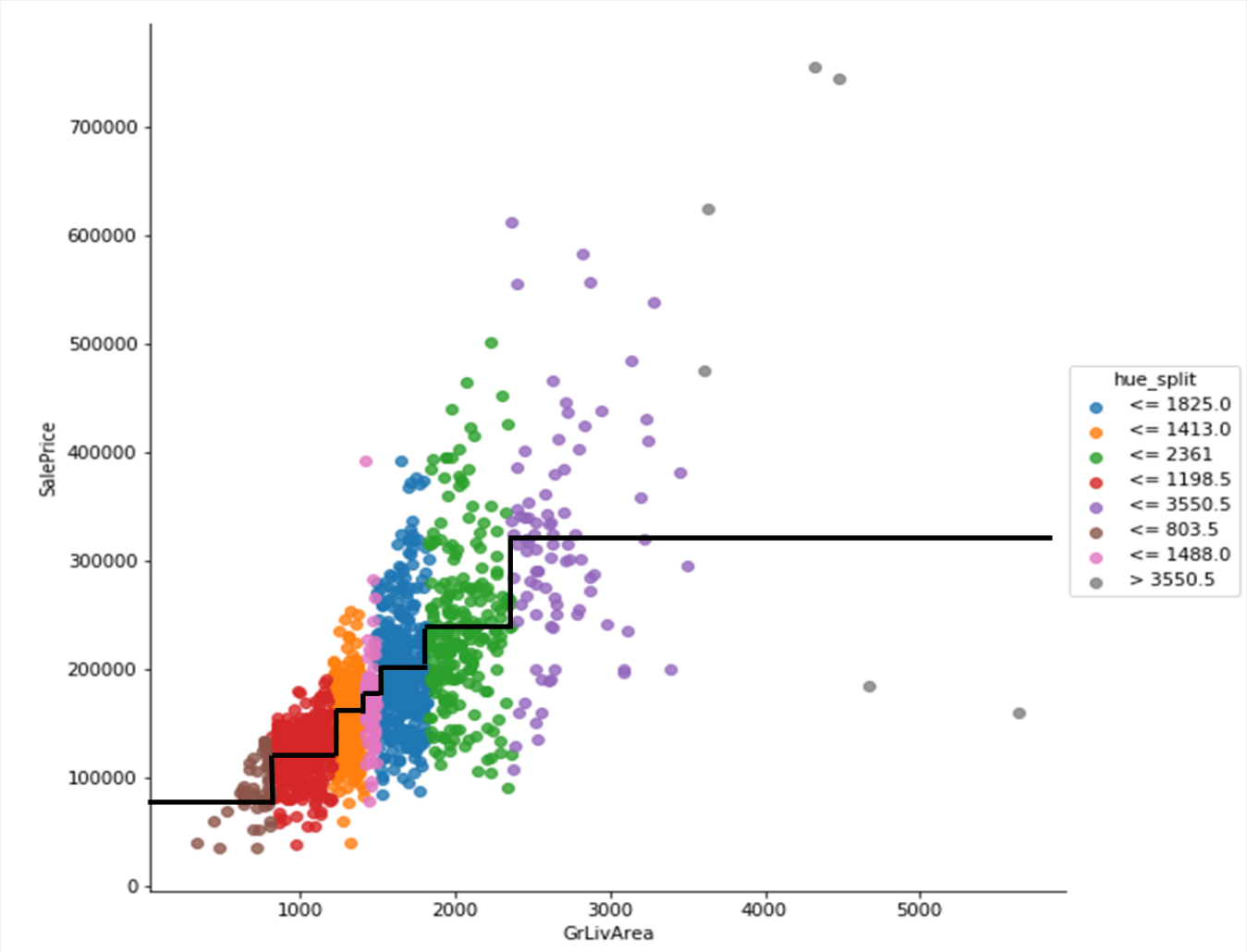
3.2 Grundlage Regressionsbäume - 2D Fall



