# Risk Analytics

Machine Learning and Optimization
for Data-Driven Decision Making

Fernando S. Oliveira

Draft version — April 29, 2026

## Chapter 7

## Interpretable Risk Segmentation with Classification Trees

# Chapter 7

# Interpretable Risk Segmentation with Classification Trees

## 7.1 Introduction

The first six chapters developed the foundations of risk analytics. Chapters 1–3 introduced the language of risk, uncertainty, perception, and preference. They showed that risk analysis is not only about measuring uncertain outcomes, but also about understanding how those outcomes are interpreted, communicated, and evaluated by decision makers. Chapters 4–6 then translated this foundation into an operational framework based on loss functions, expected loss, and decision criteria under uncertainty.

This chapter begins a new part of the book. The focus now shifts from evaluating known or constructed loss distributions to identifying risky cases from data. In many practical settings, the analyst does not begin with a fully specified loss variable. A bank does not initially know which customers will default. A procurement team does not know which suppliers will fail. A fraud analyst does not know which transactions are suspicious. An operations manager does not know which assets, projects, or processes will create future disruption. Before losses can be measured or optimized, risky cases often have to be detected, ranked, and explained.

This is the point at which supervised learning enters the risk-analytics pipeline. Classification models use historical examples to learn patterns associated with adverse outcomes. They transform observable characteristics into predictions, scores, or risk categories. In doing so, they create a bridge between raw data and managerial action. Organizations do not manage risk only through abstract probability distributions. They classify customers, prioritize alerts, monitor suppliers, investigate transactions, escalate cases, and allocate scarce attention. Classification is therefore not merely a statistical exercise. It is one of the main ways in which risk becomes operational.

 

Once classification outputs are linked to actions such as approval, rejection, monitoring, review, audit, or intervention, predictive errors acquire economic and managerial meaning [11, 4]. A false negative may leave the organization exposed to a default, fraud, failure, or disruption that should have been detected. A false positive may consume scarce resources, create unnecessary friction, or damage relationships with customers, suppliers, or partners. For this reason, a risk classifier cannot be judged only by whether it is statistically accurate. It must also be assessed in terms of the types of errors it makes, the thresholds used to trigger action, and the degree to which its recommendations can be understood and governed.

Classification trees provide a natural starting point for this predictive stage of the book. They are simple supervised-learning models, but they are especially valuable in risk analytics because they combine prediction with interpretability [2, 15, 13]. Through recursive partitioning, they transform observed characteristics into transparent if–then rules. A tree can show, for example, how recent repayment behaviour, exposure, and payment history separate lower-risk from higher-risk credit-card accounts. This makes the model useful not only for prediction, but also for risk communication, governance, and managerial learning.

Interpretability is not a cosmetic feature in high-stakes risk settings. Chapter 2 emphasized that risk is perceived, framed, communicated, and acted upon by human decision makers. A model that cannot be understood may be difficult to trust, challenge, audit, or use. Chapter 3 emphasized that probabilities alone do not determine decisions; they must be connected to objectives, preferences, and decision criteria. Classification trees contribute to this process by producing event probabilities and risk segments that can later be connected to explicit decision rules, resource constraints, and intervention policies. In this sense, they sit at the boundary between prediction and decision support.

This chapter introduces classification trees as tools for interpretable risk segmentation and decision-sensitive evaluation. It has four main objectives. First, it explains how classification trees generate risk segments from observable data. Second, it shows how alternative splitting criteria—Gini impurity, entropy, and log-loss—shape the segmentation produced by a tree. Third, it explains how classifiers should be evaluated using confusion matrices, threshold-dependent metrics, ROC curves, AUC, precision–recall analysis, log-loss, and Cohen's Kappa. Fourth, it uses the Taiwan credit-card default dataset to show how these metrics lead to different managerial interpretations of model quality.

The central message is that a classification tree should be understood as

 

more than a prediction device. In risk analytics, it is a risk-identification system: it separates cases, ranks them, explains them, and prepares them for action. Economic-loss modelling can then be built on top of those classifications once the organization specifies the relevant decisions, intervention costs, capacity constraints, and risk appetite.

## 7.2 Classification Trees and the Logic of Risk Identification

The earlier chapters of this book often began with a loss variable. Once losses were defined, it was possible to evaluate decisions directly using loss-based criteria. In many real applications, however, the analyst does not begin with a loss distribution. Instead, the first task is to identify which cases are likely to become problematic. A bank may wish to identify customers likely to default, a procurement team may want to identify suppliers likely to fail, and an operations manager may want to identify projects likely to overrun or systems likely to break down. In such settings, risk analysis begins not with a loss distribution but with a classification problem.

Let $Y \in \{0, 1\}$ denote the outcome of interest, where $Y = 1$ represents the adverse event and $Y = 0$ represents its absence. Let $X = (X_1, \ldots, X_p)$ denote the observable characteristics of the account, transaction, supplier, project, or asset. The predictive task is to use historical examples of $(X, Y)$ to learn a rule that assigns new observations either to a predicted class or to an estimated event probability. This is the basic supervised-learning setting [12, 9, 10].

This remains a risk problem, even though the analysis starts from a class label rather than a loss variable. The reason is that classification is not an end in itself. It supports decisions. Once a case is labelled as high risk, low risk, or somewhere in between, that label may trigger approval, rejection, targeting, escalation, monitoring, or further review. Those actions have consequences, and those consequences can later be translated into losses. In this sense, classification is best understood as a problem of *risk identification*: the analyst seeks to sort observations into categories that are meaningful for action.

A classification tree predicts a categorical outcome by recursively partitioning the data into increasingly homogeneous groups [2, 15, 13]. The model starts with all observations in one node and searches for a question that separates them into two child nodes with more distinct outcome profiles. The process is repeated until the tree reaches a stopping rule, such as a maximum

 

depth or a minimum leaf size. The final nodes are called leaves.

For each leaf $m$, the tree reports an estimated event probability

$$\hat{p}_m = \frac{\text{number of observations with } Y = 1 \text{ in leaf } m}{\text{number of observations in leaf } m}.$$

Thus, $\hat{p}_m$ is the observed adverse-event rate in that leaf. A new observation that falls into leaf $m$ is assigned this probability as its risk score. For example, if a leaf contains 200 historical accounts and 50 of them defaulted, then $\hat{p}_m = 0.25$. A new account assigned to that leaf receives an estimated default probability of 25%.

The questions used to split the data are typically simple threshold rules, such as whether a repayment-status variable exceeds a given value or whether a credit-limit variable lies above or below a selected cut-off. The exact algebraic form of the split is less important than the managerial interpretation: each split turns a measurable characteristic into a segmentation rule. This is why trees are useful in risk governance. They expose the logic by which the population is divided into lower-risk and higher-risk groups.

The language of classification trees becomes especially natural in risk analytics when the leaves are interpreted as *risk segments*. Rather than viewing the tree only as a prediction device, one can interpret it as a structured segmentation of the population into groups with different levels of event likelihood. In managerial settings, these segments often translate directly into action categories such as high-priority, medium-priority, and low-priority cases [7, 19].

