UNIVERSITY OF PAVIA

DEPARTMENT OF ECONOMICS AND MANAGEMENT

MASTER PROGRAMME IN FINANCE

–AGGREGATING MULTICLASS ROC CURVES WITH APPLICATIONS TO ESG AND CREDIT RISK MANAGEMENT –

Supervisor:

Prof. Paolo Giudici

Thesis by:

Rosa Carolina Rosciano

ID number:

534143

Academic Year 2024-2025

Abstract

This thesis introduces a novel methodology for constructing unified multiclass ROC curves using the multidimensional Gini index, addressing critical limitations in existing multiclass classification evaluation approaches. Traditional methods suffer from fundamental flaws: macro-averaging treats classes with different sample sizes identically, while micro-averaging allows majority classes to dominate, potentially masking failures in critical minority classes.

The proposed methodology leverages the established relationship between the Gini coefficient and ROC analysis ($G = 2 \times AUC - 1$) and extends this framework to multiclass settings through the multidimensional Gini index. The approach employs Zero-phase Component Analysis (ZCA) correlation whitening with numerical stabilization to ensure scale invariance and computational robustness. Unlike traditional frequency-based weighting, the methodology generates class-specific weights based on discriminative power, providing a unified ROC curve that reflects genuine classification ability.

The framework is validated through a comprehensive case study using 1,724 Italian Small and Medium-sized Enterprises (SMEs) spanning 2020-2022, incorporating financial metrics and ESG ratings for credit risk assessment across nine ordinal rating classes. The methodology achieves an empirical multiclass AUC of 0.85, validated through Mahalanobis distance analysis confirming genuine class separability.

Complemented the multiclass ROC curve methodology with SAFE AI RGR robustness analysis, which reveals moderate model stability (RGR values 0.54-0.55) and addresses EU AI Act requirements for reliable performance under uncertainty. Interactive visualization tools enable real-time threshold optimization supporting regulatory compliance in high-stakes financial en-

vironments.

This research provides a theoretically grounded solution to multiclass performance evaluation, particularly valuable for imbalanced datasets in regulated domains where discriminative power should take precedence over class frequency considerations.

Questa tesi introduce una metodologia innovativa per la costruzione di curve ROC multiclasse unificate utilizzando l'indice di Gini multidimensionale, affrontando le limitazioni critiche degli approcci esistenti per la valutazione della classificazione multiclasse. I metodi tradizionali soffrono di difetti fondamentali: il macro-averaging tratta classi con dimensioni campionarie diverse in modo identico, mentre il micro-averaging consente alle classi maggioritarie di dominare, potenzialmente mascherando i fallimenti nelle classi minoritarie critiche.

La metodologia proposta sfrutta la relazione consolidata tra il coefficiente di Gini e l'analisi ROC ($G = 2 \times AUC - 1$) ed estende questo framework ai contesti multiclasse attraverso l'indice di Gini multidimensionale. L'approccio utilizza l'Analisi delle Componenti a Fase Zero (ZCA) con sbiancamento di correlazione e stabilizzazione numerica per garantire invarianza di scala e robustezza computazionale. A differenza della ponderazione tradizionale basata sulla frequenza, la metodologia genera pesi specifici per classe basati sul potere discriminativo, fornendo una curva ROC unificata che riflette la genuina capacità di classificazione.

Il framework è validato attraverso un caso studio che comprende 1.724 Piccole e Medie Imprese (PMI) italiane nel periodo 2020-2022, incorporando metriche finanziarie e rating ESG per la valutazione del rischio di credito attraverso nove classi di rating ordinali. La metodologia raggiunge un AUC multiclasse empirico di 0.85, validato attraverso l'analisi della distanza di Mahalanobis che conferma la genuina separabilità delle classi.

La metodologia della curva ROC multiclasse è integrata con l'analisi di robustezza SAFE AI RGR, che rivela una moderata stabilità del modello (valori RGR 0.54-0.55) e soddisfa i requisiti dell'AI Act dell'UE per prestazioni affidabili in condizioni di incertezza. Gli strumenti di visualizzazione interattiva consentono l'ottimizzazione delle soglie decisionali in tempo reale, supportando la conformità normativa in ambienti finanziari ad alto rischio.

Questa ricerca fornisce una soluzione teoricamente fondata per la valutazione delle prestazioni multiclasse, particolarmente preziosa per dataset sbilanciati in domini regolamentati dove il potere discriminativo dovrebbe avere precedenza sulle considerazioni di frequenza delle classi.

Contents

Chapter 1

Introduction

Chapter Contents

1.1 Context and Motivation

The main aim of this thesis is to develop a novel methodology for constructing unified multiclass ROC curves using the multidimensional Gini index, addressing critical regulatory compliance and model explainability challenges in AI-driven financial applications.

The past decade has witnessed a fundamental transformation in financial services, with the emergence of technology-driven intermediaries: fintech companies, crowdfunding platforms, and peer-to-peer lenders, and traditional banks are also increasingly leveraging artificial intelligence methods including machine learning, deep learning, and even large language models to optimize credit allocation decisions [9].

The competitive advantage of AI-powered approaches over traditional statistical methods lies in their superior predictive accuracy, achieved through their ability to capture complex non-linear relationships and handle high-dimensional data effectively. Recent systematic reviews of credit risk modeling confirm that machine learning and deep learning models consistently outperform traditional statistical approaches [38]. However, this enhanced performance comes at the cost of reduced model auditability, creating what regulators term ”black box” systems that are increasingly difficult to audit and understand.

This technological evolution has not occurred in a regulatory vacuum. The European Central Bank (ECB) and other regulatory institutions have expressed growing concerns about the opacity of AI-driven credit decisions, emphasizing the need for explainable and interpretable models in systemically important financial applications [4]. The recently enacted EU AI Act further reinforces these requirements, establishing a risk-based regulatory framework that categorizes AI systems and mandates strict compliance measures for high-risk applications, including credit scoring [10].

The challenge extends beyond mere explainability to fundamental model auditability. Financial regulators require the ability to scrutinize model evaluation and performance across all risk categories independently, rather than relying solely on overall performance metrics. This granular assessment is essential

because credit models directly impact capital allocation, systemic risk, and consumer protection; areas where performance failures in specific risk segments can have cascading economic consequences.

Credit risk assessment inherently involves multiclass classification, where borrowers are categorized into multiple ordinal risk categories rather than simple binary accept/reject decisions. This multiclass nature introduces significant challenges, particularly regarding class imbalance that can severely distort traditional performance metrics.

Consider a hypothetical credit dataset where 70% of companies receive BBB ratings, 15% receive A ratings, and only 2% receive D ratings. In such severely imbalanced scenarios, a naive classifier that always predicts the majority class (BBB) could achieve 70% accuracy while completely failing to identify any defaults. This phenomenon illustrates how models can exhibit high overall accuracy while demonstrating catastrophic failure on minority classes, precisely those representing the highest-stakes financial decisions.

This reliance on accuracy as a performance metric proves particularly misleading in multiclass classification contexts, as it treats all misclassifications as equivalent errors. However, the financial consequences of different misclassification types vary dramatically.

Therefore, traditional accuracy metrics mask the true model performance where it matters most: accurately identifying high-risk entities that could trigger significant financial losses if misclassified.

A metric which can overcome those drawbacks is the Receiver Operating Characteristic (ROC) curve, a cornerstone of binary classification evaluation that plots the True Positive Rate against the False Positive Rate across all decision thresholds but lacks a universally accepted multiclass counterpart. This limitation is particularly problematic because the binary ROC curve allows for:

- **Threshold-Independent Performance Assessment**: Unlike point metrics such as accuracy or F1-score that evaluate performance at a single decision threshold, ROC curves reveal model behavior across the entire spectrum of possible decision boundaries. For credit scoring, this is crucial because different institutions may operate at different risk tolerances, some prioritizing conservative lending (high precision), others focusing on market share (high recall). Regulators need to understand how models perform across all these operational scenarios, not just at one arbitrary threshold chosen by the institution.

- **Single Summary Statistic**: The ROC curve not only provides a visual representation but also a single summary statistic, the Area Under the ROC curve (AUC), which can be useful for regulators and model validators.

- **Cost-Sensitive Analysis**: Deals with the different costs of misclassification errors through the inspection of thresholds.

- **Agnostic Measure**: Allows to measure predictive power of models in other highly regulated fields, like medicine[1].

[1] Where ROC curves are already employed aside model evaluation

The ROC curve in credit applications allows inspection of the trade-off, across all possible thresholds, between conservative lending (low False Positive Rate but potentially missing creditworthy borrowers) and aggressive lending (high True Positive Rate but potentially accepting higher-risk borrowers).

The absence of a unified multiclass ROC framework therefore represents not merely a technical limitation, but a fundamental barrier to regulatory compliance and risk management. Without the ability to visualize, quantify, compare competing models and trade-offs across all risk categories simultaneously, institutions cannot adequately demonstrate to regulators that their AI systems maintain discriminative capability across the complete risk spectrum, a requirement that becomes increasingly critical as AI adoption in credit decisions accelerates.

1.2 Existing Approaches and Their Limitations

Current approaches to multiclass ROC analysis, including One-versus-Rest (OvR) and One-versus-One (OvO) strategies, suffer from fundamental limitations. OvR approaches create severe class imbalance when treating single classes against all others, while OvO methods generate multiple pairwise comparisons that become unwieldy and difficult to interpret as the number of classes increases. Moreover, these methods do not provide a single ROC curve that regulatory authorities can easily interpret and validate. [21] [24]

1.3 Proposed Solution: The Multidimensional Gini ROC Curve

This thesis addresses these challenges by proposing a novel methodology that constructs a single, unified ROC curve for multiclass classification tasks using the multidimensional Gini index and Zero-phase Component Analysis (ZCA) whitening [2]. Our approach leverages the established relationship between the Gini coefficient and ROC analysis in binary classification ($G = 2 \times AUC - 1$), extending this framework to multiclass settings while preserving the intuitive decision-theoretic properties of binary ROC analysis.

1.4 Key Contributions

This research makes several key contributions to both the machine learning and financial risk management literature:

- **Methodological Innovation**: We develop a theoretically grounded approach to construct a single ROC curve for multiclass classification, based on a robust statistical measure of inequality which is the Gini index, which aside from aiding to the construction of the Multiclass ROC curve also provides class-specific weights based on each class's contribution to overall model discrimination between classes.
- **Interactive Visualization**: The thesis presents interactive visualization tools that enable practitioners to explore threshold selection and performance trade-offs across multiple risk categories

simultaneously.

- **Robustness Analysis**: We integrate our multiclass ROC curve framework with the SAFE AI [3] robustness measures, providing tools that comply with EU AI Act requirements and enhance model explainability in regulated environments.

1.5 Thesis Structure

This thesis is organized into two main parts. Part I (Chapters 2-3) develops the theoretical foundation, while Part II (Chapter 4-5) demonstrates practical applications and its limitations.

Chapter 2 provides a comprehensive review of ROC curve analysis, from binary foundations to multiclass extensions, examining existing approaches and their limitations. We establish the theoretical connection between the Gini index and ROC analysis that underlies our methodology.

Chapter 3 presents our core methodological contribution: the construction of multiclass ROC curves via the multidimensional Gini index.

Chapter 4 applies our methodology to credit and ESG risk assessment using Italian SME data spanning 2020-2022, incorporating both traditional financial metrics and ESG ratings, addressing the contemporary regulatory emphasis on sustainable finance. We demonstrate the visual construction and interpretation of multiclass ROC curves, and compare our approach with traditional metrics.

Chapter 5 proposes a validation procedure for the methodology, integrates robustness analysis, and summarizes the implementation steps.

The conclusion discusses the limitations and benefits of this methodology.

Part I

Using the Multidimensional Gini Index to Construct an Aggregate ROC Curve for Multiclass Classification Tasks

Chapter 2

ROC Curves: From Binary to Multiclass Classification

Chapter Contents

2.1 The ROC curve in the binary case

The **Receiver Operating Characteristic curve (ROC curve)** is a visual tool for evaluating model performance[1] across all classification thresholds, providing insights into the **trade-offs between sensitivity and specificity at various decision boundaries**[11].

This graphical representation, which can be seen in Figure 2.1, facilitates the understanding of threshold selection for class prediction in binary classification problems.

[1] A performance tool aims to detect and correct common problems during the validation stage, such as overfitting and compare models' predictive capability during the model development phase.

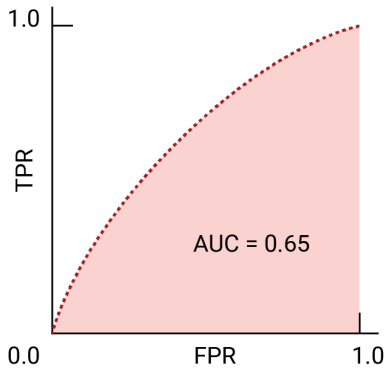

Figure 2.1: ROC and AUC of a hypothetical model.[18]

The ROC curve originated during World War II for radar signal analysis before being incorporated into signal detection theory. Beyond data science, this measure has found applications in diverse fields including medicine, psychology, and radiography. The first application of ROC analysis in machine learning was pioneered by Spackman (1989), who demonstrated its value in comparing and evaluating different classification algorithms.

Formally, the ROC curve plots the **True Positive Rate (TPR)**, also known as sensitivity, on the y-axis against the **False Positive Rate (FPR)**, equivalent to 1 - specificity, on the x-axis. The True Positive Rate represents the probability of a positive test result conditional on the subject truly belonging to the positive class, essentially measuring how many relevant items are correctly identified.

In **credit scoring**[2], for instance, a true positive occurs when the model correctly identifies a company's rating at a given threshold value. Conversely, the false positive rate quantifies the fraction of negative instances incorrectly classified as positive. In our example, companies mistakenly assigned higher ratings than they deserve, resulting in overestimation of creditworthiness at that threshold.

These error rate measurements have important relationships that decision-makers must understand:

$$\text{TPR} = 1 - \text{FNR}$$

$$\text{FPR} = 1 - \text{TNR}$$

Where FNR represents the False Negative Rate, so the proportion of positive instances incorrectly classified as negative, and TNR is the True Negative Rate, which in our case measures how well the model avoids misclassifying companies into a rating class they do not belong to. It is worth noting that TPR and FPR are complementary measures within their respective conditions, not across the entire population, thus $TPR + FPR \neq 1$.

Returning to our case under study, proper decision-making requires stakeholders to carefully **weigh which type of error is more relevant to their objectives**. Prioritizing FPR reduction helps avoid unbounded downside losses, while focusing on minimizing FNR prevents missed investment opportunities at the expense of potential profit. This asymmetry in error costs represents a fundamental consideration in practical applications of classification systems; we must choose a threshold which shows the trade-off between correctly identifying positives and incorrectly classifying negatives as positives. The ROC curves shows us exactly that.

Moreover, sensitivity and specificity address **limitations in accuracy interpretation**, particularly when comparing models with similar accuracy values but different error distributions.

Sensitivity [3] measures the ratio between true positive decisions and the total number of positive cases, while specificity is the true negative rate and calculates the ratio between true negative decisions and the total number of negative cases [31], but also measures which measures the ability to correctly identify negative cases.

The threshold selection process directly impacts these metrics. As noted by Narkhede (2018), "When

[2] Which is the problem under case study.
[3] Also known as recall.

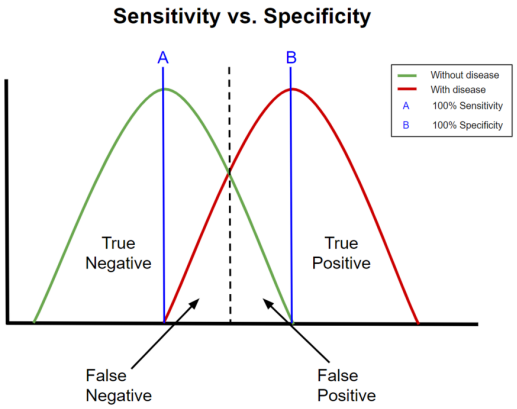

Figure 2.2: Error distributions in a medical testing case

we decrease the threshold, we get more positive predictions, thus increasing sensitivity while decreasing specificity. Similarly, when we increase the threshold, we get more negative predictions, resulting in higher specificity but lower sensitivity."

False negatives and false positives typically arise when the probability distributions of the true positive and true negative classes overlap, indicating an imperfect separation between classes, this can be better understood with Figure 2.2.

The TPR and FPR values that form the ROC curve are derived from the predicted probabilities produced by the classifier, making them functions of the classifier's discriminative capability. The curve itself depicts the fundamental trade-off between TPR and FPR across all possible thresholds.

There are several properties that make this tool paramount when developing and comparing models. In fact, according to Flach (2016), "The ROC curve can be used to address a range of problems, including: (1) determining a decision threshold that minimizes error rate or misclassification cost under given class and cost distributions; (2) identifying regions where one classifier outperforms another; (3) identifying regions where a classifier performs worse than chance; and (4) obtaining calibrated estimates of the class posterior probabilities."

2.1.1 Properties of ROC curves

ROC curves possess several important mathematical properties that enhance their utility in classifier evaluation. Flach (2016) highlights a fundamental property: "The true (false) positive rate achieved at a certain decision threshold T is the proportion of the positive (negative) score distribution to the right of the threshold". That is,

$$TPR(T) = \sum_{s>T} f(s^+)$$

and

$$FPR(T) = \sum_{s>T} f(s^-)$$

Where T is the threshold, s^+ is the positive class and s^- the negative one.

This relationship directly connects the ROC curve to the underlying score distributions produced by the classifier.

This property allows us to reconstruct the range of thresholds yielding a particular operating point by examining the score distributions. Moreover, if we connect two distinct operating points on an ROC curve by a straight line, the slope of that line segment equals the ratio of positives to negatives in the corresponding score interval[13].

This geometric interpretation provides valuable insights into the local behavior of the classifier across different regions of the score distribution.

In practice, **ROC curves are rarely perfectly smooth**, as noted by Flach (2016): "ROC curve concavities demonstrate locally sub-optimal behaviour of a classifier." These concavities indicate regions where the classifier's performance could be improved through calibration or alternative decision strategies. The presence of such concavities often suggests that the underlying score distributions are multimodal or that the classifier makes systematically different errors across different subpopulations of instances.

2.2 The Area Under the ROC curve

The ROC curve not only provides a graphical representation of classifier performance but also yields a scalar metric that quantifies its discriminative capability: the Area Under the ROC Curve (AUC). This measure possesses robust statistical properties, as it represents the probability that a randomly chosen positive sample will be assigned a higher score than a randomly chosen negative sample[11]. Therefore,

$$AUC = P(s^+ > s^-)$$

The AUC is particularly valuable for comparing classifiers **when no single model dominates across all operating points, offering a threshold-independent performance assessment**.

As Narkhede (2018) explains, "ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes." This characteristic bears a conceptual similarity to the Gini index [17][16], which will be later discussed, traditionally used in economics and social sciences to measure statistical dispersion.

In credit risk assessment applications, where **minimizing false positive rates (FPR) is often prioritized to avoid potentially catastrophic losses**, an ideal ROC curve approaches the top-left corner of the plot, indicating high True Positive Rates (TPR≈1) combined with low false positive rates (FPR≈0).

While this perfect discrimination rarely occurs in practice, it establishes that **models with larger AUC values generally demonstrate superior performance**. The "steepness" of the ROC curve, particularly in regions of low FPR, provides crucial information about a model's ability to maximize sensitivity while maintaining high specificity in the critical operating region.

