Summary

For all eight of your tree-based classifiers—XGBoost, LightGBM, CatBoost, HistGradientBoosting, ExtraTrees, BalancedRandomForest, RandomForest, and DecisionTree—**feature scaling is generally unnecessary**, because these models split on feature thresholds rather than compute distances, making them invariant to any monotonic transformation of the inputs (Why Does Tree and Ensemble based Algorithm don't need feature ..., Do Decision Trees Need Feature Scaling Or Normalization?). Only in rare scenarios—such as when you mix tree models with scaling-sensitive learners in a stacked pipeline or when extremely large feature magnitudes introduce numerical instability in gradient-based regularization—might you consider scaling (What are the implications of scaling the features to xgboost?, Does XGBoost Need Feature Scaling Or Normalization? - Forecastegy).

Why Tree-Based Models Are Scale-Invariant

- **Threshold Splits, Not Distances**
  Decision-tree algorithms recursively partition data by asking "is feature F ≤ θ?"; they never compute Euclidean or other distance metrics (Why Does Tree and Ensemble based Algorithm don't need feature ..., Do Decision Trees Need Feature Scaling Or Normalization?).
- **Monotonic Transformations Preserve Order**
  Any strictly increasing transformation (e.g. min–max, z-score) leaves the rank order of observations unchanged, so the chosen split points simply map to new thresholds with identical predictive power (Feature Scaling in Machine Learning: Which Popular Algorithms ..., Scaling/Normalization not need for tree based models).
- **Ensembles Inherit Invariance**
  RandomForest, ExtraTrees, and gradient-boosted trees (XGBoost, LightGBM, CatBoost, HistGradientBoosting) all build on single-tree logic, so they inherit this same scale-invariance property (Feature Scaling in Machine Learning: Which Popular Algorithms ..., What are the implications of scaling the features to xgboost?).

When Scaling *Can* Matter

1. **Regularization in Gradient Boosters**
   XGBoost and LightGBM apply L1/L2 penalties to leaf weights. If one feature's gradient

statistics vastly outsize others, it may be slightly over-penalized—though in practice this is rare (What are the implications of scaling the features to xgboost?, Does XGBoost Need Feature Scaling Or Normalization? - Forecastegy).

2. **Numerical Stability**
   Extremely large or tiny feature values can, in edge cases, lead to floating-point issues during split-gain calculations (Scaling to Success: Implementing and Optimizing Penalized Models).
3. **Mixed Pipelines**
   When you stack tree models with scaling-sensitive learners (e.g. SVM, KNN), you might scale first for uniformity—and accept that scaling is a no-op for the trees (Does XGBoost Need Feature Scaling Or Normalization? - Forecastegy, Feature Scaling is NOT Always Necessary - by Avi Chawla).

Model-wise Recommendations

Copy and paste the table below directly into Obsidian. The "Scaling Recommended?" column reflects the default best practice, and the "Why / When to Consider" gives any exceptions.

Model	Scaling Recommended?	Why / When to Consider
XGBoost	No	Splits only on thresholds; monotonic transforms don't change cut points. Rarely consider scaling if using linear boosters or encountering numeric instability (What are the implications of scaling the features to xgboost?, decision trees - Is it necessary to normalize data for XGBoost?).
LightGBM	No	Leaf-wise tree construction unaffected by scale. Scale only if you observe odd regularization behavior (Feature Scaling in Machine Learning: Which Popular Algorithms ..., Does XGBoost Need Feature Scaling Or Normalization? - Forecastegy).
CatBoost	No	Native handling of numerical features; scale only when mixing with non-tree models (Feature Scaling in Machine Learning: Which Popular Algorithms ..., Why Does Tree and Ensemble based Algorithm don't need feature ...).

Model	Scaling Recommended?	Why / When to Consider
HistGradientBoosting	No	Operates on binned histograms—preserves order, not magnitude (Do Decision Trees Need Feature Scaling Or Normalization?, Scaling/Normalization not need for tree based models).
ExtraTrees	No	Random splits on sorted values; invariant under any monotonic transform (Why Does Tree and Ensemble based Algorithm don't need feature ..., Feature Scaling in Machine Learning: Which Popular Algorithms ...).
BalancedRandomForest	No	Built-in resampling ignores scale; handle imbalance but don't scale □.
RandomForest	No	Majority-vote ensemble of threshold splits; scale only alongside other learners (Feature Scaling in Machine Learning: Which Popular Algorithms ..., Scaling/Normalization not need for tree based models).
DecisionTree	No	Pure threshold logic; scaling never alters predictive splits (Do Decision Trees Need Feature Scaling Or Normalization?, Why Does Tree and Ensemble based Algorithm don't need feature ...).