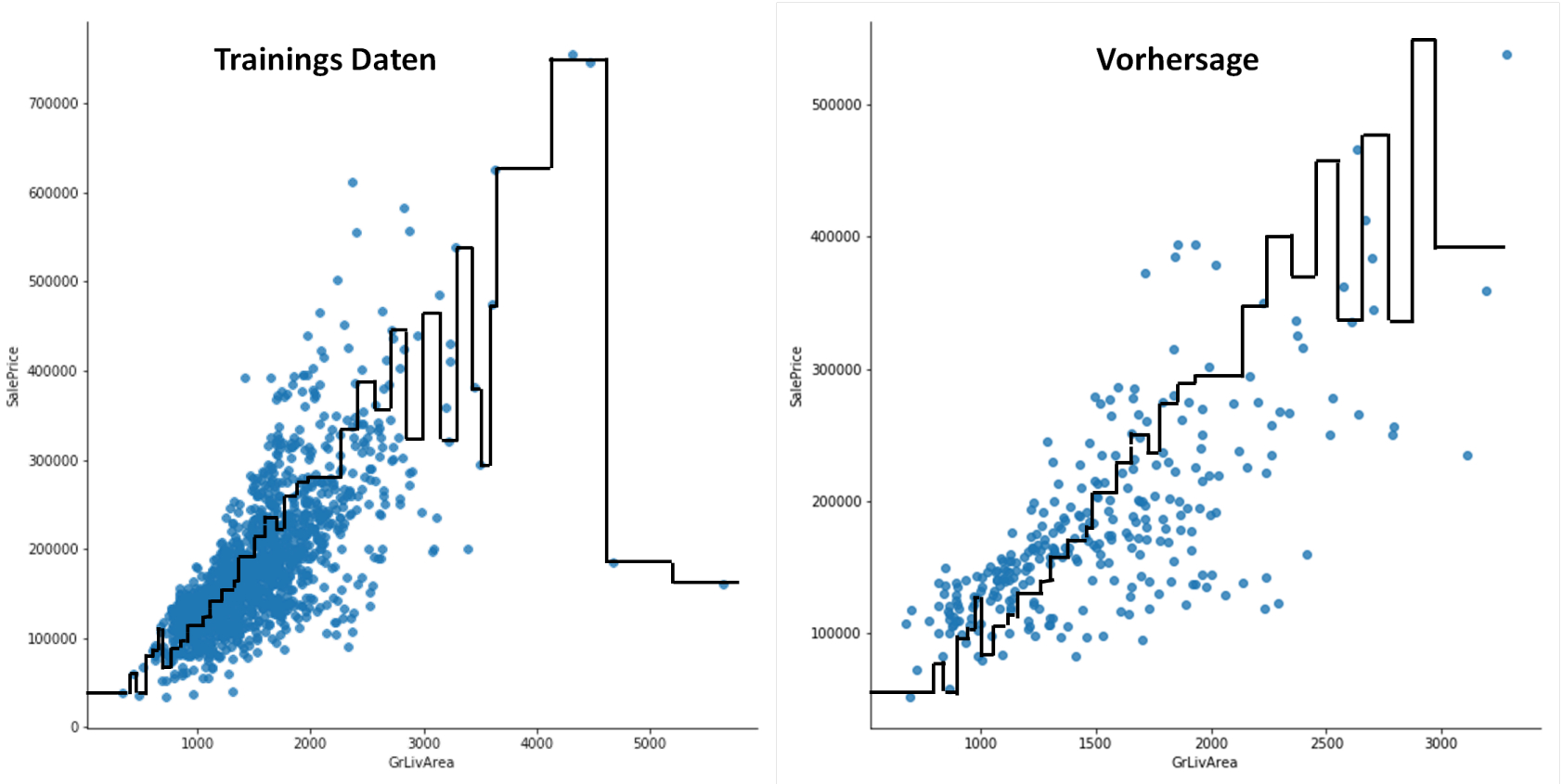
3.2 Grundlage Regressionsbäume



3.2 Regressionsbaum mit 3 Unterteilungskonditionen



4.1 Schwächen von Regressionsbäumen - Overfitting



Overfitting führt zu extrem guter Anpassung an Trainingsdatensatz aber zu schlechter Vorhersagekraft

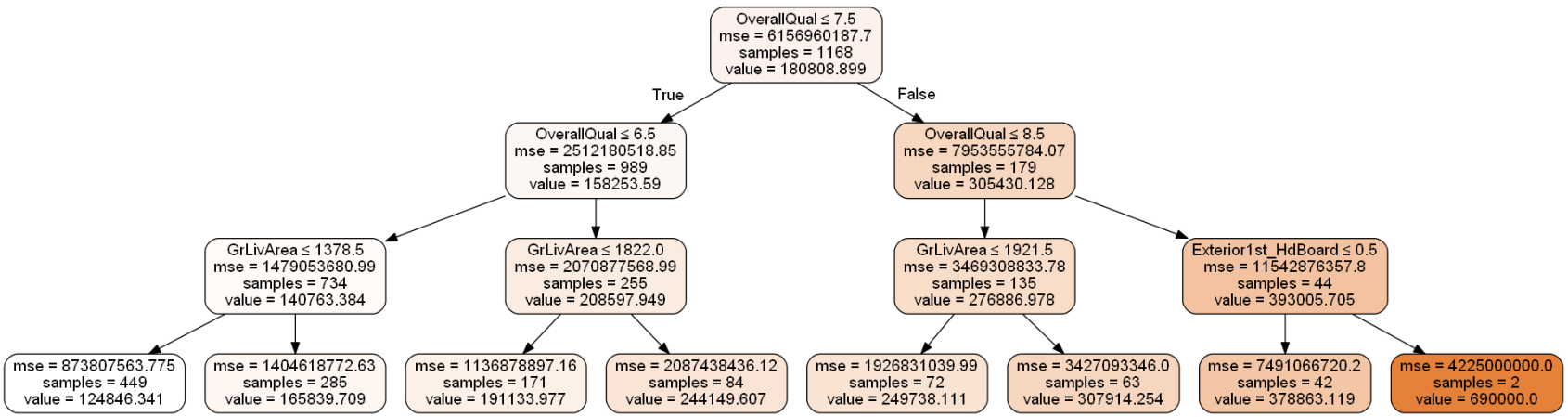
4.2 Instabilität / Varianz

Simulation

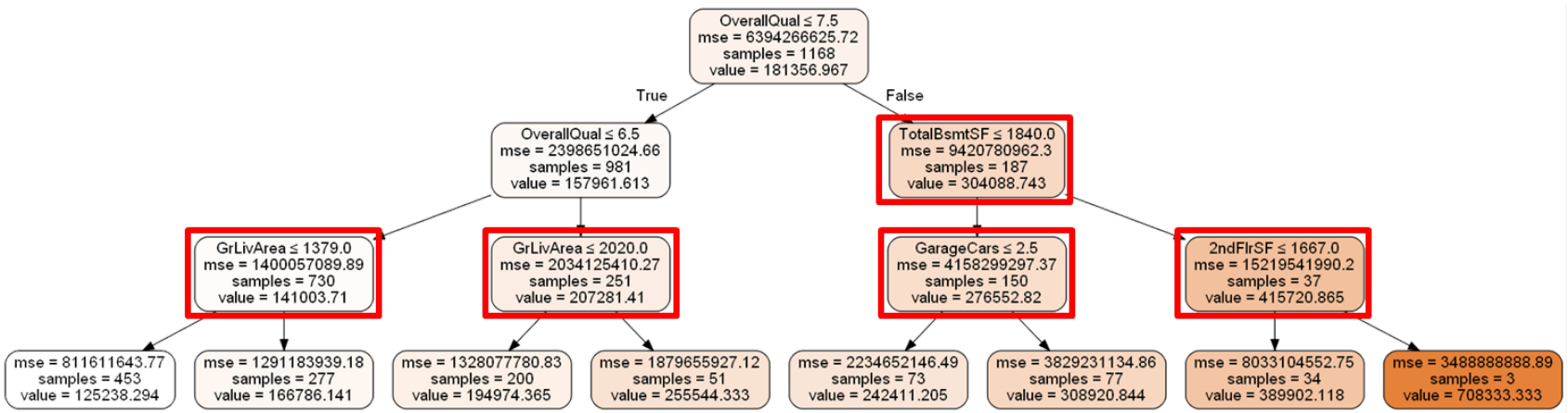
1. Ziehe Zufallsstichproben (80%) aus dem Trainingsdatensatz
2. Trainiere Regressionsbaum für jede Stichprobe
3. Visualisiere Bäume und vergleiche Unterteilungsknoten

$\Rightarrow$ Erwartung bei herkömmlichen Verfahren:
geringe Varianz der Koeffizienten und Vorhersage

