

A low-angle, upward-looking photograph of several modern skyscrapers with glass and steel facades, reaching towards a clear blue sky. The perspective creates a sense of height and architectural grandeur.

HOUSE PRICE PREDICTION

Team Insight Explorers - Präsentation

DATA EXPLORATION PROJECT



AGENDA

- › Idee & Zielsetzung
- › Setup
 - › Datensatz
 - › Technologien und Bibliotheken
 - › Generelles Vorgehen
 - › Wie setzen wir Machine Learning ein?
- › Umsetzung & Ergebnisse
 - › Exploratory Analysis
 - › Datenvorverarbeitung
 - › Gütemaß RMSE
 - › Vorstellung der Ergebnisse
 - › Kritische Reflexion

Idee & Zielsetzung

- **Vorhersage von Hauspreisen** in den USA auf Basis verschiedener Faktoren (Grundstückgröße, Baujahr, Lage...) durch den Einsatz von ML-**Regressions-Modellen**
- Wirtschaftlicher Nutzen: Die Vorhersage von Preisen führt zu einer **besseren Informationslage**, die es ermöglicht, **bessere Kaufentscheidungen** zu treffen.

Introduction

Haus 1

Wohnfläche: 109m²

Bäder: 1

Stockwerke: 1

Baujahr: 1955

Preis: 221.900\$



Haus 2

Wohnfläche: 156m²

Bäder: 2

Stockwerke: 1

Baujahr: 1987

Preis: 510.000\$

Setup

Unser Datensatz:

- Id
- Date
- Price
- Bedrooms
- Bathrooms
- Sqft_living
- Sqft_lot
- Floors
- Waterfront
- View
- Condition
- Grade
- Design
- Sqft_above
- Sqft_basement
- Yr_built
- Yr_renovated
- Zipcode
- Latitude
- Longitude
- Sqft_living15
- Sqft_lot15

Technologien und Bibliotheken

Entwicklungsumgebung: Google Collab

- Notebooks
- Gehostete Laufzeit
- Gemeinsames Coden
- Einfaches Sharing



Bibliotheken: Bekannte Python-Module

- NumPy
- Pandas
- Scikit-learn
- Matplotlib



Generelles Vorgehen

Überblick über den Datensatz verschaffen



Vorverarbeitung: Train/Test Split, Data Cleaning



Auswahl von ML-Modellen

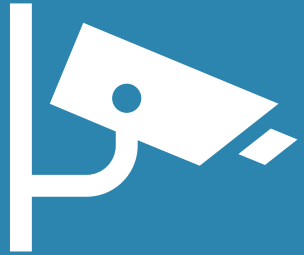


Trainieren von ML-Modellen



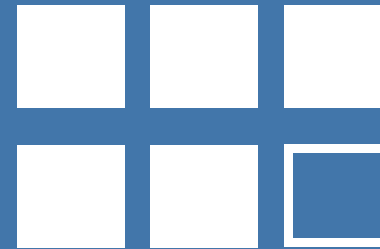
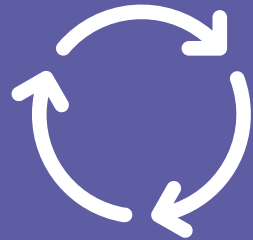
Evaluierung, mögliche Anpassung/Optimierung

Wie setzen wir Machine Learning ein?