### 7.2.1 Gini impurity, entropy, and log-loss as splitting criteria

Tree algorithms need a criterion for deciding which split is best. In a binary node with event rate $\hat{p}$, the node is pure when $\hat{p}$ is close to 0 or 1, and impure when the node contains a mixed set of zeros and ones. The splitting criterion measures this impurity or uncertainty and chooses splits that reduce it.

One commonly used measure is the Gini impurity,

$$G(\hat{p}) = 1 - \hat{p}^2 - (1 - \hat{p})^2 = 2\hat{p}(1 - \hat{p}).$$

Gini impurity is zero when a node contains only one class and is largest when the node is evenly mixed. A Gini-based tree therefore searches for splits that create class-pure child nodes [2, 9]. In risk-management terms, it tries to create segments in which the event rate is clearly high or clearly low.

Another commonly used measure is entropy,

$$H(\hat{p}) = -\hat{p}\log(\hat{p}) - (1 - \hat{p})\log(1 - \hat{p}).$$

 

Entropy has an information-gain interpretation and was central in early decision-tree algorithms such as ID3 and C4.5 [15, 16]. Entropy is high when the class label is uncertain and low when the class label is predictable. An entropy-based tree selects splits that reduce uncertainty about the adverse event. In a risk setting, this can be interpreted as selecting variables that provide the largest information gain about which cases are likely to become problematic.

A third criterion is log-loss, also called cross-entropy loss in binary classification:

$$\ell(y, \hat{p}) = -\left[y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\right].$$

Log-loss evaluates probability quality rather than only class purity. It penalizes confident wrong predictions heavily [8, 21]. This is especially relevant in risk analytics because decisions often depend not only on whether a case is classified as risky, but also on how high the estimated probability of the adverse event is.

The distinction between these criteria is important. Gini and entropy focus on class separation and node purity. Log-loss focuses more directly on probabilistic accuracy. A tree that is good at separating classes may not always produce the best probability estimates, and a tree with slightly weaker class separation may still provide better calibrated scores for decision support.

At the same time, trees must be handled carefully. A tree can always be made more complex by adding more splits. Very deep trees often fit peculiarities of the training sample rather than stable structure in the underlying problem. In that case, the model is overfitted: it looks impressive in-sample but performs poorly on new cases. To control this risk, analysts impose stopping rules such as maximum depth or minimum leaf size, or they use validation data or cross-validation to prune and tune the tree [2, 9, 10, 13].

## 7.3 Evaluating Classification Quality for Risk Analysis

Before a classification model can be used as a risk tool, its quality must be assessed in a clear and disciplined way. A classifier may appear useful while still making the wrong kinds of mistakes for the decision problem at hand. In a risk context, the analyst is interested not only in whether the model predicts correctly on average, but also in which mistakes it makes, how often they occur, and whether its predicted probabilities are reliable enough to support action. Classification quality should therefore be assessed through

 

confusion matrices, threshold-dependent metrics, threshold-independent ranking measures, probability-quality measures, and chance-adjusted agreement measures [14, 18, 6, 8].

### 7.3.1 The confusion matrix and the structure of classification error

Consider a binary classification problem in which $Y \in \{0, 1\}$ denotes the realized outcome and $\hat{Y} \in \{0, 1\}$ denotes the predicted class. The confusion matrix records the four possible predictive outcomes:

| | $\hat{Y} = 1$ | $\hat{Y} = 0$ |
|---|---|---|
| $Y = 1$ | $TP$ | $FN$ |
| $Y = 0$ | $FP$ | $TN$ |

where $TP$ denotes true positives, $TN$ true negatives, $FP$ false positives, and $FN$ false negatives.

The confusion matrix is fundamental because it reveals the structure of model errors. Accuracy alone collapses all outcomes into one number, but the confusion matrix shows whether the model mainly misses important positive cases, generates too many false alarms, or achieves a more balanced pattern of prediction. This matters in risk analytics because the costs and operational consequences of false positives and false negatives are rarely the same [18, 14].

### 7.3.2 Threshold-dependent predictive metrics and their risk meaning

Several standard metrics are derived from the confusion matrix [18, 14, 21]. Accuracy measures the share of all observations classified correctly:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Accuracy is simple and intuitive, but it can be misleading when the adverse event is rare. In default prediction, fraud detection, supplier failure, and safety screening, a classifier can obtain high accuracy by mostly predicting the majority class. Such a model may look good statistically while failing at the central risk-management task: identifying the cases that require attention.

Precision measures the reliability of positive predictions:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

 

It answers the question: among the cases flagged by the model, how many are truly positive? Precision matters when false positives are costly. A low-precision default model may overwhelm credit officers with weak alerts. A low-precision fraud model may create unnecessary investigations and customer friction. A low-precision supplier-risk model may cause procurement teams to waste time monitoring suppliers that are unlikely to fail.

Recall, also called sensitivity or the true positive rate, measures the model's ability to detect actual positives:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

It answers the question: among the truly positive cases, how many did the model identify? Recall is especially important when false negatives are costly. In credit risk, a false negative may mean approving or under-monitoring an account that later defaults. In operational risk, it may mean failing to detect a process, asset, or project that will later fail. In safety-critical systems, false negatives can be much more serious than false positives.

Specificity measures the proportion of negative cases correctly classified:

$$\text{Specificity} = \frac{TN}{TN + FP}.$$

Specificity matters when the organization wants to avoid unnecessary intervention in safe cases. In credit management, high specificity means that safe customers are less likely to be disturbed by unnecessary restrictions or reviews. In compliance monitoring, it means that limited investigative resources are not diluted across too many low-risk cases.

The false positive rate and false negative rate make the error structure explicit:

$$\text{FPR} = \frac{FP}{FP + TN} = 1 - \text{Specificity}, \qquad \text{FNR} = \frac{FN}{FN + TP} = 1 - \text{Recall}.$$

These two rates are often more useful for risk communication than accuracy. The false negative rate expresses residual exposure: the share of adverse cases that remain undetected. The false positive rate expresses intervention burden: the share of safe cases that are unnecessarily flagged.

The $F_1$-score combines precision and recall into a single harmonic-mean summary:

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}.$$

 

The harmonic mean penalizes a model that performs well on only one of precision or recall. This is useful when the analyst wants a compact summary of alert quality and detection ability. However, $F_1$ remains a predictive metric rather than an economic criterion. It gives precision and recall equal structural importance, while many risk problems do not value false positives and false negatives equally.

When class imbalance is important, a useful adjustment is balanced accuracy:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) = \frac{\text{Recall} + \text{Specificity}}{2}.$$

Balanced accuracy gives equal weight to the positive and negative classes and is often more informative than raw accuracy in rare-event settings. In risk management, it is useful when the analyst wants a simple measure that does not let the majority class dominate the evaluation.

These measures should be read as a diagnostic portfolio rather than as substitutes for one another. Accuracy describes overall correctness. Precision describes the quality of alerts. Recall describes the ability to detect the cases that create exposure. Specificity describes the ability to leave safe cases undisturbed. $F_1$ summarizes the precision–recall balance. Balanced accuracy protects against class-imbalance distortions. The relevant metric depends on the decision context, available resources, and relative costs of false positives and false negatives.