4.2 Instabilität / Varianz - Stichprobe 1



4.2 Instabilität / Varianz - Stichprobe 2



4.2 Schwächen von Regressionsbäumen

Erwartung:
geringe Varianz der Koeffizienten und Vorhersage

Aber Beobachtung: Regressionsbäume der verschiedenen Stichproben variieren in

- den Variablen die zur Unterteilung herangezogen werden
- den Werten an denen die Unterteilung durchgeführt wird

$\Rightarrow$ Problem für die Interpretation und Belastbarkeit der Vorhersagen

5. Kombinierte Schätzer - Ensemble

- Die Kombination der Schätzer verringert Varianz
  und Gefahr des Overfittings des Trainingsdatensatzes
- Vorstellung verschiedener Verfahren:
  - einfache Kombination - Bagging (Randomisierung Datensatz)
  - gewichtete Kombination (Randomisierung Datensatz)
  - Random Forest (Randomisierung erkl. Variablen)
- Nachteil:
  Einfache Darstellung bzw. Interpretation gehen verloren

5.1 Bagging - pseudo code

1. set parameters

$training\_data = train_{data}$
$test\_data = test_{data}$
$s \in [0,1]$

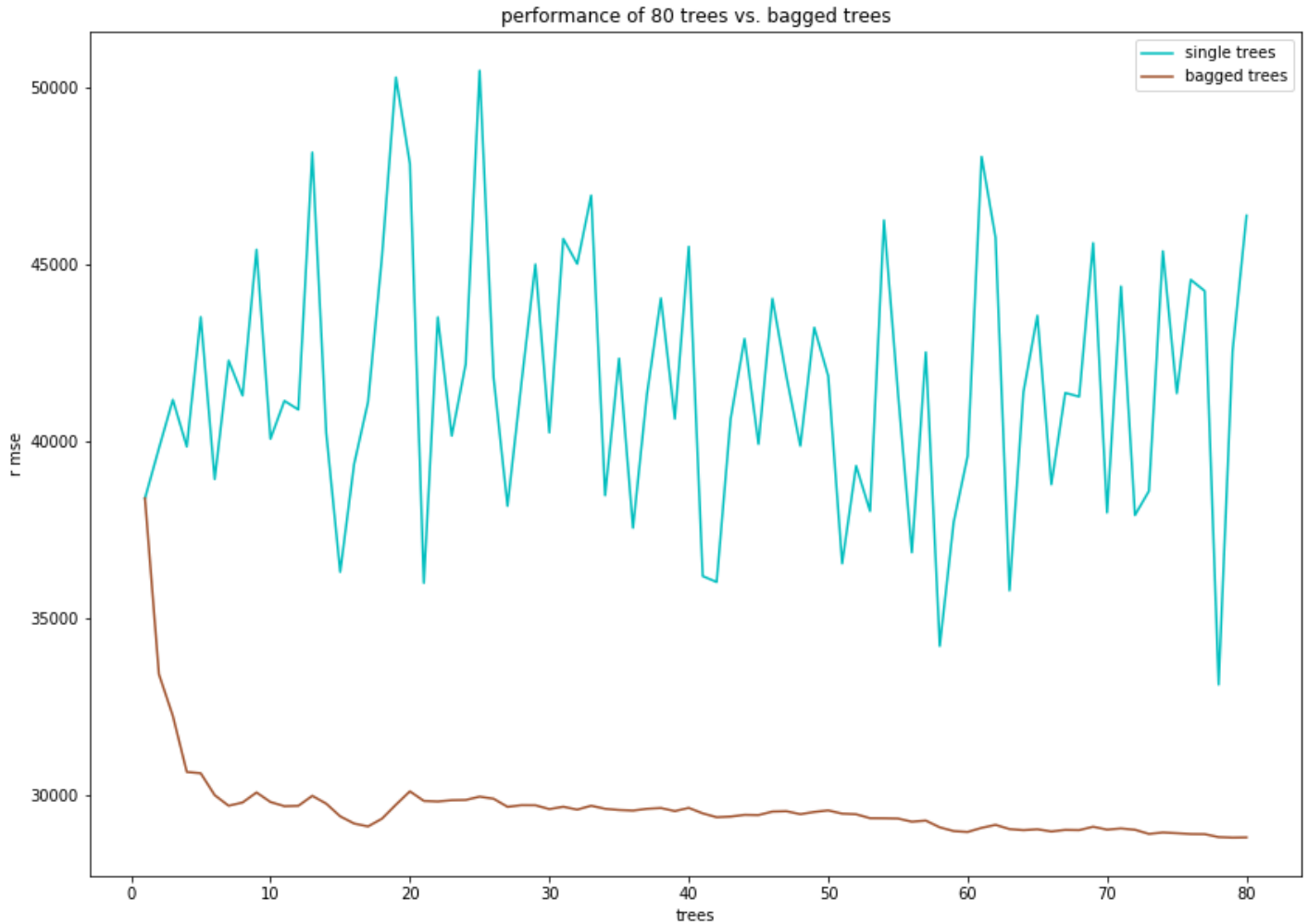
2. train single trees

for each n=1 to N:
$train_{sample}$ = random sample from $train_{data}$(size=s)
tree.fit($train_{sample}$)
$prediction_i$ = tree.predict($test_{data}$)

3. combine trees to ensemble

for each n=1 to N:
$prediciton_{ensemble,i} = \frac{1}{n}\ prediction_i + \ldots + \frac{1}{n}\ prediction_N$

5.1 Kombinierter Schätzer - Bagging



5.2 Gewichteter Schätzer - pseudo code

1. set parameters

$\text{training\_data} = \text{data}_{train}$
$\text{test\_data} = \text{data}_{test}$
$s \in [0,1]$

2. train single trees and evaluate fit

for each n=1 to N:

$\text{sample}_{train} = \text{random sample from data}_{train}(\text{size}=s)$
$\text{tree.fit}(\text{train}_{sample})$
$\text{prediction}_i = \text{tree}_i.\text{predict}(\text{data}_{test})$

$sample_{weight} = data_{train} - sample_{train}$
$prediction_{weighting,i} = tree.predict(sample_{weighting})$