The AUC threshold-independence makes it an attractive summary statistic; higher AUC values indicate better overall discriminative performance regardless of the specific decision threshold employed. An ideal model achieves AUC = 1, as in the hypothetical curve in Figure 2.3, reflecting perfect separation

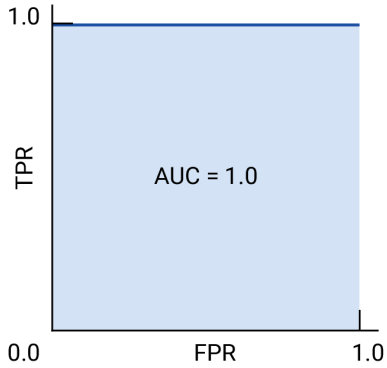

Figure 2.3: ROC and AUC of a hypothetical perfect model. [18]

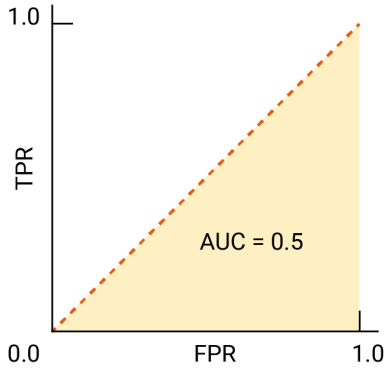

Figure 2.4: ROC and AUC of completely random guesses. [18]

between positive and negative classes when evaluated on independent test data not used during model training.

As Flach (2016) notes, ”The classifier’s performance depends on how well the per-class score distributions are separated.” This separation directly influences the AUC value, however, when we have many classes this becomes troublesome, more about this in subsequent sections.

This probabilistic interpretation captures the essence of discrimination when presented with one randomly selected positive instance and one randomly selected negative instance, the AUC represents the probability that the classifier will correctly differentiate between them.

Flach (2016) further elaborates on the mathematical properties of AUC: ”Since **the curve is located in the unit square**, we have $0 \leq AUC \leq 1$. **AUC = 1 is achieved if the classifier scores every positive higher than every negative**; AUC = 0 is achieved if every negative is scored higher than every positive. AUC = 0.5 is obtained in a range of different scenarios, including: (i) the classifier assigns the same score to all test examples, whether positive or negative, and thus the ROC curve is the ascending diagonal; (ii) the per-class score distributions are similar, which results in an ROC curve close (but not identical) to the ascending diagonal; and (iii) the classifier gives half of a particular class the highest scores, and the other half the lowest scores.”

This third case, seen in Figure 2.4, highlights an important nuance: ”Although a classifier with AUC close to 0.5 is often said to perform randomly, **there is nothing random in the third classifier: rather, its excellent performance on some of the examples is counterbalanced by its very poor performance on some others**”[13]. This observation underscores that AUC, while valuable, may obscure significant variations in classifier performance across different subsets of the data.

2.3 Limitations of the ROC curve

Despite their widespread adoption, ROC curves and the AUC metric face substantial criticisms that challenge their suitability as universal performance measures. As noted by several researchers, ”Any attempt to summarize the ROC curve into a single number loses information about the pattern of tradeoffs of the particular discriminator algorithm”[19]. This fundamental limitation undermines the completeness of AUC as a standalone evaluation metric.

A significant criticism of ROC curves concerns **the incorporation of areas with low sensitivity and low specificity (both lower than 0.5) in the calculation of the total AUC**[5].

These regions correspond to confusion matrices where binary predictions yield poor results and typically represent threshold values that would never be employed in practice. As Chicco and Jurman (2023) argue, these portions of the curve, associated with either very high or very low classification thresholds, are rarely of interest to researchers performing binary classification in any domain, yet they contribute equally to the AUC calculation.

The AUC measure is invariant to the class distribution, meaning it treats all misclassification types with equal importance regardless of their prevalence. While this property is sometimes considered advantageous, Lobo et al. (2008) argue that it can be misleading in applications where the class distribution is highly skewed and unlikely to change. In such contexts, the AUC may give undue weight to regions of ROC space that represent unrealistic operating conditions, potentially favoring models that perform well on rare cases at the expense of common ones.

Muschelli (2019) highlights another significant limitation: ”ROC and AUC say nothing about precision and negative predictive value.” Since ROC analysis focuses exclusively on sensitivity and specificity, it fails to capture the precision dimension that is crucial in many applications, particularly those involving rare events or where false positives carry significant costs, such as the credit risk management case. This omission has led to the development of alternative visualization approaches like **Precision-Recall (PR) curves**, which explicitly address this limitation by plotting precision against recall.

For classifiers that produce binary outputs rather than continuous scores, Muschelli (2019) demonstrates that the AUC can be potentially misleading. Since such classifiers produce only a single point in ROC space rather than a curve, the resulting ”AUC” represents an artificial interpolation that may not accurately reflect the classifier’s performance characteristics.

Halligan et al. (2015) argue that the AUC has limited utility for comparing the performance of imaging tests in clinical settings. They note that ”because the AUC summarizes performance across all possible thresholds, it includes many that would never be used in clinical practice.” This criticism extends to

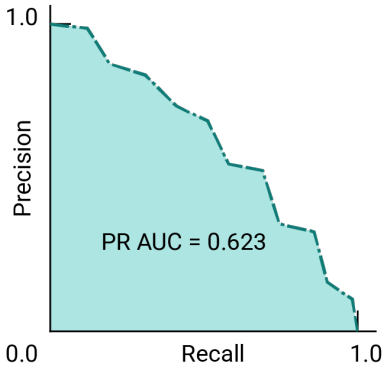

Figure 2.5: Precision-Recall curve of a hypothetical model. [18]

many other application domains where only a small range of thresholds would ever be considered viable operating points.

2.4 Alternative Performance Metrics

Given these limitations, several researchers have proposed alternative metrics. Chicco and Jurman (2023) advocate for the Matthews Correlation index (MCC) as a superior alternative to AUC for binary classification assessment. The MCC takes into account all four elements of the confusion matrix (true positives, false positives, true negatives, and false negatives) and provides a more balanced measure that is particularly suitable for imbalanced datasets.

Other alternative approaches include the Total Operating Characteristic (TOC) curve, which reveals more information than the ROC by showing absolute counts rather than ratios, and Precision-Recall (PR) curves, which focus on the trade-off between precision and recall, metrics particularly relevant in information retrieval and rare event detection scenarios.

We can also plot precision in the y-axis, and recall in the x-axis, as in Figure 2.5, in this way obtaining a curve similar to the ROC one, in which we can also analyze decision threholds as a trade-off of these two metrics. Moreover, this precision-recall curve tends to be more robust to the imbalanced cases since a classifier could achieve a high AUC score on its ROC curve while delivering poor performance when predicting the minority class in datasets with severe class imbalance. Precision-Recall curves are better at exposing these performance issues.

2.4.1 Performance metrics in the multiclass case

In supervised learning, multiclass classification involves categorizing instances into more than two classes[34]. Unlike binary classification, there is no inherent concept of ”positive” and ”negative” classes, which creates a fundamental difficulty in constructing aggregate multiclass measures for ROC analysis.

This challenge is particularly pronounced when dealing with imbalanced datasets, where the number of instances varies significantly across classes.

The core problem in multiclass classification lies in maintaining good generalization, using training data to develop classification rules that perform well on unseen data.

While binary classification allows for a clear distinction between classes and straightforward calculation of performance metrics, multiclass scenarios introduce complexity in how we measure and visualize classifier performance across multiple decision boundaries.

In binary classification, several standard performance metrics are derived from the confusion matrix elements[34]. Precision, also known as positive predictive value, is defined as the ratio of true positives to the sum of true positives and false positives:

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

Precision quantifies the classifier's ability to avoid labeling negative samples as positive [11]. It ranges from 0 to 1, with 1 indicating perfect precision where no negative samples are incorrectly classified as positive. Precision becomes particularly important in applications where false positives carry significant consequences.

Recall is calculated as the ratio of true positives to the sum of true positives and false negatives:

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

Recall measures the classifier's ability to identify all positive samples [6]. Like precision, it ranges from 0 to 1, with 1 representing perfect recall, where all positive samples are correctly classified.

When analyzing precision-recall curves, it is important to note that the terminal point typically has precision = 1.0 and recall = 0.0, corresponding to a highly conservative classifier that rarely predicts the positive class but does so with high confidence. Conversely, the initial point generally has precision equal to the class balance and recall = 1.0, representing a classifier that always predicts the positive class[36].

The F1-score represents the harmonic mean of precision and recall, providing a balanced measure that accounts for both metrics:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score ranges from 0 to 1, with higher values indicating better performance. Unlike the arithmetic mean, the harmonic mean gives more weight to lower values, making the F1-score particularly sensitive to imbalances between precision and recall [5]. This characteristic makes the F1-score valuable in scenarios with imbalanced datasets.

When extending these metrics to multiclass problems, several approaches exist, each with distinct limitations. The two predominant methods for aggregating performance across multiple classes are micro-averaging and macro-averaging[39].

In micro-averaging, metrics are calculated globally by counting the total true positives, false negatives, and false positives across all classes:

$$\text{Precision}_{\text{micro}} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} (TP_i + FP_i)}$$

$$\text{Recall}_{\text{micro}} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} (TP_i + FN_i)}$$

Where K represents the total number of classes and TP_i, FP_i, and FN_i are the true positive, false positive, and false negative counts for class i, respectively.

In contrast, macro-averaging calculates metrics independently for each class and then takes an unweighted mean:

$$\text{Precision}_{\text{macro}} = \frac{1}{K} \sum_{c=1}^{K} \frac{TP_i}{TP_i + FP_i}$$

$$\text{Recall}_{\text{macro}} = \frac{1}{K} \sum_{i=1}^{K} \frac{TP_i}{TP_i + FN_i}$$

Macro-averaging gives equal importance to each class by computing the simple average of all one-vs-rest ROC curves. This approach "gives equal importance to each class" regardless of class size [1].

Despite their widespread use, these multiclass evaluation approaches have notable limitations. Micro-averaging inherently gives more weight to classes with larger sample sizes, potentially masking poor performance on minority classes[14]. This is particularly problematic in applications where minority classes are of significant interest[4] or where class imbalance is severe.

Macro-averaging, while treating all classes equally regardless of their frequency, does not account for class imbalance and may overemphasize performance on classes with fewer samples [32]. In highly imbalanced datasets, a classifier might achieve high macro-averaged metrics despite performing poorly on important minority classes.

There is also weighted macro-averaging, which applies weights based on class frequencies to address imbalance issues. This approach is particularly useful in "multi-class classification setup with highly imbalanced classes" where micro-averaging might overly focus on majority classes [1].

Additionally, traditional multiclass extensions of precision, recall, and F1-score do not capture the complete picture of classification performance across the full spectrum of decision threshold [21]. These metrics provide point estimates at specific thresholds rather than a comprehensive view of the classifier's behavior across all possible thresholds. This can be overcome if we use a single ROC curve for the multiclass task, as it will be done.

[4] As is the case under study.

Furthermore, existing multiclass performance metrics often fail to adequately capture the nuanced aspects of misclassification costs, which can vary significantly across different class pairs in real-world applications [12]. This limitation becomes particularly pronounced in domains where certain misclassification types carry substantially different consequences, such as financial applications, where there could be systemic consequences for both clients, companies, and the whole economy.

2.5 Extending ROC Analysis to Multiclass Classification: Approaches and Limitations

While the binary ROC framework provides a well-established methodology for performance assessment, its extension to multiclass problems introduces significant conceptual and practical challenges. Unlike the binary case, **there exists no single universally accepted AUC measure that comprehensively summarizes model predictive performance in the multiclass context** [21].

Several approaches have been proposed to extend ROC analysis beyond binary classification. The most common strategies include:

- **One-versus-Rest (OvR) Approach:**

  The One-versus-Rest approach decomposes the multiclass problem into multiple binary classification problems, constructing one ROC curve for each class treated as the positive class against all others combined. This generates k different ROC curves for a k-class problem [35]. While straightforward to implement, this method suffers from several limitations:

  1. **Class Imbalance:** When treating a single class as positive against all others, severe class imbalance often arises, particularly for rare classes, potentially biasing the resulting ROC curves [11].
  2. **Information Loss:** Aggregating all non-target classes masks the model's ability to discriminate between specific class pairs, potentially obscuring important performance characteristics [24]. For instance, misclassifying class A as class B might have vastly different practical implications than misclassifying class A as class C. This nuance is difficult to capture in aggregated performance metrics.
  3. **Threshold Inconsistency:** Different thresholds may be optimal for different class-specific ROC curves, making it difficult to select a coherent set of thresholds for the overall multiclass classifier [12].
  4. **Inherited Binary Limitations:** As noted by Hand [20], the OvR approach inherits all the limitations of binary ROC analysis, including the incoherence problem and insensitivity to varying misclassification costs across different classes.

- **One-versus-One (OvO) Approach:**

  The One-versus-One strategy constructs ROC curves for each possible pair of classes, generating $(k(k-1))/2$ curves for a k-class problem. Hand and Till [21] proposed a multiclass AUC measure based on averaging all pairwise AUCs, known as the M-measure:

$$M = \frac{2}{k(k-1)} \sum_{i<j} A(i,j)$$

where $A(i,j)$ represents the AUC for discriminating between classes i and j. While this approach captures pairwise separability between classes, it presents several drawbacks:

1. **Interpretation Difficulty:** Analyzing and interpreting multiple pairwise ROC curves becomes challenging, complicating the derivation of actionable insights [13].
2. **Ignoring Class Relationships:** The pairwise approach treats all class pairs equally, potentially overlooking naturally ordered or hierarchical relationships between classes [12].
3. **Aggregation Bias:** The M-measure's simple averaging of pairwise AUCs fails to account for the varying importance of different class distinctions and can mask severe performance deficiencies in distinguishing particular class pairs [20].

Both OvR and OvI reduce the multiclass problem into multiple binary problems.

- **Volume Under the Surface (VUS):**

  It has also been proposed to extend the ROC curve to a three-dimensional ROC surface, with the Volume Under the Surface (VUS) serving as the multiclass analogue of AUC [30]. This approach can be generalized to *k* dimensions for *k* classes, but faces severe limitations:

  1. **Visualization Impossibility:** Beyond three classes, the resulting hypersurface cannot be directly visualized, severely limiting intuitive interpretation [24].
  2. **Statistical Properties:** The probabilistic interpretation of VUS becomes increasingly complex and less intuitive compared to the binary AUC [21].

Moreover, various weighted averaging schemes have been proposed to combine class-specific or pairwise ROC information, adjusting for factors such as class prevalence or misclassification costs [12]. However, these approaches still face fundamental challenges:

1. **Weight Selection:** Determining appropriate weights remains subjective and context-dependent, potentially introducing bias [24].
2. **Interpretability:** Weighted averages may obscure performance deficiencies for specific classes or class combinations [13].
3. **Threshold Selection:** Identifying optimal operating points across multiple weighted ROC curves remains problematic [12].
4. **Theoretical Foundation:** Many weighted schemes lack a coherent theoretical foundation, making their interpretation and justification difficult [20].

The limitations of existing multiclass ROC extensions highlight the need for a more unified framework that preserves the intuitive interpretation and decision-theoretic foundations of the binary case while accommodating the complexities of multiclass discrimination.

This gap motivates the development of novel approaches that can provide a single, comprehensive ROC-based visualization and quantification method for multiclass classification problems; precisely the focus of this thesis.

2.6 The Gini index and ROC Analysis

The Gini index, originally developed to measure income inequality in economics, has a well-established relationship with ROC analysis in the binary classification context. For binary classification, the Gini index can be expressed as: $G = 2 \cdot AUC - 1$. This relationship can be proven geometrically, as demonstated by Hand and Till in 2001[21].

This relationship provides an alternative perspective on classifier performance, measuring the degree of separation between class distributions[13]. The Gini index ranges from 0 to 1, where 1 indicates perfect classification and 0 a classification equal to random guessing.

From a geometric perspective, **the Gini index in ROC analysis is analogous to its application in economics through the Lorenz curve [27]**, both of which illustrate cumulative proportions of true positives (ROC) and of wealth or outcomes (Lorenz). The Gini coefficient in economics measures inequality as twice the area between the Lorenz curve and the diagonal (line of equality). Formally:

$$G = 1 - 2 \int_0^1 L(F)\, dF$$

where $L(F)$ is the Lorenz curve, that is, the cumulative share of total outcomes up to fraction F of the population.

The Lorenz curve is a statistical representation of the Cumulative Distribution Function (CDF) of the examples of the target class (in a binary problem) plotted against the fraction of the whole sample (all examples of all classes), sorted by quantile-score. It defines an area metric, the Gini index, that is twice the area between the target class CDF and the chance diagonal, corresponding to randomly sorted examples[7].

Schechtman and Schechtman demonstrated that ”the ROC curve can be presented as a relative concentration curve for a properly defined X and Y, hence the area between the ROC curve and the 45-degree line can be calculated as a Gini covariance up to constants”[37].

Geometrically, the ROC and the Lorenz curve are inverses: the factor of 2 in the equation arises from normalizing the area between the curve and the diagonal to run from 0 (no discrimination) to 1 (perfect discrimination). In economics, the amount of inequality is quantified as the area between the Lorenz curve and the line of equality; in classifier analysis, the Gini coefficient analogously measures class separability as the area between the ROC curve and the diagonal.

In our proposed framework, we exploit this relationship to construct an aggregate multiclass ROC curve that preserves the interpretability and discriminative power of the binary case while addressing the limitations of existing multiclass approaches.

Chapter 3

Methodology: The Aggregated Multiclass ROC Curve via Gini Multidimensional index

Chapter Contents

3.1 Towards a Unified Multiclass ROC Framework

Due to the deficiencies in existing methodologies, there is an obvious necessity for an integrated approach that broadens the valuable features of binary ROC assessment to multiclass applications while retaining interpretive transparency and statistical validity. The research presents a new technique that utilizes the association between the Gini index and ROC assessment to create a unified ROC curve for multiclass classification tasks. For each class i in a k-class problem, we calculate:

$$G_{\text{i}} = 2 \cdot \text{AUC}_{\text{i}} - 1 \tag{3.1}$$

Where AUC_i represents the Area Under the ROC Curve for class i using either the one-vs-all or one-vs-one approach. We then construct an aggregate ROC curve through a weighted average of the class-specific ROC curves:

$$\text{TPR}_{\text{agg}} = \sum_{i=1}^{k} w_i \cdot \text{TPR}_i \tag{3.2}$$

$$\text{FPR}_{\text{agg}} = \sum_{i=1}^{k} w_i \cdot \text{FPR}_i \tag{3.3}$$

Here, w_i denotes the weight assigned to each class i in the aggregate measurements. The determination of this vector of weights will be discussed later in this thesis.

This approach allows classes with higher discriminative power[1] to exert greater influence on the aggregate ROC curve, ensuring that the final measure reflects overall classification performance while remaining sensitive to individual class separability.

The aggregate AUC can be calculated as:

$$\text{AUC}_{\text{agg}} = \int_0^1 \text{TPR}_{\text{agg}}(\text{FPR}_{\text{agg}})\, d\text{FPR}_{\text{agg}} = \sum_{i=1}^{k} w_i \cdot \text{AUC}_i \tag{3.4}$$

Where the last equality corresponds to the discrete approximation, hence, the one we are interested in.

This formulation enables us to decompose aggregate classification performance, considering the contributions of each class to the total separability of the model. The corresponding aggregate Gini index is [2]:

$$G_{\text{agg}} = 2 \cdot \text{AUC}_{\text{agg}} - 1 \tag{3.5}$$

Where $G_{agg} = 1$ implies perfect separability between classes, $G_{agg} = 0.5$ indicates random classification, while $G_{agg} = 0$ represents no discrimination between the classes.

By building on the established connection between the Gini index and ROC analysis, our proposed framework offers a novel approach to multiclass performance evaluation that addresses the limitations of existing methods while preserving the interpretability and robustness that have made binary ROC analysis so valuable in practice.