### 7.3.3 Cohen's Kappa and value beyond the base rate

Cohen's Kappa is a chance-adjusted agreement measure [3, 20]. It is useful when accuracy may be inflated by class imbalance. Let

$$p_o = \frac{TP+TN}{TP+TN+FP+FN}$$

denote observed agreement between the classifier and the realized labels. This is the same quantity as accuracy. Let $p_e$ denote the agreement expected by chance if the predicted and realized labels were independent but had the same marginal class frequencies. In binary classification,

$$p_e = \left(\frac{TP+FP}{N} \cdot \frac{TP+FN}{N}\right) + \left(\frac{FN+TN}{N} \cdot \frac{FP+TN}{N}\right),$$

where $N = TP+TN+FP+FN$. The first term is the chance agreement on the positive class; the second term is the chance agreement on the negative

class. Cohen's Kappa is then

$$\kappa = \frac{p_o - p_e}{1 - p_e}.$$

Kappa asks whether the classifier adds classification value beyond what would be expected from the base rates alone. In credit-risk terms, a model may obtain reasonable accuracy because most customers do not default. Kappa penalizes this kind of superficial success. A low Kappa warns that the model may be reproducing the majority-class structure rather than identifying risk. A higher Kappa suggests that the model's classifications agree with observed outcomes more than chance would imply.

Kappa should not be used as a splitting criterion for growing ordinary classification trees. Unlike Gini impurity, entropy, or log-loss, it is not a local node impurity measure. It depends on the global confusion matrix and on a chosen threshold. For this reason, it is better used for validation, model comparison, threshold selection, or pruning diagnostics, not for selecting each split inside the tree. Its value in this chapter is therefore diagnostic rather than constructive: it helps assess whether a fitted tree adds useful classification structure beyond the base rate.

### 7.3.4 Thresholds, ROC curves, AUC, and risk-management trade-offs

Many classifiers do not directly output a class label. Instead, they output a score or estimated event probability $\hat{p}(x)$. The final classification is produced by comparing this score to a threshold $\tau$:

$$\hat{Y}(\tau) = \mathbf{1}\{\hat{p}(x) \geq \tau\}.$$

Here $\hat{p}(x)$ is the estimated probability of the adverse event for an observation with characteristics $x$, and $\tau$ is the operating threshold. Lowering $\tau$ makes the classifier more aggressive: more cases are classified as positive, recall tends to rise, and false positives usually rise as well. Raising $\tau$ makes the classifier more conservative: false positives tend to fall, but more truly positive cases may be missed.

A convenient way to summarize this trade-off across all possible thresholds is the receiver operating characteristic, or ROC, curve [6, 1]. The ROC curve plots the true positive rate,

$$\mathrm{TPR}(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)},$$

against the false positive rate,

$$\text{FPR}(\tau) = \frac{FP(\tau)}{FP(\tau) + TN(\tau)},$$

as the threshold $\tau$ varies. The area under this curve, AUC, summarizes how well the model ranks positive cases above negative cases. A useful interpretation is that AUC equals the probability that a randomly chosen positive case receives a higher score than a randomly chosen negative case [6].

For risk management, AUC answers a ranking question rather than an action question. A model with high AUC is good at ordering cases from safer to riskier. This is valuable when organizations prioritize accounts, inspections, suppliers, or transactions under limited attention. However, AUC does not specify which threshold should be used, how many cases should be reviewed, or what the cost of each error is. A model can have strong AUC and still be poorly suited to a particular operating point.

In imbalanced problems, precision–recall analysis is often a useful complement to ROC analysis [17, 5]. ROC curves can appear strong when the negative class is large because the false positive rate divides false positives by all negatives. Precision–recall curves focus directly on the positive class and are therefore often more revealing when the adverse event is rare. In risk-management terms, the precision–recall curve shows how alert quality deteriorates as the organization tries to detect a larger share of adverse cases.

### 7.3.5 Probability-quality measures: log-loss

If the model outputs estimated probabilities $\hat{p}_i$, one common measure of probability quality is log-loss:

$$\text{LogLoss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{p}_i) + (1 - y_i)\log(1 - \hat{p}_i)\right].$$

Here $y_i$ is the realized outcome for observation $i$, and $\hat{p}_i$ is the model's estimated probability that $Y_i = 1$. Log-loss evaluates the quality of probability forecasts rather than only final class labels. It rewards models that assign high probabilities to events that occur and low probabilities to events that do not. It is especially unforgiving when a model is very confident and wrong. This makes log-loss particularly useful when predicted probabilities will later be used for thresholds, risk scores, pricing, provisioning, or expected-loss calculations [8, 21].

In risk analytics, log-loss is important because many decisions depend not only on rank ordering, but on probability quality. A model with better

calibrated and more informative probabilities provides a stronger foundation for downstream decisions. This is why a tree with slightly lower AUC may still be attractive if it produces better probability estimates.

## 7.4 From Classification Quality to Decision-Sensitive Evaluation

Traditional classification metrics describe predictive quality, but they do not fully express the decision consequences of prediction errors. Risk analysis begins when predictive outputs are connected to actions. A threshold, an alert rule, a manual-review policy, a monitoring rule, or a prioritization procedure transforms a classification model into a decision system.

The core distinction is between false positives and false negatives. A false positive occurs when the model flags a case as risky even though the adverse event does not occur. A false negative occurs when the model treats a case as safe even though the adverse event does occur. These two errors have different risk-management meanings. False positives create intervention cost, review burden, customer friction, supplier disruption, or unnecessary managerial attention. False negatives create residual risk exposure: the organization fails to act on cases that later generate loss, disruption, default, fraud, or failure.

Accuracy can be useful as a first diagnostic, but it is not a decision rule. A model can be accurate because it predicts the majority class well while still missing many adverse cases. This is particularly dangerous in risk management because the adverse class is often the minority class. Credit defaults, fraud cases, safety incidents, and supplier failures are usually less frequent than normal outcomes, but they are also the outcomes that motivate the risk system. A model that mostly predicts "no event" may look acceptable under accuracy and yet be weak as a risk-identification tool.

Recall and the false-negative rate are central when missed events are costly. In credit screening, a false negative may mean that an account that later defaults is not reviewed or monitored. In fraud detection, it may mean that a suspicious transaction is allowed to pass. In maintenance or safety applications, it may mean that a failure-prone asset is left untreated. High recall reduces this residual exposure, but it usually does so by flagging more cases and increasing false positives.

Precision and the false-positive rate become central when intervention is costly or capacity is limited. A low-precision alerting system may overwhelm analysts with weak alerts, create customer friction, or waste scarce monitoring resources. A high-precision system produces more reliable alerts, but may

 

miss more true adverse cases. The preferred balance is therefore not purely statistical. It depends on the organization's tolerance for missed risk, its available review capacity, and the cost of unnecessary intervention.

Specificity protects the non-event population from unnecessary action. This is important when false positives damage relationships, reduce customer value, consume management attention, or produce regulatory and reputational concerns. A highly aggressive risk screen may have excellent recall but poor specificity. Such a system may be acceptable in safety-critical settings, but not in contexts where interventions are expensive or intrusive.