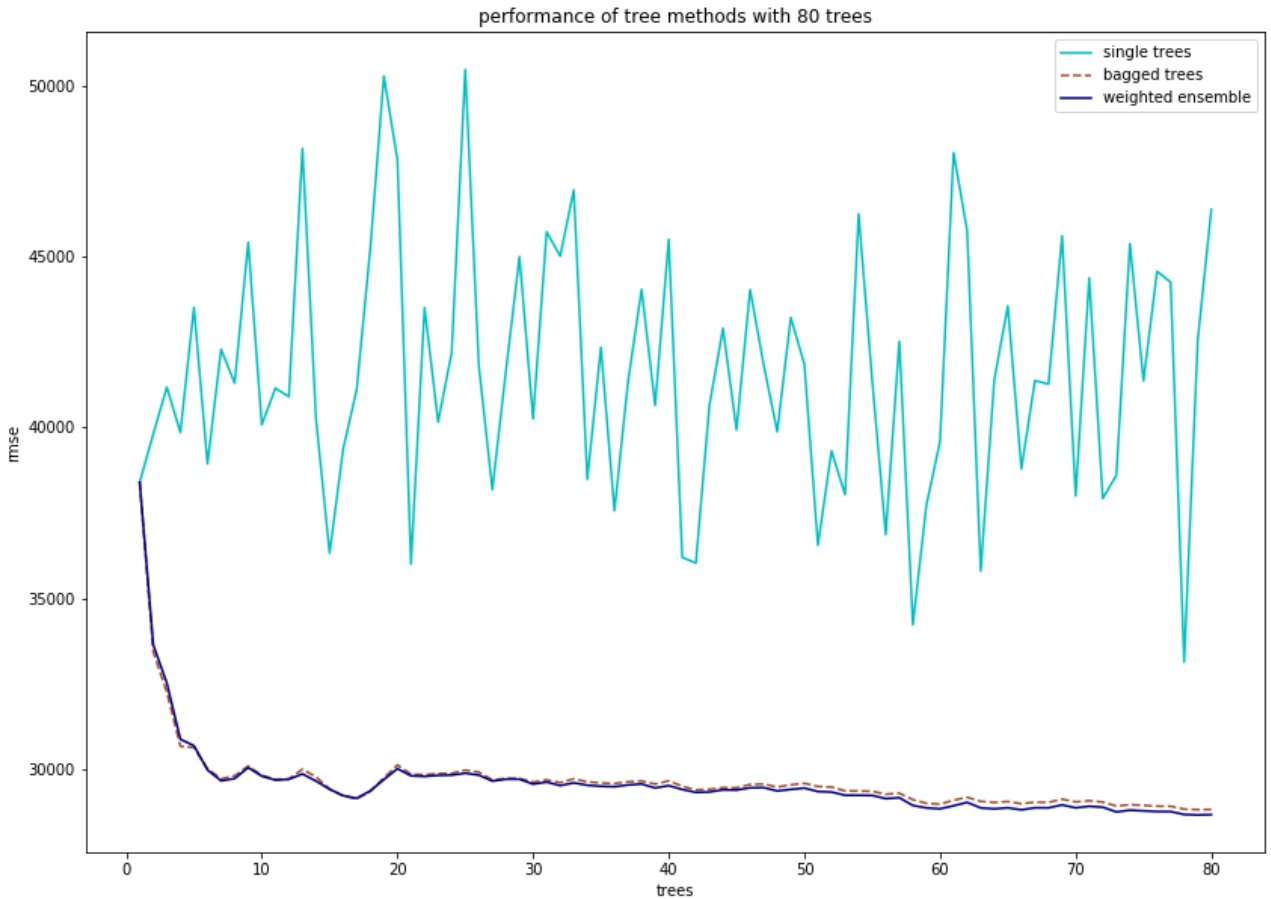
$$mse_{inverse,i} = \frac{1}{mse_i(predictions_{ensemble}, data_{test})}$$

$$weight_i = \frac{mse_{inverse,i}}{\sum_{n=1}^{N} mse_{inverse,i}}$$

3. combine trees to ensemble

for each n=1 to N:
$\text{prediciton}_{ensemble,i} = \textbf{weight}_i \times \text{prediction}_i + \ldots + \textbf{weight}_N \times \text{prediction}_N$

5.2 Gewichteter Schätzer



5.3 Random Forest - pseudo code

1. set parameters

training_data=$data_{train}$
test_data=$data_{test}$
s $\in$ [0,1]
$x \in [1, \#features]$, *usually* $\sqrt{\#features}$

2. train single trees

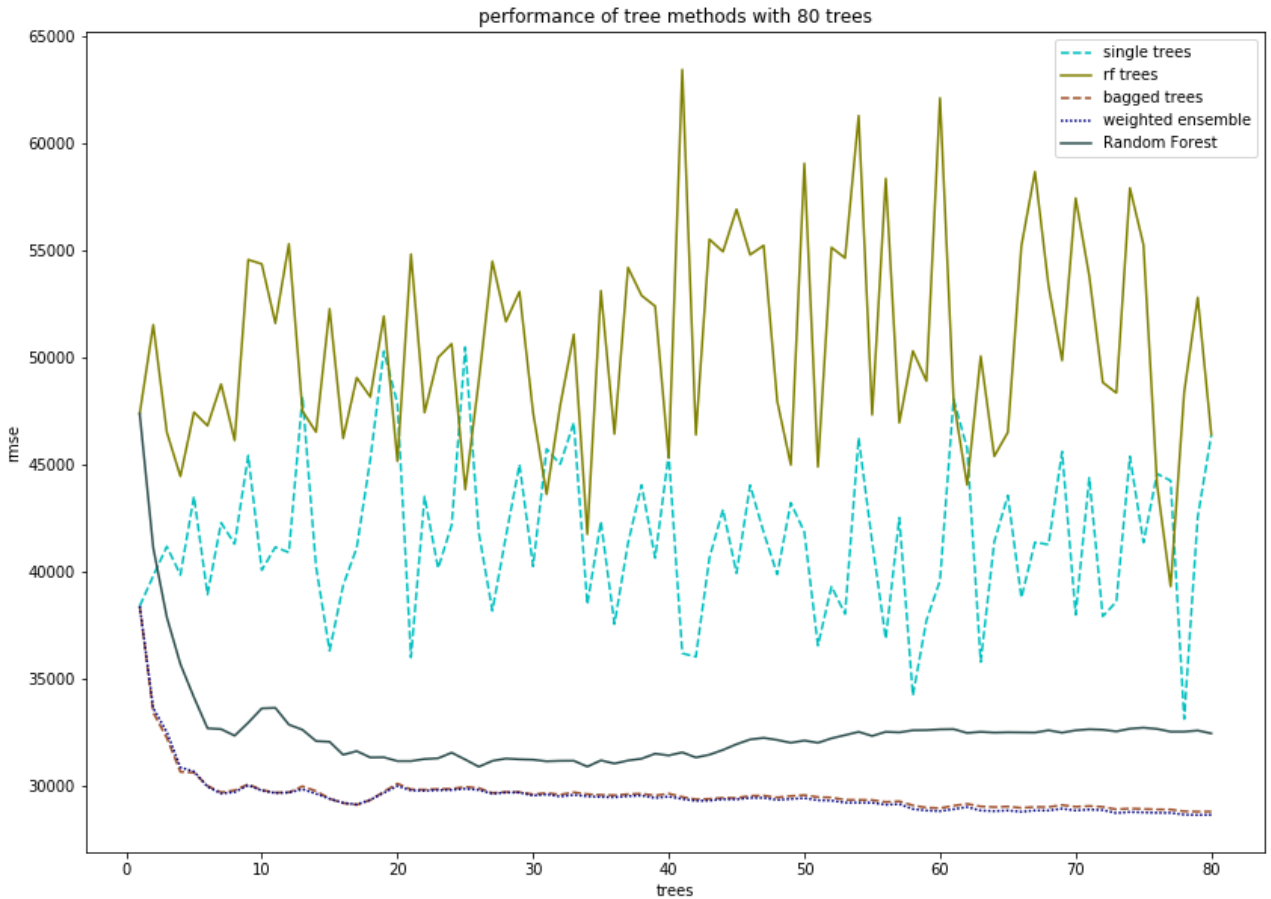
```
for each n=1 to N:
    randomly select x features
        sample_train = random sample from data_train(size=s)
        tree.fit(sample_train)

        prediction_i = tree_i.predict(data_test)
```