Überwachtes
Lernen

Multiple
Regression



Batch Learning

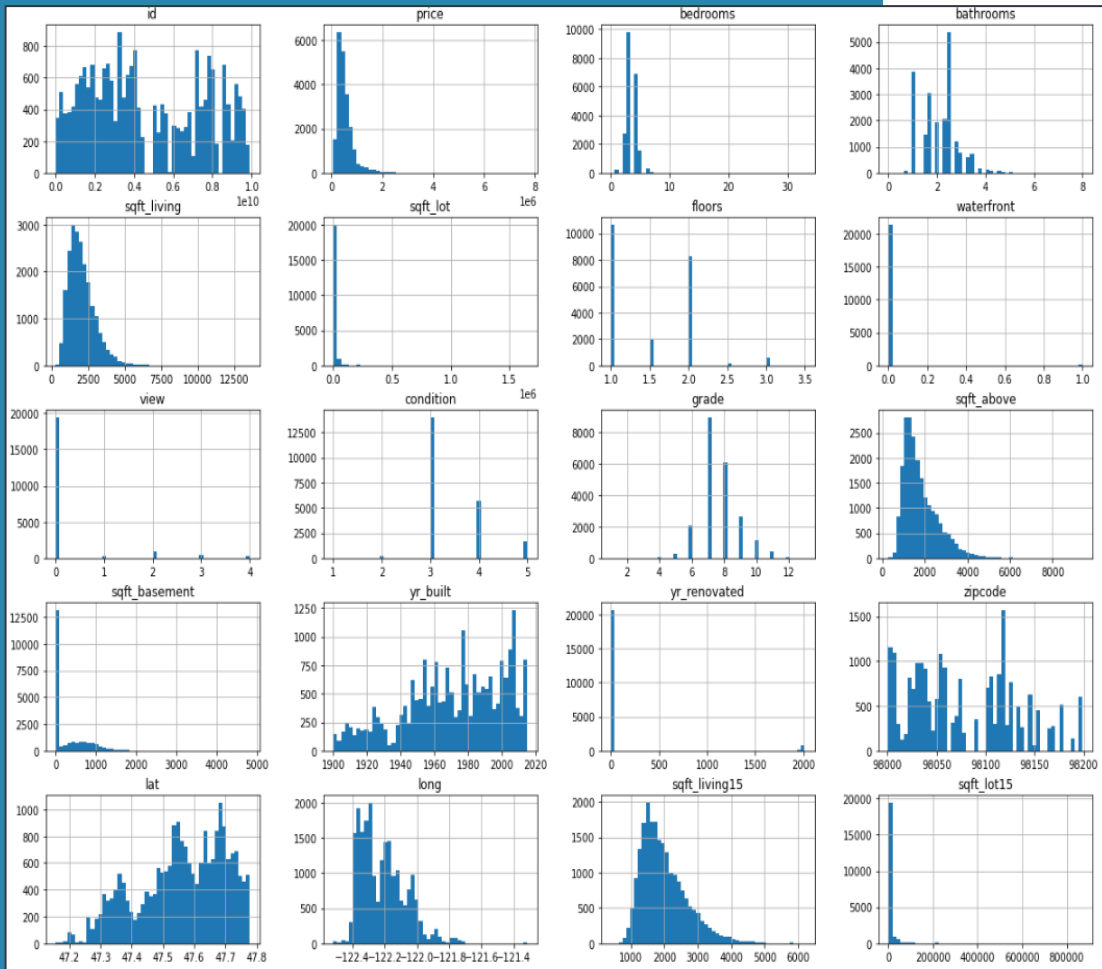
Evaluierung:
RMSE und R2-
Wert





Umsetzung & Ergebnisse



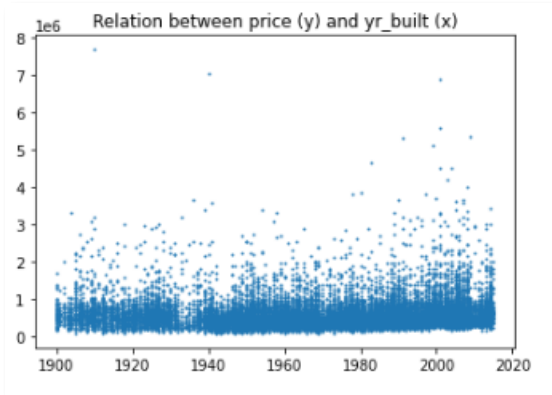


Exploratory Analysis

- Genereller Datenüberblick
- Statistischer Datenüberblick
- Histogramme
- Korrelationen
- Geografische Lage