3.2 The Multidimensional Gini index

Assessing inequality within multivariate statistical distributions poses substantial difficulties. In the Economics field such quantification is valuable for examining the concentration of wealth and income within a population, nonetheless, studying the cocentration of certain distribution via the Gini index allows to compress complex distributional information into a single, interpretable metric. While univariate inequality measures such as the Lorenz curve and the Gini index [17] [16] remain foundational tools for analyzing the mutual variability expressed by statistical distributions, their application to multivariate data is not straightforward. This limitation arises because univariate instruments disregard the inherent dependence structure among components of a multidimensional random vector [2].

A critical property that any multivariate Gini index should possess is scale invariance, which ensures that the index remains unchanged when components of the original random vector are scaled by positive constants.

This characteristic is crucial for assessing inequality within multivariate scenarios that may involve differing units of measurement. This situation arises, for instance, when in later sections we will employ the

[1] Thus, the larger the Gini index.
[2] From equation 3.1

Multidimensional Gini index to evaluate the dispersion of predicted probabilities by a classifier across various classes in the multiclass setting.

Auricchio et al. (2024) in their paper "Extending the Gini Index to Higher Dimensions via Whitening Processes"[3] proposed a new version of the multivariate Gini index that satisfies the scale invariance property by utilizing the Mahalanobis distance in place of the Euclidean distance.

While the Euclidean distance is commonly used to measure similarity or dispersion in multivariate data, it does not account for differences in scale or correlations between variables. This limitation becomes especially apparent when evaluating dispersion in high-dimensional or highly correlated datasets.

The Mahalanobis distance, on the other hand, achieves scale invariance by normalizing for variances and covariances. By using the covariance matrix for normalization, Mahalanobis distance transforms the data such that each feature (or column) has unit variance and all features are uncorrelated. As a result, the transformed matrix is of full rank, ensuring that its columns are linearly independent. This adjustment allows Mahalanobis distance to accurately capture true statistical dispersion, regardless of scale or inter-variable relationships.

Using the l_p-Mahalanobis norm, a new multivariate Gini-type index is defined as:

$$G_p(X) = \frac{1}{2\, M_p(X)} \int_{\mathbb{R}^n \times \mathbb{R}^n} \left\| W_\mu^{\text{ZCA-cor}}(x-y) \right\|_p \, \mu(dx)\, \mu(dy) \tag{3.6}$$

where $W_\mu^{ZCA-cor}$represents the correlation whitening process associated with X. x and y are two independent samples drawn from μ[4], and μ is the probability measure associated with X[2].

μ allows to define the distribution of X, and it is used to compute expectations or integrals over the space $\mathbb{R}^n$.

The l_p-Mahalanobis norm is defined as:

$$M_p(X) = M_p(QX) \tag{3.7}$$

where Q is any diagonal matrix whose diagonal contains strictly positive values, and X is any random vector.

When $p = 1$[5], this index becomes particularly interesting, as it can be expressed as a convex combination of the one-dimensional Gini indices of the components of the vector $W_\mu^{ZCA} X$:

$$G_1(X) = \sum_{i=1}^{n} \frac{|m_i^*|}{\sum_{j=1}^{n} |m_j^*|} \cdot G\left((W_\mu^{\text{ZCA}} X)_i \right) \tag{3.8}$$

[6]

[3] This whole chapter is based mainly on this article

[4] Thus, drawn from the distribution of X.

[5] When $p = 1$, the Manhattan Distance is computed, which measures distance only horizontally and vertically.

[6] For further explanations, the reader is invited to check Theorem 4 in "Extending the Gini Index to Higher Dimensions via Whitening Processes" (Auricchio et. al, 2024).

where m is the mean of X, $m_i^* = (W_\mu^{ZCA} m)_i$ and $G(W_\mu^{ZCA} X)_i$ is the one-dimensional Gini index of the $i-th$ component of $W_\mu^{ZCA} X$[2].

This equation will be useful when, in further sections, the Multiclass ROC curve will be constructed.

Furthermore, it is particularly significant as it establishes that the higher-dimensional Gini index induced by the l_1-Mahalanobis norm is a convex combination of the one-dimensional Gini indices of the whitened random variables.

Another relevant result is that when the components of X are non-negative, the index satisfies:

$$0 \leq G_1(X) \leq 1$$

This property ensures that the multidimensional Gini index remains within the same interpretable range as its univariate counterpart and the AUC of the ROC curve.

3.2.1 Properties of the G_1

The G_1 inequality measure possesses several important properties[2]:

- **Approximation Properties:** For any $\epsilon > 0$, there exist random variables X_ϵ and X'_ϵ such that $G_1(X_\epsilon) \leq \epsilon$ and $G_1(X'_\epsilon) \geq 1 - \epsilon$. Thus, G_1 can attain values arbitrarily close to both 0 and 1.
- **Scale Invariance:** $G_1(X) = G_1(XQ)$ for any diagonal matrix $Q = \text{diag}(q_1, \dots, q_n)$ with $q_j > 0$.
- **Rising Tide Property:**[7] $G_1(X + c) \leq G_1(X)$ for any strictly positive vector $c \in \mathbb{R}^n_+$. This formalizes the intuition that uniformly increasing all features reduces measured inequality.
- **Dominance:** If the leading component dominates, i.e., $|(m_1)^*| \gg \sum_{i=2}^n |(m_i)^*|$, then $\lim_{|(m_1)^*| \to \infty} G_1(X) = G(X^*)$.

The approximation properties and dominance demonstrate that this is a robust measure of inequality even in limiting cases. Furtheremore, the dominance property indicates that if one of the uncorrelated components of X^* dominates the others in mean, then the overall inequality is driven by that component.

3.2.2 Interpretation and Application

The Multidimensional Gini index presented here offers a coherent way to measure inequality in multivariate data by leveraging whitening transformations to express the same inequality, or separability, as a function of standard one-dimensional Gini indices applied to whitened components.

The weights in this convex combination depend on the relative importance of the mean of each whitened component, proportional to its mean value normalized by the sum of all mean values.

$$w_i = \frac{|m_i^*|}{\sum_{j=1}^n |m_j^*|} \tag{3.9}$$

[7] This is a particular form of monotonicity, but in the direction of reduced inequality.

This weight relates to the one in equations 3.2, 3.3 and 3.4. And will be used to construct the Multiclass Gini Multidimensional ROC curve.

The multidimensional index captures inequality across all dimensions simultaneously, offering a more comprehensive assessment than univariate measures alone [2].

3.2.3 Whitening Process

Whitening is a linear transformation that converts a random *n*-dimensional vector $\mathbf{X} = (X_1, \dots, X_n)^T$ with mean value $\mathbf{m} = (m_1, \dots, m_n)^T$ and positive definite $n \times n$ covariance matrix Σ into a new random vector $\mathbf{X}^* = W_\mu X$ whose entries are orthonormal, meaning the variance of each X_i^* is 1 and the covariance of any X_i^* and X_j^* is zero whenever $i \neq j$ [2]. This transformation can be viewed as a generalization of standardization, which is carried out by:

$$\mathbf{X}^* = V^{-1/2}\mathbf{X} \tag{3.10}$$

where the diagonal matrix $V = \text{diag}(\text{var}(X_1), \text{var}(X_2), \dots, \text{var}(X_n))$ contains the variances of X_i. While standardization ensures unitary variance for each component, it does not remove the correlations between components [23]. Therefore, this kind of whitening transformation could also aid to reduce muticollinearity within independent variables.

In its most generic form, a whitening process is a map:

$$S : P(\mathbb{R}^n) \to P_{Id}(\mathbb{R}^n) \tag{3.11}$$

that transforms an n-dimensional random vector $\mathbf{X}$ characterized by a probability measure μ, with positive mean $\mathbf{m}$ and covariance matrix Σ, into a new n-dimensional random vector whose covariance matrix is the identity matrix[2].

A whitening process S is linear if, for any given $\mathbf{X} \in P(\mathbb{R}^n)$, there exists an $n \times n$ square matrix that depends on $\mathbf{X}$ through its distribution, denoted W_μ, such that:

$$\mathbf{X}^* = S(\mathbf{X}) = W_\mu \mathbf{X} \tag{3.12}$$

The matrix W_μ is known as the whitening matrix associated with S [23]. If the covariance matrix Σ of $\mathbf{X}$ is invertible, then the whitening matrix must satisfy:

$$W_\mu \Sigma W_\mu^T = I \tag{3.13}$$

which simplifies to:

$$W_\mu^T W_\mu = \Sigma^{-1} \tag{3.14}$$

This condition does not uniquely determine the linear application that transforms $\mathbf{X}$ into a whitened vector,

allowing for rotational freedom. Given a whitening matrix W_μ, any $\tilde{W}_\mu$ of the form:

$$\tilde{W}_\mu = ZW_\mu \tag{3.15}$$

is also a whitening matrix as long as Z is an orthogonal[8] matrix, i.e., $Z^T Z = I$ [26].

3.2.4 Zero-phase Components Analysis (ZCA) Whitening

The Zero-phase Components Analysis (ZCA) whitening transformation, also known as Mahalanobis whitening, is characterized by the matrix:

$$W_\mu^{ZCA} = \Sigma^{-1/2} \tag{3.16}$$

Since Σ is symmetric and positive definite, it can be decomposed[9] as:

$$\Sigma = Z\Theta Z^T \tag{3.17}$$

where Z is the eigenmatrix associated with Σ, and Θ is the diagonal matrix containing the positive eigenvalues of Σ. Therefore:

$$\Sigma^{-1/2} = Z\Theta^{-1/2}Z^T \tag{3.18}$$

For a Gaussian random vector, this transformation yields a new Gaussian random vector with uncorrelated (and thus independent) components[2].

If the whitening process is not scale stable, it entails that the whitened random vector can change if components of the original vector are scaled by positive constants [2].

3.2.5 The ZCA-Correlation Whitening

The correlation whitening, also known as Zero-Components Analysis ($ZCA-cor$), employs a whitening matrix derived from the correlation matrix. For a random vector $\mathbf{X}$, the whitening matrix is defined as:

$$W_\mu^{ZCA-cor} = P^{-1/2}V^{-1/2} \tag{3.19}$$

where P is the correlation matrix of $\mathbf{X}$, and V is the diagonal matrix containing the variances of each component[2].

Since the correlation matrix P is scale invariant, the $ZCA-cor$ whitening process is scale stable:

$$S(\mathbf{X}) = S(Q\mathbf{X}) \tag{3.20}$$

regardless of how the square root of P^{-1} is selected [2]. This for every random vector $\mathbf{X}$ and any diagonal matrix Q with positive diagonal elements.

[8] An orthogonal, or orthonormal matrix, is a real square matrix whose columns and rows are orthonormal vectors.
[9] This is an application of the Spectral Decomposition Theorem.

Among the various whitening processes, both Cholesky and ZCA-correlation whitening are scale stable, while Principal Component Analysis (PCA) whitening is not, as demonstrated through counterexamples by Auricchio et al. (2024)[2].

Another important property for certain applications is the preservation of non-negativity. When working with non-negative data, the whitening transformation should preserve this property. Auricchio et al. (2024) demonstrated that:

For a random vector $\mathbf{X}$ with invertible correlation matrix P, the random vector:

$$O\Lambda^{-1/2}O^T V^{-1/2}\mathbf{X} \tag{3.21}$$

is non-negative if and only if $\mathbf{X}$ is non-negative, where O is the matrix containing the eigenvectors of P^{-1}, Λ is the diagonal matrix containing the eigenvalues of P^{-1}, and V the diagonal matrix containing the variances of $\mathbf{X}$.

This property is particularly useful when dealing with economic data or other domains where variables are inherently non-negative.

For defining a multivariate Gini index, the appropriate choice of whitening matrix is critical. Based on the properties of scale stability and non-negativity preservation, Auricchio et al. (2024) recommend using[10]:

$$W_{\mu}^{ZCA} = O\Lambda^{-1/2}O^T V^{-1/2} \tag{3.22}$$

where O is the orthonormal matrix containing all the eigenvectors of P^{-1}, Λ is the diagonal matrix containing the eigenvalues of P^{-1}, and V is the diagonal matrix containing the variances of $\mathbf{X}$.

Furthermore, it satisfies the rising tide property, which states that adding a positive constant to all observations reduces inequality.

Finally, it also allows to preserve the range [0,1] when applied to non-negative data, consistent with the univariate Gini index, and enabling the construction of a multivariate Gini index that maintains desirable properties while accounting for the correlation structure among variables.

In the realm of machine learning and image processing, whitening transformations are often employed as preprocessing steps to decorrelate features and normalize their variances, which can improve the performance of subsequent models. The ZCA whitening, in particular, has the advantage of preserving the orientation of the data in the original space, making it useful for applications where interpretability is important [23].

Therefore, the ZCA whitening process when applied to the correlation matrix offers a mathematically sound method for transforming multivariate data into orthonormal components while preserving the key properties enlisted in section 3.2.1. This approach enables the construction of a robust measure of dispersion in higher dimensions that remains consistent with its univariate counterpart.

[10] For further explanations, the reader is invited to check Theorem 3 in "Extending the Gini Index to Higher Dimensions via Whitening Processes" (Auricchio et. al, 2024).

3.3 Computing the Multiclass Gini Multidimensional ROC curve

The first step to construct the Multiclass ROC curves is to get the predicted probabilites, we obtain them from models[11] and then apply the whitening transformation. For evaluation purposes, we utilize each model's probability estimation capability to obtain K-class probability vectors for each of the L test samples, resulting in the probability matrix X_{proba}.

It is important to note that for models which do not utilize the One-vs-Rest strategy, such as multiclass logistic regression, the predicted probabilities for each instance are constrained to sum to one across all classes.

Whitening the predicted probabilities serves to remove inter-class correlations and ensure that the associated weights of the Multidimensional Gini index w_i from Equation 3.8 properly reflect the underlying discriminability of each class, so we normalize the scale of probabilistic outputs.

The vector of weights $\{w_i\}$ provides valuable diagnostic information beyond its role in computing G_{agg}. Specifically, a large w_i indicates that the model places disproportionate predictive emphasis on class i. This may suggest:

- Greater inherent separability of class i.
- Model bias toward certain classes.
- Potential overfitting to class-specific features.

These insights can guide model refinement and highlight potential areas for improvement in multiclass classification systems.

The ZCA-correlation whitening transform, is therefore a technique designed to decorrelate features while preserving their original orientation in feature space. Unlike principal component analysis (PCA), which rotates the data into a new basis, ZCA whitening performs decorrelation while maintaining interpretability in the original feature space.

Whitening the predicted probabilities is a crucial step to ensure interpretable and robust multiclass evaluation. Without this step, diagnostic signals, such as disproportionately large weights for specific classes, could be distorted or confounded by underlying correlations or scale differences that originate from the probabilistic outputs of the classifier. This adjustment guarantees that subsequent analysis focuses on genuine predictive structure, rather than on artifacts introduced by the model's probabilistic representation.

In a nutshell, the ZCA-correlation whitening procedure comprises three sequential operations:

1. **Diagonalization**: The empirical correlation matrix $P = \text{corr}(X^{\text{std}})$ is decomposed into its eigenvalue and eigenvector components:

$$P = V \Lambda V^T \tag{3.23}$$

[11] Which may include logistic regression, random forests, or neural networks.

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_D)$ contains the eigenvalues and V contains the corresponding eigenvectors.

2. **Rescaling**: Each principal direction is rescaled by the inverse square root of its corresponding eigenvalue, effectively equalizing the variance along each principal component:

$$\Lambda^{-1/2} = \text{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2}, \ldots, \lambda_D^{-1/2}) \tag{3.24}$$

3. **Back Rotation**: The data is rotated back to the original feature space, yielding whitened data:

$$X^{\text{white}} = X^{\text{std}} W^T \tag{3.25}$$

where $W = V\Lambda^{-1/2}V^T$ is the whitening matrix.

The resultant whitened dataset exhibits two fundamental properties:

- Every feature maintains unit variance.
- All pairwise correlations between features are zero.

Hence, following the $ZCA - cor$ whitened transformation, the covariance matrix of the predicted probabilities will equal the identity matrix.

Theorem 3.1: For a ZCA-whitened matrix X^{white}, the covariance matrix $\Sigma(X^{\text{white}}) = I$, where I is the identity matrix.

Proof: Let $W = V\Lambda^{-1/2}V^T$ be our whitening matrix and X^{std} our standardized data with correlation matrix P. Then:

$$\begin{aligned}
\Sigma(X^{\text{white}}) &= \frac{1}{N}(X^{\text{white}})^T X^{\text{white}} \\
&= \frac{1}{N}(X^{\text{std}} W^T)^T (X^{\text{std}} W^T) \\
&= \frac{1}{N} W (X^{\text{std}})^T X^{\text{std}} W^T \\
&= WPW^T \\
&= V\Lambda^{-1/2}V^T \cdot V\Lambda V^T \cdot V\Lambda^{-1/2}V^T \\
&= V\Lambda^{-1/2} \cdot \Lambda \cdot \Lambda^{-1/2}V^T \\
&= V \cdot I \cdot V^T \\
&= VV^T \\
&= I
\end{aligned}$$

This mathematical proof confirms that ZCA whitening effectively decorrelates the features while preserving their original orientation in the feature space.

To ensure robust and unbiased model evaluation, the standardized data are partitioned into training and test sets using stratified sampling to maintain class distributions.

Returning to the calculation of the Multidimensional Gini for the purposes of constructing a Multiclass ROC curve, if W_{proba} represents the whitening matrix computed from the predicted probabilities matrix, which is a matrix of $L \times K$ dimensions[12], we apply:

$$X_{\text{proba}}^{\text{whitened}} = X_{\text{proba}} W_{\text{proba}}^{T} \tag{3.26}$$

Empirically, in order to construct the Multivariate Gini index, we will leverage the Gini Mean Difference (GMD) which serves as our foundational measure of dispersion and separation. It is used to compute each univariate Gini index for each class. For a univariate sample $z \in \mathbb{R}^n$, the GMD is defined as[37]:

$$\text{GMD}(z) = \frac{1}{n(n-1)} \sum_{1 \le a < b \le n} |z_a - z_b| \tag{3.27}$$

This formulation is equivalent to the expectation of absolute differences between independent identically distributed random variables:

$$G(X) = \mathbb{E}[|X_1 - X_2|] \tag{3.28}$$

where X_1 and X_2 are i.i.d. draws from the distribution of X.

Theorem 3.2: For binary classification with true class labels $y \in \{0, 1\}$ and predicted probabilities p, the relationship between the GMD of p and the Area Under the ROC Curve (AUC) is given by $G = 2 \cdot \text{AUC} - 1$.

Proof: Let p_0 be the vector of probabilities for samples with true label 0, and p_1 for samples with true label 1. Then:

$$\begin{aligned} \text{AUC} &= \Pr(p_1 > p_0) \\ &= \mathbb{E}[\mathbf{1}_{p_1 > p_0}] \end{aligned} \tag{3.29}$$

In the binary case, the AUC is the probability that a randomly selected positive sample receives a higher score than a randomly selected negative sample.

The indicator function $\mathbf{1}_{p_1 > p_0}$ is 1 when $p_1 > p_0$, and 0 otherwise. In other words, it counts all cases where a positive sample's score exceeds a negative sample's score.

For the Gini index:

[12] where L is the number of observations in the test set, and K is the number of classes.

$$
\begin{aligned}
G &= \frac{\mathbb{E}[|p_1 - p_0|]}{\mathbb{E}[p_1] + \mathbb{E}[p_0]} \\
&= \mathbb{E}[\mathbf{1}_{p_1 > p_0}(p_1 - p_0) + \mathbf{1}_{p_0 > p_1}(p_0 - p_1)] \\
&= \mathbb{E}[\mathbf{1}_{p_1 > p_0}] - \mathbb{E}[\mathbf{1}_{p_0 > p_1}] \\
&= \text{AUC} - (1 - \text{AUC}) \\
&= 2 \cdot \text{AUC} - 1
\end{aligned} \tag{3.30}
$$

This completes the demonstration that the Gini coefficient is a linear transformation of the AUC, and highlights the role of the indicator function in counting favorable pairwise comparisons.