For observations indexed by $i = 1, \ldots, n$, let $Y_i \in \{0, 1\}$ denote the realized outcome and $\hat{Y}_i \in \{0, 1\}$ denote the classification-based decision. A simple observation-level loss can be written as

$$L_i = C_{FP,i}\mathbf{1}\{\hat{Y}_i = 1, Y_i = 0\} + C_{FN,i}\mathbf{1}\{\hat{Y}_i = 0, Y_i = 1\},$$

where $C_{FP,i} > 0$ is the cost of a false positive and $C_{FN,i} > 0$ is the cost of a false negative.

This expression clarifies why a single predictive score cannot determine the preferred model. If $C_{FN,i}$ is large relative to $C_{FP,i}$, the decision maker will usually prefer a lower threshold and higher recall. If $C_{FP,i}$ is large relative to $C_{FN,i}$, the decision maker will usually prefer a higher threshold and higher precision or specificity. If review capacity is limited, the task may become ranking rather than binary classification: the model should order cases well so that the highest-risk cases are reviewed first. In that setting, AUC and precision–recall curves become more informative than accuracy.

Exposure also matters. A false negative on a large, active, or systemically important account is more serious than a false negative on a small account. A false positive on a valuable customer or critical supplier may be more costly than a false positive on a low-value relationship. This means that serious risk analytics often requires moving from constant error costs to heterogeneous, case-level costs. The present chapter introduces this logic conceptually, but its empirical focus remains on the earlier and more foundational question: how well does the tree identify, rank, and explain risk?

The practical implication is that model evaluation should be plural. Accuracy, precision, recall, specificity, AUC, log-loss, and Cohen's Kappa answer different questions. A defensible risk model is not necessarily the model that wins on one metric. It is the model whose pattern of errors, ranking quality, probability quality, and interpretability fit the decision context.

 

## 7.5 Numerical Illustration: Credit-Card Default Classification

### 7.5.1 Purpose of the case study

The empirical illustration uses the Taiwan credit-card default dataset [22]. The dataset contains 30,000 credit-card accounts and variables describing granted credit limits, repayment status, bill amounts, payment amounts, and customer characteristics. The target variable records whether the account defaults in the following month. The case is therefore a natural risk-classification problem: the analyst is not predicting a neutral label, but identifying accounts that may require closer monitoring.

The purpose of the case is to compare classification trees as interpretable risk-identification systems. The analysis asks how tree depth and splitting criterion affect predictive performance, probability quality, and managerial interpretation. The emphasis is not on finding the best possible credit-scoring model. More sophisticated methods will be introduced later. The emphasis here is on understanding what a classification tree is doing, how alternative criteria change the model, and how standard classification metrics should be interpreted in risk-management terms.

### 7.5.2 Data structure and descriptive evidence

The dataset is moderately imbalanced. Approximately 22% of accounts default in the following month, while roughly 78% do not. This imbalance is large enough to make raw accuracy potentially misleading. A conservative classifier can obtain reasonable accuracy by mostly predicting non-default, while still missing many accounts that matter for risk management.

The predictors also show substantial heterogeneity in exposure and account activity. Credit limits vary from small limits to very large limits, bill amounts are highly dispersed, and payment amounts contain many low-activity accounts as well as a smaller number of very active accounts. This heterogeneity matters because some accounts are both riskier and more economically consequential than others.

Figure 7.1 shows the class imbalance directly. The non-default class is much larger than the default class. This explains why accuracy must be interpreted cautiously: a model can look good under accuracy by favouring the majority class.

Figure 7.2 confirms that recent repayment status is strongly associated with default risk. Accounts with worse recent repayment status have much

Table 7.1: Selected descriptive statistics for the credit-card default dataset

| Variable | Mean | Std. Dev. | Min | Median | Max |
|---|---|---|---|---|---|
| `LIMIT_BAL` | 167,484 | 129,748 | 10,000 | 140,000 | 1,000,000 |
| `AGE` | 35 | 9 | 21 | 34 | 79 |
| `BILL` | 44,977 | 63,261 | -56,043 | 21,052 | 877,314 |
| `PAY` | 5,275 | 10,138 | 0 | 2,397 | 627,344 |
| `PAY_STATUS_MEAN` | 0 | 1 | -2 | 0 | 6 |

Notes: `BILL` and `PAY` denote six-month averages of bill amounts and payment amounts, respectively.

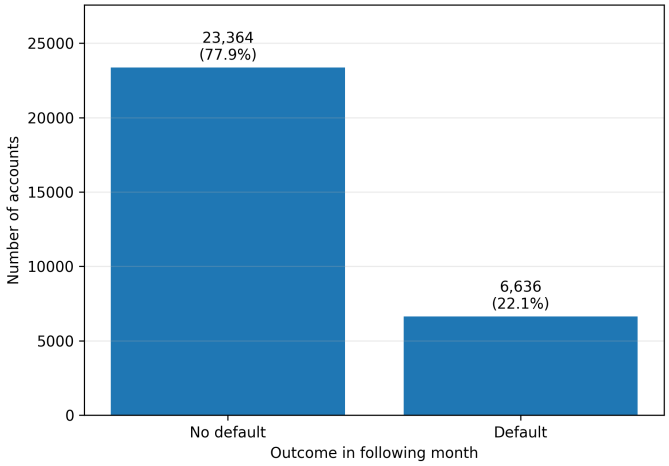


Figure 7.1: Class distribution in the Taiwan credit-card default dataset.

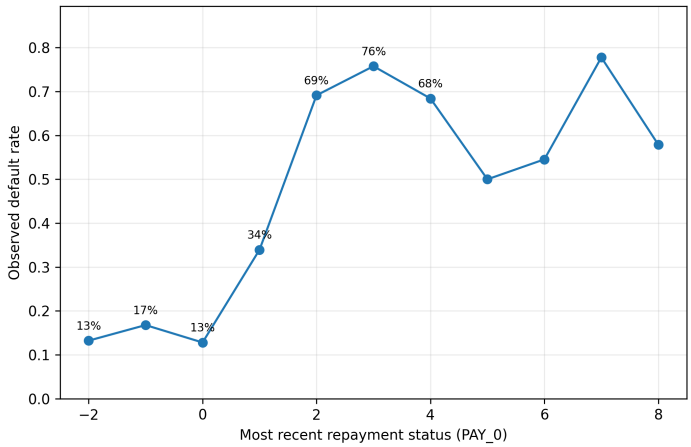


Figure 7.2: Observed default rate by most recent repayment status, `PAY_0`.

higher observed default rates. This explains why repayment variables tend to appear early in interpretable tree structures.

Figure 7.3 compares credit-limit distributions by outcome. The figure is not intended to show that credit limit alone determines default. Rather, it reminds the reader that exposure varies substantially across accounts, so the practical importance of classification errors can also vary across accounts.

### 7.5.3 Modelling design

The empirical design uses three separate data partitions: a training set, a validation set, and a test set. The training set is used only to estimate the candidate trees. For each combination of tree depth and splitting criterion, the tree structure and terminal-node probabilities are learned from the training observations. The validation set is then used to choose the operating threshold for each fitted tree. In the reported results, the threshold is selected to maximize the validation-set $F_1$-score, which provides a balanced compromise between precision and recall. Finally, the test set is used only once, after the tree and threshold have already been fixed, to report the final performance metrics in Table 7.2. All figures and tables reported in this Chapter are based on the test set, after the tree structure has been estimated on the training set and the operating threshold has been selected on the validation set.