Überblick über den Datensatz verschaffen

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000

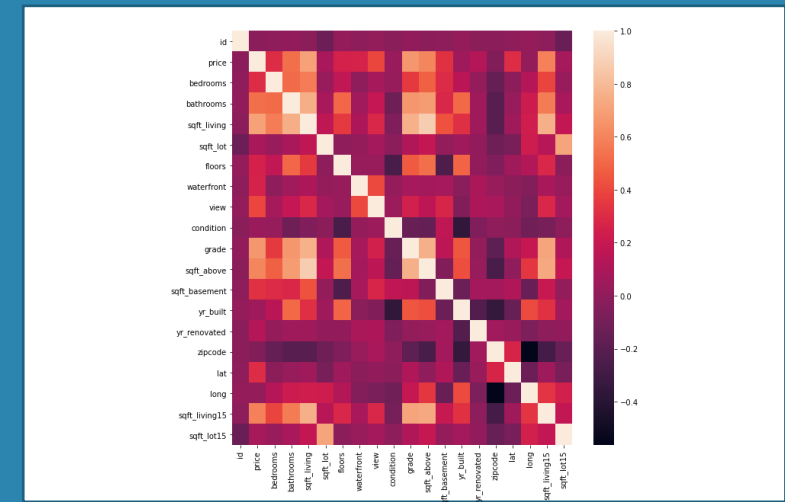
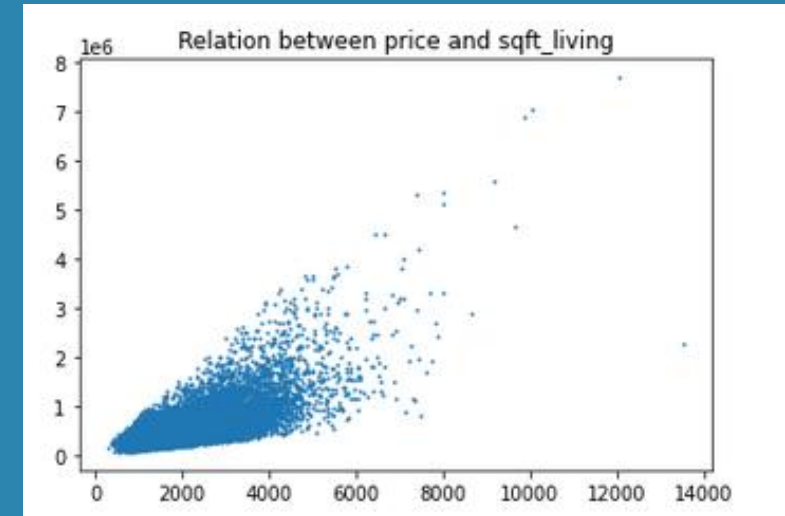
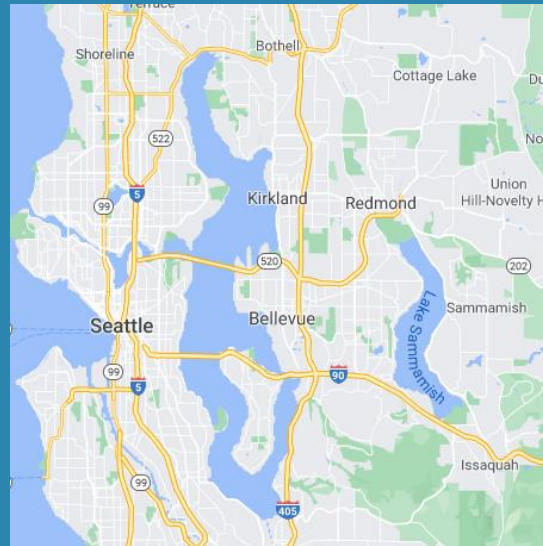
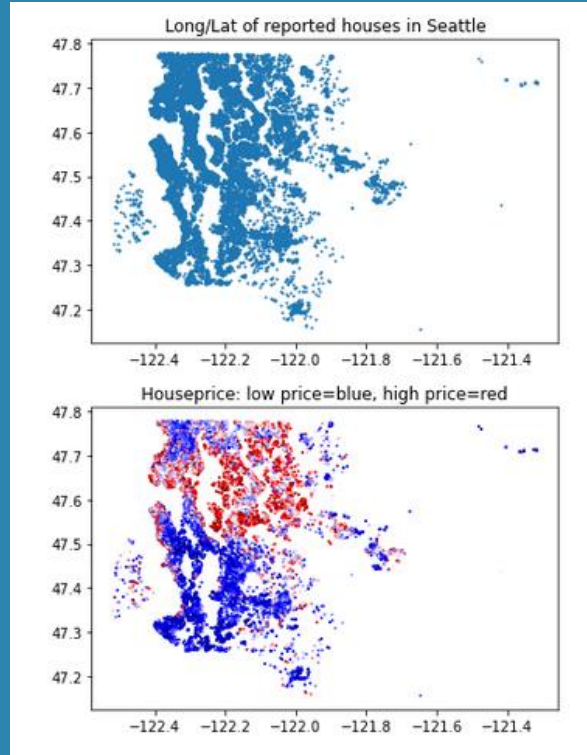


