Ein Vergleich zwischen Random Forest und Gradient Boosting als geeignetes Prognoseverfahren anhand von Immoblilienwerten in Boston.

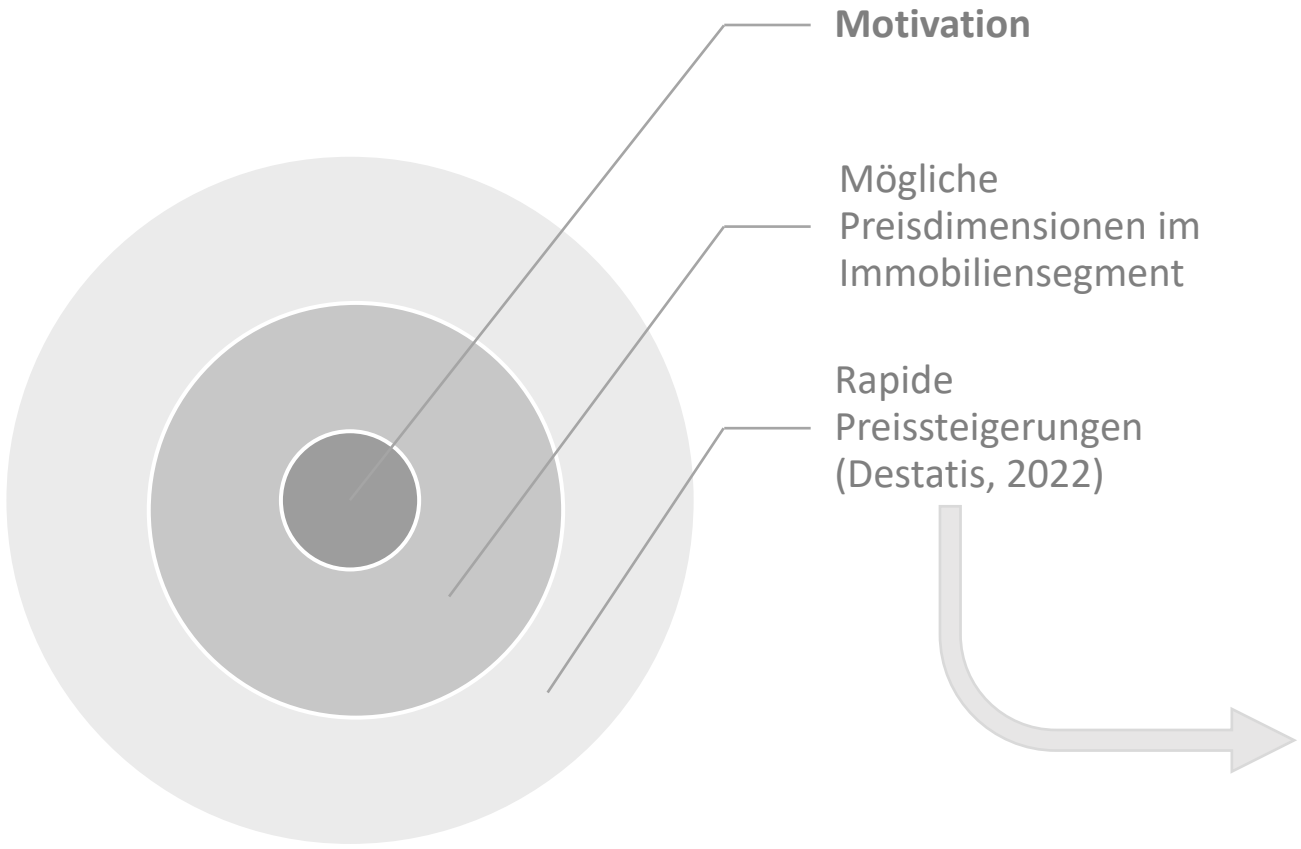
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