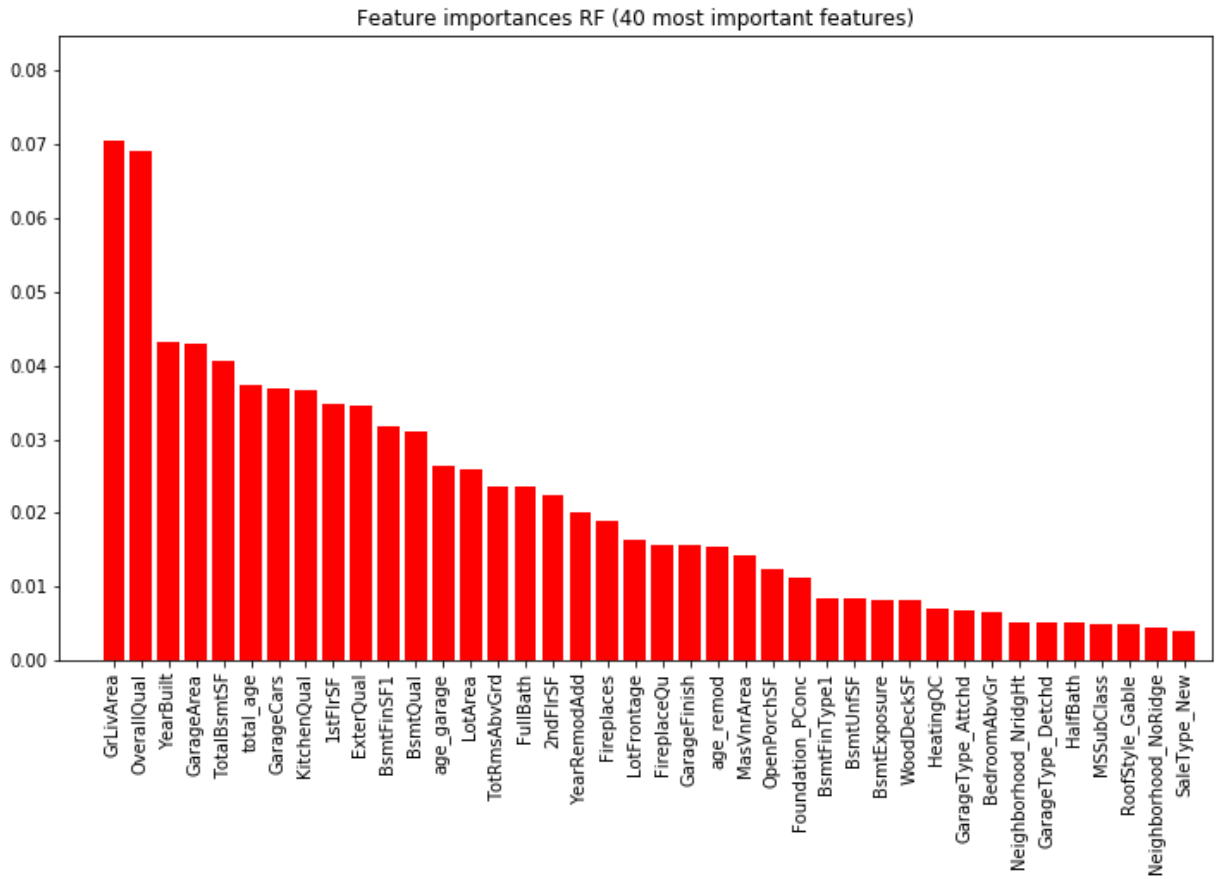
3 combine trees to ensemble

```
for each n=1 to N:
        prediciton_ensemble,i = 1/n × prediction_i + ... + 1/n × prediction_N
```

5.3 Random Forest



5.3 Random Forest - Einfluss der erklärenden Variablen



Einfluss: Summe der totalen MSE Reduktion der Variable gewichtet mit der Anzahl der unterteilten Beobachtungen und Gesamtzahl der Bäume

6. Zusammenfassung

Vorhersagekraft der angewandten Methoden

	Einzelbaum	**Kombination**	**Gewichtet**	**Random Forest**
RMSE	41.567	29.381	29.136	30.960
R^2	0,773	0,892	0,893	0,875

6. Zusammenfassung

- Regressionsbäume stellen ein einfach zu implementierendes Maschinelles Verfahren dar
- Regressionsbäume könne auch nichtlineare Zusammenhänge gut modellieren
- Hohe Varianz bei kleinen Veränderungen des Trainingsdatensatzes birgt Probleme
- Randomisierung der Variablen oder Daten und Kombination der Schätzer verbessern Schätzung enorm

Questions

Vielen Dank für die Aufmerksamkeit
Fragen?

