3_Klassifikation_des_objekttyps

January 11, 2024

cml1 - Immobilienrechner

1 Klassifikation des Objekttyps

Das Ziel dieses Notebooks ist es, der Typ des Objekts anhand verschiedener Modelle zu klassifizieren. Die verwendeten Daten stammen aus dem `datawrangling.ipynb` Notebook, wobei die Daten gesäubert und aufbereitet wurden.

1.1 Metriken

In der Welt der Datenanalyse und des maschinellen Lernens sind Metriken unerlässlich, um die Leistungsfähigkeit und Genauigkeit von Klassifikationsmodellen zu bewerten. Sie ermöglichen es uns, die Effektivität unserer Modelle objektiv zu messen und Bereiche für Verbesserungen zu identifizieren. Im Folgenden werden einige der wichtigsten Metriken vorgestellt, die üblicherweise zur Bewertung von Klassifikationsmodellen verwendet werden:

1.1.1 Accuracy

Die Accuracy ist ein Mass für den Gesamtanteil der korrekt klassifizierten Objekte im Verhältnis zur Gesamtanzahl der Objekte. Sie ist besonders hilfreich in Situationen, in denen die Klassen gleichmässig verteilt sind. In Situationen mit unausgewogenen Klassenverteilungen, wo es wichtig ist, die Minderheitsklasse korrekt zu erkennen, kann der F1-Score daher eine bessere Einschätzung der Modellleistung bieten als die reine Genauigkeit.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

1.1.2 Precision

Die Precision bewertet, wie genau die Vorhersagen des Modells sind, indem sie misst, welcher Anteil der Objekte, die als zu einer bestimmten Klasse gehörig vorhergesagt wurden, tatsächlich zu dieser Klasse gehören. Diese Metrik ist besonders wichtig in Szenarien, wo die Kosten für falsch positive Ergebnisse hoch sind.

$$Precision = \frac{TP}{TP + FP}$$

1.1.3 Recall

Der Recall gibt an, welcher Anteil der tatsächlichen Objekte einer Klasse vom Modell korrekt identifiziert wurde. Diese Metrik ist kritisch in Fällen, in denen das Übersehen von tatsächlichen Fällen schwerwiegende Konsequenzen hat.

$$Recall = \frac{TP}{TP + FN}$$

1.1.4 F1-Score

Der F1-Score ist eine wichtige Metrik, die den harmonischen Mittelwert aus Präzision (Precision) und Sensitivität (Recall) bildet. Er bietet einen ausgeglichenen Überblick über die Leistungsfähigkeit eines Modells, indem er sowohl die Genauigkeit der Vorhersagen als auch die Vollständigkeit der erfassten relevanten Fälle berücksichtigt. Der F1-Score reicht von 0 bis 1, wobei 1 für perfekte Präzision und Recall steht.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Erweiterungen des F1-Scores sind die Varianten Micro, Macro und Weighted. Diese werden genutzt, um unterschiedliche Aspekte bei der Bewertung von Modellen, insbesondere in ungleich verteilten Datensätzen, zu berücksichtigen:

- Micro Average: Berechnet globale Metriken, indem alle True Positives, False Positives und False Negatives aggregiert werden. Geeignet für Datensätze mit Klasseungleichgewichten.

$$Micro\ F1 = \frac{TP}{TP + \frac{1}{2} \cdot (FP + FN)}$$

- Macro Average: Berechnet für jede Klasse separat Metriken und nimmt dann das ungewichtete Mittel. Dieser Ansatz betrachtet alle Klassen gleich, unabhängig von ihrer Häufigkeit.

$$Macro\ F1 = \frac{\sum_{i=1}^{n} F1\ Score_i}{n}$$

- Weighted Average: Ähnlich wie der Macro F1-Score, jedoch gewichtet nach der Häufigkeit der einzelnen Klassen. Dies bietet eine Balance zwischen Micro- und Macro-Ansätzen.

$$Weighted\ F1 = \sum_{i=1}^{N} w_i \times F1\ Score_i$$

Towards Data Science - Micro, Macro & Weighted Averages of F1 Score, Clearly Explained

1.1.5 Konfusionsmatrix

Die Konfusionsmatrix bietet eine detaillierte Darstellung der korrekten und falschen Vorhersagen für jede Klasse und ist ein nützliches Werkzeug zur visuellen Analyse der Leistung eines Klassifikationsmodells.

$$Confusion\ Matrix = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

- **True Positive (TP)**: Ein Objekt wurde als positiv vorhergesagt und ist tatsächlich positiv.
- **True Negative (TN)**: Ein Objekt wurde als negativ vorhergesagt und ist tatsächlich negativ.
- **False Positive (FP)**: Ein Objekt wurde als positiv vorhergesagt, ist aber tatsächlich negativ.
- **False Negative (FN)**: Ein Objekt wurde als negativ vorhergesagt, ist aber tatsächlich positiv.

1.1.6 ROC

Die ROC-Kurve (Receiver Operating Characteristic) ist eine grafische Darstellung der Leistung eines binären Klassifikators, der die Trade-offs zwischen den wahren positiven Raten (TPR) und den falsch positiven Raten (FPR) veranschaulicht. Sie wird häufig verwendet, um die Leistungsfähigkeit eines Klassifikationsmodells zu bewerten, wenn die Klassen ungleichmässig verteilt sind.

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

1.1.7 AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

Die AUC (Area Under Curve) misst die gesamte zweidimensionale Fläche unterhalb der gesamten ROC-Kurve und bietet damit ein aggregiertes Mass der Leistung über alle möglichen Klassifikationsschwellen hinweg. Der ROC-AUC-Wert reicht von 0 bis 1, wobei 1 für ein perfektes Modell und 0.5 für ein Modell steht, das nicht besser ist als eine zufällige Schätzung. Eine hohe AUC-ROC deutet darauf hin, dass das Modell eine gute Trennung zwischen den Klassen erreicht, was bei ungleicher Klassenverteilung entscheidend ist.

1.1.8 Matthews Korrelationskoeffizient (MCC)

Der Matthews Korrelationskoeffizient ist eine Metrik, die die Qualität der binären (zwei Klassen) Klassifikationen bewertet. Er ist besonders nützlich, wenn die Klassen ungleichmässig verteilt sind. Der MCC reicht von -1 bis +1. Ein Koeffizient von +1 steht für eine perfekte Vorhersage, 0 ist nicht besser als eine zufällige Vorhersage, und -1 bedeutet, dass Vorhersage und Beobachtung völlig auseinanderklaffen.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

1.2 Metrik Auswahl

Bei der Bewertung von Klassifikationsmodellen mit einer ungleichen Verteilung der Immobilienarten sind bestimmte Metriken besonders aussagekräftig und sollten daher bevorzugt verwendet werden. In solchen Szenarien ist die herkömmliche Genauigkeitsmetrik (Accuracy) oft irreführend, da sie die Leistung des Modells in Bezug auf die überwiegende Klasse überbetont und die Minderheitenklasse vernachlässigt.

Wir haben uns also entschieden, uns auf diese drei spezifische Metriken zu konzentrieren, um die Leistung unserer Klassifikationsmodelle zu bewerten:

Eine der wichtigsten Metriken in diesem Kontext ist der `F1-Score`. Da er den harmonischen Mittelwert von Precision und Recall darstellt, bietet der F1-Score eine ausgewogene Bewertung der Modellleistung, indem er sowohl die Fähigkeit des Modells berücksichtigt, tatsächlich positive Fälle korrekt zu identifizieren (Recall), als auch die Genauigkeit seiner positiven Vorhersagen (Precision). Diese Balance ist besonders wichtig in Situationen, in denen sowohl die Vermeidung von falsch positiven als auch von falsch negativen Ergebnissen von Bedeutung ist.