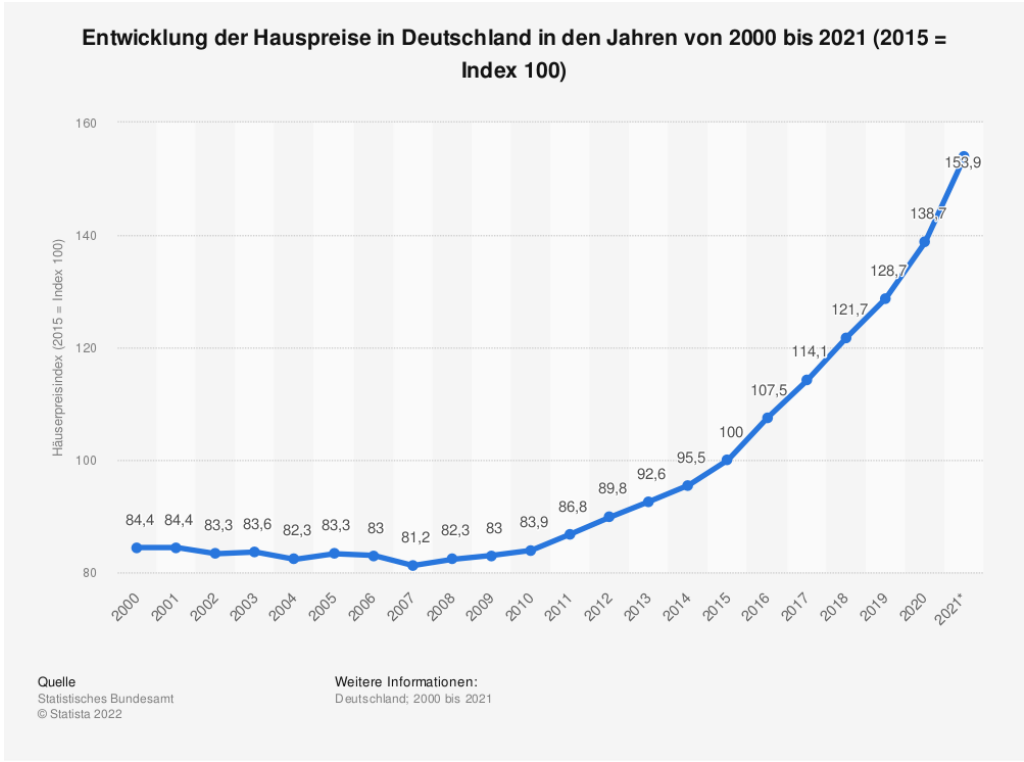
JG|U

AGENDA

1. Einleitung
2. Theoretischer Rahmen
3. Experimenteller Aufbau
4. Ergebnisse
5. Diskussion
6. Fazit

IMMOBILIENPREISE





METHODIK

Was wird nicht betrachtet:	
Ermittlung von Kausalitäten	Traditionelle, ökonometrische Modelle

Was wird hingegen untersucht:	
Bestimmung von validen Prognosen	Random Forest und Gradient Boosting

ABGRENZUNG DER MODELLE

Ensemble-Methoden

- Sowohl bei RF, als auch bei GB werden eine Reihe von Klassifizierern generiert, die wiederum neue Datenpunkte erzeugen (Dietterich, 2000)
- Der Hauptunterschied zwischen beiden Konzepten liegt in der Generierung und Aggregation der Klassifizierer → Entscheidungsbäume

Random Forest (RF)

Entscheidungsbäume durch Bagging generiert (Breiman, 2001)

Gradient Boosting (GB)

Entscheidungsbäume erfolgt die Generierung additiv (Friedman, 2001)

AKTUELLER FORSCHUNGSSTAND

Stärken:

- Präzise Vorhersagen im Trainings- und Testverfahren (Ho et al., 2020; Hjort et al., 2022)
- Leistungssteigerungen möglich durch Verwendung von:
  - Angepasste Verlustfunktionen (Hjort et al., 2022)
  - Verwendung von Hybridmodellen (Hjort et al., 2022; Lu et al., 2017; Truong et al., 2020)
- Abseits vom Wohnungsmarkt weitere Anwendungen möglich (Yoon, 2021; Callens et al., 2020)

Limitationen:

- Blackbox-Eigenschaften (Ho et al., 2020)
- RF neigen stärker dazu, Daten überanzupassen als GB (Truong et al., 2020)
- RF haben eine höhere Laufzeit gezeigt als GB bedingt durch gewählte Struktur (Truong et al., 2020)
- Exogene Schocks werden tendenziell nicht erfasst (Yoon, 2021)

WIE FINDET MAN DAS BESTE MODELL?