Therefore, $\text{AUC} = \frac{G+1}{2}$, establishing the direct relationship between the Gini index and AUC.

Multiclass probability outputs X_{proba} inherently satisfy the constraint $\sum_{i=1}^{K} y_{l,i} = 1$ for each sample l, which introduces structural correlations among the K dimensions. To handle these correlations, we apply ZCA-correlation whitening to the probability matrix, as done before:

$$
\widetilde{X} = X_{\text{proba}} W^T \tag{3.31}
$$

$$
W = P^{-1/2} V^{-1/2} \tag{3.32}
$$

where P is the empirical correlation matrix of X_{proba} and V is its diagonal variance matrix. This transformation ensures that each column $\widetilde{X}[:, i]$ is uncorrelated with others and has unit variance, enabling independent treatment of each class dimension in subsequent analyses.

The computation of the multidimensional Gini index proceeds through three sequential steps:

1. **Per-class Gini Calculation**: For each class $i = 1, \ldots, K$, compute:

$$
G_i = \text{GMD}(\widetilde{X}[:, I]) \tag{3.33}
$$

2. **Class Weight Assignment**: For each class, we assign a weight proportional to the absolute mean of its whitened scores:

$$
w_i = \frac{|\bar{\tilde{x}}_i|}{\sum_{j=1}^{K} |\bar{\tilde{x}}_j|} \tag{3.34}
$$

where $\bar{\tilde{x}}_i$ denotes the sample mean of $\widetilde{X}[:, i]$. This weighting scheme ensures that classes with larger average whitened scores, indicating greater separability, contribute more substantially to the aggregate measure.

3. **Aggregate Gini Computation**: The Multidimensional Gini index is calculated as the convex combination:

$$
G_{\text{agg}} = \sum_{i=1}^{K} w_i G_i, \quad G_{\text{agg}} \in [0, 1] \tag{3.35}
$$

Recall that each G_i comes from the whitened predicted probabilities from a classifier trained and tested, this allows us to construct the Multidimensional Gini index, which is equivalent to the one in formula. Drawing from the established relationship in binary classification where $G = 2 \cdot \text{AUC} - 1$, we define an aggregate multiclass AUC via:

$$\text{AUC}_{\text{agg}} = \frac{G_{\text{agg}} + 1}{2} \tag{3.36}$$

This formulation yields a single, interpretable performance statistic that is directly comparable to the standard ROC-AUC while respecting the full multiclass structure. The whitening procedure ensures that Gini computations on each dimension truly reflect class separability, uncontaminated by spurious inter-class dependencies.

Having obtained the chapter's main result, which was to showcase how to calculate the aggregate AUC via the Multidimensional Gini index, we now turn to its geometric interpretation. In doing so, we pave the way for a more intuitive grasp of how the measure captures class separability and for having model-agnostic explanations about model behavior, which is much needed in financial applications.

As it has been mentioned, this procedure allows to remove inter-variable correlations in X_{proba}, we apply a ZCA-whitening transformation to its sample correlation matrix P. First, we compute the spectral decomposition as in equation 3.23, where V is an orthonormal matrix whose columns are the eigenvectors of P, and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ contains the corresponding positive eigenvalues. The ZCA-whitening matrix is the one in equation 3.22.

Which performs three steps in succession:

1. Rotate into the principal-axis frame via $O^\top$;
2. Scale each principal component by $1/\sqrt{\lambda_i}$ (the entries of $\Lambda^{-1/2}$);
3. Rotate back to the original coordinate system via O.

Because all $\lambda_i > 0$, P is invertible[13], ensuring that $W^{\text{ZCA-corr}}$ exists. Finally, setting

$$X^* = W_\mu^{ZCA-cor} X_{\text{proba}} \tag{3.37}$$

yields $\text{cov}(X^*) = I$, so all off-diagonal covariances vanish and each whitened variable has unit variance. In practice, this justifies treating each dimension independently, for instance, when computing univariate Gini indices on X^*.

Orthonormality means that a set of vectors all point in mutually "right-angle" directions and each one has length one. Concretely, if we take any two different vectors from the set and compute their dot-product, we get zero, which is the orthogonal part, and if we take a vector and dot it with itself we get one, which is the "normal" part. This combination is convenient because orthonormal bases let us measure lengths

[13] Equivalently, $\det P = \prod_i \lambda_i > 0$, so P has full rank and no pair of original variables is perfectly collinear.

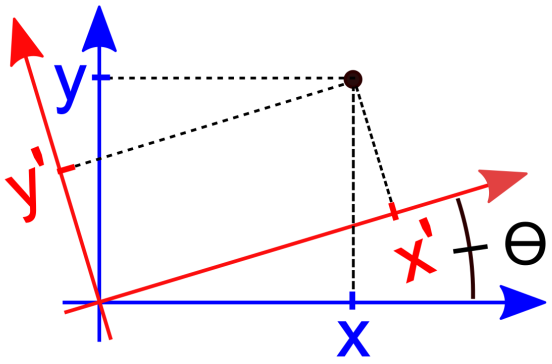

Figure 3.1: Rotation of axes in two dimensions[8]

and angles very simply: projecting onto one basis vector does not interfere with any of the others, and we can recover the original vector by adding up its independent projections.

When we talk about a "rotation" in this context, we mean spinning the coordinate axes around, just like rotating the x- and y-axes in the plane, without stretching or squeezing anything, as can be seen in Figure 3.1. A rotation preserves both distances and angles, which guarantees that the statistical relationships (variances and covariances) in the data are preserved except for how they are "viewed" in the new axes.

We need rotations in whitening because they let us reorient our X_{proba} so that its new axes line up exactly with the directions of greatest and least variability; after that, scaling each axis to unit length completely removes correlation.

The directions we rotate into are called the principal axes. Each principal axis is an eigenvector of the correlation matrix of the standardized data; mathematically, it is the direction along which the data varies the most.[14] Once we have identified those axes, the coordinates of our X_{proba} measured along each axis are the principal components. In practical terms, a principal component is just "how far we have to travel along that principal axis to reach our data point." Because different principal axes are uncorrelated by construction, the principal components themselves are uncorrelated random variables, which makes downstream tasks, like computing univariate statistics or visualizing multi-dimensional data, far simpler.

After decorrelating and rescaling the data along the principal axes, we typically rotate back to the original coordinate system by multiplying with the matrix O, whose columns are the original eigenvectors. This step is essential for ZCA whitening, as it ensures that the transformed data remains uncorrelated and possesses unit variance, while its representation is expressed in terms of the original features rather than abstract principal directions. In other words, rather than leaving the data in the often unintuitive axes of principal components, we restore its orientation so each new vector can be directly interpreted and compared against the original variables. This process preserves geometric and statistical relationships with the raw data, making the whitened variables as close as possible to the original ones, but now guaranteed to be orthogonal and standardized. The combination of rotation and rescaling followed by re-rotation maintains both interpretability and mathematical rigor, allowing the analysis to benefit from the properties of orthogonalization while keeping the context of each feature intact.

[14] Or next most, and so on.

Part II

Case study: Credit ratings multiclass classification task

Chapter 4

Applying the constructed Multiclass ROC curve for Credit and ESG risk management purposes

Chapter Contents

4.1 The Dataset

This study employs a comprehensive panel dataset encompassing financial, credit risk, and Environmental, Social, and Governance (ESG) metrics for 1,724 Small and Medium-sized Enterprises (SMEs) operating within the Italian market. The dataset spans three consecutive years (2020-2022) and comprises 38 variables.

The dataset is composed by two main components, the financial one and the ESG one. The financial dimension constitutes the most substantial component of the dataset, encompassing 23 variables derived from Balance Sheet and Income Statement data. These accounting metrics, expressed in thousands of euros, provide comprehensive coverage of financial health and company's solvency. The financial variables capture both stock measures such as total assets, shareholders' funds, and current liabilities, and flow measures including operating revenue, EBITDA, and net income throughout the three-year observation period.

This temporal dimension provides valuable insights into how historical financial performance influences current credit assessments.

Then there is the credit assessment component which incorporates three temporal variables representing annual credit ratings from 2020 to 2022. These ratings adhere to the conventional credit rating scale employed by major rating agencies, comprising ten distinct classes arranged in descending order of creditworthiness: AAA, AA, A, BBB, BB, B, CCC, CC, C, and D.

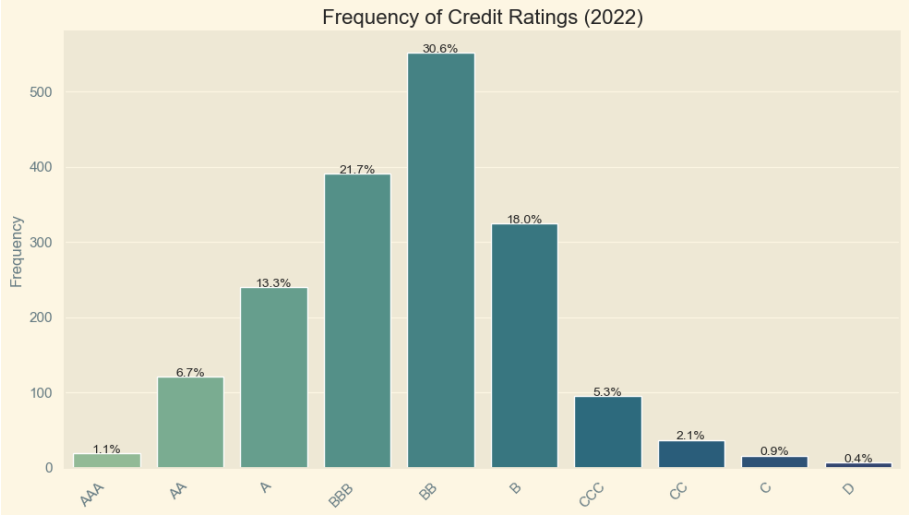

Figure 4.1: 2022 Credit-Rating Frequency

This rating system reflects an ordinal risk assessment framework, where AAA-rated companies demonstrate the highest level of financial solvency and lowest default probability, while D-rated entities indicate companies experiencing default or near-default conditions.

The 2022 credit rating variable serves as the dependent variable for the classification task. As shown in Figure 4.1, the frequency distribution reveals a concentration of observations within intermediate rating categories, reflecting the typical landscape of Italian SMEs. Notably, the distribution demonstrates significant class imbalance, with very few instances at the extremes (AAA, CC, C, and D ratings). This skewed pattern accurately mirrors the typical risk profile for established SMEs, where both exceptionally high-quality credits and default events are uncommon.

Returning to the dataset, in order to ensure model robustness and in spite of categories with relative low frequencies, we decided to exclude synthetic data generation techniques such as SMOTE (Synthetic Minority Oversampling Technique), reflecting the stringent regulatory constraints and interpretability requirements inherent in financial risk modeling. Given the critical role of credit risk models in maintaining financial system stability and the rigorous explainability requirements imposed by financial regulators, preserving the authentic distributional characteristics of the data takes precedence over achieving artificial class balance.

Then, we can find the ESG framework, which comprises eight variables that capture both ordinal and continuous measures of sustainability and reputational performance. The system employs a dual methodological approach: ordinal ratings ranging from S1 to S7 [1] and corresponding numerical scores for each of the three ESG pillars, as well as a composite ESG score for 2022.

The ESG rating frequency distribution provides valuable insights into the adoption, maturity and sustainable practices of Italian SMEs, with the highest concentration of ratings being in between the S2 and S4 classes, with S3 being the one with the highest frequency. The structural composition of ESG ratings and their corresponding scores is comprehensively illustrated in Figure 4.2, which demonstrates both the categorical distribution and the continuous score patterns across the dataset.

[1] S1 represents superior ESG performance.

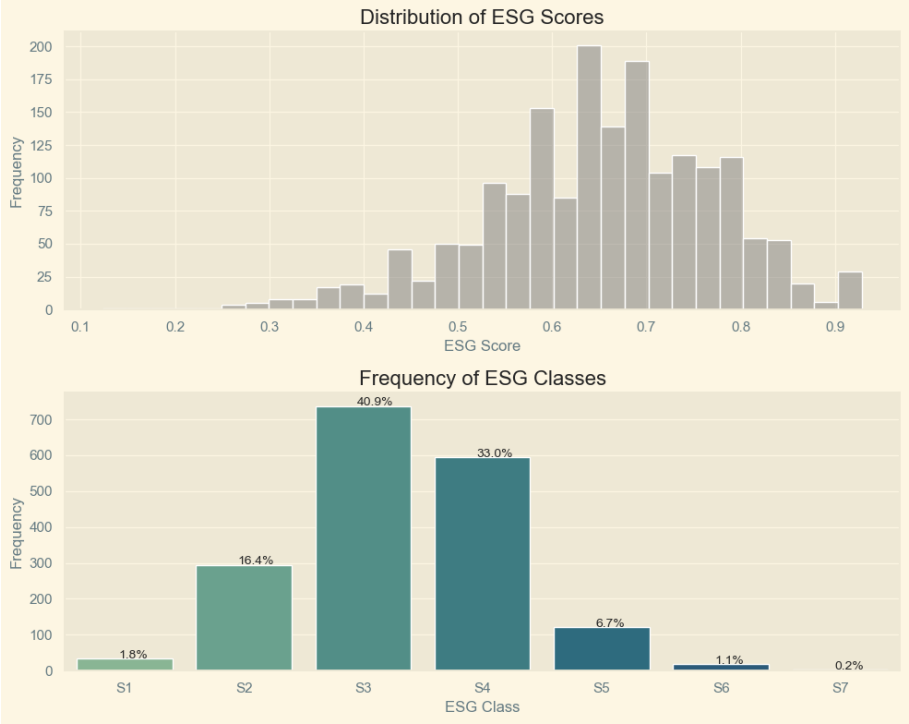

Figure 4.2: ESG class and score frequency and distribution

The dataset also incorporates geographic and sectoral classification variables, including company identifiers, sectoral assignments, and regional headquarters locations. To ensure statistical robustness while maintaining analytical interpretability, both geographic and sectoral variables underwent a consolidation process.

The original 20 Italian regions were consolidated into five macro-regions based on established geographic and economic criteria. The regional consolidation follows the framework presented in Table 4.1:

Table 4.1: Regional Grouping of Italian Companies

Macro Region	Regions
Islands	Sicilia, Sardegna
South	Abruzzo, Basilicata, Calabria, Campania, Molise, Puglia
Northeast	Emilia-Romagna, Friuli-Venezia Giulia, Trentino-Alto Adige, Veneto
Northwest	Liguria, Lombardia, Piemonte, Valle D'Aosta
Center	Lazio, Marche, Toscana, Umbria

The sectoral classification underwent a similar consolidation process, where numerous industry categories were aggregated into five economically coherent macrosectors. This approach ensures that each macrosector contains companies with similar business models, comparable risk profiles, and analogous ESG challenges. The sectoral consolidation framework is detailed in Table 4.2:

As can be seen in Figure 4.3, the sectoral distribution analysis reveals manufacturing as the predominant sector within the dataset, comprising 41.3% of the total sample, followed by the consumer sector at 23.3%. This distribution accurately reflects the industrial composition of the Italian SME landscape,

Table 4.2: Sectoral Grouping into Five Macrosectors

Macro Sector	Sectors
Consumer	Agriculture, Horticulture & Livestock; Retail; Travel, Personal & Leisure; Printing & Publishing; Wholesale
Financials	Banking, Insurance & Financial Services; Property Services; Business Services
Health and Utilities	Public Administration, Education, Health Social Services; Utilities; Transport, Freight & Storage
Manufacturing	Chemicals, Petroleum, Rubber & Plastic; Construction; Industrial, Electric & Electronic Machinery; Leather, Stone, Clay & Glass products; Metals & Metal Products; Mining & Extraction; Textiles & Clothing Manufacturing; Wood, Furniture & Paper Manufacturing; Miscellaneous Manufacturing; Transport Manufacturing; Waste Management & Treatment; Food & Tobacco Manufacturing
Tech.Com	Biotechnology and Life Sciences; Communications; Computer Hardware; Computer Software; Media & Broadcasting

where manufacturing traditionally represents the economic backbone of the country's small and medium enterprise sector.

On the other hand, from a geographical perspective, the Northwest region demonstrates the highest representation at 33.4% of the sample, which aligns with the concentration of Italy's manufacturing activities in the Lombardy region. The South region constitutes the second-largest segment at 22.5%. This is illustrated in Figure 4.4.

Moreover, a data quality assessment was also carried out and revealed varying degrees of completeness across variables, particularly within financial metrics containing "n.a." entries and ESG ratings exhibiting heterogeneous availability patterns. The missing data treatment involved exclusion of observations with incomplete critical variables, resulting in a final analytical sample of 1,724 observations from 1,896.

4.1.1 Multicollinearity Challenge

A challenge while working with financial data is the predominance of balance sheet and income statement variables within the dataset. This feature introduces significant multicollinearity challenges, as numerous accounting figures are intrinsically related through fundamental accounting identities. For instance, total assets and current assets are mathematically linked through basic accounting principles.

The temporal structure of the dataset amplifies these multicollinearity concerns, as consecutive years' financial figures exhibit high correlation indices due to business continuity principles. This phenomenon is illustrated in the correlation matrix presented in Figure 4.5.

The correlation matrix reveals several relationships within the dataset. Financial variables demonstrate high "autocorrelation" across temporal periods, with variables such as shareholders' funds and total assets showing strong correlations between 2020 and 2021 values.