Wir werden besonders auf den `Weighted F1-Score` achten, da unserer Meinung nach, die Genauigkeit bei den häufigsten Klassen besonders wichtig ist.

Für eine umfassendere Bewertung ist auch die `Area Under the Receiver Operating Characteristic Curve (AUC-ROC)` von grossem Wert. Die AUC-ROC misst, wie gut das Modell zwischen den Klassen unterscheidet, unabhängig von der Schwellenwertwahl.

Zusätzlich ist der `Matthews Korrelationskoeffizient (MCC)` eine robuste Metrik, die auch bei stark ungleich verteilten Daten effektiv ist. Der MCC berücksichtigt alle vier Werte der Konfusionsmatrix (TP, FP, TN, FN) und liefert einen ausgewogenen Score, der sowohl die Stärken als auch die Schwächen des Modells in Bezug auf alle Aspekte der Klassifikation widerspiegelt.

1.3 Fehlerabschätzung für neue Daten

Um eine gute Abschätzung des Fehlers für neue Daten zu erhalten, insbesondere in Kontexten wie maschinellem Lernen und statistischer Modellierung, gibt es verschiedene wichtige Aspekte, die berücksichtigt werden sollten:

1. Vermeidung von Overfitting: Um Overfitting zu vermeiden, muss ichergestellt werden, dass das Modell nicht nur auf den Trainingsdaten gut funktioniert, sondern auch auf neuen, unbekannten Daten.

2. Cross-Validation: Kreuzvalidierung ist eine wichtige Methode, um eine bessere Schätzung der Modellleistung zu erhalten.

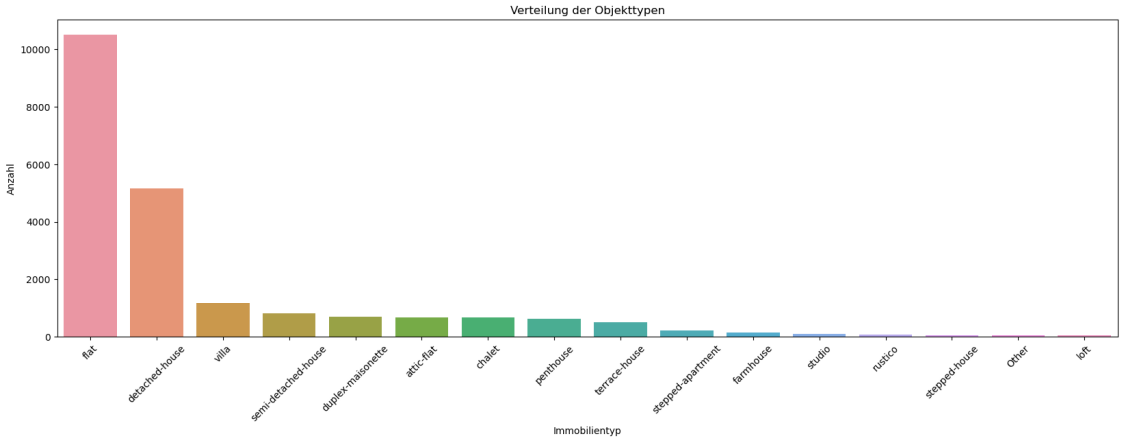
3. Ausgewogenheit der Klassen: Ausserdem sollte darauf geachtet werden, dass falls der Datensatz eine starke Unausgewogenheit in den Klassen aufweist, sollte man die Leistung des Modells anhand der Metriken `F1-Score`, `AUC-ROC` und `MCC` bewerten, da diese Metriken die Leistung des Modells in Bezug auf die Minderheitenklasse besser widerspiegeln.

4. Feature-Importance: Schliesslich sollte analysiert werden, welche Merkmale am meisten zur Klassifizierung beitragen, um die Feature-Importance zu verstehen.

1.4 Daten laden und betrachten

```
Anzahl der Zeilen:  21466
Anzahl der Spalten:  57
```

Da es in diesem Notebook um die Art der Immobilien geht, wollen wir nochmals einen Blick auf deren Verteilung werfen.



Das Diagramm zeigt eine sehr ungleichmässige Verteilung der Immobilientypen. `flat` ist die häufigste Kategorie, gefolgt von `detached house`, wobei die Häufigkeit der anderen Typen deutlich abnimmt. Eine solche Verteilung bedeutet folgendes im Sinne eines Klassifikationsmodells:

1. **Modell Bias**: Das Modell kann zu Vorhersagen für die Mehrheitsklassen (`flat` und `detached house`) neigen, da es mehr Datenpunkte gibt, aus denen das Modell für diese Klassen lernen kann.
2. **Performance Metriken**: Die Standardgenauigkeit ist möglicherweise keine gute Leistungskennzahl, da die Vorhersage von überwiegend `flat` zwar eine höhere Genauigkeit, aber einen geringen Modellnutzen ergeben könnte. Metriken, die einen besseren Einblick in die klassenspezifische Leistung geben, wie `F1-Score`, `AUC-ROC` oder der `Matthews-Korrelationskoeffizient`, wären besser geeignet.
3. **Resampling-Techniken**: Techniken wie das Oversampling der Minderheitsklassen, das Undersampling der Mehrheitsklassen oder die Verwendung synthetischer Daten (wie SMOTE) können notwendig sein, um ein gut verallgemeinertes Modell zu trainieren.
4. **Klassengewichte**: Die Anpassung der Klassengewichte im Modell könnte dazu beitragen, die unausgewogenen Daten zu kompensieren, indem den Minderheitsklassen beim Training mehr Bedeutung beigemessen wird.
5. **Modellbewertung**: Das Modell sollte anhand von Metriken bewertet werden, die das Klassenungleichgewicht berücksichtigen, wie z. B. der `ROC-AUC-Score`, um sicherzustellen, dass es in allen Klassen und nicht nur in der Mehrheitsklasse gute Leistungen erbringt.

1.5 Daten vorbereiten

Hier bereiten wir die Daten für die Modellierungen vor. Die Vorbereitung der Daten erfolgt in den gleichen Schritten wie bei der Aufgabe 2.2 (Bestmögliches Regressionsmodell).

1. Spalten löschen
2. Kategorische und numerische Spalten trennen
3. Kategorische Spalten mit `LabelEncoder` kodieren.
4. Ausreisser mittels Quantile entfernen

1.6 Train Test Split

Nun werden die Daten in 80% Trainings- und 20% Testdaten aufgeteilt. Die Zielvariable ist `type_unified` und wird ebenfalls getrennt.

```
X_train shape:  (17172, 25)
y_train shape:  (17172,)
X_test shape:  (4294, 25)
y_test shape:  (4294,)
```

1.7 Imputation

Für die Imputation der fehlenden Werte verwenden wir den `KNNImputer` von `sklearn`. Dieser Imputer verwendet die k-Nearest Neighbors Methode, um fehlende Werte zu imputieren. In unserem Fall verwenden wir die 5 nächsten Nachbarn.

1.8 Standardisierung

Hier verwenden wir den `StandardScaler` von `sklearn`, um die Daten zu standardisieren. Dieser Skalierer standardisiert Merkmale, indem er den Mittelwert auf 0 und die Standardabweichung auf 1 setzt. Wir berücksichtigen dabei nur die numerischen Spalten.