Quellen

Sammlung der Python Codes:
www.github.com/Davekofski/seminar_paper
Datensatz:
www.kaggle.com/c/
house-prices-advanced-regression-techniques
Dokumentation: ww2.amstat.org/publications/jse/v19n3/
decock/DataDocumentation.txt
Literatur:
Breiman, L. Machine Learning (1996): 24: 123 "Bagging predictors".
Hastie, T; R. Tibshirani, J. Friedman (2009): "Elements of Statistical Learning", Chap. 8, 9, 15.
James, G, D. Witten, T. Hastie, R. Tibshirani (2013): "An Introduction to Statistical Learning", Chap. 8.

Appendix

APPENDIX

3.1 Theoretische Grundlage Regressionsbäume

Schätzung mit Regressionsbäumen

- Unterteilung des Datensatzes:
  - $\min_{j,s}[\min_{c_m} \sum_{x_i \in R_m(j,s)} (y_i - c_m)^2 + \min_{c_m+1} \sum_{x_i \in R_{m+1}(j,s)} (y_i - c_{m+1})^2]$
    mit j als Split-Variable und s als Split-Punkt
  - $\hat{c}_m = ave(y_i | x_i \in R_m(j,s))$
- Vorhersage:
  - $\hat{f}(X) = \sum_{m=1}^{N} c_m I\{(X_1, X_2) \in R_m\}$

Vorbereitung der Variablen

Unterteilung der Variablen

1. kontinuierliche Variablen(30)
2. kategorische Variablen (18):
   - ordinale "Qualitätsvariablen" (10)
   - andere ordinale Variablen (8)
3. nominale Variablen (27)
4. Datumsangaben (4)

$\Rightarrow$ nach Bearbeitung insgesamt 342 Variablen (viele Dummy Variablen)

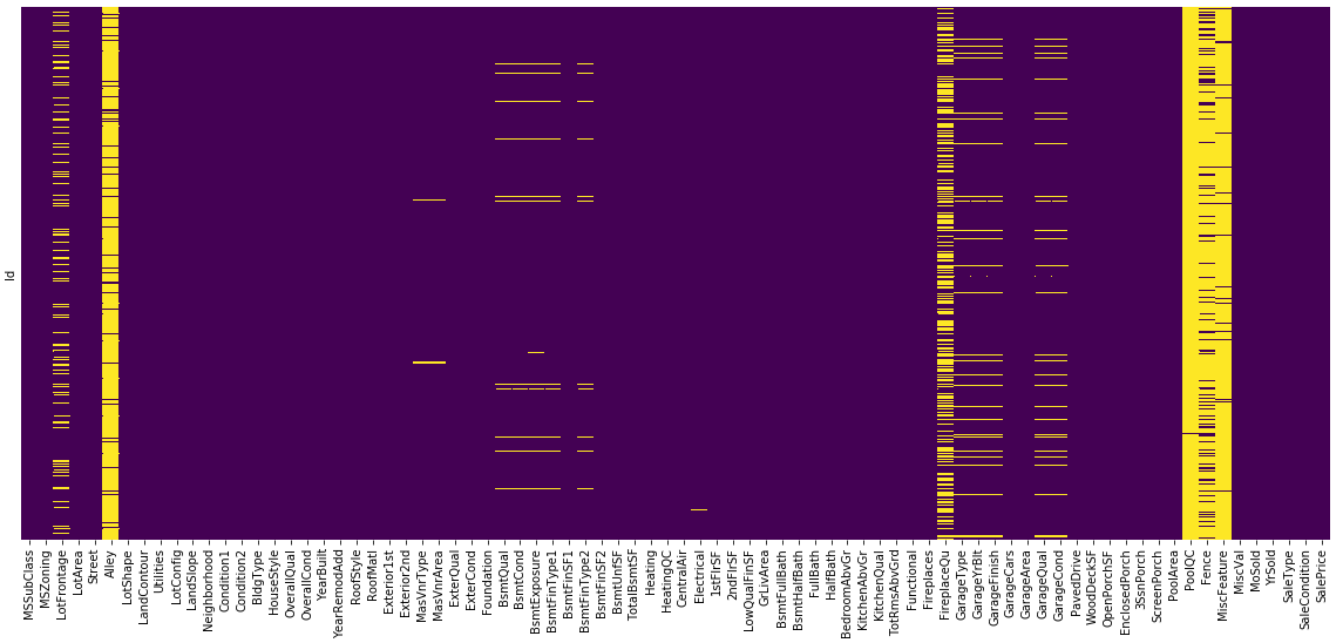
Vorbereitung der Variablen

Unterteilung der Variablen

1. kontinuierliche Variablen(30): keine weitere Bearbeitung
2. kategorische Variablen (18):
   - ordinale "Qualitätsvariablen" (10):
     Excellent: 4, Good: 3, Average: 2, Fair: 1, NA: 0
   - andere ordinale Variablen (8):
     individuelle Codierung
3. nominale Variablen (27): Dummy Variablen (OneHotEncoding)
4. Datumsangaben (4): neue "Altersvariablen"
   + Dummys für Jahr (Verkauf, Erbauung) und Monat (Verkauf)

$\Rightarrow$ insgesamt 342 Variablen (viele Dummy Variablen)