This separation is important because tree construction, threshold selection, and final evaluation are distinct tasks. If the same observations were used to estimate the tree, choose the threshold, and report final performance, the

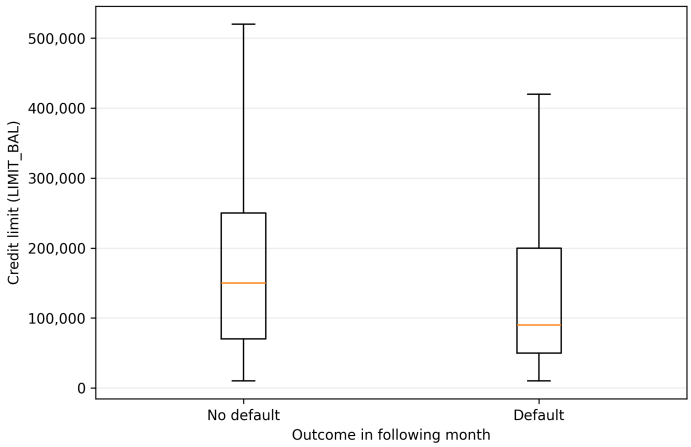


Figure 7.3: Credit-limit distribution by default outcome.

results would overstate the model's ability to generalize. The training set answers the question, "What tree should be learned?" The validation set answers the question, "Where should the operating threshold be placed?" The test set answers the question, "How well does the resulting classification policy perform on new data?"

The empirical comparison varies both tree depth and splitting criterion:

$$\text{Depth} \in \{2, 3, 4, 5, 6, 7, 8, 10\}, \qquad \text{Criterion} \in \{\text{Gini}, \text{Entropy}, \text{Log-loss}\}.$$

Depth 7 is included because it fills the gap between depth 6 and depth 8. Depth 10 is included as a diagnostic for whether additional complexity continues to improve performance or begins to produce weaker probability quality or unstable threshold behaviour. The comparison therefore contains shallow, moderate, and relatively deep trees.

All trees use a minimum leaf size of 50 observations. This stabilizes terminal-node default-rate estimates while still allowing the tree to identify local risk segments. Thresholds are selected on the validation set to maximize $F_1$. This choice gives a balanced operating point between precision and recall and avoids the arbitrary use of the conventional 0.5 threshold in an imbalanced risk problem.

 

### 7.5.4 The shallow tree as an interpretable risk-segmentation model

The shallow tree remains the teaching model because it is simple enough to display and interpret. It should not be presented as the best model by assumption. It should be presented as the most transparent model, against which more complex trees can be compared.

 

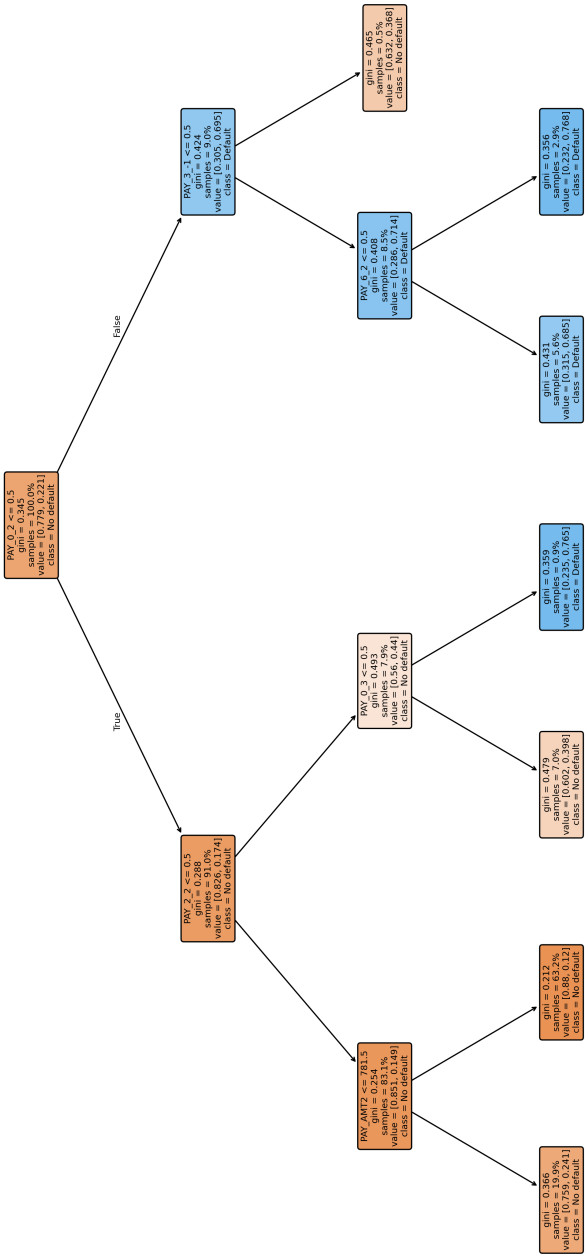


Figure 7.4: Shallow classification tree for the credit-card default case.

Figure 7.4 displays the depth-3 Gini tree without an internal title because the caption already identifies the object. The tree shows how recent repayment behaviour, billing information, and payment behaviour are translated into a set of interpretable risk segments. This is the main governance advantage of a shallow tree: even if a deeper tree performs better on some metrics, the shallow tree is easier to explain, audit, and communicate.

### 7.5.5 Performance comparison across depth and splitting criterion

Table 7.2 reports the classification performance of the candidate trees. The thresholds are selected on the validation set to maximize $F_1$, and the reported metrics are computed on the test set.

The table should therefore be read as an out-of-sample evaluation. The thresholds shown in the third column were not chosen on the test set; they were selected on the validation set and then applied unchanged to the test set.

 