1.9 Scorers

Hier definieren wir unsere `Scorers` für die Modellbewertung. Wie oben bereits erwähnt verwenden wir die Metriken `F1-Score`, `ROC-AUC-Score` und `Matthews-Korrelationskoeffizient`.

```
roc_auc make_scorer(roc_auc_score, needs_proba=True, multi_class=ovr)
f1 make_scorer(f1_score, average=weighted)
mcc make_scorer(matthews_corrcoef)
```

1.10 Modellierung

Wir werden fünf verschiedene Modelle verwenden, um die Immobilienobjekte zu klassifizieren:

1. Logistic Regression (Linear Model)
2. Random Forest Classifier (Ensemble)
3. Hist Gradient Boosting Classifier (Ensemble)
4. Support Vector Classifier (Support Vector Machine)
5. KNN Classifier (Nearest Neighbors)

Um das Verfahren für alle drei Modelle zu beschreiben, hier eine detaillierte Erklärung des gesamten Modellierungsprozesses:

Definition der Parameter-Grids: Für jedes Modell wird ein separates `param_grid` definiert, das die zu optimierenden Hyperparameter enthält. Diese Grids variieren je nach den spezifischen Hyperparametern, die wir für jedes Modell als relevant erachten.

Durchführung des Grid Search: Mit `GridSearchCV` wird für jedes Modell und jede Bewertungsmetrik ein Prozess durchgeführt. Dabei werden verschiedene Kombinationen der Parameter aus den jeweiligen `param_grids` ausprobiert, um die beste Kombination gemäss der jeweiligen

Bewertungsmetrik zu finden. Eine 5-fache Kreuzvalidierung (`cv=5`) wird verwendet, um die Modellleistung zu bewerten.

Modelltraining und Evaluierung: Jedes Modell wird mit den verschiedenen Parameterkombinationen trainiert. Nach dem Grid-Search Prozess wird das beste Modell für jede Metrik identifiziert.

Speichern der Ergebnisse: Die Ergebnisse, einschliesslich der besten Parameter, der erzielten Scores und der Vorhersagen, werden für jede Metrik in einem separaten Dictionary für jedes Modell gespeichert. Für den `roc_auc` Score werden die Wahrscheinlichkeiten und die Klassen-Vorhersagen gespeichert, während für andere Metriken die direkten Vorhersagen gespeichert werden.

Ergebnisanalyse: Nach Abschluss der Grid-Search Prozesse für alle Modelle werden die gespeicherten Ergebnisse analysiert mittels Confusion Matrix und Classification Report, um die Leistung der Modelle für das Identifizieren der Immobilienobjekte zu vergleichen.

1.10.1 Logistic Regression

Die logistische Regression ist ein lineares Klassifikationsmodell, das zur Vorhersage binärer oder multiklassiger Ergebnisse verwendet wird. Es modelliert die Wahrscheinlichkeit des Eintretens eines bestimmten Ereignisses als Funktion der unabhängigen Variablen. Es verwendet die logistische Funktion (auch Sigmoid-Funktion genannt), um Wahrscheinlichkeiten zu schätzen und eine Entscheidungsgrenze zu ziehen.

scikit-learn.org - Logistic Regression

Hyperparameter-Tuning mittels verschiedener Metriken

```
Optimierung mit roc_auc
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Optimierung mit f1
Fitting 5 folds for each of 36 candidates, totalling 180 fits
Optimierung mit mcc
Fitting 5 folds for each of 36 candidates, totalling 180 fits


Ergebnisse für roc_auc:
Beste Parameter: {'C': 1, 'class_weight': 'balanced', 'max_iter': 100,
'penalty': 'l2'}
Score: 0.769139671138015

Ergebnisse für f1:
Beste Parameter: {'C': 0.001, 'class_weight': 'balanced', 'max_iter': 100,
'penalty': 'l2'}
Score: 0.5284115649365345

Ergebnisse für mcc:
Beste Parameter: {'C': 0.5, 'class_weight': None, 'max_iter': 100, 'penalty':
'l2'}
Score: 0.3994444474540297
```

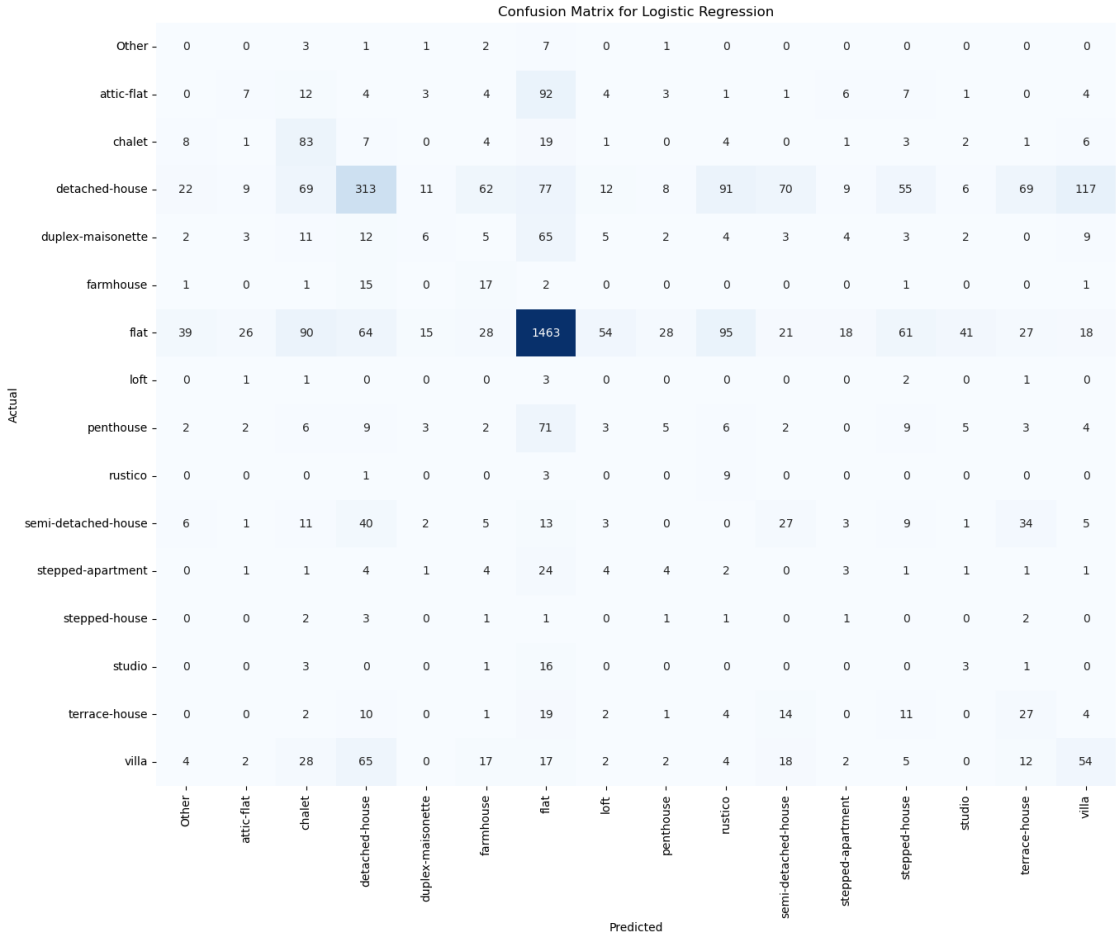
1. **ROC AUC (Receiver Operating Characteristic Area Under Curve) Wert: 0.769**

- Ein Wert von 0.769 ist recht gut und deutet darauf hin, dass das Modell gut in der Lage ist, zwischen den verschiedenen Klassen in Ihrem Datensatz zu unterscheiden.
- In einer Mehrklassenumgebung wird diese Punktzahl über alle Klassen hinweg gewichtet und gibt die Gesamteffektivität des Modells über alle Kategorien hinweg an.

