Random Forest:

* Random Forest ist ein Klassifikations- und Regressionsverfahren
* Besteht aus mehreren unkorrelierten Entscheidungsbäumen
* Alle Entscheidungsbäume sind unter einer bestimmten Art von Randomisierung gewachsen
* Jeder Baum darf eine Entscheidung treffen und die Klasse mit den meisten Stimmen entscheidet über die endgültige Klassifikation

Eigenschaften:

* Klassifikator trainiert sehr schnell 🡪 kurze Trainingszeit eines einzelnen Baums und steigt linear mit der Anzahl der verwendeten Bäume
* Evaluierung geschieht auf jeden Baum einzeln 🡪 Parallelisierbar
* Effizient für sehr große Datenmengen
* Wichtige Klassen können identifiziert werden.
* Zusammenhang zwischen Klassen kann erkannt werden

Funktionsweise:

Diverse Ansätze und Verfahren: 🡪 beispielsweise welche Entscheidungsbäume verwendet werden und welche maximale Tiefe sie besitzen

1. Es werden n-Bootstrap-Samples gezogen
2. Von den M Merkmalen (Features oder Dimensionen) werden an jedem Knoten im Baum m<<M Merkmale zufällig ausgewählt, die als Kriterium für den Schnitt gewählt werden
   1. Die anschließende Auswahl eines Merkmals aus dieser Menge kann zum Beispiel mittels der Minimierung der Entropie geschehen.
3. Der Baum wird voll ausgebaut und nicht zurückgeschnitten

Zur Klassifizierung einer Eingabe wird diese in jedem Baum ausgewertet.

Klasse mit den meisten Stimmen wird als Ergebnis ausgegeben.

**SKLEARN/ Skitlearn – Bibliothek Python**

* Freie Software-Bibliothek zum maschinellen Lernen
* Es bietet verschiedene Klassifikations-, Regressions- und Clustering-Algorithmen, darunter Support-Vektor-Maschinen, Random Forest, Gradient Boosting(wie XGBoost), k-means und DBSCAN. Sie basiert als SciKit (Kurzform für SciPy Toolkit), wie beispielsweise auch Scikit-image, auf den numerischen und wissenschaftlichen Python-Bibliotheken NumPy und SciPy.

Random Forest in skitlearn

* Enthält zwei Mittellungsalgorithmen, die auf randomisierten Entscheidungsbäumen basieren
* Durch die Einführung von Zufälligkeit in die Klassifikatorenkonstruktion wird eine vielfältige Gruppe von Klassifikatoren erstellt.
* Die Vorhersage des Ensemble ergibt aich aus der gemittelten Vorhersage der einzelnen Klassifikatoren
* Forest-Klassifikator muss mit zwei Arrays ausgestattet werden.
  + Array X der Form (n\_samples, n\_features), dass die Trainingsdaten enthält und einem weiteren Array Y der Form (n\_samples), das die Zielwerte (Klassenlabel) für die Trainingsstichprobe enthält

Bei Random Forests (siehe RandomForestClassifier und RandomForestRegressor) wird jeder Baum des Ensembles aus einer mit Zurücklegen gezogenen Stichprobe (d. h. einer Bootstrap-Stichprobe) aus der Trainingsmenge erstellt.

Darüber hinaus wird bei der Aufteilung jedes Knotens während der Erstellung eines Baums die beste Aufteilung entweder aus allen eingegebenen Merkmalen oder einer zufälligen Teilmenge der Größe max\_features gefunden. (Weitere Einzelheiten finden Sie in den Richtlinien zur Parametereinstellung).

Der Zweck dieser beiden Zufallsquellen besteht darin, die Varianz des Forest-Schätzers zu verringern. Einzelne Entscheidungsbäume weisen nämlich in der Regel eine hohe Varianz auf und neigen zu einer Überanpassung. Die in die Forests eingespeiste Zufälligkeit führt zu Entscheidungsbäumen mit leicht entkoppelten Vorhersagefehlern. Durch die Bildung eines Durchschnitts aus diesen Vorhersagen können sich einige Fehler ausgleichen. Zufallswälder erzielen eine geringere Varianz, indem sie verschiedene Bäume kombinieren, manchmal um den Preis einer geringfügigen Zunahme der Verzerrung. In der Praxis ist die Verringerung der Varianz oft signifikant und führt zu einem insgesamt besseren Modell.

Die wichtigsten Parameter, die bei der Verwendung dieser Methoden angepasst werden müssen, sind n\_estimators und max\_features. Ersterer ist die Anzahl der Bäume im Wald. Je größer, desto besser, aber auch desto länger dauert die Berechnung. Darüber hinaus ist zu beachten, dass die Ergebnisse ab einer kritischen Anzahl von Bäumen nicht mehr signifikant besser werden. Letztere ist die Größe der zufälligen Teilmengen von Merkmalen, die bei der Aufteilung eines Knotens zu berücksichtigen sind. Je kleiner sie ist, desto größer ist die Verringerung der Varianz, aber desto größer ist auch der Anstieg der Verzerrung. Empirisch gute Standardwerte sind max\_features=1.0 oder äquivalent max\_features=None (immer alle Merkmale statt einer zufälligen Teilmenge berücksichtigen) für Regressionsprobleme und max\_features="sqrt" (eine zufällige Teilmenge der Größe sqrt(n\_features) verwenden) für Klassifikationsaufgaben (wobei n\_features die Anzahl der Merkmale in den Daten ist). Der Standardwert von max\_features=1.0 entspricht den Bagged Trees; mehr Zufälligkeit kann durch die Einstellung kleinerer Werte erreicht werden (z. B. 0.3 ist ein typischer Standardwert in der Literatur). Gute Ergebnisse werden oft erzielt, wenn max\_depth=None in Kombination mit min\_samples\_split=2 gesetzt wird (d. h. wenn die Bäume vollständig entwickelt werden). Beachten Sie jedoch, dass diese Werte in der Regel nicht optimal sind und zu Modellen führen können, die viel Arbeitsspeicher verbrauchen. Die besten Parameterwerte sollten immer durch eine Kreuzvalidierung ermittelt werden. Beachten Sie außerdem, dass bei Random Forests standardmäßig Bootstrap-Stichproben verwendet werden (bootstrap=True), während die Standardstrategie für Extra-Bäume darin besteht, den gesamten Datensatz zu verwenden (bootstrap=False). Bei der Verwendung von Bootstrap-Stichproben kann der Verallgemeinerungsfehler auf den ausgelassenen oder Out-of-Bag-Stichproben geschätzt werden. Dies kann durch die Einstellung oob\_score=True aktiviert werden.

Übersetzt mit www.DeepL.com/Translator (kostenlose Version)