Methoden *info()* und *describe()* für schnelle Beschreibung des Datensatzes

Plotten von Histogrammen

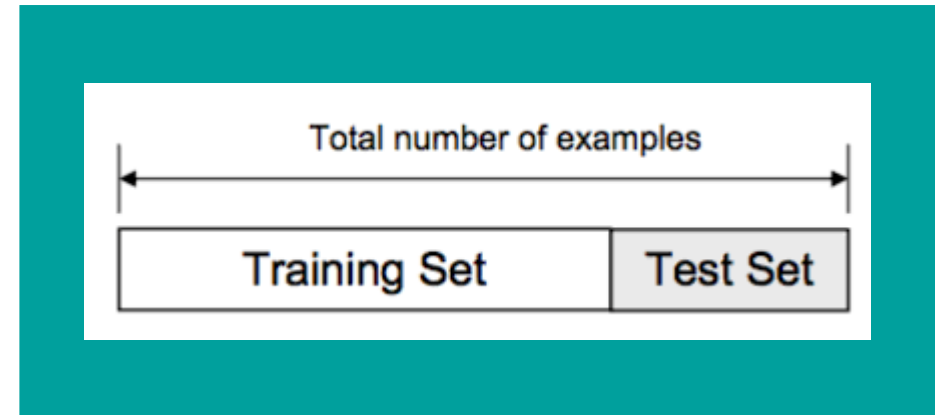
Suche nach Korrelationen mit *corr()*

Exploratory Analysis

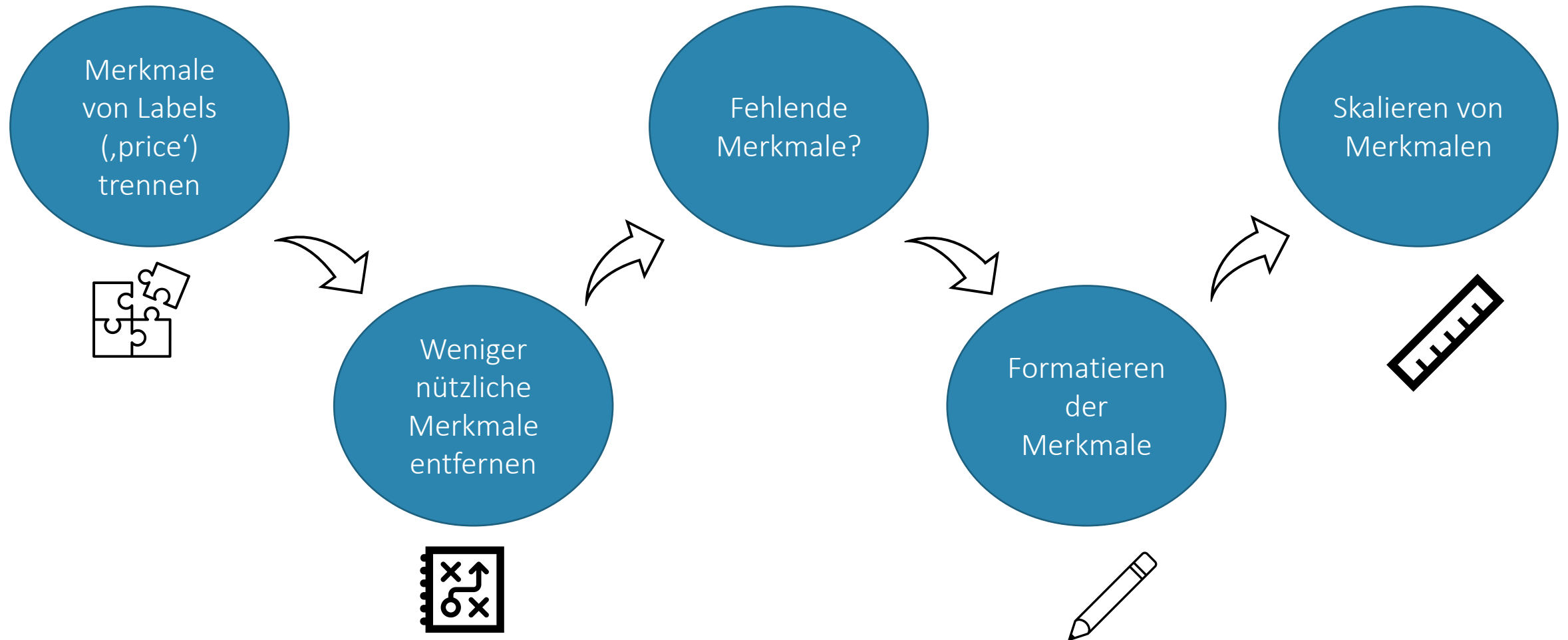


Vorverarbeitung

- › Train-Test-Split:
80% Trainingsdaten, 20% Testdaten
- › Split mit scikit-learn Funktion
- › Cross-Validierung erfolgt später