Optimierung

Hyperparameter

Bias-variance tradeoff

Random Forest

Gradient Boosting

k-fold cross validation

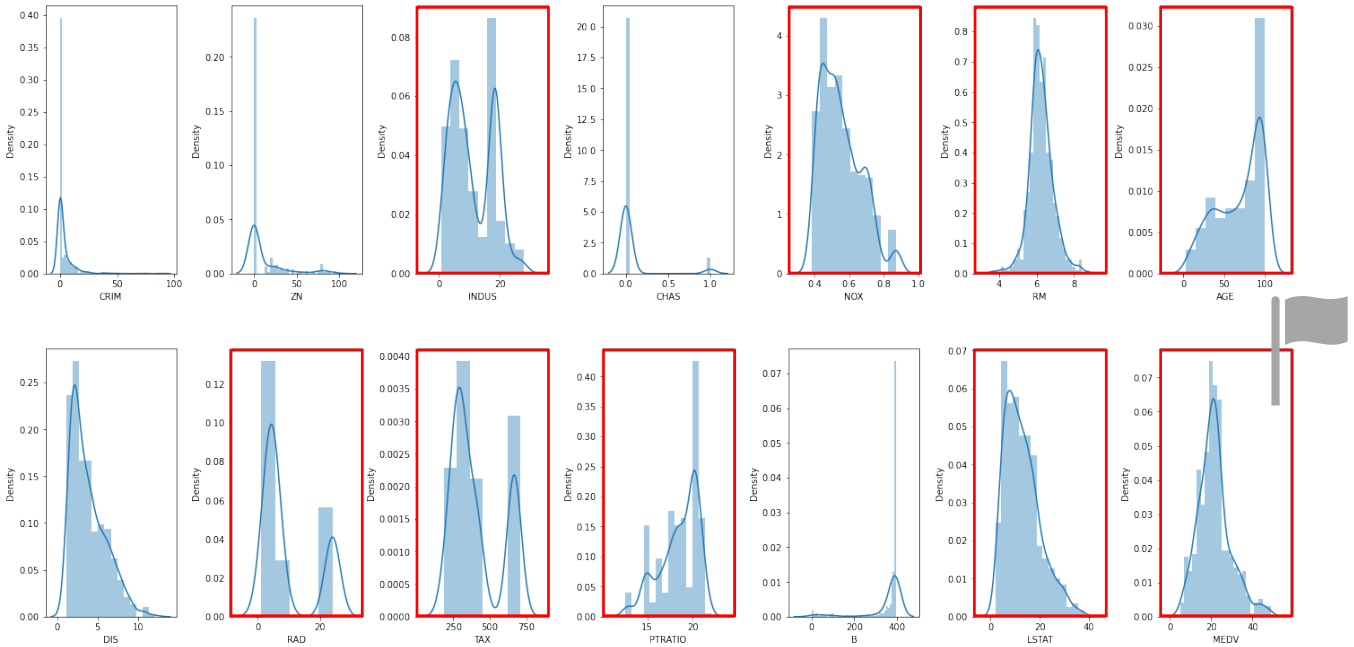
Anzahl Entscheidungsbäume

Anzahl Entscheidungsbäume

Lernrate

BOSTON HOUSING DATA

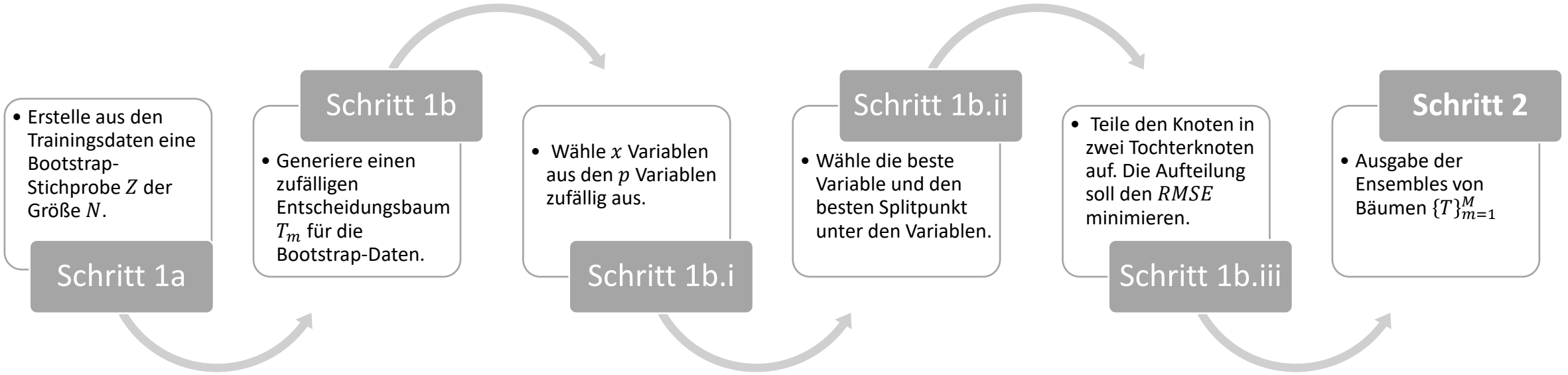
- Datensatz mit 506 Beobachtungen
- Ausschluss von Beobachtungen mit einem Median (MEDV) ≥ 50.000 $
  - 490 Realisationen übrig
- Berücksichtigung von Variablen mit hohen Erklärungsgehalt für MEDV
- Standardisierung der verbliebenen, unabhängigen Variablen



ALGORITHMUS ZU RANDOM FOREST

Ermittlung einer RF-Prognose gemäß folgender Schritte (Breiman, 2001; Yoon, 2021)

Schritt 1. Für $m = 1$ bis M:



ALGORITHMUS ZU GRADIENT BOOSTING

Ermittlung einer RF-Prognose gemäß folgender Schritte (Friedman, 2001; Yoon, 2021)

Input. Ein Datensatz $\{[x_i, y_i]\}_{i=1}^{n}$ mit einer differenzierbaren Verlustfunktion $L(y_i, F(x))$:

Schritt 2. Für $m = 1$ bis M:

Schritt 1
- Initialisiere Modell mit einem konstanten Startwert
- Hier der Mittelwert

$$F_0(x) = \underset{\gamma}{argmin} \sum_{i=1}^{n} L(\gamma_i, y)$$

Schritt 2a
- Berechne die Residuen $\tilde{y}_{im}$

$$\tilde{y}_{im} = -\left[\frac{\partial L(\gamma_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

für $i = 1 \ldots N$

Schritt 2b
- Passe einen Regressionsbaum an die Werte von $\tilde{y}_{im}$ an und erstelle endgültige Bereiche R_{jm} für $j = 1 \ldots J_m$

Schritt 2c
- Für $j = 1 \ldots J_m$, berechne die Outputs der Regressions-bäume γ_{jm}

$$\gamma_{jm} = \underset{\gamma}{argmin} \sum_{x_j \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

Schritt 2d
- $F_m(x)$ wird durch den Output seines Vorgängers $F_{m-1}(x)$ addiert mit den aggregierten Residuen γ_{jm} korrigiert mit der Lernrate v ermittelt

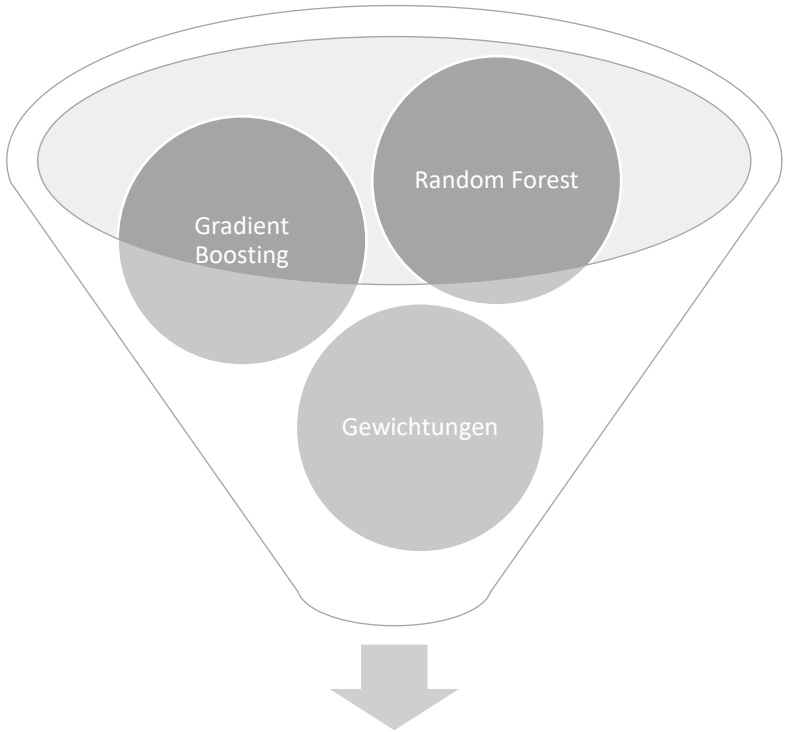
$$F_m(x) = F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} I(x_j \in R_{ij})$$

Schritt 3
- Ausgabe von $F_m(x)$, soblad alle Iterationen abgeschlossen sind

DAS HYBRIDMODELL

Hybridmodell: Voting Regression

- Kombination von gegebenen Schätzern zu einem Meta-Schätzer
- Gewichtungen der Einzelschätzer möglich
- Diversifikation von Schwächen der gegebenen Einzelschätzer möglich
- Finale Gewichtung: $\frac{5}{6}\ GB + \frac{1}{6}\ RF$



TRAINING- UND TESTVERFAHREN

Der höchsten Score bzw. der geringsten Fehler:

- Training
  - Random Forest
- Test
  - Voting Regression

Tabelle 4: Ergebnisse der Vorhersagen

Modell	R^2		RMSE	
	Train	Test	Train	Test
Random Forest	**0.96749**	0.84397	**1.35815**	3.4552
Gradient Boosting	0.951740	0.86452	1.65491	3.21963
Voting Regression	0.95658	**0.86560**	1.5695	**3.20686**