Table 7.2: Classification performance by tree depth and splitting criterion

| Criterion | Depth | Threshold | Accuracy | Precision | Recall | Specificity | $F_1$ | AUC | Log-loss | $\kappa$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Gini | 2 | 0.150 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.674 | 0.463 | 0.373 |
| Gini | 3 | 0.250 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.720 | 0.452 | 0.373 |
| Gini | 4 | 0.220 | 0.783 | 0.509 | 0.507 | 0.861 | 0.508 | 0.731 | 0.449 | 0.368 |
| Gini | 5 | 0.160 | 0.744 | 0.443 | 0.612 | 0.781 | 0.514 | 0.746 | 0.445 | 0.346 |
| Gini | 6 | 0.270 | 0.787 | 0.518 | 0.542 | 0.857 | 0.530 | 0.755 | 0.444 | 0.392 |
| Gini | 7 | 0.240 | 0.777 | 0.496 | 0.560 | 0.839 | 0.526 | 0.754 | 0.446 | 0.381 |
| Gini | 8 | 0.290 | 0.779 | 0.501 | 0.561 | 0.841 | 0.529 | 0.759 | 0.446 | 0.385 |
| Gini | 10 | 0.300 | 0.771 | 0.485 | 0.579 | 0.825 | 0.528 | 0.756 | 0.503 | 0.378 |
| Entropy | 2 | 0.150 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.674 | 0.463 | 0.373 |
| Entropy | 3 | 0.250 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.720 | 0.452 | 0.373 |
| Entropy | 4 | 0.220 | 0.783 | 0.509 | 0.507 | 0.861 | 0.508 | 0.731 | 0.450 | 0.368 |
| Entropy | 5 | 0.310 | 0.806 | 0.577 | 0.455 | 0.905 | 0.509 | 0.748 | 0.446 | 0.390 |
| Entropy | 6 | 0.270 | 0.780 | 0.502 | 0.567 | 0.840 | 0.533 | 0.757 | 0.442 | 0.389 |
| Entropy | 7 | 0.310 | 0.806 | 0.575 | 0.476 | 0.900 | 0.521 | 0.760 | 0.459 | 0.401 |
| Entropy | 8 | 0.280 | 0.800 | 0.553 | 0.490 | 0.888 | 0.520 | 0.760 | 0.461 | 0.394 |
| Entropy | 10 | 0.280 | 0.768 | 0.481 | 0.591 | 0.819 | 0.530 | 0.754 | 0.501 | 0.379 |
| Log-loss | 2 | 0.150 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.674 | 0.463 | 0.373 |
| Log-loss | 3 | 0.250 | 0.804 | 0.575 | 0.429 | 0.910 | 0.491 | 0.720 | 0.452 | 0.373 |
| Log-loss | 4 | 0.220 | 0.783 | 0.509 | 0.507 | 0.861 | 0.508 | 0.731 | 0.450 | 0.368 |
| Log-loss | 5 | 0.310 | 0.806 | 0.577 | 0.455 | 0.905 | 0.509 | 0.748 | 0.446 | 0.390 |
| Log-loss | 6 | 0.270 | 0.780 | 0.502 | 0.567 | 0.840 | 0.533 | 0.757 | 0.442 | 0.389 |
| Log-loss | 7 | 0.310 | 0.806 | 0.575 | 0.476 | 0.900 | 0.521 | 0.760 | 0.459 | 0.401 |
| Log-loss | 8 | 0.280 | 0.800 | 0.553 | 0.490 | 0.888 | 0.520 | 0.760 | 0.461 | 0.394 |
| Log-loss | 10 | 0.280 | 0.768 | 0.481 | 0.591 | 0.819 | 0.530 | 0.754 | 0.501 | 0.379 |

Notes: thresholds are selected on the validation set to maximize $F_1$ and metrics are reported on the test set. Log-loss is better when lower; all other reported performance metrics are better when higher.

The table should not be read as a search for one universal winner. Gini depth 8 has the highest AUC among the Gini trees. Entropy depth 7 has the highest AUC among the entropy trees. The log-loss criterion produces its lowest test log-loss at depth 6 in this run. These results illustrate the chapter's main point: different metrics select different trees because they answer different risk-management questions.

The table also shows why depth 10 is worth trying as a diagnostic but not necessarily worth using. It can improve some threshold-dependent measures, but additional depth may also make the model harder to interpret and may not improve ranking or probability quality sufficiently to justify the loss of simplicity. Depth should therefore be treated as a modelling choice, not as a mechanical path toward improvement.

### 7.5.6 ROC and precision–recall analysis

The following figures report out-of-sample test-set performance. Threshold-free figures use the full test-set score distribution, whereas threshold-dependent figures use the validation-selected operating thresholds.

Figure 7.5 compares a small set of representative curves rather than all candidate trees. Showing every tree would make the graph unreadable. The selected curves are: the interpretable Gini depth-3 tree, the best Gini tree by AUC, the best entropy tree by AUC, and the log-loss tree with the lowest test log-loss. This selection deliberately contrasts interpretability, ranking quality, and probability-quality objectives.

The ROC curve answers a ranking question: how well does the tree assign higher scores to accounts that default than to accounts that do not? In credit-risk management, this is important when accounts are prioritized for review. The diagonal line represents random ranking. Curves farther above the diagonal indicate better separation between defaulting and non-defaulting accounts.

Figure 7.6 focuses on the default class. Precision is high at very low recall because the model can identify a small number of accounts that are especially likely to default. As recall increases, the model must include more marginal cases in order to capture more defaults. Those additional cases contain more non-defaulting accounts, so precision falls. At very high recall, the curves move toward the base default rate because almost all accounts must be flagged to capture almost all defaults. This is why the curves converge at large recall values: once the threshold becomes very low, the classifier behaves increasingly like a broad population screen rather than a selective alerting system.

 

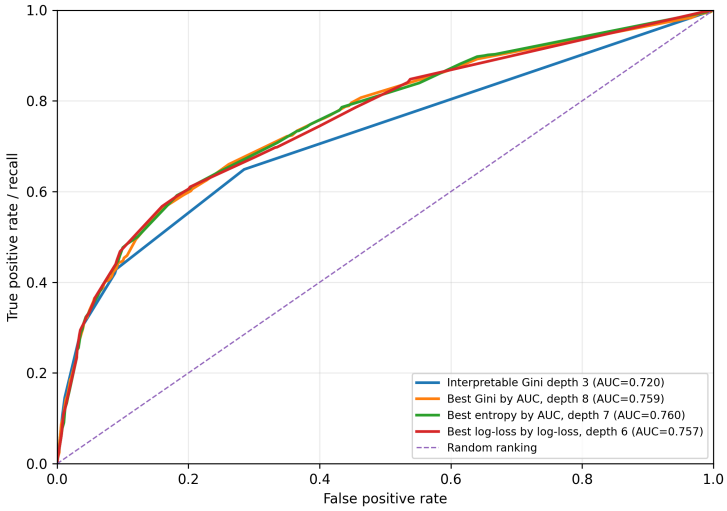


Figure 7.5: ROC curves for selected classification trees.

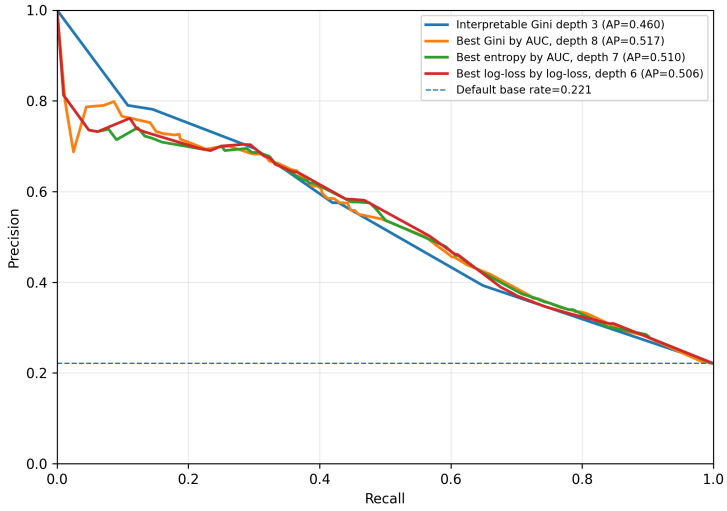


Figure 7.6: Precision–recall curves for selected classification trees.