Vorverarbeitung



Gütemaß RMSE

RMSE als zentrales Maß

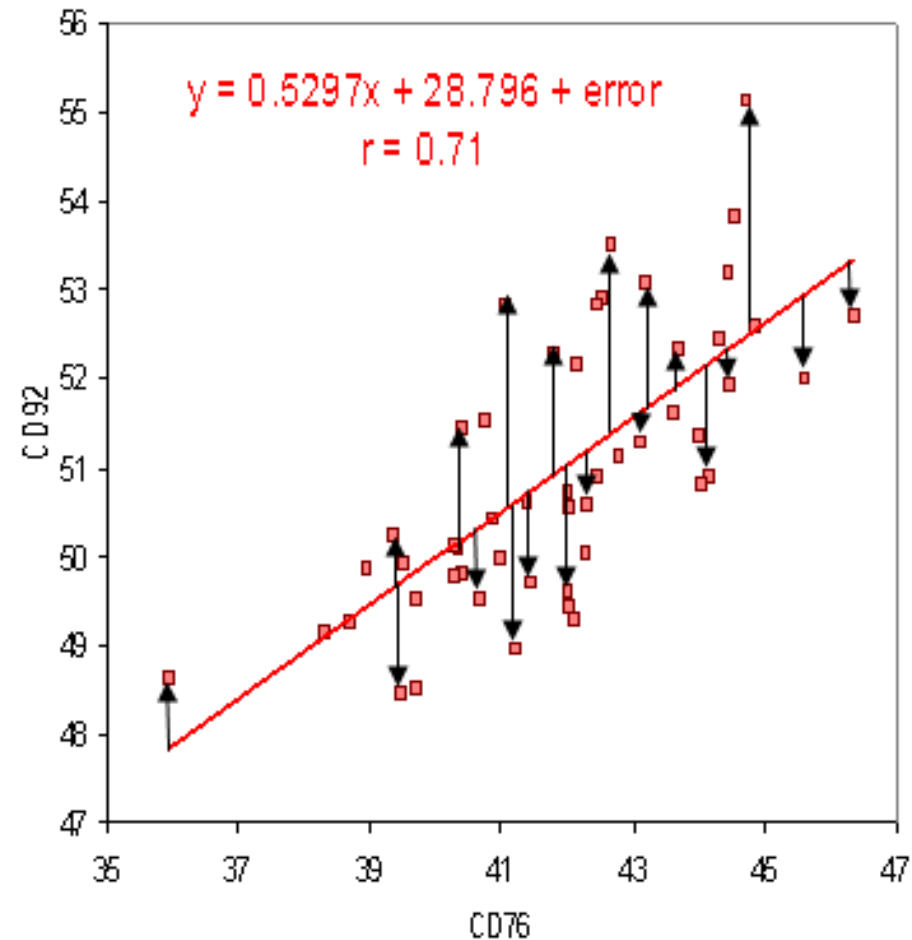
-> RMSE: Root Mean Square Error

$$= \sqrt{(f - o)^2}$$

(f = Vorhersagen, o = reale Daten)



„Um wieviel Dollar liegt die Vorhersage im Schnitt daneben?“



Vorstellung unserer Ergebnisse

Lineare Regression

RMSE:

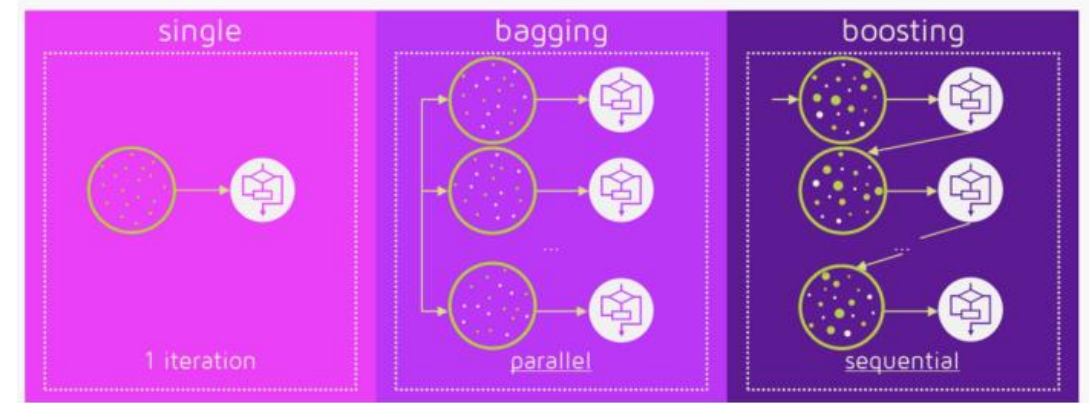
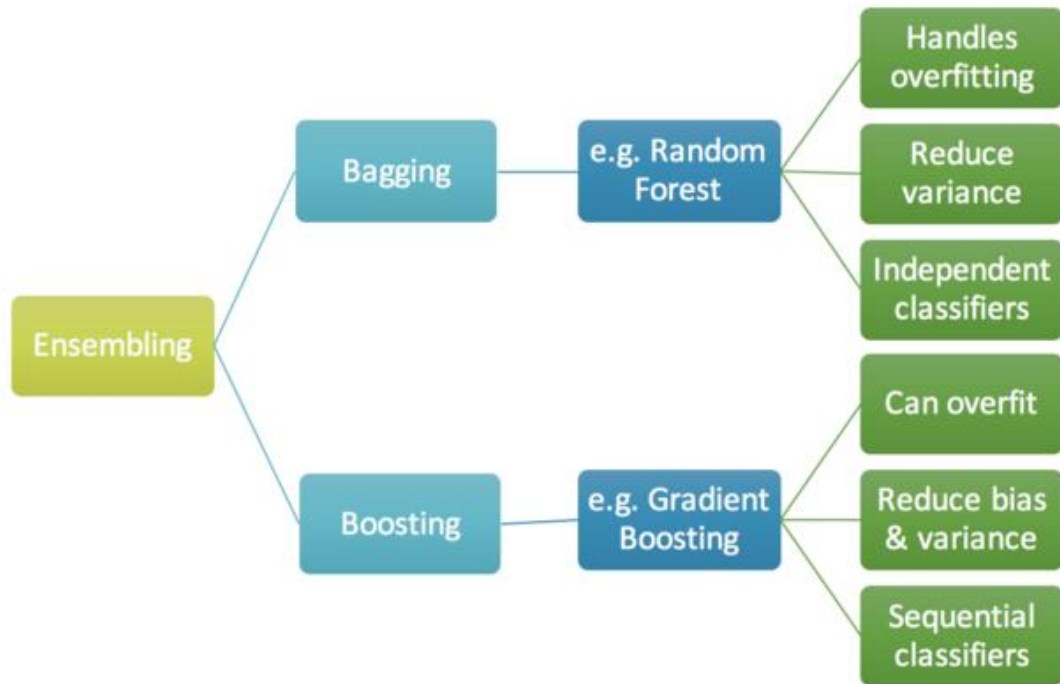
Train data	X-Validation	Test data
190 195	190 736	189 827

R2-Wert auf Testdaten: 0,71

- Underfitting, Modell zu einfach
- Zu viele Features sind nicht linear

```
Correlations to price
price          1.000000
sqft_living    0.702035
grade          0.667434
sqft_above     0.605567
sqft_living15  0.585379
bathrooms      0.525138
view           0.397293
sqft_basement  0.323816
bedrooms       0.308350
lat            0.307003
waterfront     0.266369
floors         0.256794
yr_renovated   0.126434
sqft_lot       0.089661
sqft_lot15     0.082447
yr_built       0.054012
condition      0.036362
long           0.021626
id             -0.016762
zipcode        -0.053203
Name: price, dtype: float64
```

Ensemble Methoden



Quelle: ML Review. Gradient Boosting from scratch. Abrufbar unter <https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d>