Additionally, contemporaneous financial figures exhibit substantial intercorrelation, exemplified by the

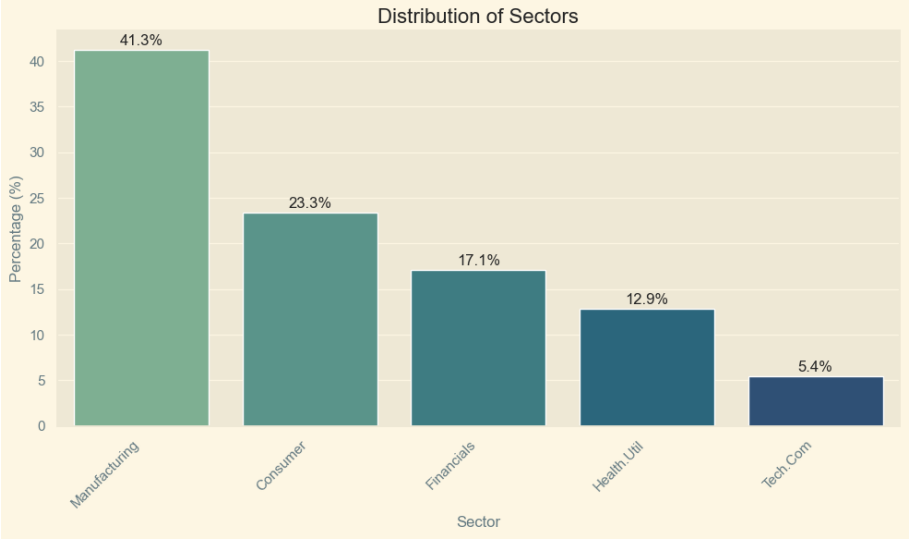

Figure 4.3: Sector distribution

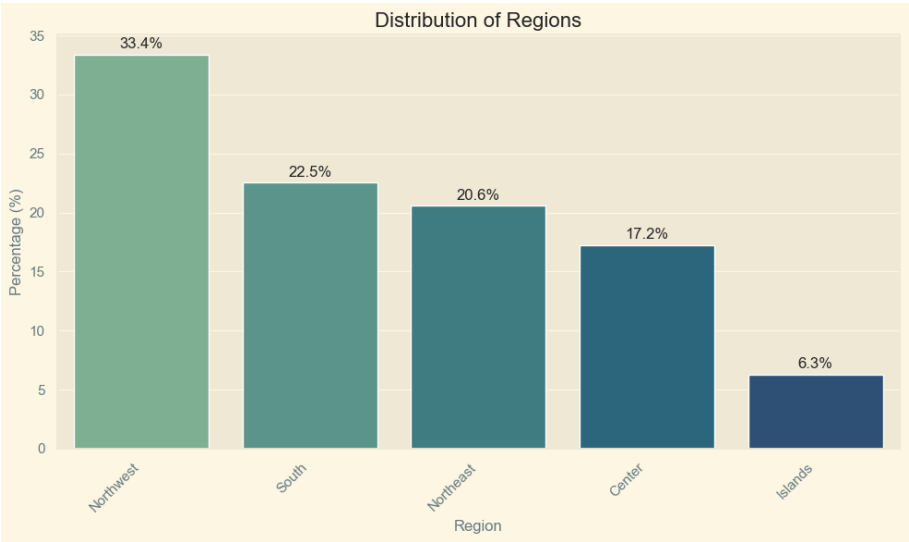

Figure 4.4: Region distribution

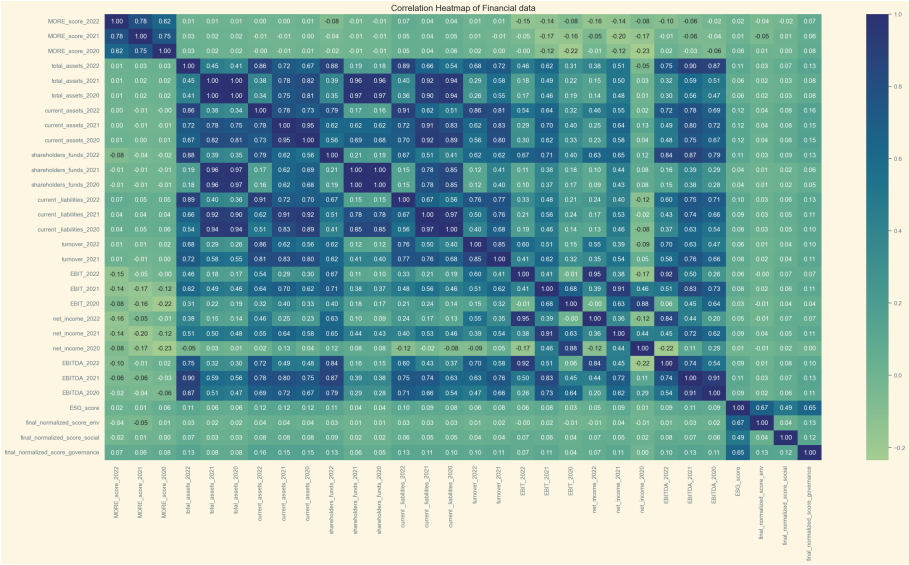

Figure 4.5: Correlation Matrix

0.91 and 0.92 correlation indices between current liabilities and current assets in 2022, and between EBIT 2022 and net income 2022.

Another relevant point we can extract from the correlation matrix is that among the ESG dimensions, the Governance pillar demonstrates the strongest correlation patterns with financial variables, even if at relatively modest levels. Governance scores show the highest correlations with variables such as current assets across all three years [2], shareholders' funds 2022, EBITDA 2021, current liabilities 2022, and total assets 2022. These patterns suggest that governance practices may have measurable relationships with firm financial structure and performance. And it makes sense since more transparent companies with better governance structures and practices, can achieve stronger financial results.

The correlation between the Governance pillar and financial variables has been empirically documented in recent academic research. Ahelegbey and Giudici (2024) in their paper "Multidimensional Inequality Metrics for Sustainable Business Development" provide evidence supporting this relationship, demonstrating how governance metrics correlate with various financial performance indicators in corporate sustainability assessments.

The dataset has 30 features, many of them with the potential of being redundant, and we want to keep only those that have any predicting power. Therefore, multicollinearity will be addressed through dimensionality reduction via a developed algorithm that dynamically selects between sequetial backward and forward elimination procedures. The algorithm optimizes the selection criterion based on the modeling technique employed: for traditional classification models, it minimizes the selected loss function, while for tree-based models, it minimizes Gini impurity.

It is crucial to distinguish between Gini impurity and the Gini index, as these are fundamentally different concepts. Gini impurity serves as a node splitting criterion in decision tree algorithms, measuring the heterogeneity of class distributions within tree nodes. The metric quantifies the probability of incorrectly classifying a randomly selected instance if it were labeled according to the class distribution at that

[2] correlation indicesof 0.16, 0.15, and 0.15 respectively

node. Mathematically, Gini impurity seeks to create nodes with homogeneous class distributions, ideally achieving pure nodes where all instances belong to the same class. [33]

To adress collinearity issues we could also apply ZCA whitening following an initial standardization of the design matrix[3] X. Each feature is first transformed to have zero mean and unit variance, ensuring that subsequent operations focus on the true correlation structure between variables rather than differences in scale or mean. Once standardized, ZCA whitening utilizes the correlation matrix to decorrelate features, transforming the data so that the covariance matrix of the whitened variables becomes the identity. This step directly mitigates collinearity by producing new features that are both uncorrelated and equally scaled.

In the present study, a misclassification occurs when the estimator assigns a company to the BBB rating category, while it actually belongs to the CC category. This exemplifies one of the central challenges of multiclass classification: typical loss functions do not capture the varying severity of misclassification errors. Consequently, there is a clear need for performance metrics specifically designed for multiclass tasks. The methodology proposed here addresses this gap, making it particularly relevant for applications in regulated fields such as finance and medicine, where accurate and nuanced classification is essential. Unlike conventional metrics such as macro and micro averaging, which primarily consider observation frequency, the proposed approach focuses on the discriminative power[4] of each class within the overall classification problem. This offers a meaningful alternative to the standard metrics commonly employed in multiclass contexts.

4.2 Model Development Framework

The modeling approach employs a nested stratified cross-validation with randomized search, which employs a train-test split with a 70-30 ratio, ensuring that the training set contains sufficient observations for robust parameter estimation while reserving an adequate holdout sample for out-of-sample validation, in this way avoiding biases.

A nested approach means that there is an outer loop which has five folds, so we will split our dataset five times without replacement, into a training and test set, and then we have an inner loop with three folds that will take the training fold, and we will further divide it into a smaller training and a validation fold.

In the inner fold we will tune the hyperparameters, while in the outer loop, we will train the models with optimal parameters. With cross-validation alone, we are estimating the generalization of our model in the test set, but with nested cross-validation we can also try to optimize the different hyperparameters.

Moreover, the stratified approach ensures that each credit rating class maintains proportional representation in both training and testing sets, preventing bias that could arise from random sampling in the presence of class imbalance.

Meanwhile, randomized grid search means that we will not have an exhaustive search of our hyperparameters, which are the parameters not learned during training.

[3] The matrix containing the explanatory variables.
[4] Which is, in practice, indirectly influenced by class frequency within the sample

It is also worth highlighting that due to the extreme class imbalance in this dataset, XGBoost classifier training encountered convergence difficulties despite parameter tuning attempts.

Prior to model estimation, all continuous variables underwent standardization [5] to ensure that variables measured in different units contribute equally to the modeling process. Dummy variables representing regional and sectoral affiliations were excluded from standardization, as their binary nature already ensures comparable scales. Moreover, to avoid redundancy, variables such as ESG score were not considered for the study since each of the pillar scores are already included.

Standardization is particularly crucial in this context because financial variables span several orders of magnitude[6], and ESG scores operate on entirely different scales. Without standardization, variables with larger numerical ranges would dominate the model estimation process, potentially obscuring the contributions of equally important predictors measured on smaller scales.

This whole approach ensures that the subsequent machine learning models can effectively identify patterns across all variable types without being biased, thereby enhancing both model performance and interpretability. However, given the imbalanced nature of the dataset, overfitting is a relevant challenge for modelling purposes.

Finally, classes C and D were consolidated into a single category, C&D, due to their extremely sparse representation in the dataset. The limited sample sizes for both classes prevented models from establishing reliable decision boundaries, while the conceptual similarity between these categories, both representing various stages of default or near-default conditions, justified their aggregation from a credit risk perspective. This consolidation reduces the classification framework from ten to nine distinct rating classes, ensuring sufficient sample representation for robust model training while maintaining meaningful distinctions between default and non-default categories.

4.3 Implementation of the Multiclass ROC Curve

Following the data preprocessing and model training phases, we can compute the AUC and plot the multiclass ROC curve. The fundamental challenge in multiclass ROC curve construction lies in aggregating information across multiple classes while preserving the interpretability and statistical properties that make ROC curves valuable for model evaluation.

The Multidimensional Gini approach addresses this challenge by providing a weighting scheme that reflects the relative importance of each class in the overall classification problem.

The initial step in constructing a multiclass ROC curve involves extracting predicted probabilities from the selected estimator and then applying ZCA whitening to preserve scale invariance. This whitening step is crucial because if one class's predicted probabilities exhibit very low variance, standard macro/micro averaging approaches can over-penalize or ignore that class whitening neutralizes this bias. This study employs Scikit-Learn's `predict_proba` built-in function, which returns probability estimates indicating the likelihood that each observation belongs to each class. The reliance on probability estimates is

[5] z-score normalization.
[6] From thousands to millions of euros.

fundamental to this approach, as ROC curves inherently depend on the ranking of predictions rather than hard classifications.

It is important to note that this methodology requires estimators that possess the `predict_proba` method or any equivalent function capable of generating predicted probabilities. This requirement excludes certain algorithms that only provide hard classifications, thereby limiting the scope of applicable models but ensuring that the resulting ROC curves maintain their probabilistic interpretation.

The complete implementation of all functions discussed in this section is available on the author's GitHub repository[7].

4.3.1 Numerical Stabilization for Whitening Predicted Probabilities

In order to whiten predicted probabilities it is paramount to apply a stabilization to them since **unstable whitening can transform reasonable probability values (like 0.8) into unusable extreme values (like 850,000)**, making them meaningless for any practical classification task.

While ZCA whitening has been successfully applied to multidimensional Gini index computation, implementations encountered numerical instabilities when applied to classifier-generated probability matrices, particularly those produced by ensemble methods with softmax outputs. These instabilities manifest in several distinct ways that compromise the reliability and interpretability of the results.

Without stabilization, **whitened probabilities explode to millions** because:

1. **Near-zero eigenvalues**: When classifiers produce high-confidence predictions, the predicted probabilities covariance matrix becomes near-singular with eigenvalues approaching zero. Taking the inverse square root ($\lambda^{-1/2}$) creates enormous scaling factors. For example, if an eigenvalue is 1×10^{-10}, then $1/\sqrt{1 \times 10^{-10}} \approx 31{,}622$.
2. **Near-zero variances**: Classes with consistently confident predictions have variance approaching zero. Again, taking the inverse square root creates massive scaling factors.
3. **Compound multiplication**: The ZCA whitening formula 3.22 multiplies these large scaling factors together, amplifying the explosion.

When we apply this unstable whitening matrix to probability values in $[0, 1]$, we get transformed values in the **millions range**, making them:

- Unusable for threshold-based analysis.
- Numerically unstable for further computations.
- Meaningless for ROC curves and performance metrics.

The first and most fundamental challenge arises from the mathematical properties of predicted probabilities matrices themselves. Unlike arbitrary data matrices, predicted probabilities exhibit constrained

[7] https://github.com/rosacrg

ranges and often display strong correlations between classes due to the underlying softmax normalization constraint.

When a machine learning classifier makes predictions, it can output results in two ways:

- **Hard classifications**: Direct class labels like "AAA" or "CC".
- **Soft classifications**: Probability estimates like $[0.2, 0.8]$ meaning "20%AAA, 80% CC".

Hence the need of **Softmax normalization**, which is the mathematical process that ensures these probabilities sum to 1.0 across all classes:

Another challenge encountered was when classifiers achieved high confidence on certain predictions, making probability vectors become nearly deterministic, with one class approaching probability 1.0 while others approach 0.0. This creates near-singular covariance matrices where some eigenvalues approach zero, leading to catastrophic numerical failure during the eigendecomposition required for ZCA whitening.

It is relevant to recall that a **singular matrix** is one where some rows/columns are nearly identical or one dimension provides no new information. If a classifier always predicts the same pattern (high confidence for class A, low for class B), the correlation between classes approaches perfect negative correlation (-1.0). Mathematically, this creates **eigenvalues approaching to zero**.

This arose the need to create the `whitening_process_stable` function, which takes as parameters the correlation matrix, the diagonal matrix containing the variances of the predicted probabilities, and an optional regularization parameter (defaulting to 1×10^{-6}), addressing these challenges through a comprehensive regularization strategy that maintains the theoretical properties of ZCA whitening while ensuring numerical robustness.

Our stabilization prevents the explosion through four mechanisms:

1. **Eigenvalue regularization**: Floors eigenvalues at 1×10^{-6}.
2. **Variance regularization**: Floors variances at 1×10^{-6}.
3. **Maximum scaling cap**: Limits scaling factors to 100.0.
4. **Final clipping**: Ensures the whitening matrix stays bounded.

The implementation begins with eigenvalue regularization, where any eigenvalues below the specified regularization threshold are elevated to that minimum value. This choice balances numerical stability with minimal bias introduction, following established practice in covariance matrix regularization [25].

This prevents the computation of infinite scaling factors that would otherwise result from taking the inverse square root of near-zero eigenvalues: the regularization effectively introduces a small bias toward the identity transformation when the data approaches singularity, which represents mathematically sound behavior for near-degenerate cases.

```python
def whitening_process_stable(corr, var, regularization=1e-6):
    # Handle problematic values
    corr = np.nan_to_num(corr, nan=0.0, posinf=1.0, neginf=0.0)
    var = np.nan_to_num(var, nan=1.0, posinf=1.0, neginf=1e-12)
    # STABILIZATION 1: Eigenvalue regularization
    eigvals, eigvecs = np.linalg.eigh(corr)
    eigvals = np.maximum(eigvals, regularization)  # Floor at 1e-6
    var_reg = np.maximum(var, regularization)

    # STABILIZATION 2: Maximum scaling limits
    max_scaling = 100.0
    eigvals_scaled = np.minimum(eigvals**(-1/2), max_scaling)
    var_scaled = np.minimum(var_reg**(-1/2), max_scaling)

    # STABILIZATION 3: Apply ZCA formula with bounded values
    D = np.diag(eigvals_scaled)
    D_var = np.diag(var_scaled)
    W = eigvecs @ D @ eigvecs.T @ D_var

    # STABILIZATION 4: Final safety clipping
    W = np.clip(W, -max_scaling, max_scaling)

    return W
```

Beyond eigenvalue regularization, the function implements variance regularization to address the common scenario where certain classes exhibit extremely low prediction variance. This situation frequently occurs when classifiers demonstrate strong separation for specific classes, resulting in consistently high confidence predictions:

Without regularization, the diagonal variance matrix would contain values approaching zero, again leading to explosive scaling factors during the whitening transformation. The variance floor ensures that all classes receive reasonable scaling treatment, preventing any single class from dominating the whitened space through numerical artifacts.

The introduction of maximum scaling limits represents another crucial stabilization mechanism. Even with eigenvalue and variance regularization, the combination of multiple small values can still produce scaling factors that, while finite, are sufficiently large to create numerical overflow or precision loss in subsequent computations. The maximum scaling parameter is hardcoded to 100.0 to cap the maximum allowable scaling factor, ensuring that the whitening transformation remains within numerically stable ranges while preserving the essential decorrelation properties

Furthermore, the `whitening_predicted_proba_stable` function extends the stabilization approach to the transformation application phase and implements an additional preprocessing step to enhance numerical stability. Before computing the whitening matrix, a small amount of Gaussian noise (with scale 1×10^{-8}) is added to both training and test probability matrices:

This noise injection serves a dual purpose: breaking perfect ties that commonly occur at probability extremes (0.0 and 1.0) and eliminating perfect correlations that can arise when classifiers produce identical probability patterns across multiple samples. While this introduces minimal perturbation to the original probabilities, it significantly improves the conditioning of the covariance matrix computation.

After applying the whitening matrix, the transformed probability values can span arbitrary ranges, often

```python
def whitening_predicted_proba_stable(X_test, X_train, est, y_test, multilabel=None):
    # Get probabilities
    y_proba_test = est.predict_proba(X_test)
    y_proba_train = est.predict_proba(X_train)
    # Add stabilizing noise
    noise_scale = 1e-8
    y_proba_train_array += np.random.normal(0, noise_scale,
                                            y_proba_train_array.shape)
    y_proba_test_array += np.random.normal(0, noise_scale,
                                           y_proba_test_array.shape)

    # Without noise: [0.0, 0.0, 1.0, 1.0] - many ties
    # With noise: [0.0000001, 0.0000003, 0.9999998, 0.9999999] - ties broken
```

extending far beyond the original $[0, 1]$ interval. These extreme values render threshold-based metrics meaningless, as decision thresholds calibrated for probability-like ranges become inappropriate for the whitened space.

The rank-based rescaling mechanism embedded within `whitening_predicted_proba_stable` provides a solution to this interpretability challenge. **Pairwise ordering** means: if sample A originally had a higher probability than sample B, this relationship should be preserved after transformation. This is crucial because **ROC curves depend entirely on ranking**, they measure, in the binary case, how well a model ranks positive examples above negative ones.

By computing the rank of each whitened value across the entire transformed matrix and rescaling these ranks to the $[0, 1]$ interval, the function preserves all pairwise orderings essential for ROC curve analysis while restoring threshold interpretability:

This transformation is mathematically equivalent to applying the empirical cumulative distribution function of the whitened values, creating a uniform distribution on $[0, 1]$ that maintains the complete ranking structure required for area-under-curve calculations.

The rank-based rescaling approach offers several theoretical advantages over alternative normalization strategies. Unlike min-max scaling, which is sensitive to outliers, or z-score normalization, which assumes specific distributional properties, rank-based transformation is **monotonic invariant** and robust against extreme values.

A transformation is **monotonic invariant** if it preserves order relationships regardless of how the data is stretched or compressed. This is essential because:

- ROC curves only care about ranking, not absolute values.
- Different classifiers produce different probability scales.
- We need robustness across diverse model types.

```python
# Original probabilities: A=0.7, B=0.3 (A > B)
# After unstable whitening: A=234, B=891 (B > A) - WRONG!
# After stable whitening + ranking: A=0.8, B=0.2 (A > B) - CORRECT!
```

```
    # Compute stable whitening matrix from training predicted probabilities only
    var_train = np.diag(np.cov(y_proba_train_array, rowvar=False))
    corr_train = np.corrcoef(y_proba_train_array, rowvar=False)
    W = whitening_process_stable(corr_train, var_train)

    # Apply to both sets and rescale
    whitened_test_proba = (W @ y_proba_test_array.T).T
    whitened_train_proba = (W @ y_proba_train_array.T).T

    # Rank-based rescaling back to [0,1]
    from scipy.stats import rankdata
    whitened_test_proba = rankdata(whitened_test_proba.ravel()).reshape(
        whitened_test_proba.shape) / len(whitened_test_proba.ravel())
    whitened_train_proba = rankdata(whitened_train_proba.ravel()).reshape(
        whitened_train_proba.shape) / len(whitened_train_proba.ravel())

    return whitened_test_proba, whitened_train_proba
```

This robustness is particularly valuable when dealing with prediction matrices from diverse classifier types, each potentially exhibiting different distributional characteristics in their probability outputs.

The tie-handling mechanism within the rank rescaling process deserves particular attention, as probability matrices often contain numerous tied values, especially at the extremes (0.0 and 1.0). The implementation employs average ranking for tied values, ensuring that the transformation remains smooth and that no artificial discontinuities are introduced into the whitened space. This careful treatment of ties prevents the creation of artificial threshold effects that could bias subsequent ROC calculations. The preprocessing noise injection works together with this tie-handling approach by reducing the frequency of perfect ties before the whitening process begins.