Quelle: Eigene Darstellung

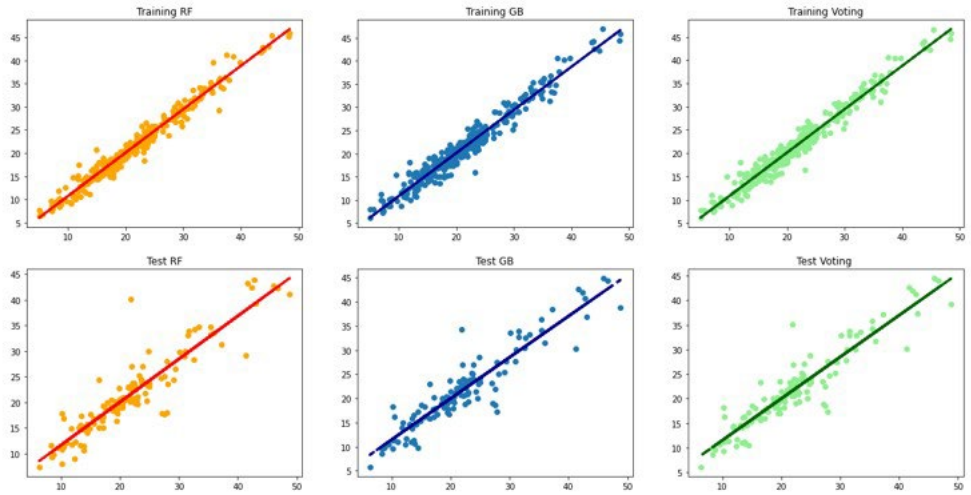


Abbildung 5: Graphen zu Training- und Testdaten

Quelle: Eigene Darstellung

STATISTISCHE EVALUIERUNG

Wie verhalten sich die Modelle bei wiederholten Training bzw. Testen?

- Nach 150x Wiederholungen bleibt das Ergebnis unverändert
- Mittelwertdifferenzen bei fast allen Kombinationen mindestens auf dem 1%-Niveau signifikant
- Lediglich GB vs. Vote ist nicht signifikant von Null verschieden

Tabelle 5: Schätzung der Gütemaße mit n =150 Iterationen

Modell	R^2		RMSE		R^2: Kreuzvalidierung	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Random Forest	**0.9672**	0.0002	**1.3637**	0.0055	0.8498	0.0658
Gradient Boosting	0.9517	0.0	1.6549	0.0000	0.8583	0.0562
Voting Regression	0.9566	0.0000	1.5691	0.0007	**0.8603**	0.0569

Quelle: Eigene Darstellung

Tabelle 6: T-test mit df = 149 für die Gütemaße

Nullhypothese	R^2		RMSE		R^2: Kreuzvalidierung	
	t-Value	p-Value	t-Value	p-Value	t-Value	p-Value
$\mu_{RF} = \mu_{GB}$	710.98**	0.0	641.78**	0.0	3.84**	0.0001
$\mu_{RF} = \mu_{Vote}$	481.97**	0.0	448.98**	0.0	4.70**	0.0
$\mu_{GB} = \mu_{Vote}$	1504.92**	0.0	1460.84**	0.0	0.96	0.3351

$p < 0.01$ entspricht **
$p < 0.05$ entspricht *

Quelle: Eigene Darstellung

STÄRKEN UND LIMITATIONEN

- Performance von allen Modellen insgesamt als sehr gut zu bewerten!
- Was es noch zu beachten gilt:

Random Forest	Hyperparameter	Hybridmodell	Weitere Verbesserungen
• Neigt eher zu Overfitting (Truong et al., 2020) • Laufzeit schneller als GB aufgrund der Parallelisierung (Carreira-Perpiñán & Zharmagambetov, 2020)	• Gittersuche als Vertreter der uninformierten Methoden • Verwendung von informierten Methoden, um optimale Kombination sicherzustellen (Callens et al., 2020).	• Konnte homogene Modelle übertreffen (Hjort et al., 2022; Lu et al., 2017; Truong et al., 2020). • Möglichkeit, Schwächen der einzelnen Schätzer zu diversifizieren • Um die Informationen optimal mit dem Meta-Schätzer zu erfassen, sollten die einzelnen Schätzer ebenfalls optimiert sein	• Verwendung von größeren Datensätzen • Untersuchung von weiteren Verlustfunktionen (Hjort et al., 2022)