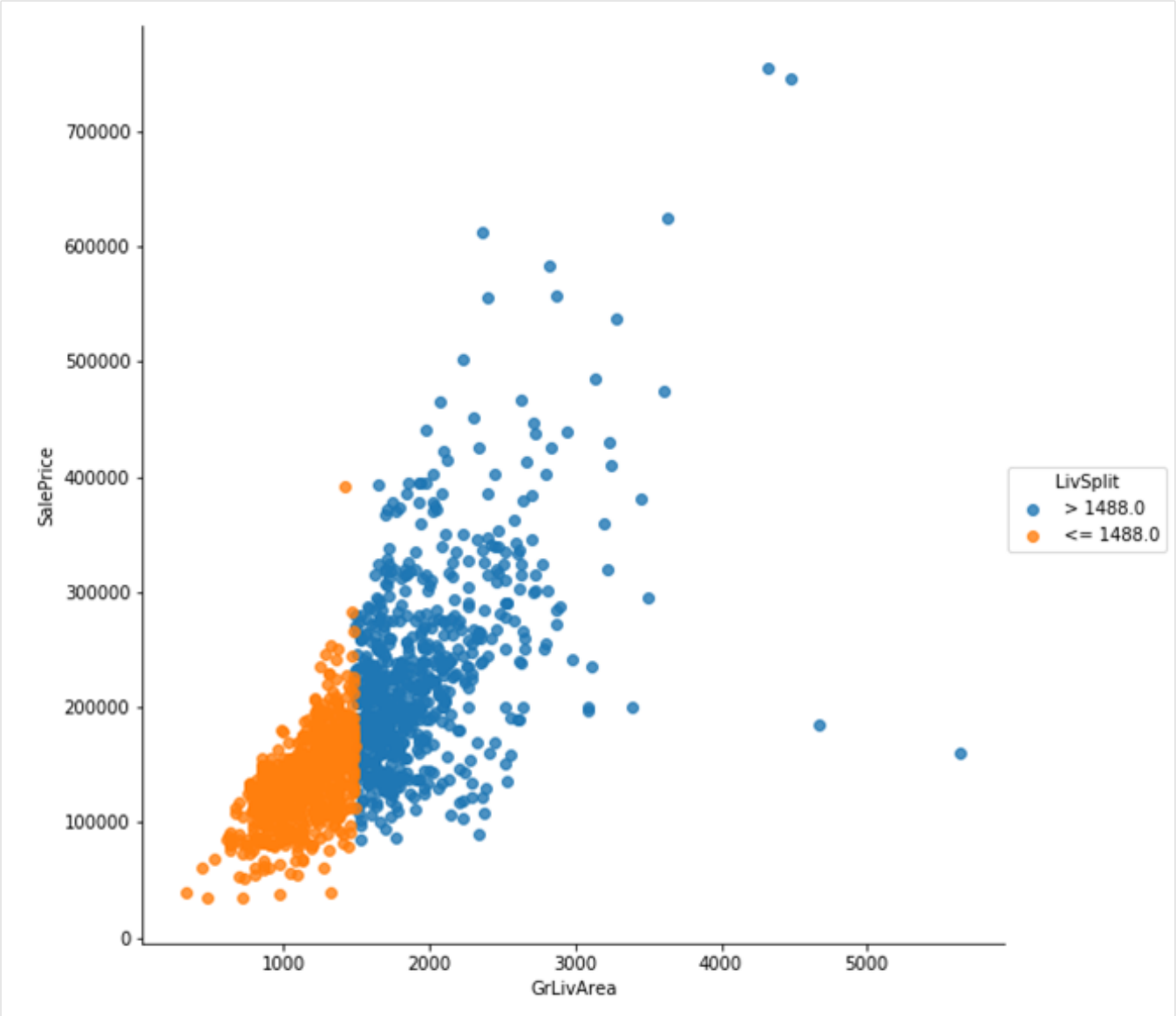
Fehlende Beobachtungen



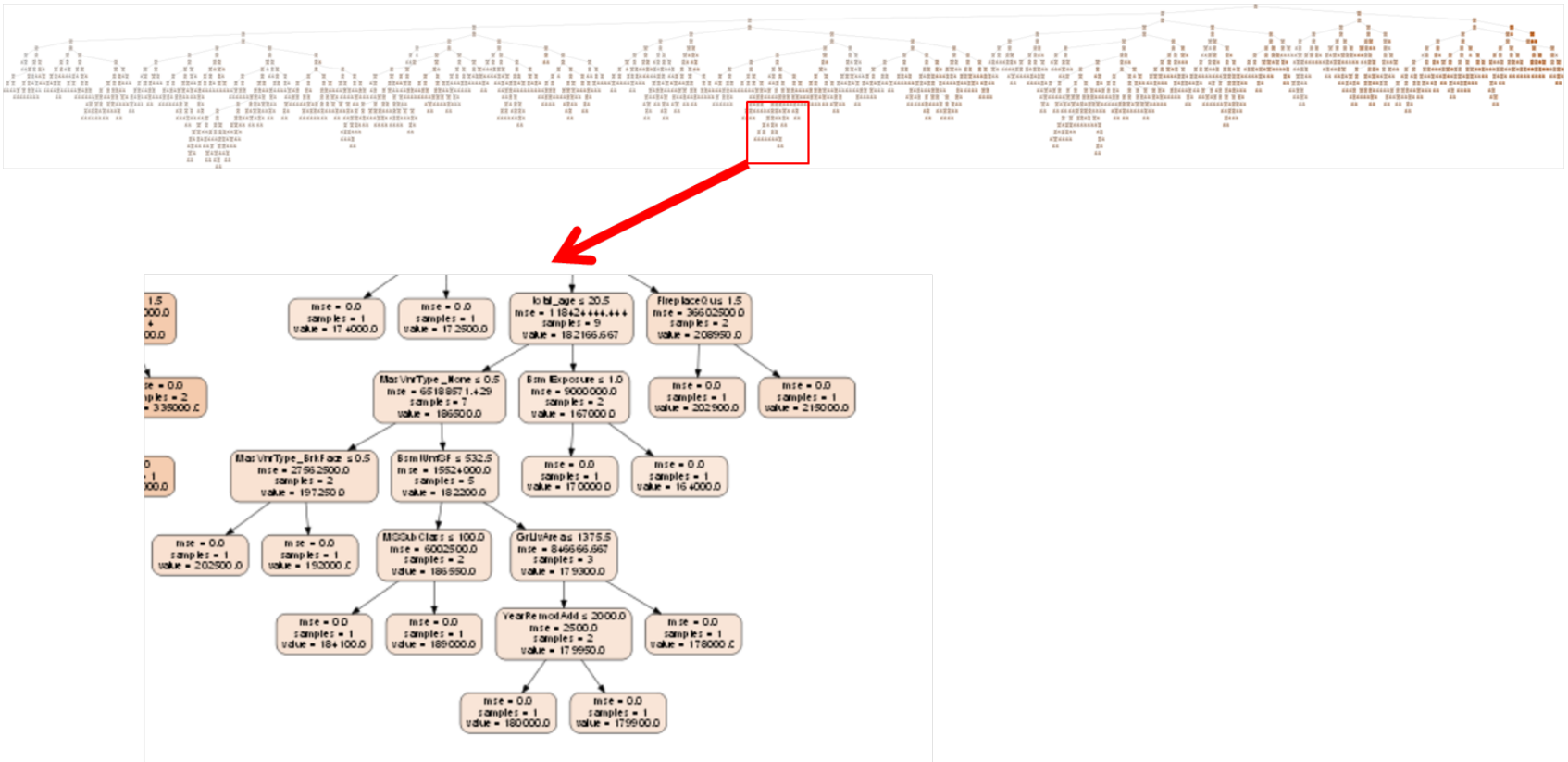
Behandlung fehlender Variablen

- "Qualitäts- und Zustandsvariablen"
  (Keller, Küche, Garage, Pool, Elektrik, Zaun,...)
  fehlende Variablen $\Rightarrow$ nicht vorhanden: mit 0 ersetzen
- numerische Variablen: Median einsetzen