The combined effect of these stabilization mechanisms ensures that the whitening process achieves its theoretical objectives, decorrelation and equal variance scaling, while remaining numerically robust and producing interpretable outputs suitable for threshold-based analysis. The resulting whitened probability matrices maintain their essential ranking properties while exhibiting the covariance structure required for meaningful Gini index computation, enabling the reliable calculation of multidimensional Gini indices even in challenging numerical scenarios that would cause standard implementations to fail catastrophically.

The **rank-based rescaling** brings everything back to $[0, 1]$ while preserving the essential ordering properties needed for ROC analysis. This comprehensive approach to numerical stabilization represents a crucial methodological advancement that bridges the gap between theoretical mathematical methodology and practical computational reliability, ensuring that the multidimensional Gini framework remains applicable across the diverse landscape of multiclass classification scenarios encountered in real-world applications.

4.3.2 Multiclass ROC curve visualization

The construction of our multiclass ROC curve rests on a solid theoretical foundation: the multidimensional Gini index from Equation 3.8. Think of this index as a sophisticated way to measure how well our model separates different classes, similar to how the traditional Gini coefficient measures inequality in

```python
def multidim_gini(whitened_data):
    # Compute the whitened mean for each class
    m_star = np.mean(whitened_data, axis=0)
    abs_means = np.abs(m_star)
    # Calculate Gini for each class
    gini_components = np.zeros(n_features)
    for i in range(n_features):
        gini_components[i] = gmd(whitened_data[:, i])

    # Calculate weights based on class separability
    weights = abs_means / np.sum(abs_means)

    # Compute multidimensional Gini
    multidimensional_gini = np.sum(weights * gini_components)

    return multidimensional_gini, weights
```

economics.

From a machine learning perspective, the Gini index serves as a measure of model prediction dispersion. Gini indices approaching zero indicate that all predictions have similar probabilities, suggesting poor discriminative capability and uniform prediction patterns. Conversely, indices approaching unity indicate perfect class separation, representing ideal classifier performance with superior ability to distinguish between classes and establish effective decision boundaries. The multidimensional Gini index provides a unified metric for comprehensive multiclass performance assessment.

Our implementation computes not just the overall multidimensional Gini index, but also individual Gini weights for each class (Equation 3.9). These weights tell us something intuitive: classes that are easier to distinguish get more influence in our final ROC curve.

Classes with higher mean predicted probabilities receive greater weight in the final aggregation, which intuitively reflects their greater impact on the overall classification performance. This adaptive weighting mechanism ensures that the resulting multiclass ROC curve appropriately represents the relative contribution of each class to the overall model performance and separability.

This makes intuitive sense: if our model consistently assigns high probabilities to a particular class, that class likely has strong distinguishing features and should influence our overall performance assessment more heavily.

After computing the Gini index, we calculate the theoretical AUC using Equation 3.5. This gives us a single number that summarizes how well our multiclass classifier performs across all classes, weighted by their relative importance.

Detailed Implementation: Per-Class ROC Construction and Aggregation

The practical implementation of our multiclass ROC methodology requires careful handling of numerical edge cases. Our approach systematically addresses these challenges through a robust pipeline that transforms the multiclass problem into manageable binary subproblems before aggregating the results.

Individual class ROC curves are generated using Scikit-Learn's built-in `roc_curve` function, which computes the ROC curve by calculating the FPR against the TPR at various decision thresholds. While this

```
# Clean and interpolate thresholds
thresholds_test = np.nan_to_num(thresholds_test,
                                nan=0.0, posinf=1.0, neginf=0.0)
thresholds_test = np.clip(thresholds_test, 0, 1)
```

function is restricted to binary classification tasks, it serves perfectly for the class-specific decomposition approach employed in our methodology. The function returns the False Positive Rate, True Positive Rate, and decision thresholds, which are subsequently used for interpolation and aggregation rather than direct plotting.

The implementation incorporates comprehensive threshold cleaning procedures to address three potential extreme scenarios that can occur during ROC curve construction:

- **NaN Thresholds:** These typically occur when the estimator achieves perfect class separation, such that all positive samples receive higher scores than all negative samples. This situation can also arise when multiple samples have identical predicted probabilities or due to numerical precision issues. The function addresses this by converting NaN values to zeros, ensuring computational stability.
- **Positive Infinity Thresholds:** This scenario arises when the estimator requires a threshold higher than any observed probability to achieve certain FPR/TPR combinations. This commonly occurs when attempting to achieve TPR=0 (no true positives), requiring an infinitely high threshold, or when predicted probabilities reach their maximum value (1.0). The function replaces positive infinity values with 1.0, maintaining probabilistic interpretation.
- **Negative Infinity Thresholds:** These occur in the opposite scenario, when the estimator requires an infinitely low threshold to achieve TPR=1 (capturing all true positives). The function replaces negative infinity values with 0.0, ensuring that all thresholds remain within the valid probability range of [0,1].

This comprehensive threshold cleaning procedure ensures that all decision boundaries remain within valid probability ranges, preventing numerical instabilities that could compromise the integrity of the final multiclass ROC curve.

Interpolation to Common FPR Grid:

The interpolation of TPR values and thresholds to a common FPR grid represents a critical step in enabling meaningful aggregation across classes. Since each class in the multiclass problem generates its own ROC curve with potentially different FPR sampling points, which can vary dramatically in density and range depending on class separability and sample size, interpolation creates the unified framework necessary for subsequent aggregation.

The implementation employs one-dimensional linear interpolation with the `bounds_error` parameter set to `False`, enabling extrapolation beyond the original FPR range when necessary. The `fill_value` parameter is configured as `(0,1)` for TPR extrapolation, implementing the mathematically sound principle that if the FPR array contains points below the minimum original FPR, the TPR should be filled with 0, while points above the maximum original FPR should be filled with TPR=1.

```
# Define common FPR grid
common_fpr = np.linspace(0, 1, n_points)

# Interpolate TPR and thresholds to common grid
tpr_interp_test = interp1d(fpr_test, tpr_test,
                           bounds_error=False,
                           fill_value=(0, 1))(common_fpr)

threshold_interp_test = interp1d(fpr_test, thresholds_test,
                                 bounds_error=False,
                                 fill_value=(thresholds_test[0],
                                             thresholds_test[-1]))(common_fpr)
```

```
# Normalize weights to ensure they sum to 1 (to handle computational imprecision
↪  cases)
if np.sum(gini_weights_test) != 1:
    gini_weights_test_clean = gini_weights_test/np.sum(gini_weights_test)
else:
    gini_weights_test_clean = gini_weights_test
# Weighted averaging across all classes
agg_tpr_test = np.average(all_tpr_interp_test, axis=0,
                          weights=gini_weights_test_clean)
agg_thresholds_test = np.average(all_threshold_interp_test, axis=0,
                                 weights=gini_weights_test_clean)
# Calculate final AUC using trapezoidal rule
agg_auc_test = auc(common_fpr, agg_tpr_test)
```

This extrapolation strategy reflects the mathematical properties of ROC curves: at FPR=0, the absence of false positives should correspond to TPR=0, while at FPR=1, the classification of all negatives as positive should correspond to TPR=1. These boundary conditions ensure that the interpolated curves maintain their probabilistic interpretation at the extremes.

Gini-Weighted Aggregation Process:

The Gini weights represent the core innovation of the methodology, as they determine the relative influence each class exerts on the final multiclass ROC curve. The weighting scheme ensures that classes with higher discrimination power (as measured by their Gini indices) contribute more significantly to the overall performance assessment.

Figure 4.6 illustrates how these weights distribute across classes in our credit rating example. Class A receives the highest weight (0.18), indicating it exhibits the strongest discriminative power and contributes most significantly to the aggregate ROC curve construction, while class CC receives the lowest weight (0.07), showing limited separability from other classes and exerting minimal influence on the overall performance assessment.

Numerical Integration and Uncertainty Quantification:

The weighted averaging occurs at each FPR point across all classes, creating a smooth aggregated curve that reflects the collective performance of the multiclass classifier. Following the aggregation, the Area Under the Curve (AUC) is calculated using Scikit-Learn's built-in `auc` function, which employs the trapezoidal rule for numerical integration. This numerical approach enables precise quantification of classifier performance across multiple classes, providing a robust foundation for multiclass evaluation.

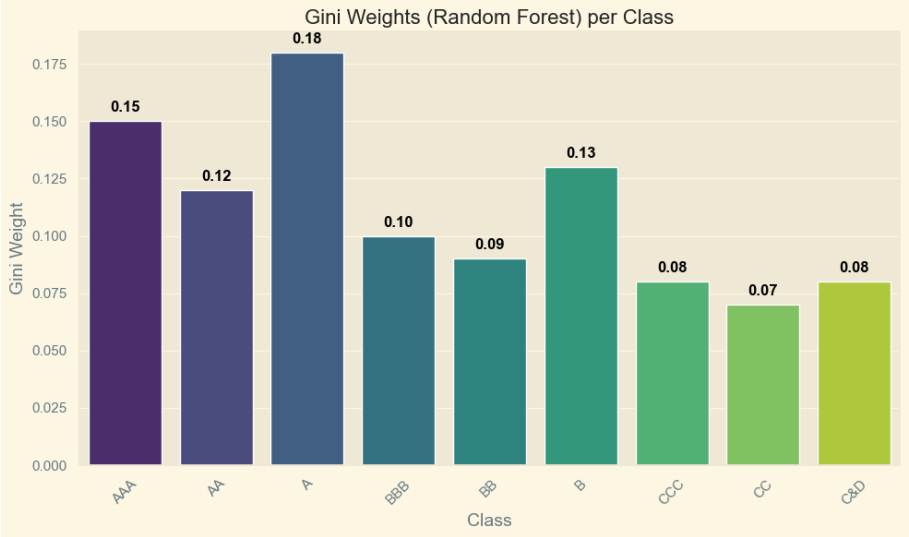

Figure 4.6: Gini weights for a Random Forest classifier showing relative class contributions to the aggregate ROC curve

```
# Compute weighted variance across classes
var_test = np.average((all_tpr_interp_test - agg_tpr_test)**2,
                      axis=0, weights=gini_weights_test_clean)
std_test = np.sqrt(var_test)
```

Confidence intervals are computed using the weighted variance and standard deviation of TPR values are calculated across all classes at each FPR point. This approach provides a quantitative measure of uncertainty and variability in the aggregated ROC curve, effectively indicating the degree to which individual class ROC curves deviate from the aggregated representation:

The inclusion of standard deviation bands enables assessment of methodological robustness and provides insight into model reliability across different classification thresholds, which proves particularly valuable for regulatory compliance in financial applications.

Gini-Weighted Performance Metrics Analysis:

Our multiclass ROC framework extends beyond visualization to provide a comprehensive suite of performance metrics that reflect the true discriminative capability of the classifier. All metrics are weighted by individual class Gini coefficients, directly linking model performance to the actual separability of each class rather than relying solely on class frequency considerations.

This approach differs fundamentally from traditional micro and macro averaging methodologies. While micro-averaging weights classes by their sample frequency and macro-averaging treats all classes equally, our Gini-weighted approach weights classes by their actual discriminative power as measured by the multidimensional Gini index.

The Gini-weighted methodology proves particularly valuable for imbalanced datasets, a common characteristic in multiclass classification scenarios, because it addresses several critical limitations:

- **Frequency Bias Mitigation:** Traditional micro-averaging can be dominated by majority classes, potentially masking poor performance on minority classes that may be of critical importance (such

as default categories in credit risk).

- **Discriminative Power Focus:** Unlike macro-averaging, which treats poorly-separable classes equally with well-separable ones, Gini weighting emphasizes classes where the model demonstrates genuine discriminative ability.
- **Realistic Performance Assessment:** By weighting based on separability rather than frequency, the metrics reflect the model's actual capability to distinguish between classes in practice.

Furthermore, all metrics are weighted by individual class Gini weights, directly linking model performance to correct classification of each class. Consequently, performance metrics such as precision, accuracy, and recall represent weighted averages of individual class metrics, computed using scikit-learn's built-in performance functions before aggregation.

Threshold Selection and Credit Risk Applications:

The threshold selection in our visualization (threshold = 0.55) was specifically chosen to maximize the F1-score, which holds particular relevance for credit risk management applications. This metric provides robust model evaluation by balancing both False Positive Rate (FPR) and False Negative Rate (FNR) while minimizing bias toward specific classes.

Table 4.3 presents a comprehensive comparison between classical metrics and our Gini-weighted approaches for the Random Forest classifier:

Table 4.3: Performance Metrics Comparison: Traditional vs. Gini-Weighted Approaches

Metric	Accuracy	Precision	Recall	F1-score
Gini-Weighted	0.44	0.20	0.77	0.30
Traditional (Macro)	0.49	0.56	0.52	0.53

The significant differences in performance metrics between the two approaches reflect fundamentally different evaluation philosophies:

- **Conservative Precision:** The Gini-weighted approach yields lower precision (0.20 vs 0.56) because it weights poorly-separable classes more realistically, avoiding optimistic estimates that traditional averaging might produce when dominated by well-performing majority classes.
- **Enhanced Recall:** Higher recall in the Gini-weighted approach (0.77 vs 0.52) indicates better performance in identifying true positive cases across classes where the model demonstrates genuine discriminative ability.
- **Balanced F1-Score:** Despite the individual metric differences, both approaches converge to similar F1-scores, suggesting that the Gini-weighted method provides a more realistic assessment of the precision-recall trade-off.

This divergence reflects that traditional metrics may overestimate performance by giving equal weight to classes where the model performs well due to chance or class imbalance, while Gini weighting provides a more conservative and realistic assessment based on actual separability.

Cross-Classifier Performance Analysis:

Table 4.4 presents the comprehensive performance comparison across different classifiers using our multidimensional Gini-weighted metrics[8]:

Table 4.4: Performance Metrics Using the Gini-weights

Classifier	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.34	0.19	0.91	0.30
Logistic Regression	0.44	0.28	0.77	0.30
Random Forest	0.44	0.20	0.77	0.30
Bagging Classifier	0.14	0.14	0.98	0.22

Model Selection and Performance Assessment:

For our credit rating case study, we needed to select the best-performing classifier from several candidates. Our evaluation went beyond simple accuracy scores to include learning curves, confusion matrices, and individual class ROC curves computed using the One-vs-Rest strategy, which can be seen at 4.7.

Why not just use accuracy? In our highly imbalanced dataset, a naive classifier could achieve high accuracy by always predicting the most common rating class, while completely failing to identify defaults. This is precisely why we need more sophisticated evaluation methods.

The comparative analysis revealed the performance characteristics shown in Table 4.5:

Table 4.5: Classifier Performance Comparison

Classifier	Train Score	Test Score	Fitting Time	Accuracy	Precision	Recall	F1-Score
Decision Tree	0.55	0.55	41.12	0.48	0.54	0.52	0.51
Logistic Regression	0.60	0.51	84.63	0.35	0.53	0.53	0.51
Random Forest	0.70	0.62	58.70	0.50	0.56	0.53	0.54
Bagging Classifier	0.52	0.48	331.04	0.40	0.52	0.44	0.46

Understanding the challenging context: Our dataset presents a perfect storm of difficulties that would challenge any classifier: limited sample size, severe multicollinearity among financial variables, and extreme class imbalance. These factors collectively constrain the achievable performance of all classification algorithms. However, this creates an excellent opportunity to test how our multiclass ROC curve behaves under adverse conditions.

The Random Forest emerged as our optimal choice, achieving a theoretical AUC of 0.51 for both training and test sets. While this might seem modest, it represents the best balance across multiple evaluation criteria while maintaining computational efficiency.

Our multidimensional Gini calculation also produces individual Gini indices for each credit rating class, offering valuable insights into class-specific model performance and intraclass variability in predicted probabilities.

[8] Performance measured at the threshold yielding the highest F1-score for each classifier.

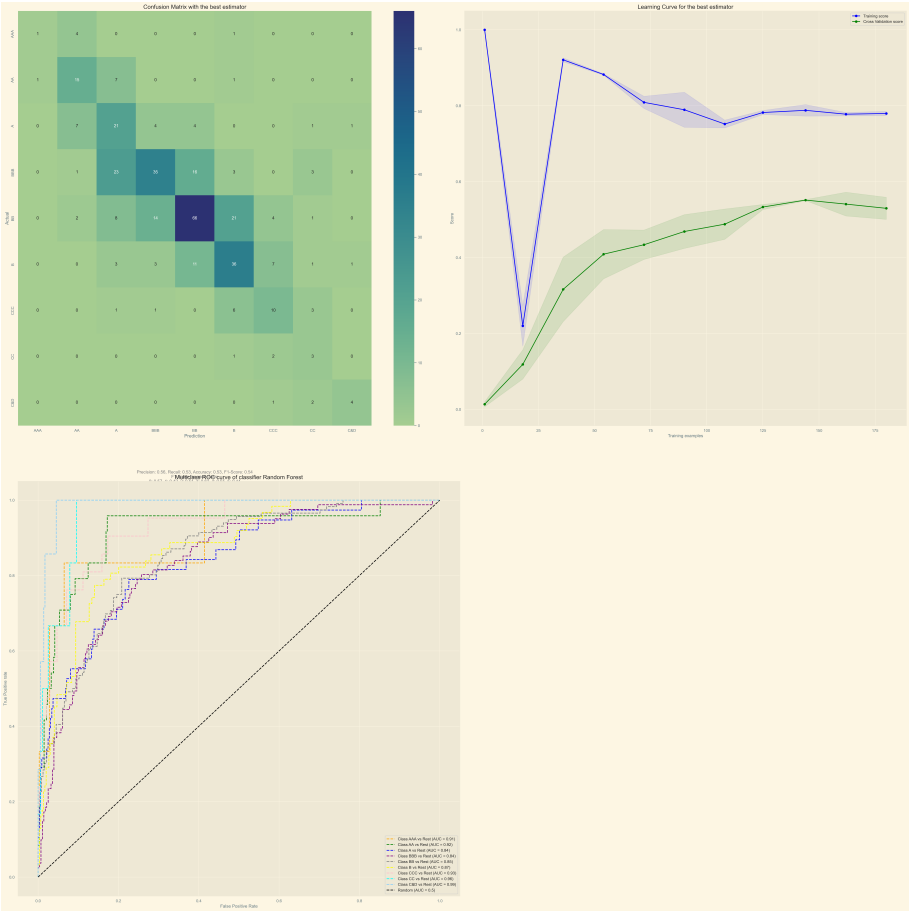

Figure 4.7: Confusion matrix, Learning curve, and ROC curves for Random Forest

Table 4.6: Unidimensional Gini Coefficients by Credit Rating Class

Credit Rating	Unidimensional Gini Coefficient
AAA	0.0186
AA	0.0186
A	0.0188
BBB	0.0187
BB	0.0186
B	0.0186
CCC	0.0186
CC	0.0186
C&D	0.0198

These results tell an interesting story. The observed values, ranging between 0.0186 and 0.0198, indicate almost no discrimination across all rating classes. Notably, the C&D rating category shows the highest coefficient (0.0198). This aligns perfectly with credit risk theory: companies in distress are naturally more distinguishable from healthy companies than, say, an AA-rated company is from an A-rated one.

Higher unidimensional Gini indices indicate better discrimination ability for that specific class. Lower indices might signal either inherent difficulty in separating that class or insufficient training data.

In its unidimensional form, the Gini index represents the average of absolute differences between all pairs of values in a distribution. The index averages the absolute deviations between every possible pair of values in the dataset, with this average subsequently normalized by its mean to ensure scale-invariance and cross-distribution comparability. Values approaching zero indicate minimal average deviation, intermediate values represent moderate dispersion relative to the mean, and a value of one signifies maximum inequality when average differences are maximized.

The unidimensional Gini index is mathematically expressed as:

$G = \frac{1}{2n^2\mu} \sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|$

Where:

- n represents the number of observations.
- x_i, x_j are individual values in the distribution.
- μ denotes the mean of the distribution.