BEFUNDE UND WEITERER AUSBLICK

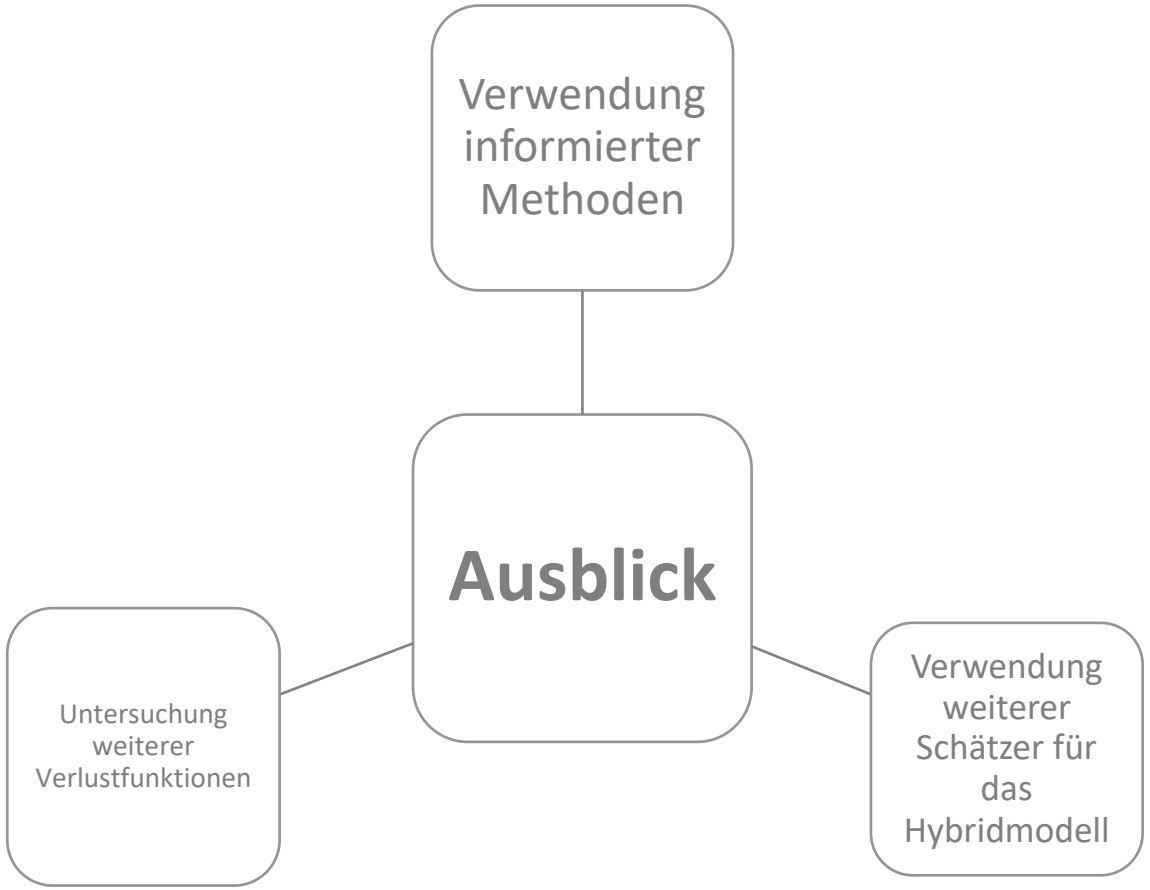
Befunde

RF erzielt das beste Resultat im Training, dafür Overfitting

GB erzielt bessere Resultate für Testdaten als RF

Voting Regression erzielt das beste Resultat für die Testdaten bzw. die Kreuzvalidierung

→ Das Hybridmodell ist zu bevorzugen



VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

LITERATURVERZEICHNIS

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32.

Callens, A., Morichon, D., Abadie, S., Delpey, M. & Liquet, B. (2020). Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. *Applied Ocean Research*, *104*, 102339.

Carreira-Perpiñán, M. Á. & Zharmagambetov, A. (2020). Ensembles of Bagged TAO Trees Consistently Improve over Random Forests, AdaBoost and Gradient Boosting. In J. Wing & D. Madigan (Hrsg.), *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference* (S. 35–46). ACM.

Destatis. (2022). *Häuserpreisindex, Preisindex für Bauland: Deutschland, Jahre.* Statistisches Bundesamt. https://www-genesis.destatis.de/genesis//online?operation=table&code=61262-0001&bypass=true&levelindex=0&levelid=1652035420387#abreadcrumb

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In J. Kittler & F. Roli (Hrsg.), *Lecture Notes in Computer Science: Bd. 1857. Multiple classifier systems: First international workshop, MCS 2000* (Bd. 1857, S. 1–15). Springer. https://doi.org/10.1007/3-540-45014-9_1

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5).

G, T. R., Bhattacharya, S., Maddikunta, P. K. R., Hakak, S., Khan, W. Z., Bashir, A. K., Jolfaei, A. & Tariq, U. (2020). Antlion re-sampling based deep neural network model for classification of imbalanced multimodal stroke dataset. *Multimedia Tools and Applications.* Vorab-Onlinepublikation. https://doi.org/10.1007/s11042-020-09988-y

Harrison, D. & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*(1), 81–102. https://doi.org/10.1016/0095-0696(78)90006-2

Hastie, T., Tibshirani, R. & Friedman, J. H. (2017). *The elements of statistical learning: Data mining, inference, and prediction*. *Springer Series in Statistics*. Springer.

Hjort, A., Pensar, J., Scheel, I. & Sommervoll, D. E. (2022). House price prediction with gradient boosted trees under different loss functions. *Journal of Property Research*, 1–27.

Ho, W. K., Tang, B.-S. & Wong, S. W. (2020). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70.

LITERATURVERZEICHNIS

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013). An Introduction to Statistical Learning (Bd. 103). Springer New York.

