Instructions

1. Load an imbalanced binary dataset (for example, the *Breast Cancer* dataset from `sklearn.datasets`, modified to increase the imbalance between classes).
2. Train a `DecisionTreeClassifier` using the `class_weight` parameter:
   - Start with the default model (no weighting),
   - Then retrain the model with `class_weight='balanced'`.
3. Evaluate the models using:
   - the **ROC curve**,
   - the **Precision–Recall curve**.
4. Determine the **optimal decision threshold** that maximizes the F1-score or recall, depending on the application context (e.g., medical diagnosis, fraud detection).
5. Compare model performance before and after adjusting the threshold, and discuss how this affects false positives and false negatives.
6. Visualize the decision tree structure to interpret how class weighting influences the learned decision boundaries.

Optional Extension

- Implement a custom **cost function** where false negatives are penalized ten times more than false positives.
- Use this function to manually compute a *cost-based evaluation metric* and compare it with the standard F1-score.

Project Instruction: apply these techniques to your data

Considering your data, here you have a structured guideline you can apply to your project:

1. Diagnose the imbalance by calculating the ratio: we can consider three levels of imbalanced data situations

Level of imbalance	Typical Ratio	Technique
weak	1:2-1:0	use class_weight=balanced
moderate	1:10-1:50	requires oversampling or SMOTE
severe	> 100	hybrid sampling

2. Understand your data type

Data Type	Recommanded technique
Tabular, numeric	RandomOversampling, SMOTE
Mixed (Numeric and Categorical)	SMOTE for Mixed Types (as SMOTENC), RandomOversampling
Time series	class weighting, sliding window
Images	image Augmention Techniques

3. Consider the model(s) you use

Model type	Suggested approach
Tree-based models as RandomForest, XGBoost	class_weight or Balanced Random Forest
Linear models (SVM, LR, LogR)	Class weighting or SMOTE
Ensemble Models	Hybrid resampling or Class weighting

4. Validate the impact of the resampling technique by evaluation with different metrics robust to imbalance problem:
    - Precision, Recall, F1-score
    - ROC-AUC, PR-AUC
    - confusion Matrix
    - Cost Sensitive evaluation
    - Use stratifiedKFold to preserve class ratio in cross-validation

Recommandation for your project progress

Before testing your Machine Learning models, it is worthfully to :

1. Understand your data by doing the Exploratory Data Analysis (EDA)
2. Visualize your features using as examples histograms, boxplots, Correleation matrix,etc.
3. Analyse the imbalance issue in your dataset.