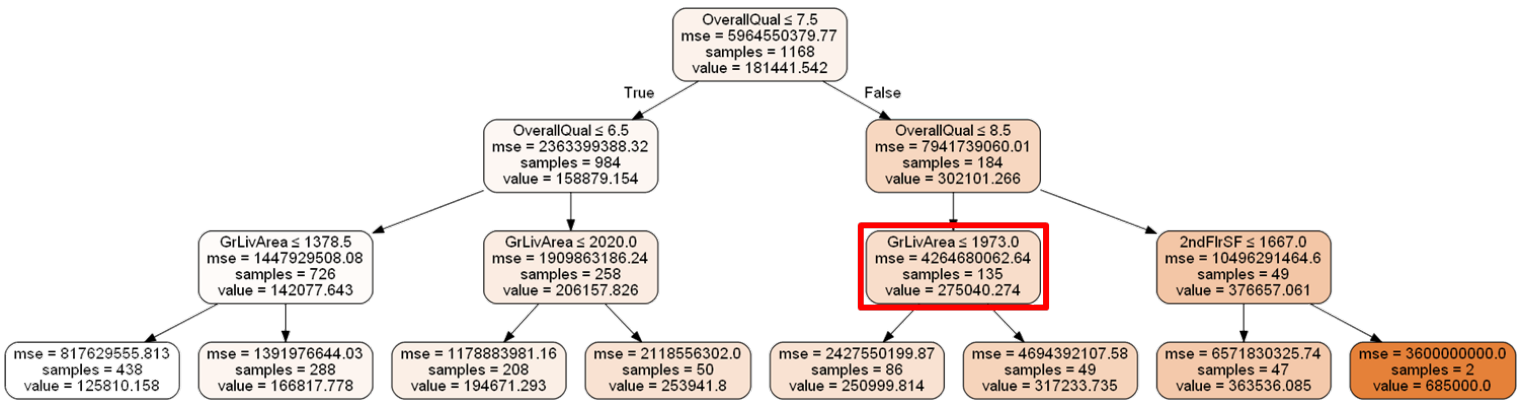
Grundlage Regressionsbäume



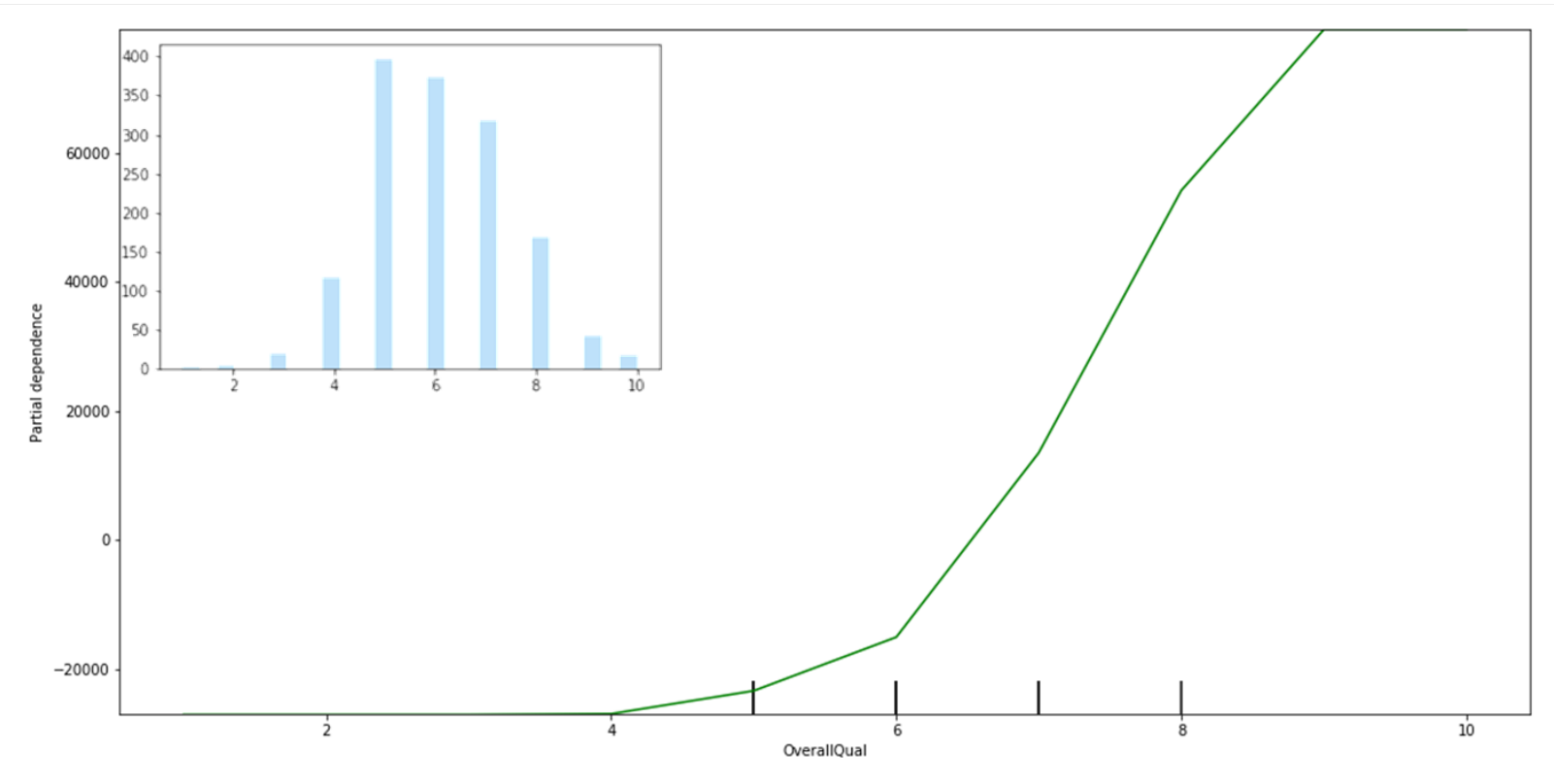
4.1 Overfitting



Instabilität / Varianz - Stichprobe 3



Random Forest - Marginaler Effekt OveralQual



Random Forest - Marginaler Effekt GrLivArea