Klassifikation mit Decision Tree und Random Forest

Welches Pre-Tuning ist für deinen Entscheidungsbaum/-wald sinnvoll? Womit würdest du starten?

- **Maximale Tiefe des Baums (max_depth)**: Begrenzen der Tiefe verhindert Overfitting. Startwerte: None (keine Begrenzung) oder Werte wie 3-10.
- **Mindestanzahl von Datenpunkten in einem Blatt (min_samples_leaf)**: Startwert: 1 oder 2. Höhere Werte fördern Generalisierung.
- **Anzahl der Features für Splits (max_features)**: Testen von sqrt, log2 oder einer festen Anzahl.
- **Kriterium (criterion)**: gini oder entropy (Entscheidungsbaum-Algorithmus).
- Für Random Forests: **Anzahl der Bäume (n_estimators)**, Startwert: 100.

Wie gut funktioniert das Modell? Welchen Einfluss haben unterschiedliche Werte beim Pre-Tuning?

- Die Modellleistung wird üblicherweise durch Metriken wie **Genauigkeit**, **Precision**, **Recall** oder **F1-Score** bewertet, abhängig von der Problemstellung.
- Mit den aktuellen Daten scheint es ein Klassifikationsproblem zu sein, basierend auf den Merkmalen length, weight, und w_l_ratio.
- Ein einfacher Train-Test-Split kann initiale Performance-Ergebnisse liefern. Ein Baseline-Test hilft einzuschätzen, wie weitreichend das Pre-Tuning das Modell verbessern kann.

Hast du noch andere Parameter zum Optimieren von Decision Trees und Random Forest gefunden, die für ein gutes Ergebnis hilfreich sind?

- **Kleine maximale Tiefe (max_depth)**: Kann zu einem unteranpassenden Modell führen (Underfitting), insbesondere bei komplexeren Daten.
- **Große maximale Tiefe**: Kann zu Overfitting führen, wenn das Modell zu spezifisch auf das Training reagiert.
- **Hohe Werte bei min_samples_leaf**: Fördern Generalisierung, verringern aber die Auflösung des Modells.
- **Random Forest n_estimators**: Zu wenige Bäume liefern instabile Ergebnisse, während zu viele Bäume Rechenzeit beanspruchen.

Was sind die Unterschiede zwischen den Modellen, die du trainiert hast?

- **Entscheidungsbaum**: Einzelner Baum, anfällig für Overfitting.
- **Random Forest**: Ensemble-Ansatz, der mehrere Bäume kombiniert. Stärker gegen Overfitting, aber rechenintensiver.

Welche Features sind für dein Modell relevant (Feature Importance)? Was passiert, wenn du unwichtige Features weglässt? Hast du dieses Ergebnis erwartet?

- Random Forests bieten eine intuitive Methode, die **Feature-Wichtigkeit** zu bewerten. Dies kann durch die Reduktion von Gini-Impurity oder durch Permutation-Methoden erfolgen.
- **Was passiert, wenn unwichtige Features weggelassen werden?**
    - In der Regel steigt die Modellperformance, da redundante oder irrelevante Daten vermieden werden.
    - Die Ergebnisse hängen davon ab, ob ein Feature tatsächlich wenig Information enthält oder ob es mit anderen stark korreliert ist.