2. **F1-Score: 0.528**
    - Ein Wert von 0.528 ist mässig niedrig.
    - Dieser niedrigere Wert im Vergleich zum ROC AUC könnte darauf hindeuten, dass das Modell zwar die positive Klasse besser als die negative Klasse einstuft, aber möglicherweise nicht so effektiv bei der genauen Identifizierung echter Positiver ist oder eine erhebliche Anzahl falsch positiver oder falsch negativer Ergebnisse erzeugt.
3. **Matthews Korrelationskoeffizient: 0.399**
    - Ein Wert von 0.399 ist zwar positiv, aber relativ niedrig. Dies deutet darauf hin, dass das Modell eine bescheidene Qualität der Vorhersagen aufweist.
    - MCC gilt als ausgewogenes Mass, auch wenn die Klassen sehr unterschiedlich gross sind, so dass dieser Wert besonders nützlich ist, wenn der Datensatz unausgewogen ist.

Für die weitere Analysen des Logistischen Regressionsmodells haben wir uns entschieden, die `ROC AUC` Metrik zu verwenden, da diese Metrik die beste Leistung zu liefern scheint.

Confusion Matrix Für die Konfusionsmatrix müssen wir die Vorhersagen für den `roc_auc` Score auswählen.



Classification Report

	precision	recall	f1-score	support
Other	0.00	0.00	0.00	15
attic-flat	0.13	0.05	0.07	149
chalet	0.26	0.59	0.36	140
detached-house	0.57	0.31	0.40	1000
duplex-maisonette	0.14	0.04	0.07	136
farmhouse	0.11	0.45	0.18	38
flat	0.77	0.70	0.74	2088
loft	0.00	0.00	0.00	8
penthouse	0.09	0.04	0.05	132
rustico	0.04	0.69	0.08	13
semi-detached-house	0.17	0.17	0.17	160
stepped-apartment	0.06	0.06	0.06	52
stepped-house	0.00	0.00	0.00	12
studio	0.05	0.12	0.07	24

```
terrace-house       0.15      0.28      0.20        95
        villa       0.24      0.23      0.24       232

     accuracy                           0.47      4294
    macro avg       0.17      0.23      0.17      4294
 weighted avg       0.55      0.47      0.50      4294
```

Interpretation der Ergebnisse Anhand der Konfusions-Matrix und Klassifikationsbericht können wir sehen, dass das Modell Verbesserungsbedarf hat, um die Klassifizierung der Immobilienobjekte zu verbessern. Wenn wir die Diagonale der Konfusionsmatrix betrachten, können wir sehen, dass die Immobilienobjekte hauptsächlich nur als die häufigsten Klassen (`flat` und `detached-house`) klassifiziert werden. Die anderen Klassen werden nicht gut erkannt.

Auffällig ist das Immobilienobjekt `chalet`, welches nur 140 mal vorkommt, im Vergleich zu den anderen Klassen gut klassifiziert wird (83 Treffer). Was auch ins Auge sticht, ist dass `penthouse` mit 132 Fällen, nur 5 mal richtig klassifiziert wurde. Dies liegt wahrscheinlich daran, dass die Klasse `penthouse` sehr ähnlich zu `flat` ist und somit das Modell diese beiden Klassen nicht gut unterscheiden kann.

1.10.2 Random Forest Classifier

Der Random Forest Classifier ist ein Ensemble-Lernmodell, das aus einer Menge von Entscheidungsbäumen besteht. Es erstellt mehrere Entscheidungsbäume auf zufällig ausgewählten Teilmengen der Trainingsdaten und kombiniert ihre Vorhersagen, um Overfitting zu reduzieren und genauere Vorhersagen zu erzielen. Es eignet sich gut für Klassifikationsaufgaben.

scikit-learn.org - Random Forest Classifier

Hyperparameter-Tuning mittels verschiedener Metriken

```
Optimierung mit roc_auc
Fitting 5 folds for each of 162 candidates, totalling 810 fits
Optimierung mit f1
Fitting 5 folds for each of 162 candidates, totalling 810 fits
Optimierung mit mcc
Fitting 5 folds for each of 162 candidates, totalling 810 fits


Ergebnisse für roc_auc:
Beste Parameter: {'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf':
2, 'min_samples_split': 10, 'n_estimators': 500}
Score: 0.8767396465051003

Ergebnisse für f1:
Beste Parameter: {'max_depth': None, 'max_features': 'log2', 'min_samples_leaf':
1, 'min_samples_split': 2, 'n_estimators': 200}
Score: 0.6682668566782995

Ergebnisse für mcc:
```

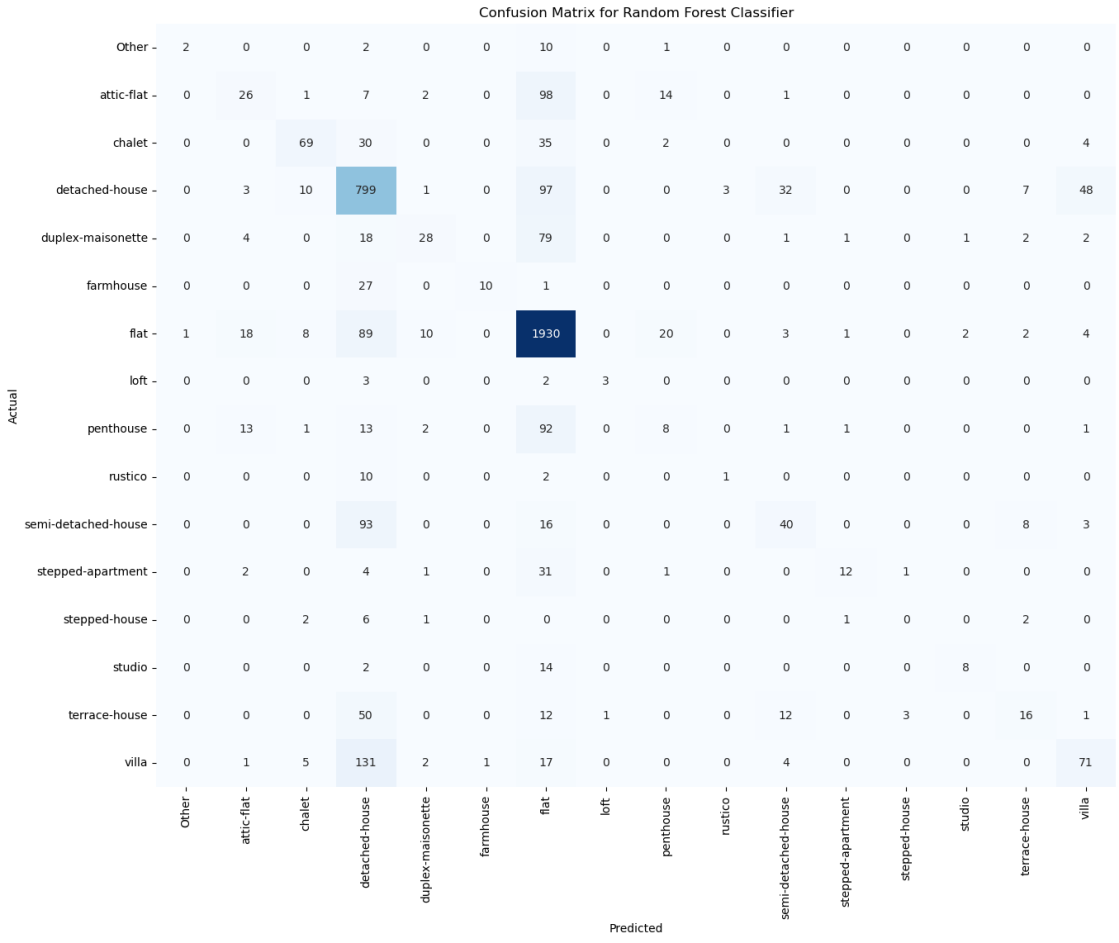
```
Beste Parameter: {'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf':
1, 'min_samples_split': 2, 'n_estimators': 500}
Score: 0.555019282610499
```

1. **ROC AUC-Wert: 0.876**: Zeigt eine gute Fähigkeit an, zwischen den Klassen zu unterscheiden. Um einiges effektiver als das Modell der logistischen Regression.
2. **F1 Ergebnis: 0.668**: Dies ist ebenfalls eine Verbesserung gegenüber dem logistischen Regressionsmodell und deutet auf ein besseres Gleichgewicht zwischen Präzision und Recall hin.
3. **MCC: 0.555**: Eine deutliche Verbesserung gegenüber dem logistischen Regressionsmodell, was auf eine bessere Gesamtvorhersagequalität hindeutet.