 

### 7.5.7 Metric profiles and model choice

Figure 7.7 compares AUC, log-loss, $F_1$, and Cohen's Kappa across depth and splitting criterion.

 

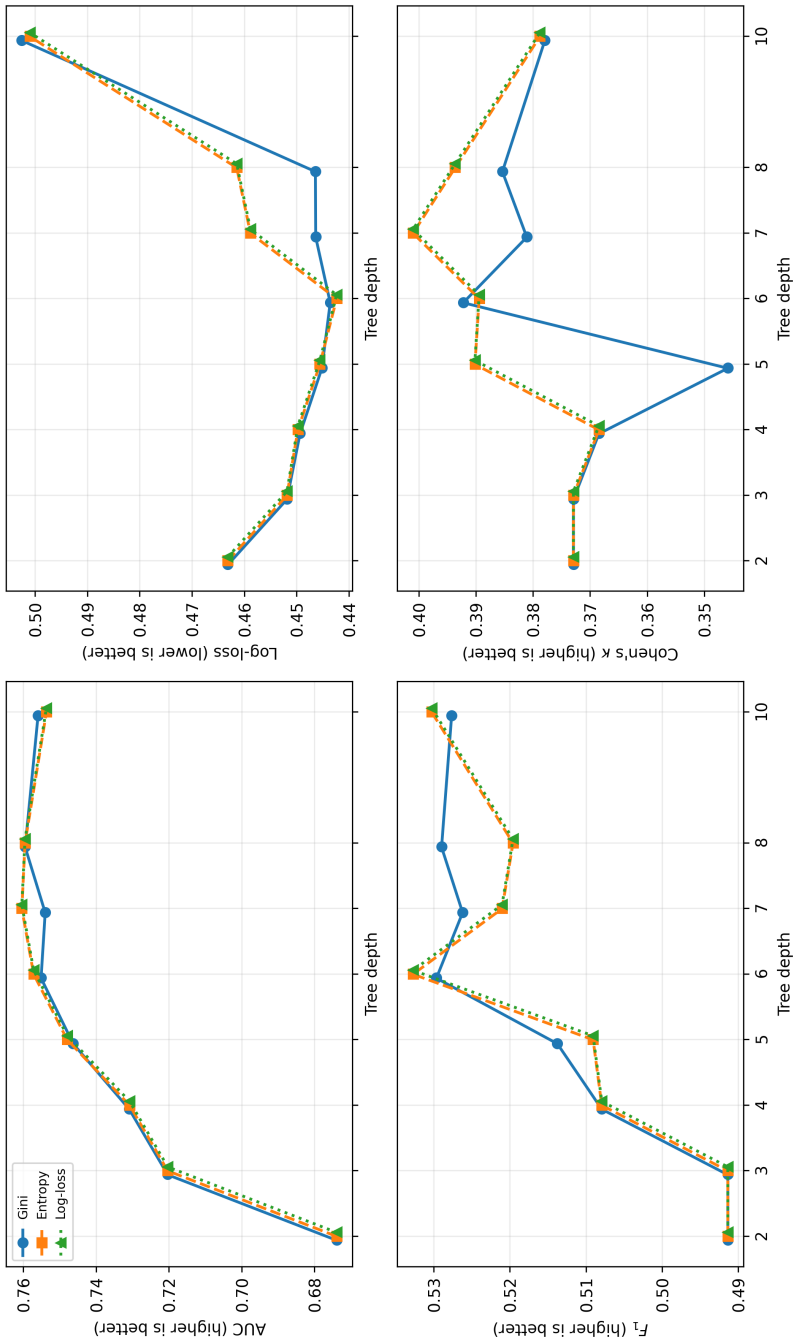


Figure 7.7: Metric profiles by tree depth and splitting criterion.

The log-loss panel uses the regular scale: lower values indicate better probability forecasts. This is intentionally not inverted, so the reader can interpret the metric directly. In this implementation, the entropy and log-loss criteria often produce very similar or identical results because both are based on cross-entropy/information-gain logic for binary classification. Where the curves overlap, this is not a plotting error; it means the criteria produced the same or nearly the same tree behaviour for those depths. The figure uses different markers and small horizontal offsets to make overlapping profiles easier to see.

The profiles show why model choice depends on the purpose of the risk system. If the objective is ranking accounts for review, AUC is central. If the objective is a binary alerting rule, $F_1$ and Kappa help summarize threshold-dependent performance. If predicted probabilities will be used downstream, log-loss is important. If interpretability is essential, the shallow tree may still be preferred even when deeper trees improve some metrics.

### 7.5.8 Confusion matrices at validation-selected thresholds

The thresholds used in Figure 7.8 are not fixed at 0.5. They are selected on the validation set to maximize $F_1$. This is why the reported thresholds lie around 0.23–0.33. In an imbalanced default problem, the base default rate is around 22%, and many useful risk scores lie below 0.5. A threshold of 0.5 would classify only the most extreme accounts as risky, producing high specificity but low recall. The lower validation-selected thresholds create a more balanced trade-off between identifying defaulting accounts and avoiding unnecessary alerts.

 