The double summation encompasses all pairwise comparisons within the dataset, providing a comprehensive measure of distributional dispersion.

The Multidimensional Gini index extends this concept as a weighted average, where each unidimensional Gini index is weighted by the relative magnitude of its whitened mean, thereby accounting for the contribution of each dimension to overall variability.

This weighting scheme represents a fundamental innovation: instead of treating all classes equally (macro-averaging) or weighting by frequency (micro-averaging), we weight by actual discriminative power.

4.3.3 From Theory to Visualization: Building the ROC Curve

The real challenge lies in translating our theoretical AUC measure into a visual ROC curve that practitioners can understand and use. This requires maintaining the interpretability of traditional ROC curves while incorporating our sophisticated Gini-based weighting scheme.

Our `metrics_multiroc` function serves as the bridge between theory and practice. Here is how it works:

The random guessing problem: In binary classification, random guessing follows the diagonal line (AUC = 0.5). But in multiclass problems, this changes dramatically. For our 9-class credit rating problem, random guessing has an AUC of only 1/9 = 0.111. This makes our achieved empirical AUC of 0.85

```python
def metrics_multiroc(y_test, whitened_proba_test, gini_weights_test, n_points=100):
    """
    Compute aggregated multiclass ROC metrics using Gini-weighted averaging.

    The magic happens in three steps:
    1. Compute individual ROC curves for each class
    2. Interpolate all curves to a common FPR grid
    3. Aggregate using Gini weights
    """
    n_classes = len(np.unique(y_test))

    # Normalize weights to ensure they sum to 1
    if np.sum(gini_weights_test) != 1:
        gini_weights_test_clean = gini_weights_test/np.sum(gini_weights_test)
    else:
        gini_weights_test_clean = gini_weights_test

    # Storage for interpolated values
    all_tpr_interp_test = np.zeros((n_classes, n_points))
    all_threshold_interp_test = np.zeros((n_classes, n_points))

    # Common FPR grid for interpolation
    common_fpr = np.linspace(0, 1, n_points)

    # Process each class individually
    for i in range(n_classes):
        # Convert multiclass to binary for this class
        true_labels_test = (y_test == i).astype(int)

        # Compute ROC curve
        fpr_test, tpr_test, thresholds_test = roc_curve(
            true_labels_test, whitened_proba_test[:, i])

        # Interpolate to common grid
        tpr_interp_test = interp1d(fpr_test, tpr_test,
                                   bounds_error=False,
                                   fill_value=(0, 1))(common_fpr)

        all_tpr_interp_test[i] = tpr_interp_test

    # Weighted averaging - this is where the Gini weights matter
    agg_tpr_test = np.average(all_tpr_interp_test, axis=0,
                              weights=gini_weights_test_clean)

    # Calculate final AUC
    agg_auc_test = auc(common_fpr, agg_tpr_test)

    return common_fpr, agg_tpr_test, agg_thresholds_test, agg_auc_test, std_test
```

even more impressive; it is a five-fold improvement over random chance. Nonetheless, in high-stakes applications analysts should try to consider models with a Multiclass ROC curve with an AUC>0.5.

Why different baselines matter: The choice of baseline depends on your classifier type:

For a multiclass model, an AUC of 0.85 means that, on average, the model has an 85% chance of ranking a randomly chosen instance from a given class higher (in its predicted score for that class) than a randomly chosen instance from another class, when considering all possible pairs of classes and averaging these probabilities.

- For native multiclass algorithms (like multinomial logistic regression) [9]: use the 45-degree diagonal.
- For ensemble methods using One-vs-Rest strategies: use $y = x/\text{number\_of\_classes}$.

Our implementation automatically detects the appropriate baseline by analyzing the probability structure.

4.3.4 Visualization and Performance Analysis

Our multiclass ROC curve implementation provides multiple visualization options. Figure 4.8 shows the result for our Random Forest classifier:

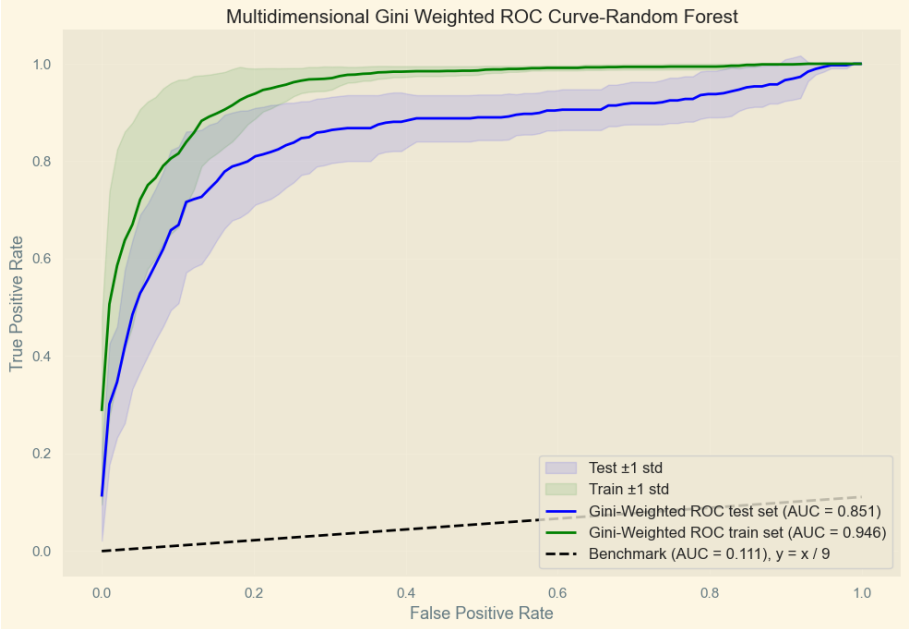

Figure 4.8: Multiclass ROC curve for a Random Forest classifier

Key insights from the visualization:

- The empirical AUC reaches 0.85, far exceeding the 0.111 random baseline.
- Confidence bands show the variability across classes.

[9] It calculates the probability of each class using the logistic function and normalizes these values across all classes.

- The slight gap between training[10] and test curves indicates minimal overfitting.

As it has been mentioned, visualization remains a critical component of our methodological approach. To provide a truly interactive analytical experience, we developed a Plotly-based multiclass ROC curve implementation, at Figure 4.9, featuring a shaded aggregate operating characteristic curve. A dynamic red marker can be positioned along the curve to inspect key performance metrics, accuracy, precision, recall, and F1-score, at any selected threshold. The slider empowers analysts to explore alternative trade-offs and optimize threshold selection based on specific application requirements.

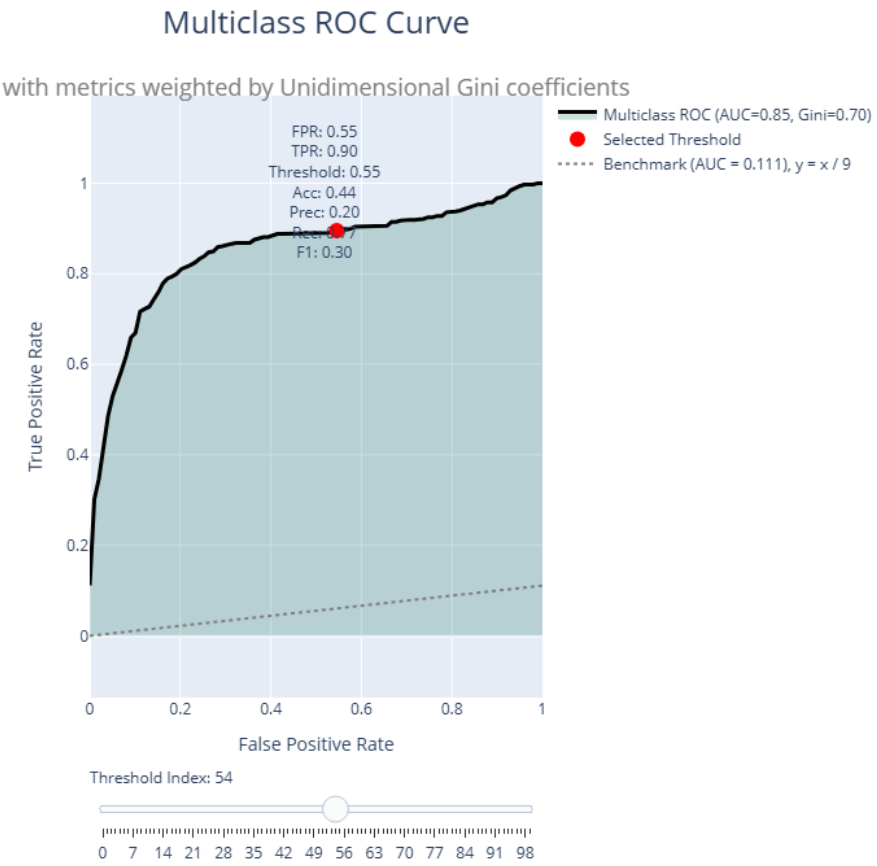

Figure 4.9: Interactive multiclass ROC curve utilizing the Multidimensional Gini index

The primary advantage of this dynamic visualization lies in its ability to demonstrate how performance metrics vary with decision threshold selection. This functionality proves invaluable for identifying thresholds that maximize specific performance metrics, enabling data-driven decision-making in threshold optimization. Analysts can select the threshold that benefits them the most, considering the type of error which has a higher cost to them.

Performance Comparison: Traditional vs. Gini-Weighted Metrics:

Table 4.7 reveals fascinating patterns when comparing our approach to traditional methods:

[10] Calculating AUC values of the training set is not standard practice in model evaluation, but was interesting to consider.

Table 4.7: Comparison of AUC and Gini Metrics Across Different Classifiers

Classifier	Theoretical Gini-AUC	Empirical Gini-AUC	Macro AUC	Micro AUC	Theoretical Gini	Empirical Gini
Decision Tree	0.51	0.78	0.84	0.87	0.022	0.57
Logistic Regression	0.50	0.51	0.79	0.87	0.032	0.02
Random Forest	0.51	0.85	0.90	0.91	0.018	0.70
Bagging Classifier	0.56	0.48	0.84	0.85	0.124	0.03

Consistently, micro-AUC values exceed macro-AUC values across all classifiers, indicating that models perform better on majority classes than on minority classes, a typical pattern in imbalanced datasets where micro-averaging is influenced by the dominant classes. **A particularly noteworthy pattern emerges when examining the convergence between empirical Gini-AUC and traditional AUC metrics**: high-performing classifiers like Random Forest (empirical Gini-AUC: 0.85 $\approx$ macro-AUC: 0.90 $\approx$ micro-AUC: 0.91) and Decision Tree show close alignment between these metrics, while lower-performing classifiers like Logistic Regression (empirical Gini-AUC: 0.51 $<<$ macro-AUC: 0.79) and Bagging Classifier exhibit substantial divergences.

This convergence pattern validates the empirical Gini-AUC methodology, suggesting that when classifiers achieve robust performance, the stabilization mechanisms preserve the underlying discriminative structure and maintain consistency with traditional AUC calculations.

Notably, the empirical Gini-AUC tends to be slightly lower than traditional metrics even for well-performing models, which reflects the **conservative nature of the multidimensional approach**. This is not a bug; it is a feature. This occurs because the rank-based rescaling and stabilization processes remove artificial correlations and overfitting artifacts that may inflate traditional AUC values, while the class-specific weighting scheme demands genuine discriminative power across all rating categories rather than relying on dominant classes[11].

This conservative estimation proves advantageous for credit risk applications, as it provides more reliable performance bounds and reduces the risk of deploying overconfident models in high-stakes financial decisions. The Gini-based approach could also offer superior robustness to class imbalance since it weights classes according to their actual separability, rather than frequency.

Moreover, this approach can also serve as a quality indicator where convergence with traditional metrics signals deployable classifier performance. Conversely, large discrepancies indicate poor class separability, numerical instabilities during whitening, or fundamental model inadequacy for the given dataset, making the empirical Gini-AUC both a performance metric and a model validation tool essential for regulatory compliance in financial risk management.

Understanding the Gini Indices Discrepancy

A significant difference exists between our theoretical Gini index (0.0187) and empirical Gini index (0.70), which approaches unity and suggests effective discrimination between classes. This is not an

[11] More about this in the next sections.

error, it reflects different mathematical spaces:

- **Theoretical Gini:** Operates in "whitened probability space" after ZCA transformation, provides insight into the fundamental separability characteristics of the transformed feature space, offering a mathematically pure measure of class distinction that remains consistent with the Multidimensional Gini theoretical framework.
- **Empirical Gini:** Functions in "rank space" derived from $G = 2 \times \text{AUC} - 1$, proving more practically relevant for classification applications, as it operates within the same probabilistic space where actual classification decisions will be implemented. This empirical measure directly correlates with ROC curve performance, accurately reflects the decision-making capability across various threshold settings, and enables meaningful comparison with other classification methods operating within the same analytical framework.

The difference between the two Gini indices has to do with both the stabilization process, the accuracy of the visual approximation through trapezoidal integration, and weighted averages.

This difference was not so drastical if I did not apply the stabilization of the whitening process. The stabilization process applies rank-based rescaling that preserves all pairwise orderings (essential for ROC analysis) while completely changing distributional properties. The empirical Gini proves more practically relevant for classification decisions, as it operates in the same space where actual predictions occur.

The theoretical Gini formula assumes the original distributional characteristics of the whitened data. After rank rescaling, this becomes a measure of uniformly distributed ranks rather than the original probability differences.

This divergence between theoretical and empirical values represents a methodologically sound trade-off that resolves critical computational challenges while preserving the essential discriminative information required for practical classification analysis.

Another relevant visualization employed has to do with per-class ROC curves, which are computed using the One-vs-Rest strategy that involves treating each class as the positive class while grouping all remaining classes as the negative class. These per-class ROC curves are plotted against the Multidimensional one computed. Figure 4.10 demonstrates how our multiclass ROC curve relates to individual class performance.

As expected, the constructed multiclass ROC curve lies between the individual per-class curves, which aligns with theoretical expectations since the Gini index functions as an averaging mechanism in the Cauchy mean value theorem sense. [12]

Moreover, a relevant aspect when carrying out model selection is understanding the characteristic shapes of ROC curves encountered in practice, this is more relevant for per-class ROC curves, since the averaging procedure makes the multiclass ROC curve smoother by construction. This provides valuable insights into model performance and data characteristics:

Smooth curves typically indicate:

[12] In other words, the ROC curve is between the per-class ROC curve with the highest AUC value, and the per-class ROC curve with the lowest AUC value.

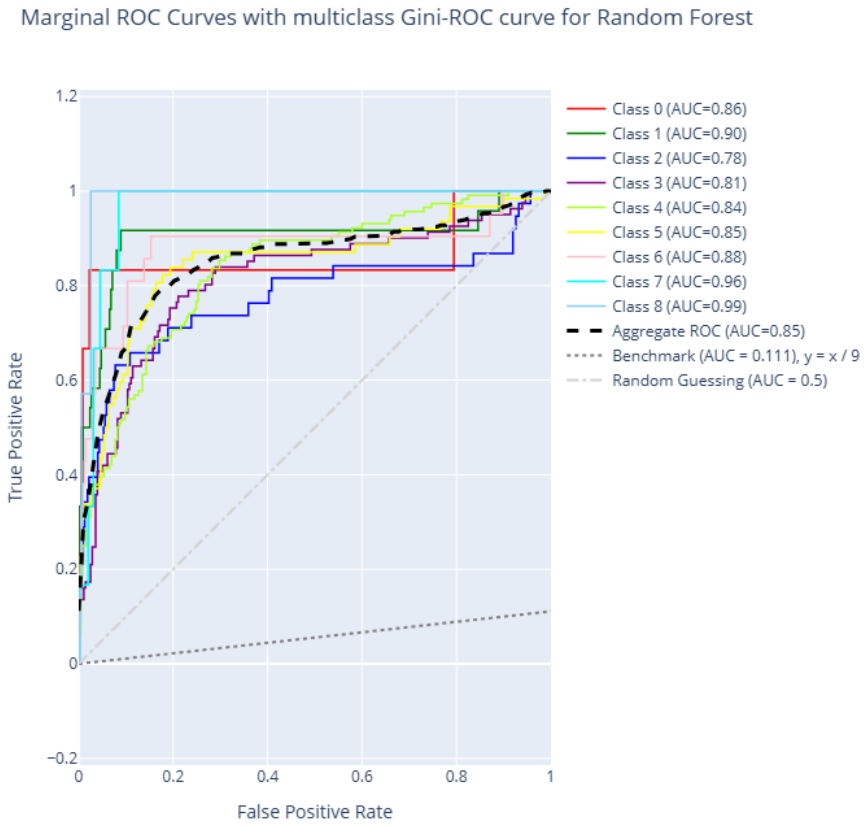

Figure 4.10: Per-class ROC curves and multiclass ROC curve comparison for the Random Forest

- Large sample sizes creating finer threshold granularity.
- Continuous probability distributions: Well-calibrated models producing diverse prediction scores.
- Balanced class representation with sufficient positive and negative cases.
- Effective model discrimination with clear separation capabilities between classes.

Erratic curves typically result from:

- Limited sample sizes constraining threshold options.
- Discrete probability outputs from models producing limited distinct values.
- Severe class imbalance creating sparse sampling in certain FPR/TPR regions.
- Poor model calibration with probabilities clustered around specific values.

Sharp edges occur when:

- Threshold discontinuities: Large gaps between consecutive probability values.
- Perfect classification regions: Model achieving perfect separation for specific score ranges.
- Data clustering: Sample concentration around specific probability values.
- Model artifacts: Decision trees or similar models creating discrete decision boundaries.

This is evident in Figure 4.10, where class 4 (BB rating) exhibits the smoothest curve and represents the highest frequency in the sample at 30.6%, while class 0 (AAA rating) displays a highly discontinuous curve despite representing only 1.1% of the sample.

These characteristics provide diagnostic information, indeed, smooth curves with high AUC indicate robust, well-calibrated models. Meanwhile, jagged curves with high AUC may suggest overfitting or insufficient data. On the other hand, sharp edges could reveal critical decision thresholds particularly relevant in credit assessment applications.

Per-class ROC curves facilitate detailed comparisons between individual credit rating classes, while computing a single multiclass ROC curve enables assessment of overall classifier discriminatory power. The multiclass ROC curve exhibits greater smoothness compared to individual per-class curves because we interpolate each class' ROC onto a common false-positive-rate grid and weight them by their Gini-derived importances. This methodology generates smooth curves whose dispersion envelopes highlight model stability and robustness.

4.3.5 The Precision-Recall Gini-weighted Curve

Recognizing that ROC analysis does not always accurately reflect performance in imbalanced class scenarios, we extend the same Gini-weighted philosophy to Precision-Recall (PR) curves, shown in Figure 4.11. We compute micro- and macro-average PR curves using conventional methods, then overlay a third, Gini-weighted PR curve that prioritizes classes proportionally to their contribution to overall dispersion. A visual marker denotes the point where precision equals recall, providing an additional perspective on threshold selection. This implementation returns a comprehensive metrics dictionary for threshold optimization.

Traditional multiclass Precision-Recall curves, constructed using macro and micro averaging techniques, exhibit greater variability and sensitivity to class sample sizes. Given that our dataset contains relatively few samples, this sensitivity likely explains the reduced robustness observed in these conventional approaches. This observation is confirmed when examining per-class precision-recall curves, which demonstrate considerable variability in contrast to the smoother precision-recall curve generated using the multidimensional Gini index approach. We should recall that Precision-recall curves positioned closer to the upper-right corner indicate superior performance, characterized by both high precision and high recall values simultaneously.