Gesamtinterpretation: Das Random-Forest-Modell zeigt im Vergleich zum logistischen Regressionsmodell eine ausgewogene Leistung über alle Metriken hinweg, mit deutlich besseren F1- und MCC-Werten, was auf eine bessere Gesamteffektivität der Klassifizierung hindeutet.

Schauen wir mittels der Konfusionsmatrix und des Klassifikationsberichts etwas genauer hin. Wir werden hierzu die Vorhersagen für den `f1` Score auswählen.

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
Other	0.67	0.13	0.22	15
attic-flat	0.39	0.17	0.24	149
chalet	0.72	0.49	0.58	140
detached-house	0.62	0.80	0.70	1000
duplex-maisonette	0.60	0.21	0.31	136
farmhouse	0.91	0.26	0.41	38
flat	0.79	0.92	0.85	2088
loft	0.75	0.38	0.50	8
penthouse	0.17	0.06	0.09	132
rustico	0.25	0.08	0.12	13
semi-detached-house	0.43	0.25	0.31	160
stepped-apartment	0.75	0.23	0.35	52
stepped-house	0.00	0.00	0.00	12
studio	0.73	0.33	0.46	24
terrace-house	0.43	0.17	0.24	95
villa	0.53	0.31	0.39	232
accuracy			0.70	4294
macro avg	0.55	0.30	0.36	4294
weighted avg	0.67	0.70	0.67	4294

Interpretation der Ergebnisse Die Analyse des Random Forest Klassifikators zeigt eine gute Gesamtleistung mit einer Genauigkeit von 70%, wobei besonders hohe Werte bei der Klassifizierung der häufigsten Klasse `flat` erreicht werden. Allerdings offenbaren die Ergebnisse auch Schwächen in der Klassifizierung seltenerer Klassen, wie z.B. `stepped-house` oder `penthouse`, was sich in niedrigen Präzisions- und Recall-Werten manifestiert. Die Diskrepanz zwischen dem Macro Average und dem Weighted Average deutet darauf hin, dass das Modell bei weniger häufigen Klassen Verbesserungspotenzial hat.

1.10.3 Hist Gradient Boosting Classifier

Der Histogram Gradient Boosting Classifier ist ein leistungsstarker Ensemble-Klassifikator, der auf dem Gradienten-Boosting-Algorithmus basiert. Er erstellt eine Kombination von Entscheidungsbäumen, indem er schrittweise Schwachstellen (in diesem Fall Bäume) anpasst, um den Fehler zu minimieren. Dieser Classifier verwendet Histogramm-basierte Techniken, um die Berechnung zu beschleunigen und ist für grosse Datensätze geeignet.

scikit-learn.org - Histogram Gradient Boosting Classifier

Hyperparameter-Tuning mittels verschiedener Metriken

```
Optimierung mit roc_auc
Fitting 5 folds for each of 144 candidates, totalling 720 fits
Optimierung mit f1
```

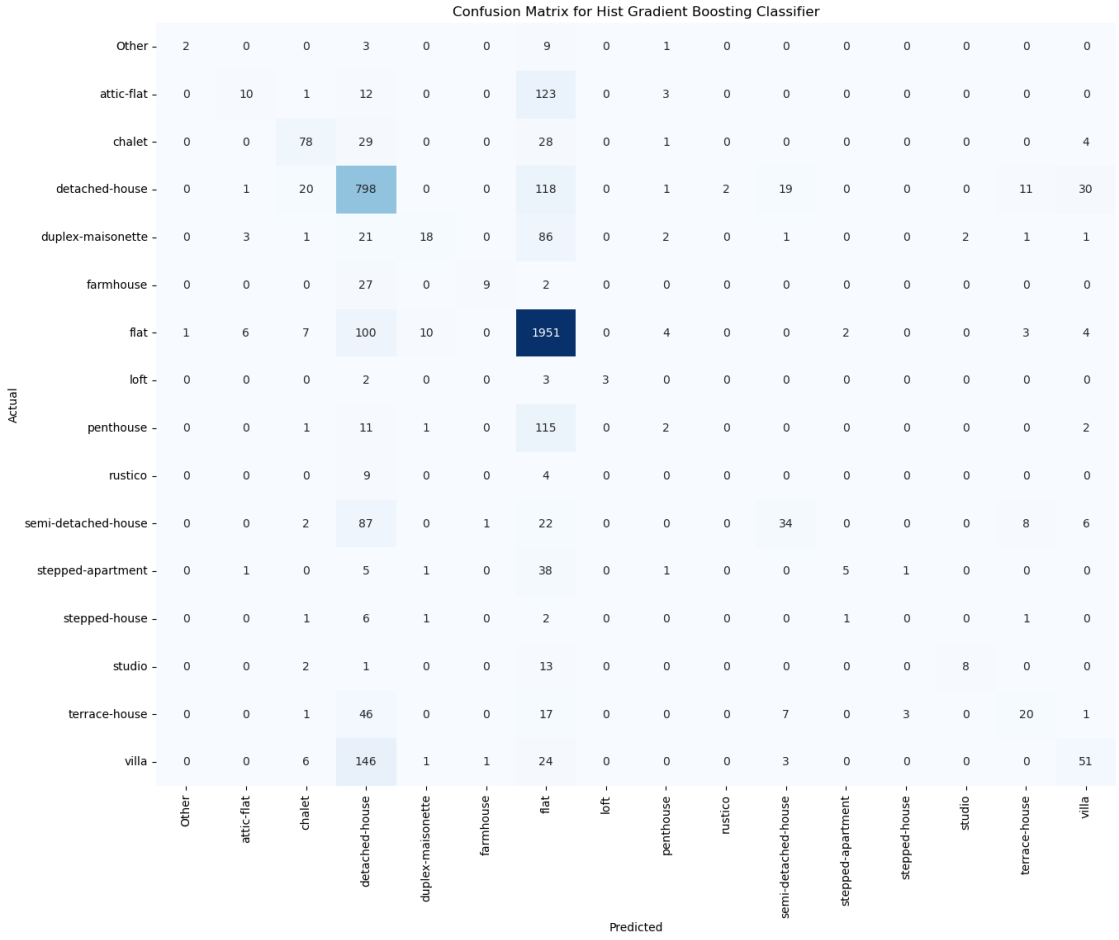
```
Fitting 5 folds for each of 144 candidates, totalling 720 fits
Optimierung mit mcc
Fitting 5 folds for each of 144 candidates, totalling 720 fits


Ergebnisse für roc_auc:
Beste Parameter: {'l2_regularization': 0.5, 'learning_rate': 0.1, 'max_depth':
20, 'max_iter': 100, 'min_samples_leaf': 50}
Score: 0.8584043441758009

Ergebnisse für f1:
Beste Parameter: {'l2_regularization': 1, 'learning_rate': 0.1, 'max_depth':
None, 'max_iter': 100, 'min_samples_leaf': 50}
Score: 0.6483233528361109

Ergebnisse für mcc:
Beste Parameter: {'l2_regularization': 1, 'learning_rate': 0.1, 'max_depth':
None, 'max_iter': 100, 'min_samples_leaf': 30}
Score: 0.534944787968145
```

1. **ROC AUC-Wert: 0.858**: Vergleichbar mit dem Random-Forest-Modell, was auf eine gute Klassentrennungsfähigkeit hinweist.
2. **F1 Ergebnis: 0.648**: Ähnlich wie das Random Forest Modell, was auf eine mässige Ausgewogenheit von Präzision und Recall hinweist.
3. **MCC: 0.534**: Etwas niedriger als das Random-Forest-Modell, aber immer noch eine mässige Vorhersagequalität.