Kim, J., Won, J., Kim, H. & Heo, J. (2021). Machine-Learning-Based Prediction of Land Prices in Seoul, South Korea. Sustainability, 13(23), 13088. https://doi.org/10.3390/su132313088

Lu, S., Li, Z., Qin, Z., Yang, X. & Goh, R. S. M. (2017). A hybrid regression technique for house prices prediction. In 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (S. 319–323). IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825--2830.

Truong, Q., Nguyen, M., Dang, H. & Mei, B. (2020). Housing Price Prediction via Improved Machine Learning Techniques. Procedia Computer Science, 174, 433–442.

The University of Toronto. (1996). The Boston Housing Dataset: A Dataset derived from information collected by the U.S. Census Service concerning housing in the area of Boston Mass. The University of Toronto. https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

Wang, R., Ma, H. & Wang, C. (2022). An Ensemble Learning Framework for Detecting Protein Complexes From PPI Networks. Frontiers in genetics, 13, 839949.

Wu, J., Chen, S. & Liu, X. (2020). Efficient hyperparameter optimization through model-based reinforcement learning. Neurocomputing, 409, 381–393. https://doi.org/10.1016/j.neucom.2020.06.064

Yoon, J. (2021). Forecasting of Real GDP Growth Using Machine Learning Models: Gradient Boosting and Random Forest Approach. Computational Economics, 57(1), 247–265.

BESCHREIBUNGEN ZU BOSTON HOUSING DATA

Tabelle 1: Beschreibung von Boston Housing Data

Nummerierung	Variable	Beschreibung
1	CRIM	Pro-Kopf-Verbrechensrate nach Stadt
2	ZN	Anteil der Wohnbauflächen, die für Grundstücke mit einer Größe von mehr als 25.000 Quadratmetern ausgewiesen sind
3	INDUS	Anteil der Flächen für Nicht-Einzelhandelsunternehmen je Stadt
4	CHAS	Charles River Dummy-Variable (= 1, wenn der Trakt an den Fluss grenzt, sonst 0)
5	NOX	Stickstoffoxidkonzentration (Teile pro 10 Millionen)
6	RM	durchschnittliche Anzahl der Zimmer pro Wohnung
7	AGE	Anteil der Eigentumswohnungen, die vor 1940 gebaut wurden
8	DIS	gewichtete Entfernungen zu fünf Bostoner Beschäftigungszentren
9	RAD	Index der Erreichbarkeit von Radialautobahnen
10	TAX	Vollwertiger Grundsteuersatz pro 10.000 Dollar
11	PTRATIO	Schüler-Lehrer-Verhältnis nach Stadt
12	B	Anteil der Schwarzen in der Population
13	LSTAT	% unterer Status der Bevölkerung
14	MEDV	Medianwert von Eigenheimen in $1000s

Quelle: Übersetzt nach Harrison & Rubinfeld (1978, S. 96 & 97)