Vorstellung unserer Ergebnisse

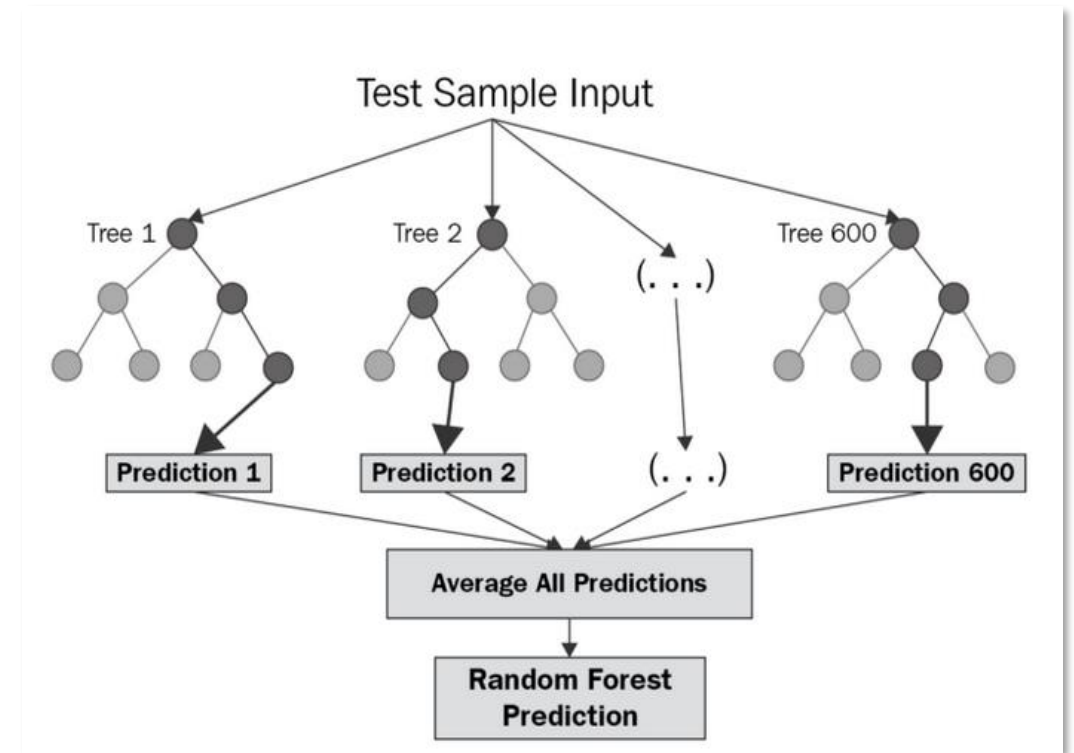
Random Forest Regression

RMSE:

Train data	X-Validation	Test data
45 617	122 635	120 174

R2-Wert auf Testdaten: 0,88

- Overfitting
- Deutlich besser als linear Regression



Quelle: Start it up. Random Forest Regression. Abrufbar unter:
<https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f>

Vorstellung unserer Ergebnisse

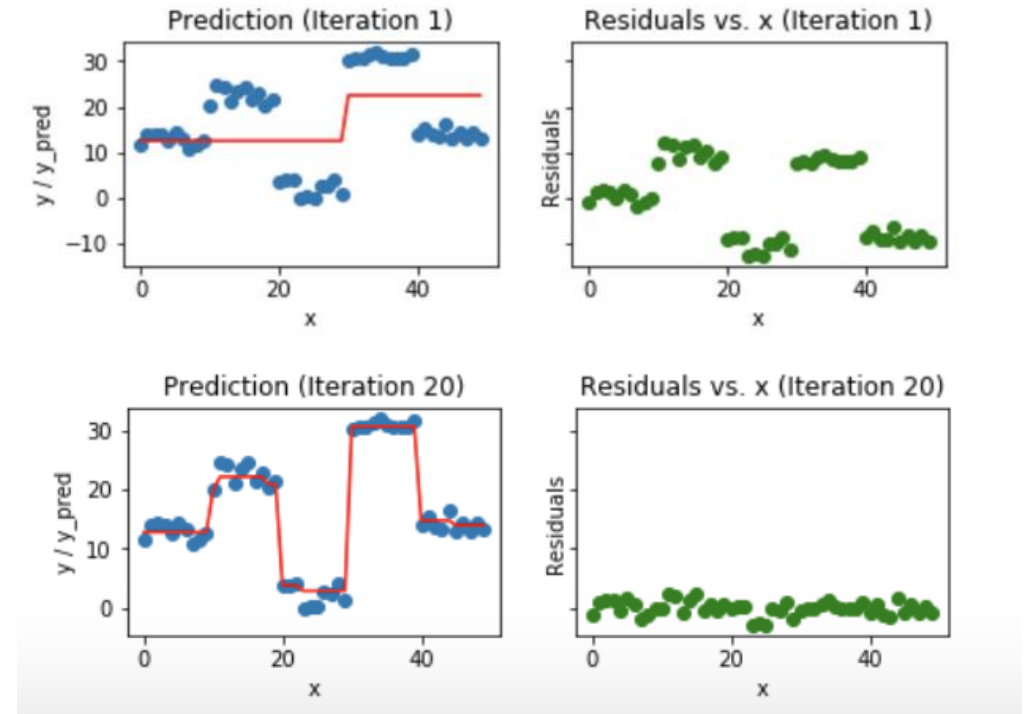
Gradient Boosting Regression

RMSE:

Train data	X-Validation	Test data
115 619	127 740	127 794

R2-Wert auf Testdaten: 0,87

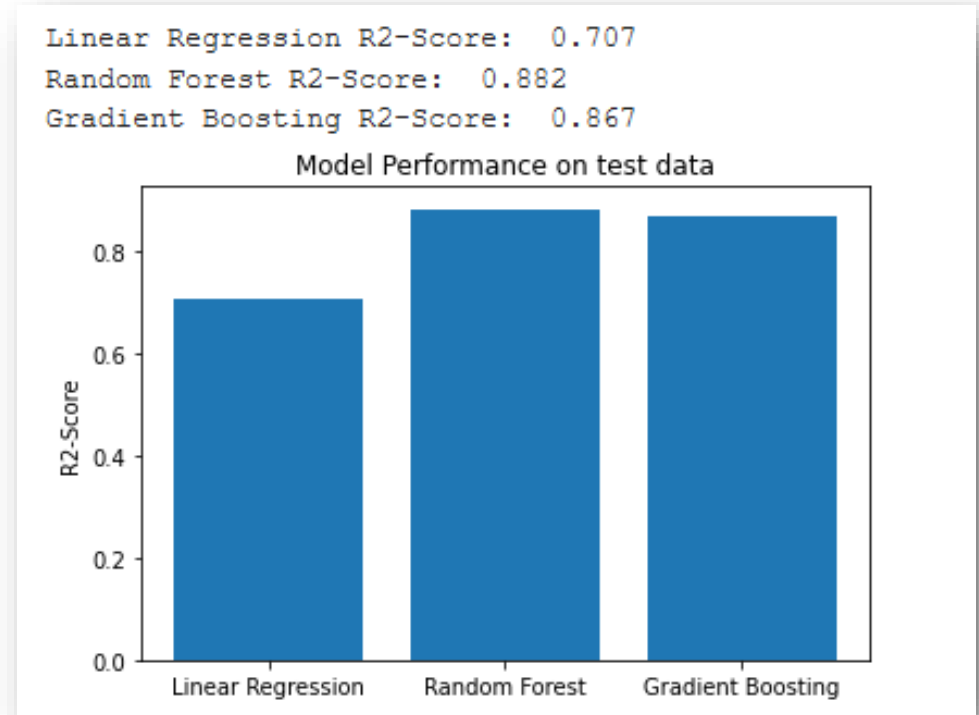
- Leichtes Overfitting
- Gute Alternative zum Random Forest



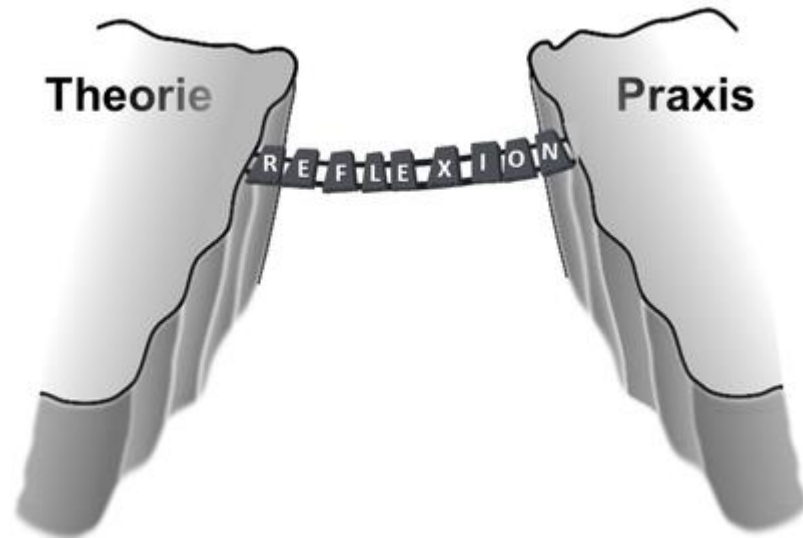
Quelle: ML Review. Gradient Boosting from scratch. Abrufbar unter <https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d>

Vorstellung unserer Ergebnisse: Vergleich der Modelle

- Random Forest Regressor mit der höchsten Performance
- Gradient Boosting ähnlich passend und zudem mit weniger Overfitting als der Random Forest
- Mit Hyperparameter-Optimierung (Grid Search) konnten wir nur einen kleinen Einfluss auf die Performance nehmen



Kritische Reflexion



Lessons Learned:

- Erlernte Theorie konnte erfolgreich in Praxis umgesetzt werden
- Modell-Performance noch nicht optimal
- Zeitlicher Aspekt wird nicht berücksichtigt

→ Dataset könnte noch größer sein

→ Dataset sollte Kaufpreise mehrerer Jahre beinhalten



Vielen Dank!

Insight Explorers