Gesamtinterpretation: Der Hist Gradient Boosting Classifier zeigt eine ausgewogene Leistung mit einem guten ROC-AUC-Wert und moderaten F1- und MCC-Werten. Er ist konkurrenzfähig mit dem Random-Forest-Modell.

Wir werden nun mit den `MCC` Scores die Konfusionsmatrix und den Klassifikationsbericht erstellen.

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
Other	0.67	0.13	0.22	15
attic-flat	0.48	0.07	0.12	149
chalet	0.65	0.56	0.60	140
detached-house	0.61	0.80	0.69	1000
duplex-maisonette	0.56	0.13	0.21	136
farmhouse	0.82	0.24	0.37	38
flat	0.76	0.93	0.84	2088
loft	1.00	0.38	0.55	8
penthouse	0.13	0.02	0.03	132
rustico	0.00	0.00	0.00	13
semi-detached-house	0.53	0.21	0.30	160
stepped-apartment	0.62	0.10	0.17	52
stepped-house	0.00	0.00	0.00	12
studio	0.80	0.33	0.47	24

```
terrace-house       0.45      0.21      0.29        95
        villa       0.52      0.22      0.31       232

     accuracy                           0.70      4294
    macro avg       0.54      0.27      0.32      4294
 weighted avg       0.65      0.70      0.65      4294
```

Interpretation der Ergebnisse Das Hist Gradient Boosting Classifier Modell kann Immobilienobjekte zu 70% korrekt klassifizieren. Dies ist ein relativ guter Wert, zeigt aber auch, dass Raum für Verbesserungen besteht. Das Modell hat Schwierigkeiten, seltene Klassen zu erkennen, wie z.B. `rustico` oder `stepped-house`. Auffallend ist aber, dass die Klasse `loft` 3 von 8 mal korrekt klassifiziert wurde (Wie beim Random Forest Classifier) und ein anderes Objekt nie als `loft` klassifiziert wurde. Da der Macro Average immer noch sehr tief ist, performt das Modell bei häufigen Klassen besser als bei seltenen Klassen.

1.10.4 K-Nearest Neighbors (KNN)

Der K-Neares Neighbor Algorhitmus auch KNN oder k-NN genannt, ist ein supervised-learning Klassifikationsalgorhitmus, welcher anhand "proximity" - also nähe der values klassifikationen oder vorhersagen zur grupppierung eines Datenpunktes macht. KNN kann für klassifikations und regressionsprobleme verwendet werden, wwird aber typischerweise vorallem für ersteres verwendet.

Ein Datenpunkt wird also genommen und ein Radius um den Punkt berechnet (es kann zwischen diversen Distanzberechnungen gewählt werden, z.B. Euklidische D, Manhattan D, Minkowski D, etc.). Danach wird der Datenpunkt zur Gruppe genommen welche innerhalb dieses Radius am meisten vertreten ist.

scikit-learn.org - K-Nearest Neighbors

Hyperparameter-Tuning mittels verschiedener Metriken

```
Optimierung mit roc_auc
Fitting 5 folds for each of 48 candidates, totalling 240 fits
Optimierung mit f1
Fitting 5 folds for each of 48 candidates, totalling 240 fits
Optimierung mit mcc
Fitting 5 folds for each of 48 candidates, totalling 240 fits


Ergebnisse für roc_auc:
Beste Parameter: {'algorithm': 'auto', 'n_neighbors': 20, 'weights': 'distance'}
Score: 0.7783028991991482

Ergebnisse für f1:
Beste Parameter: {'algorithm': 'kd_tree', 'n_neighbors': 5, 'weights':
'distance'}
Score: 0.6284621424613963

Ergebnisse für mcc:
```

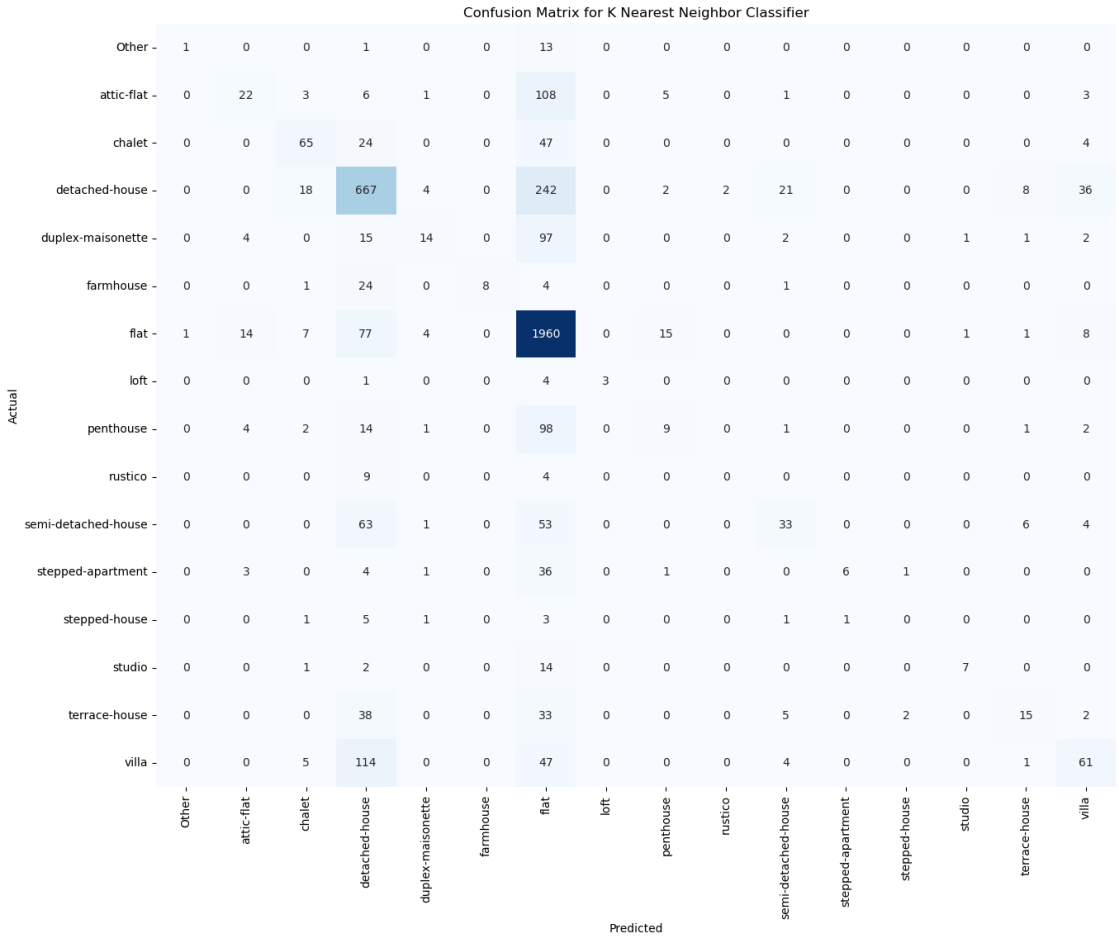
```
Beste Parameter: {'algorithm': 'auto', 'n_neighbors': 15, 'weights': 'distance'}
Score: 0.48617338570613516
```

