h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbmn
FACHBEREICH MATHEMATIK
UND NATURWISSENSCHAFTEN

Prof. Dr. H. Zisgen
Markus Schepers
Fachbereich Mathematik und Naturwissenschaften
Sommersemester 2020

Data Mining 1

Pflichtaufgaben – Teil III
- Ensemblemethoden -

Aufgabe 1 (Ensemblemethoden mit Python)
In dieser Aufgabe soll der australische Wetterdatensatz verwendet werden (Aufteilung wie in Booklet-Aufgabenteil I). Weiterführende Links zu Dokumentationen und User Guides der im folgenden genannten Python-Klassen können Sie im Anhang finden.

Für die folgenden Verfahren sollen Modelle angepasst werden:

- **Bagging** mit der Klasse `sklearn.ensemble.BaggingClassifier` und einem Klassifizierungsbaum als Basisklassifizierer:
  - **Klassifizierungsbaum**
    Nutzen Sie die Klasse `sklearn.tree.DecisionTreeClassifier`.
- **Random Forest** mit der Klasse `sklearn.ensemble.RandomForestClassifier`.
- **AdaBoost** mit der Klasse `sklearn.ensemble.AdaBoostClassifier`. und einem Klassifizierungsbaum als Basisklassifizierer:
  - **Klassifizierungsbaum**
    Nutzen Sie die Klasse `sklearn.tree.DecisionTreeClassifier`.

Achten Sie darauf bei allen Verfahren den `random_state` zu setzen.

a) Führen Sie für jeweils für die in Tabelle 1 aufgeführten Kombinationen Hyperparameteroptimierungen mit 10-facher Kreuzvalidierung durch und visualisieren Sie Ihre Ergebnisse jeweils gemeinsam in geeigneten Grafiken. Für die Hyperparameteroptimierungen mit Kreuzvalidierung kann die Klasse `sklearn.model_selection.GridSearchCV` genutzt werden.

b) Führen Sie jeweils für alle Verfahren (incl. Decision Tree) eine Hyperparameteroptimierung über einem geeigneten Parametergitter mit 10-facher Kreuzvalidierung durch. In welchem Bereich liegen die Parameter der besten Klassifikatoren?

Parameter	**Modelle**
`n_estimator`	Bagging, Random Forest, AdaBoost
`criterion`	Bagging, Random Forest, AdaBoost, DecisionTree
`max_depth`	Bagging, Random Forest, AdaBoost, DecisionTree
`min_samples_split`	Bagging, Random Forest, AdaBoost, DecisionTree
`min_samples_leaf`	Bagging, Random Forest, AdaBoost, DecisionTree
`min_weight_fraction_leaf`	Bagging, Random Forest, AdaBoost, DecisionTree
`max_features`	Bagging, Random Forest, AdaBoost, DecisionTree
`max_leaf_nodes`	Bagging, Random Forest, AdaBoost, DecisionTree
`min_impurity_decrease`	Bagging, Random Forest, AdaBoost, DecisionTree
`min_impurity_split`	Bagging, Random Forest, AdaBoost, DecisionTree

Tabelle 1: Parameter-Modell-Kombinationen

Anhang

Dokumentationen und User Guides zu den Klassen

- – Documentation: `sklearn.model_selection.GridSearchCV` (Hyperlink)
  – User Guide: Tuning the hyper-parameters of an estimator (Hyperlink)
  – Beispiel: Demonstration of multi-metric evaluation on cross_val_score and GridSearchCV (Hyperlink)
- – Documentation: `sklearn.tree.DecisionTreeClassifier` (Hyperlink)
  – User Guide: Decision Trees (Hyperlink)
- – Documentation: `sklearn.ensemble.BaggingClassifier` (Hyperlink)
  – User Guide: Bagging meta-estimator (Hyperlink)
- – Documentation: `sklearn.ensemble.RandomForestClassifier` (Hyperlink)
  – User Guide: Forests of randomized trees (Hyperlink)
- – Documentation: `sklearn.ensemble.AdaBoostClassifier` (Hyperlink)
  – User Guide: AdaBoost (Hyperlink)
- Klassenübersicht `model_selection` (Hyperlink)