KORRELATIONSMATRIX

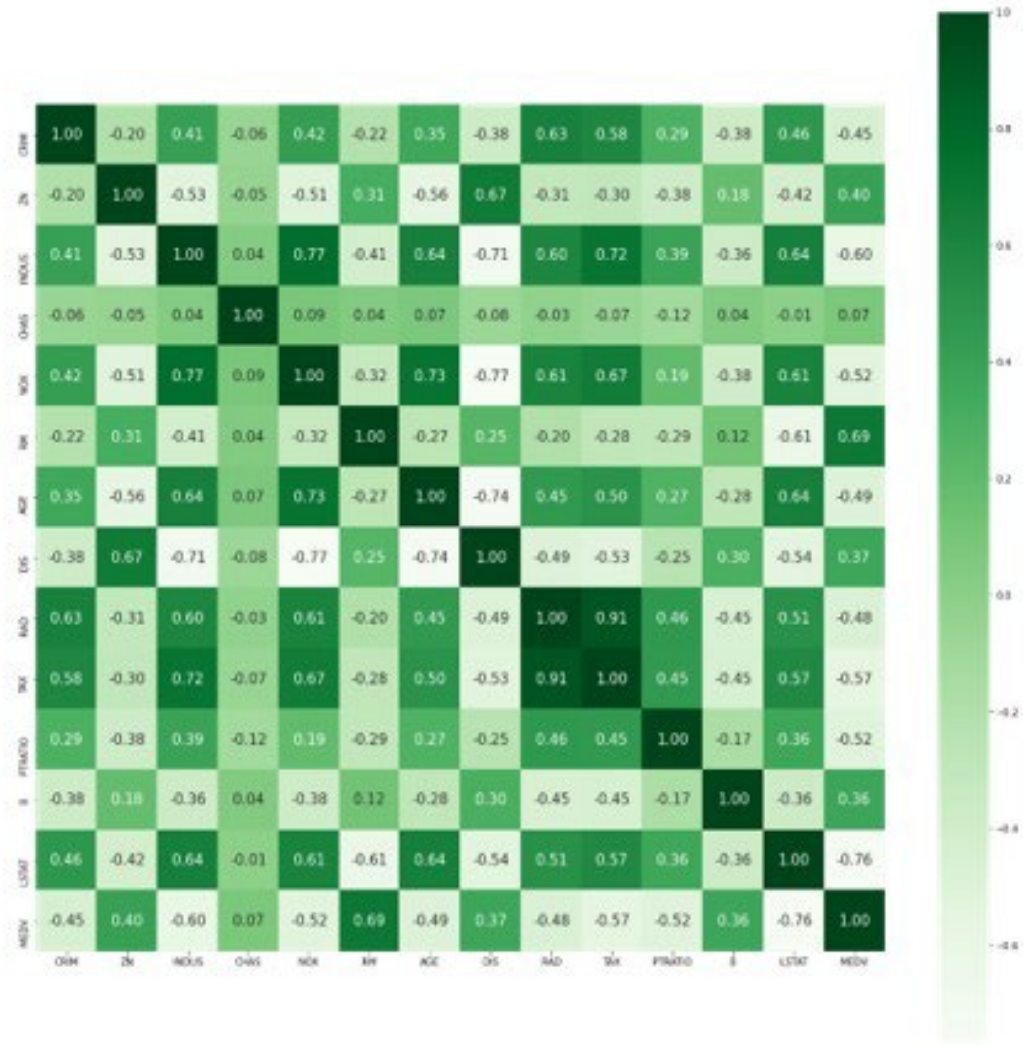


Abbildung 3: Korrelationsmatrix

Quelle: Eigene Darstellung

GETESTETE KOMBINATIONEN UND VERWENDETE SCHÄTZER

Tabelle 2: Getestete Hyperparameter mit GridSearchCV

Modell	Hyperparameter	k-Fold
Random Forest	max_depth= [2,4,6,8,10,12,14,16,18,20] n_estimators = [100, 200 ,500, 900, 1000] min_samples_leaf = [2,4,6,8,10]	$k = 10$
Gradient Boosting	learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.3] n_estimators = [100, 200 ,500, 900, 1000] max_depth = [2,4,6,8,10,12,14,16,18,20] min_samples_leaf = [2,4,6,8,10]	$k = 10$
Voting Regression	weights = [(1,1), (2,1), (3,1), (4,1), (5,1), (1,2), (1,3), (1,4), (1,5)] estimators = [("Random Forest", RandomForestRegressor (max_depth=20, n_estimators=900, min_samples_leaf = 2), ("Gradient Boosting", GradientBoostingRegressor (max_depth = 2, n_estimators = 200, learning_rate = 0.1, min_samples_leaf = 6)]	$k = 10$

Quelle: Eigene Darstellung

Tabelle 3: Schätzer mit optimalen Hyperparametern

Modell	Schätzer mit optimalen Hyperparameter
Random Forest	RandomForestRegressor (max_depth=20, min_samples_leaf=2, n_estimators=900)
Gradient Boosting	GradientBoostingRegressor (max_depth=2, min_samples_leaf=6, n_estimators=200)
Voting Regression	VotingRegressor (estimators= [('Random Forest', RandomForestRegressor (max_depth=20, min_samples_leaf=2, n_estimators=900)), ('Gradient Boosting', GradientBoostingRegressor (max_depth=2, min_samples_leaf=6, n_estimators=200))], weights= (1, 5))

Quelle: Eigene Darstellung

DESKRIPTIVE STATISTIKEN

VAL	LSTAT	INDUS	NOX	PTRATIO	RM	TAX	DIS	AGE	MEDV
count	490.00	490.00	490.00	490.00	490.00	490.00	490.00	490.00	490.00
mean	3.3760E-17	6.5985E-16	6.4200E-16	2.6464E-16	2.2465E-16	9.5389E-16	-1.7832E-16	1.8953E-16	21.64
std	1.0010E+00	1.0010E+00	1.0010E+00	1.0010E+00	1.0010E+00	1.0010E+00	1.0010E+00	1.0010E+00	7.87
min	-1.55	-1.52	-1.45	-2.81	-4.11	-1.32	-1.28	-2.32	5.00
0.25	-0.79	-0.87	-0.90	-0.53	-0.56	-0.76	-0.82	-0.84	16.70
0.50	-0.18	-0.21	-0.14	0.28	-0.09	-0.46	-0.27	0.30	20.90
0.75	0.59	1.03	0.60	0.80	0.51	1.54	0.65	0.91	24.68
max	3.54	2.44	2.72	1.65	3.88	1.80	3.93	1.13	48.80

Abbildung 4: Deskriptive Statistiken für Trainings- und Testdatensatz

Quelle: Eigene Darstellung