1. **ROC AUC-Wert: 0.778**: Der ROC ist gerade noch so in Ordnung und ist ähnlich wie der Wert bei der Logistischen Regression.
2. **F1 Ergebnis: 0.628**: Der F1 Wert bewegt sich in der gleichen umgebung wie die von RF und HGB und ist somit nicht optimal aber auch nicht sehr schlecht, da es eine gute Balance zwischen Precision und Recall vermuten lässt.
3. **MCC: 0.486**: Der MCC befindet sich zwischen den Werten von RF, HGB auf der einen und Logistischer Regression auf der anderen. Es ist also moderat gut, es ist kein sehr gutes Ergebnis, zeigt aber ein gewisses Mass von Verlässlichkeit der Klassifizierung (nicht total random).

Gesamtinterpretation: Der KNN Classifier performt also "moderat", hat eine gute Balance zwischen Recall und Precision und eine moderate Abweichung von Prediction und effektivem Ergebnis.

Wir werden nun mit den `MCC` Scores die Konfusionsmatrix und den Klassifikationsbericht erstellen.

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
Other	0.50	0.07	0.12	15
attic-flat	0.47	0.15	0.22	149
chalet	0.63	0.46	0.53	140
detached-house	0.63	0.67	0.65	1000
duplex-maisonette	0.52	0.10	0.17	136
farmhouse	1.00	0.21	0.35	38
flat	0.71	0.94	0.81	2088
loft	1.00	0.38	0.55	8
penthouse	0.28	0.07	0.11	132
rustico	0.00	0.00	0.00	13
semi-detached-house	0.48	0.21	0.29	160
stepped-apartment	0.86	0.12	0.20	52
stepped-house	0.00	0.00	0.00	12
studio	0.78	0.29	0.42	24
terrace-house	0.45	0.16	0.23	95
villa	0.50	0.26	0.34	232
accuracy			0.67	4294
macro avg	0.55	0.25	0.31	4294
weighted avg	0.63	0.67	0.62	4294

Interpretation der Ergebnisse Auch hier ist diese Klassifizierung mit 67% Accuracy zwischen der Logistischen Regression und den anderen beiden bisherigen Classifier. Die prominenteste Klasse `flat` wird wieder am meisten predicted, dies vermutlich weil Wohnungen in so vielen Arten vorkommen und somit dann auch schnell in KNN Radius die Mehrzahl bilden. Klassen mit weniger Einträgen werden schlechter vorhergesagt.

1.10.5 Support Vector Machine (SVM)

Support Vecor Machines ist eine Sammlung von supervied-learning Methoden, welche sich für Klassifizierungs-, Regressions- und Ausreissererkennungsprobleme eignen.

Wir verwenden hier SVC -> C-Support Vector Classification.

Die Klassifizierung eines Datenpunktes hängt von der Berechnung eines "Hyperplanes" ab, der im Wesentlichen eine "Entscheidungsgrenze" darstellt. Während des Trainingsprozesses wird dieser Hyperplane optimiert und durch Hilfsvektoren, die sich in der Nähe dieser Hyperplane befinden, ergänzt. Die endgültige Entscheidung über die Zugehörigkeit eines Datenpunktes zu einer bestimmten Klasse basiert auf der Berechnung der Distanz zwischen diesem Datenpunkt und der Hyperplane.

scikit-learn.org - Support Vector Machine

Hyperparameter-Tuning mittels verschiedener Metriken Die Anwendung von Non-Linear Support Vector Classification (SVC) auf eine grosse Anzahl von Datenpunkten in X_Train ist

rechnerisch aufwendig und erfordert eine erhebliche Zeit. Andererseits erweist sich LinearSCV als unzureichend präzise für unsere komplexe Klassifizierungsaufgabe.

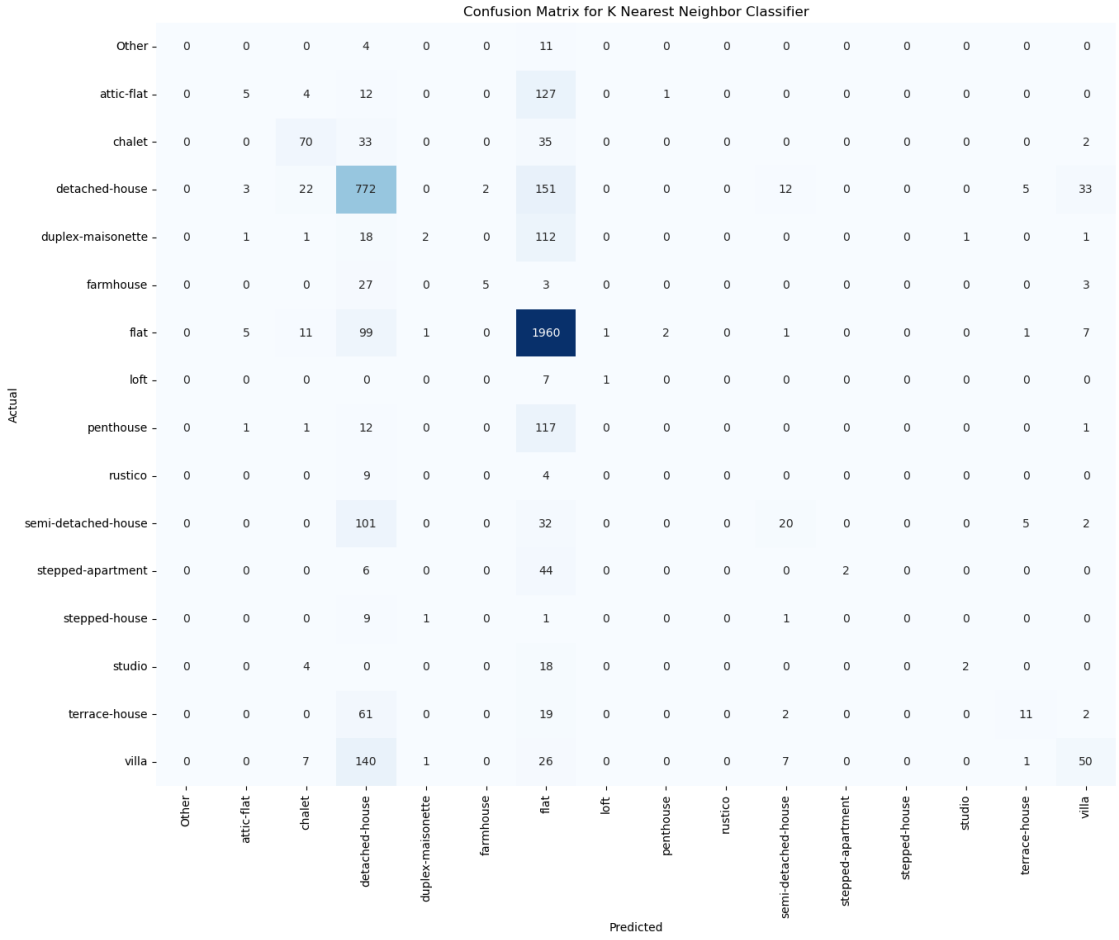
```
Optimierung mit roc_auc
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Optimierung mit f1
Fitting 5 folds for each of 12 candidates, totalling 60 fits
Optimierung mit mcc
Fitting 5 folds for each of 12 candidates, totalling 60 fits


Ergebnisse für roc_auc:
Beste Parameter: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Score: 0.8425419272480281

Ergebnisse für f1:
Beste Parameter: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}
Score: 0.6093283722171565

Ergebnisse für mcc:
Beste Parameter: {'C': 10, 'gamma': 'auto', 'kernel': 'rbf'}
Score: 0.49658537090034094
```