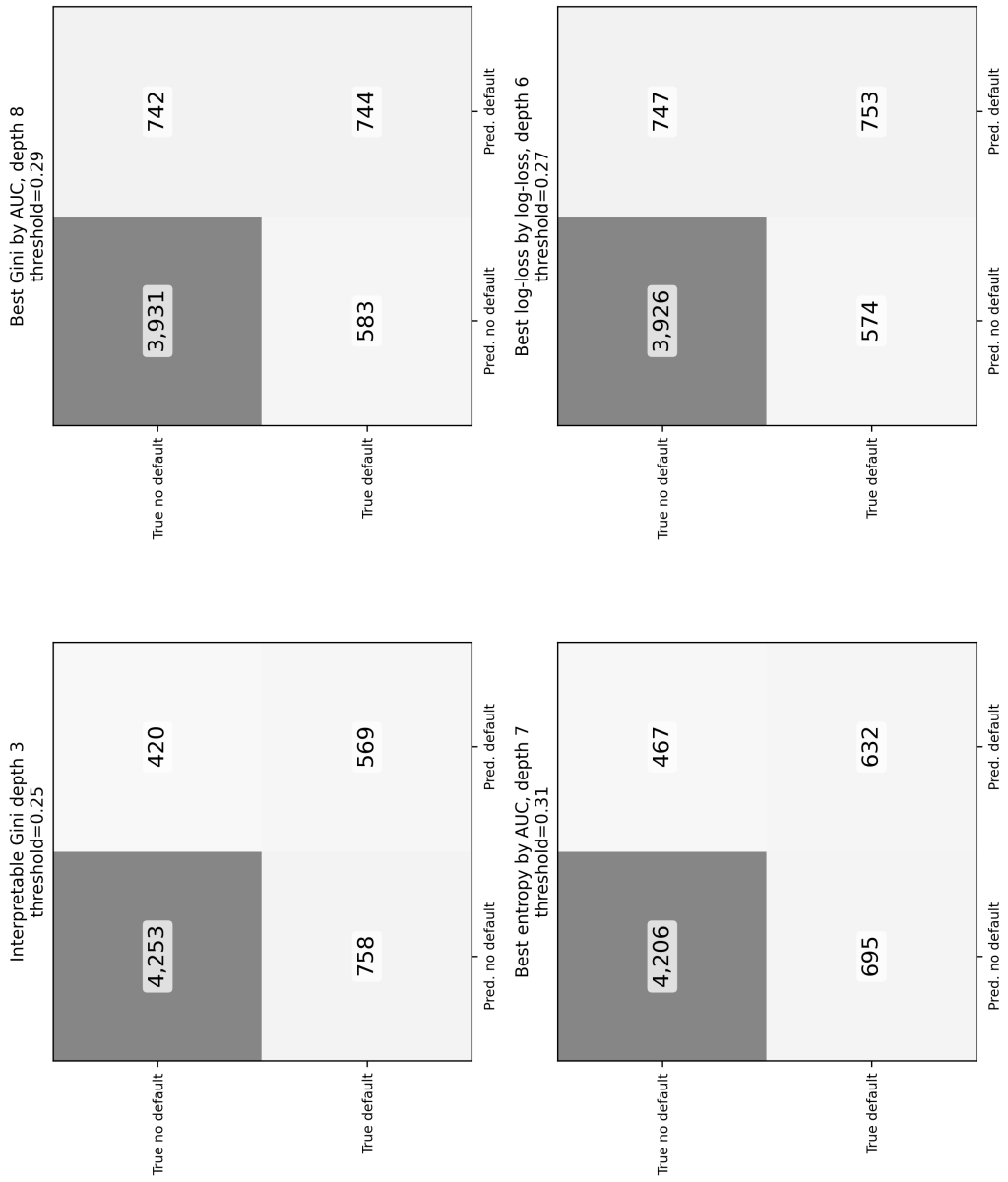


Figure 7.8: Confusion matrices for selected trees at validation-selected thresholds.

The confusion matrices make the operating consequences concrete. The interpretable Gini depth-3 tree flags 989 accounts as default-risk cases: 569 true defaults and 420 false alerts. It also misses 758 defaults. This is a conservative alerting system relative to deeper alternatives. The Gini depth-8 tree captures 686 defaults and misses 641, but it creates 554 false alerts. The entropy depth-7 tree captures 632 defaults and misses 695, with 467 false alerts. The log-loss depth-6 tree captures 753 defaults and misses 574, but at the cost of 747 false alerts.

These differences are managerially meaningful. A risk manager focused on reducing missed defaults may prefer the log-loss depth-6 tree because it captures more defaulting accounts. A manager with limited review capacity may prefer a more precise model with fewer false alerts. A governance-oriented manager may prefer the shallow tree because it is easier to explain. The confusion matrices therefore convert abstract metrics into operational trade-offs.

### 7.5.9 Managerial interpretation

The main managerial lesson is that model choice depends on the purpose of the risk system. If the objective is communication, governance, and transparent segmentation, a shallow tree may be preferred even if its AUC is not the highest. If the objective is ranking accounts for review, AUC and precision–recall performance become central. If predicted probabilities feed into pricing, provisioning, or expected-loss calculations, log-loss becomes more important. If the analyst wants to know whether the classifier adds value beyond the base rate, Cohen's Kappa provides a useful diagnostic.

The case also shows why different tree-building criteria matter. Gini focuses on node purity. Entropy and log-loss emphasize information gain and probabilistic separation. In this empirical run, entropy and log-loss are often very close, which is itself informative: the practical difference between criteria may be smaller than the difference created by depth, threshold choice, or the intended risk-management use.

## 7.6 Summary and Key Takeaways

This chapter introduced classification trees as the first predictive AI model in the book and positioned them as tools for interpretable risk segmentation. Rather than starting from a fully specified loss distribution, the analyst starts from observable features and an event label, then uses recursive partitioning to identify groups with different levels of event likelihood.

 

The central message is that predictive quality and decision usefulness are not the same thing. A classifier may achieve acceptable accuracy while missing important risky cases, produce strong AUC while offering weak probability estimates, or improve recall only by creating too many false alarms. For this reason, risk classifiers should be evaluated through a portfolio of metrics rather than a single headline score.

The credit-default case study illustrates this logic. The shallow tree provides a transparent risk map, while the broader comparison shows how tree depth and splitting criterion affect accuracy, precision, recall, specificity, $F_1$, AUC, log-loss, and Cohen's Kappa. Gini impurity, entropy, and log-loss define different ways of constructing risk segments. Confusion-matrix metrics describe operational error trade-offs. ROC and AUC summarize ranking quality. Precision–recall curves clarify alert quality when adverse events are relatively rare. Log-loss evaluates probability quality, and Cohen's Kappa asks whether the classifier adds value beyond chance agreement and the base-rate structure.

The chapter therefore prepares the transition to Chapter 8, where the predictive toolkit expands beyond single trees, and to Chapter 9, where prediction is turned more explicitly into score-based decision support.

### Key takeaways

- Classification trees are risk-identification tools: they segment, rank, and explain risky cases.
- Their main advantage is interpretability, since they translate data into visible if–then rules that can support action and governance.
- Gini impurity, entropy, and log-loss construct tree-based risk segments in different ways.
- Accuracy, precision, recall, specificity, $F_1$, AUC, log-loss, and Cohen's Kappa answer different evaluation questions.
- Accuracy can be misleading under class imbalance; balanced accuracy and Cohen's Kappa provide useful complements.
- Threshold choice is part of the decision system because it controls the trade-off between false positives and false negatives.
- AUC evaluates ranking quality across thresholds, while precision–recall analysis is especially useful when adverse events are relatively rare.

 

- Log-loss evaluates probability quality and is important when predicted probabilities feed into downstream risk scores or decisions.
- Cohen's Kappa is useful for validation and comparison, but not as a local split criterion for growing ordinary classification trees.
- In the credit-default case, the preferred tree depends on the risk-management purpose: explanation, alerting, ranking, or probability estimation.

 

# Bibliography

[1] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[2] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, 1984.

[3] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[4] Jr. Cox, Louis Anthony. *Risk Analysis of Complex and Uncertain Systems*. Springer, New York, 2009.

[5] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 233–240, 2006.

[6] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[7] Andrew Gemino, Chris Sauer, and Blaize Horner Reich. Using classification trees to predict performance in information technology projects. *Journal of Decision Systems*, 19(2):189–214, 2010.

[8] Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2 edition, 2009.

 

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer, New York, 2 edition, 2021.

[11] Patrick Kelliher. Risk classification system article. *The Actuary*, 2011. Article describing the Risk Classification Working Party and the development of a common risk language for actuaries.

[12] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268, 2007.

[13] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[14] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[15] J. Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[16] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[17] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015.

[18] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

[19] Bart van Liebergen. Machine learning: A revolution in risk management and compliance? *Journal of Financial Transformation*, 47:60–67, 2018.

[20] Anthony J. Viera and Joanne M. Garrett. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37(5):360–363, 2005.

[21] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9(2):187–212, 2022.

[22] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480, 2009.

 