1. **ROC AUC-Wert von 0.842**: Mit einem Wert von 0.842 erzielt das Modell eine solide Leistung. Obwohl dieser Wert nicht im Bereich "sehr gut" liegt, zeigt er dennoch, dass die Vorhersagen des Modells insgesamt gut sind. Es gibt jedoch Raum für Verbesserungen. Die Leistung liegt nur geringfügig unter der von Ensemble-Methoden.
2. **F1-Ergebnis von 0.609**: Das F1-Ergebnis des Modells beträgt 0.609, was zwar etwas schlechter ist als bei K-Nearest Neighbors, aber dennoch als moderat gut angesehen werden kann. Dies deutet darauf hin, dass das Modell eine einigermassen ausgewogene Balance zwischen Genauigkeit und Präzision aufweist.
3. **MCC-Wert von 0.496**: Der MCC-Wert liegt fast auf dem gleichen Niveau wie bei KNN und kann ebenfalls als moderat gut bewertet werden. Dies zeigt eine akzeptable Zuverlässigkeit in der Klassifizierung der Daten.

Gesamtinterpretation: Das Modell bewegt sich in einem ähnlichen Bereich wie KNN. Obwohl der ROC AUC-Wert etwas besser ist, sind die F1- und MCC-Werte etwas schlechter.

Wir werden nun mit den `MCC` Scores die Konfusionsmatrix und den Klassifikationsbericht erstellen.

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
Other	0.00	0.00	0.00	15
attic-flat	0.33	0.03	0.06	149
chalet	0.58	0.50	0.54	140
detached-house	0.59	0.77	0.67	1000
duplex-maisonette	0.40	0.01	0.03	136
farmhouse	0.71	0.13	0.22	38
flat	0.73	0.94	0.82	2088
loft	0.50	0.12	0.20	8
penthouse	0.00	0.00	0.00	132
rustico	0.00	0.00	0.00	13
semi-detached-house	0.47	0.12	0.20	160
stepped-apartment	1.00	0.04	0.07	52
stepped-house	0.00	0.00	0.00	12
studio	0.67	0.08	0.15	24

```
terrace-house       0.48      0.12      0.19        95
        villa       0.50      0.22      0.30       232

     accuracy                           0.68      4294
    macro avg       0.44      0.19      0.22      4294
 weighted avg       0.62      0.68      0.61      4294
```

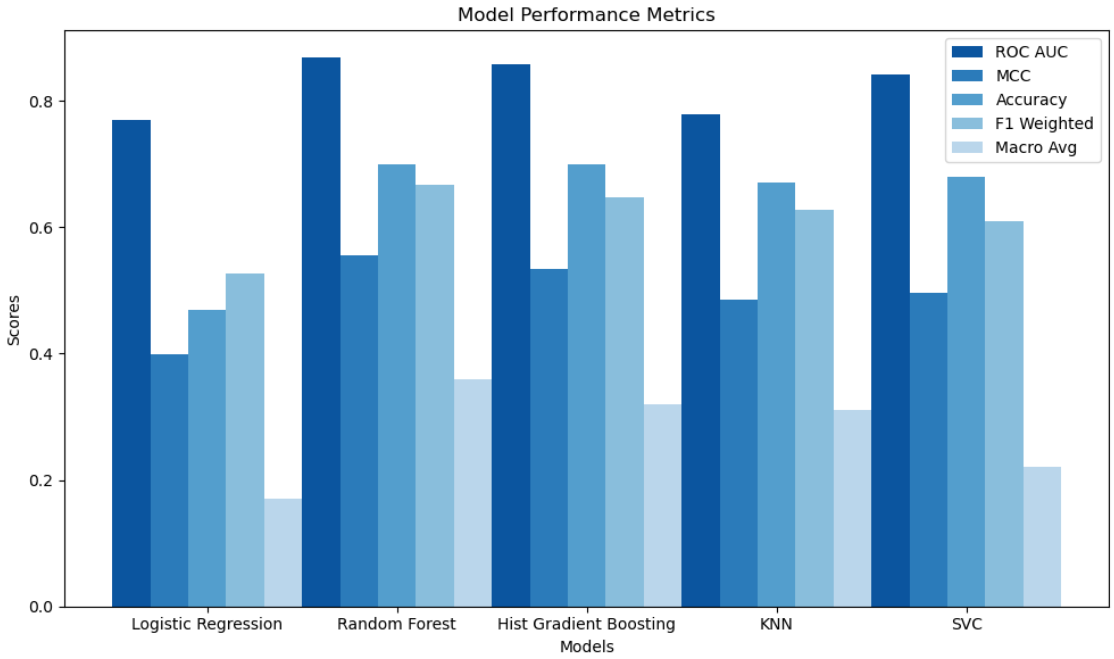
Interpretation der Ergebnisse Es ist hier ebenfalls eine Genaugkeit von 68%, jedoch scheinen die kleineren Kategorien klar schlechter predicted zu werden im Vergleich zu den Ensemble Modelle. Die prominenten Kategorien werden wiederum am besten predicted.

1.11 Modellbewertungen

Wenn wir die Metriken der fünf Modelle auf einem Blick vergleichen, können wir sehen, dass das Random Forest Modell die beste Leistung erzielt. Es hat die höchste `ROC AUC`, `MCC`, `Accuracy`, `F1 Weighted` und `Macro Avg` Werte. Das Hist Gradient Boosting Classifier Modell ist dem Random Forest Modell sehr ähnlich. Das logistische Regressionsmodell ist das schlechteste Modell, da es die niedrigsten Werte für alle Metriken aufweist und hat einen hohen Bias für die häufigste Klasse `flat`.

Modell	ROC AUC	MCC	Accuracy	F1 Weighted	Macro Avg
Logistische Regression	0.769	0.399	0.47	0.526	0.17
Random Forest Classifier	0.868	0.555	0.70	0.668	0.36
Hist Gradient Boosting Classifier	0.858	0.534	0.70	0.648	0.32
K Neares Neighbor Classifier	0.778	0.486	0.67	0.628	0.31
SVM - SVC Classifier	0.842	0.496	0.68	0.609	0.22

Die Resultate stellen wir hier nochmals schöner dar:



1.12 Ausblick

Folgende Punkte könnten in einer weiteren Iteration des Projekts untersucht werden:

1. Erweiterte Hyperparameter-Tuning: Für den Random Forest und den Hist Gradient Boosting Classifier könnten weitere Hyperparameter in den Tuning-Prozess einbezogen werden. Auch eine grössere Bandbreite an Werten für die bereits getesteten Hyperparameter könnte untersucht werden.
2. Weitere Modelle testen: Es könnten weitere Modelle getestet werden, wie z.B. ein Multi-Layer Perceptron oder XGBoost Classifier.
3. Behandlung von Klassenungleichgewichten: Da das logistische Regressionsmodell einen hohen Bias für die Klasse `flat` aufweist, könnte eine Über- oder Unterstichprobierung der Klassen oder der Einsatz von Techniken wie SMOTE (Synthetic Minority Over-sampling Technique) hilfreich sein, um das Klassenungleichgewicht zu mindern.