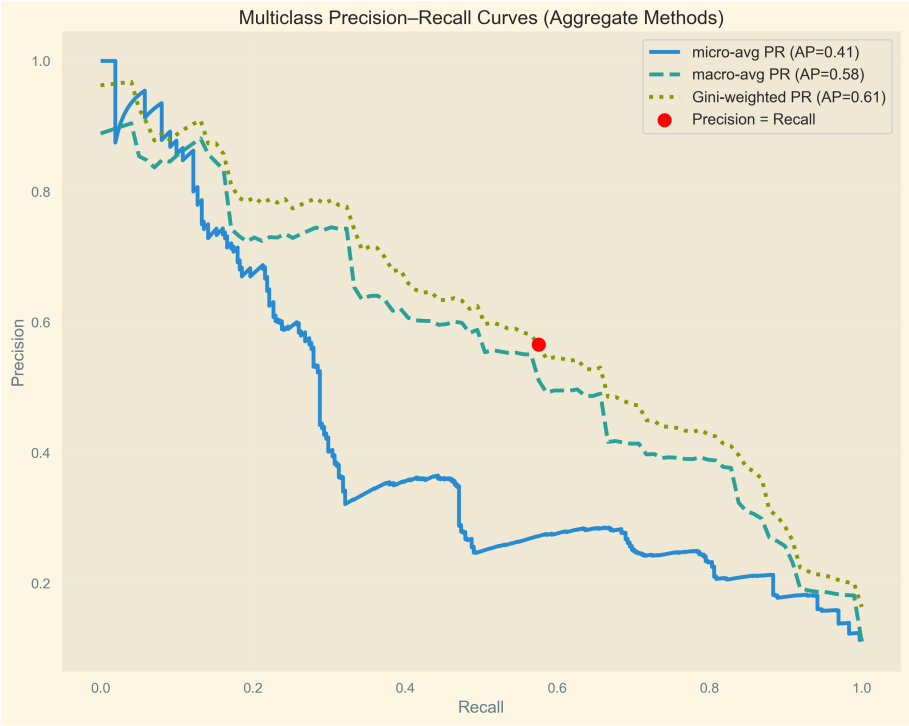

Figure 4.11: Multidimensional Gini Multiclass Precision-Recall curve against standard multiclass Precision-Recall curves

Precision-recall curves prove particularly valuable when prioritizing True Positive Rate (TPR) over True Negative Rate (TNR) in the analytical framework. The per-class PR curves plotted in Figure 4.12 alongside the Gini-weighted PR curve exhibit considerable variability, suggesting that individual class-based precision-recall analysis may not provide a robust framework for model validation in this multiclass context.

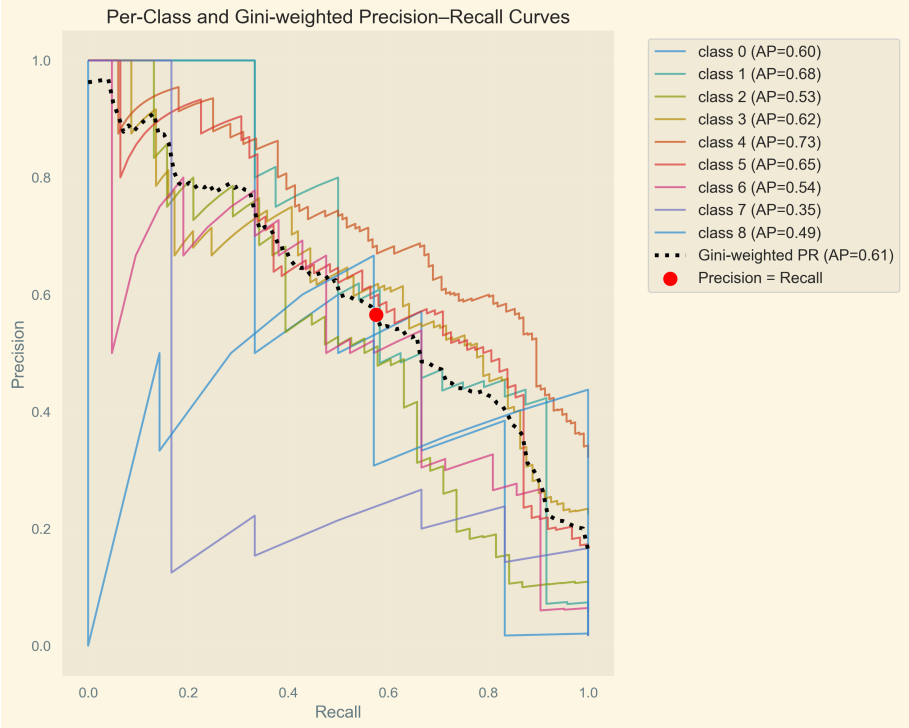

Figure 4.12: Per-class Precision-Recall curves against Multidimensional Gini Multiclass Precision-Recall curve

Notably, the Gini-weighted PR curve consistently positions itself between the individual class curves, demonstrating its averaging behavior. In Figure 4.11, the Gini-weighted PR curve achieves a higher PR-AUC, of 0.61, while following a trajectory remarkably similar to the micro-averaging approach across different thresholds. This similarity stems from both methods being influenced by the dominant classes in the dataset, though the Gini-weighted approach provides additional stability through its discriminative power weighting scheme, which helps mitigate erratic behavior observed in traditional averaging methods for imbalanced multiclass scenarios.

Chapter 5

Model Validation and Analysis

5.1 Class Separability Analysis: Mahalanobis Distance Assessment

To validate the reliability of our Multidimensional Gini-based ROC metrics, we conducted a geometric separability analysis using Mahalanobis distances between class centroids, or means, in the predicted probability space. This analysis provides crucial diagnostic information about whether our multiclass performance metrics reflect genuine discriminative ability or are inflated by class overlap and noise.

Geometric separation refers to how distinctly classes are positioned relative to each other. Unlike simple accuracy metrics that only count correct classifications, geometric separation measures the *spatial relationships* between class distributions. When classes are well-separated geometrically, it indicates that the model has learned meaningful boundaries that can reliably distinguish between different risk categories.

We decided to employ the Mahalanobis distance rather than Euclidean distance because it accounts for the covariance structure and scaling differences between features [28]. The Mahalanobis distance between two class centroids μ_i and μ_j is defined as:

$$d_M(\mu_i, \mu_j) = \sqrt{(\mu_i - \mu_j)^T S^{-1} (\mu_i - \mu_j)} \tag{5.1}$$

where S is the pooled covariance matrix. This metric is particularly valuable because:

- **Scale invariance**: It normalizes for different measurement units across predicted probabilities.
- **Correlation adjustment**: It accounts for correlations between class probability predictions.
- **Geometric interpretation**: Values > 2.0 typically indicate well-separated classes, while values < 1.0 suggest significant overlap.

Figure 5.1 presents the pairwise Mahalanobis distances between all credit rating classes in our predicted probability space.

Our Mahalanobis results prove that when our model assigns:

- Company A: 85% probability of AAA rating.
- Company B: 15% probability of AAA rating.

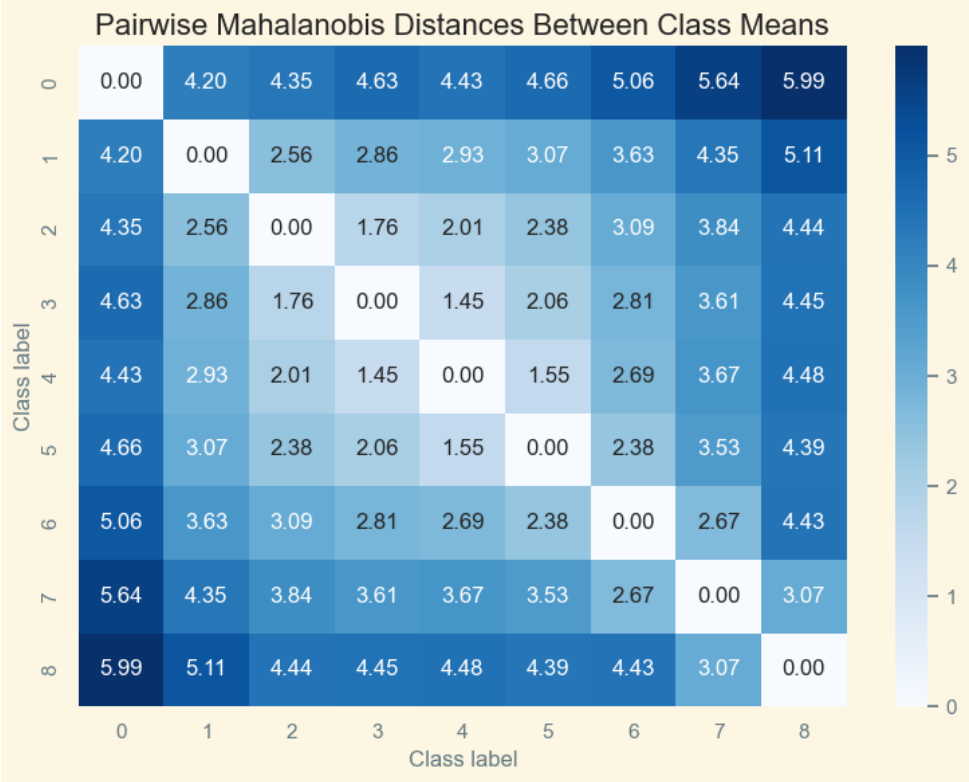

Figure 5.1: Pairwise Mahalanobis distances between class means

This difference reflects **genuine risk characteristics** rather than random model behavior, because AAA-rated companies cluster in a statistically distinct region of probability space (distance 5.99 from C&D companies).

Key Observations:

- **Diagonal elements** (0.00): Distance from each class to itself, confirming mathematical consistency
- **Extreme class separation**: The distance between AAA (class 0) and C&D (class 8) reaches 5.99, indicating excellent separation between the highest and lowest credit quality categories, demonstrating that our model has identified systematic differences between risk categories, not random patterns.
- **Neighboring class relationships**: Adjacent credit ratings show moderate distances (typically 1.45-2.93), reflecting the natural ordinal progression in credit risk
- **Overall separation quality**: Most pairwise distances exceed 2.0, indicating robust class discrimination according to established statistical guidelines [22].
- **Evidence Against Overfitting:** Since no class pairs exhibit distances lower than 1.0 (indicating poor separation), our AUC results do not stem from overfitting to noise or random fluctuations. Well-separated classes (distances >2.0) indicate that the model has learned genuine patterns rather than memorizing training data, ensuring that performance should generalize well to new companies.

Validation of Multiclass ROC Metrics

The substantial Mahalanobis distances observed provide strong validation for our multidimensional Gini-based ROC approach for several reasons:

1. Theoretical Foundation: According to [21], reliable multiclass performance metrics require classes that are genuinely separable rather than overlapping. Our distances confirm that this prerequisite is met.

2. Noise Resistance: High inter-class distances (particularly the 3.0+ values between non-adjacent ratings) indicate that our performance metrics reflect true discriminative ability rather than random classification in overlapping regions [15].

3. Ordinal Structure Validation: The model demonstrates authentic discriminative capability by systematically learning credit risk patterns. It has identified features that genuinely distinguish risk levels. The distance pattern shows appropriate progression from adjacent classes (moderate distances) to distant classes (large distances), confirming that our model correctly captures the ordinal nature of credit ratings; neighboring classes share similarities.

4. Robustness Indicator: Classes with clear geometric separation are unlikely to result from random classification or noise [22], indicating the model has learned robust discriminative features and supporting the reliability of our robustness analysis results.

Implications for Credit Risk Assessment

The clear class separation revealed by this analysis has practical implications:

- **Decision boundary reliability**: Well-separated classes indicate stable decision boundaries that are less likely to misclassify companies near rating transitions
- **Regulatory confidence**: Clear geometric separation provides evidence of systematic risk differentiation rather than arbitrary classification
- **Model interpretability**: Distinct class positioning facilitates understanding of what distinguishes different risk levels

In conclusion, the substantial pairwise Mahalanobis distances (ranging from 1.45 to 5.99) provide strong empirical evidence that our multidimensional Gini-based ROC curve and associated AUC metrics accurately reflect genuine model discriminative ability. This validation is particularly significant since our moderately high AUC for the Random Forest (0.85) is supported by clear geometric separation rather than statistical artifacts.

The Mahalanobis distance analysis validates our approach because it provides independent geometric evidence that our model has learned to place different credit rating classes in distinct regions of probability space. This separation could not occur by chance; it requires the model to have discovered systematic relationships between company characteristics and credit risk.

Well-separated classes ensure that decision boundaries lie in regions with genuine distributional differences rather than in overlapping areas where classification would become arbitrary. The model can re-

liably differentiate between companies with different risk profiles because each rating class occupies a statistically distinct region in probability space.

Protection Against Alternative Explanations: If classes had overlapped heavily (low Mahalanobis distances), high AUC scores could have resulted from several problematic scenarios:

- **Random fluctuations** in overlapping regions where chance classifications appear successful
- **Noise-driven classifications** where the model guesses correctly by accident rather than systematic learning
- **Artificial boundaries** that reflect data artifacts rather than genuine risk differences

However, our substantial inter-class distances eliminate these concerns, confirming that our Random Forest has developed genuine discriminative ability through systematic learning of credit risk patterns. This geometric validation supports the trustworthiness of our performance assessments and confirms that our methodology produces meaningful and interpretable results for credit risk classification applications.

5.2 Robustness Analysis: an application of the Multiclass ROC curve

In the risk management sphere, backtesting is heavily regulated due to its critical importance. Models must demonstrate robustness to dramatic changes in both idiosyncratic market conditions and systemic macroeconomic events, which occur without warning and can dramatically alter the landscape within brief time periods. Without adequate robustness, financial institutions rapidly lose liquidity, placing their solvency at risk and making bank runs and contagion imminent threats. We need to know if our credit risk model still makes good decisions when the data gets ”bumpy” due to market volatility or data noise.

This need for robust models has become even more critical with the European Union’s AI Act, which requires financial institutions to demonstrate that their AI systems remain stable and reliable under various conditions.

Robustness testing asks a simple question: ”If we slightly change the input data, do our model’s predictions change dramatically or stay relatively stable?”.

In our credit rating context, imagine a company’s total assets change by 5% due to market fluctuations. A robust model should still rank this company similarly relative to other companies; it should not suddenly jump from being classified as ”high risk” to ”low risk.”

To measure robustness, we use the SAFE AI framework [3], which focuses on rankings rather than exact prediction values. The SAFE AI framework was specifically designed to help organizations comply with these regulatory requirements by providing quantitative metrics for measuring AI system robustness, fairness, explainability, and accuracy. This is particularly crucial for high-risk AI applications in sectors such as credit scoring. Here is why rankings matter:

Example: Suppose our model assigns credit scores to three companies:

- Company A: 0.8 (good credit)
- Company B: 0.6 (moderate credit)
- Company C: 0.3 (poor credit)

The ranking is: A > B > C. Now, if we slightly perturb the data and get new scores:

- Company A: 0.7
- Company B: 0.5
- Company C: 0.2

The ranking remains: A > B > C. This indicates a robust model. But if perturbation gives us:

- Company A: 0.4
- Company B: 0.7
- Company C: 0.6

Now the ranking is: B > C > A. This dramatic change suggests poor robustness.

The SAFE AI framework introduces two key metrics:

Rank Graduation Accuracy (RGA)

RGA measures how well rankings are preserved between two sets of predictions:

$$\text{RGA} = \frac{\text{Number of correctly ordered pairs}}{\text{Total number of possible pairs}} \tag{5.2}$$

Where RGA ranges from 0 to 1:

- **RGA = 1.0**: Perfect ranking preservation (all pairs maintain their relative order).
- **RGA = 0.0**: Complete ranking disruption (all pairs flip their relative order).

Rank Graduation Robustness (RGR)

RGR applies RGA to measure stability under perturbations:

$$\text{RGR} = \text{RGA}(\text{Original Rankings}, \text{Perturbed Rankings}) \tag{5.3}$$

Interpretation:

- **RGR = 1.0**: Perfect robustness, rankings unchanged despite perturbations.

- **RGR = 0.5**: Moderate robustness, some ranking changes but structure partially preserved.
- **RGR = 0.0**: No robustness, rankings completely scrambled.

The RGR (Rank Graduation Robustness) quantifies model stability by measuring how well prediction rankings are preserved under input perturbations. The methodology operates through a systematic process that integrates seamlessly with multiclass ROC analysis.

The RGR framework follows a four-step process: **1. Baseline Ranking Establishment:** Extract original probability predictions from the unperturbed model, creating a reference hierarchy against which stability is measured. **2. Systematic Perturbation:** Apply controlled modifications (typically 5% changes) to input variables, simulating real-world data uncertainty. The perturbation percentage determines sensitivity assessment granularity while identifying variables that cause maximum instability when modified. **3. Rank Comparison:** Analyze whether the relative ordering of instances remains consistent after perturbation, for example, if Company A was originally ranked higher than Company B in creditworthiness, does this relationship persist? **4. Robustness Quantification:** Employ RGA (Rank Graduation Accuracy) to mathematically compare original and perturbed rankings, producing RGR scores between 0 and 1, where higher values indicate maintained ranking decisions despite input variations.

Integration with Multiclass ROC Analysis

Our approach extends traditional RGR by testing robustness *at each decision threshold*, creating a comprehensive robustness profile across the entire ROC curve:

1. Convert probability predictions to binary decisions at threshold t: decision $=$ (probability $\geq t$)
2. Apply identical thresholds to both original and perturbed predictions
3. Calculate RGR between resulting binary rankings using: RGR $=$ RGA(original ranks, perturbed ranks)
4. Generate robustness profiles showing RGR variation across all thresholds

The methodology conducts both individual and grouped perturbation analysis:

Individual Variable Perturbation: Each variable is modified in isolation while others remain unchanged, isolating specific contributions to model instability. This enables:

- Performance degradation assessment quantifying how each variable affects overall stability.
- Threshold-specific impact analysis revealing varying sensitivity across decision boundaries.
- Comparative ranking of variables by their robustness impact.

Grouped Perturbation: All variables are simultaneously modified to assess cumulative instability effects under realistic uncertainty scenarios.

Implementation

This comprehensive approach[1] provides threshold-specific robustness assessment essential for credit risk

[1] All functions available in the file `robustness`.

```python
def analyze_multiclass_perturbation(xtest, y_test, model, variables,
                                    whitened_proba_orig, gini_weights_test,
                                    perturbation_percentage=0.05):
    """
    Comprehensive RGR analysis across decision thresholds.
    Returns variable-specific robustness profiles.
    """
    # Establish baseline metrics
    fpr_orig, tpr_orig, thresholds, auc_orig, std = metrics_multiroc(
        y_test, whitened_proba_orig, gini_weights_test)

    perturbation_results = {}

    # Test each variable individually
    for var in variables:
        xtest_pert = perturb(xtest.copy(), var, perturbation_percentage)
        whitened_proba_pert = model.predict_proba(xtest_pert)

        # Calculate RGR at each threshold
        rgr_values = []
        for thresh in thresholds:
            orig_rank = whitened_proba_orig >= thresh
            pert_rank = whitened_proba_pert >= thresh
            rgr_values.append(core.rga(orig_rank.flatten(),
                                       pert_rank.flatten()))

        perturbation_results[var] = {
            'rgr_profile': rgr_values,
            'max_rgr': max(rgr_values),
            'optimal_threshold': thresholds[np.argmax(rgr_values)]
        }

    return perturbation_results
```

applications, where different decision boundaries may exhibit varying stability under market uncertainty. The methodology serves multiple purposes: introducing controlled noise for generalization assessment, preventing overfitting detection, and simulating market volatility effects in financial applications.

5.2.1 Results

Our robustness analysis revealed several important patterns, summarized in Table 5.1.

Table 5.1: Robustness Analysis Results: Maximum RGR Values and Optimal Thresholds

Index	Variable	Max RGR	Threshold
0	final_normalized_score_env	0.546869	0.611845
1	final_normalized_score_social	0.546760	0.608805
2	final_normalized_score_governance	0.546760	0.608805
3	MORE_score_2021	0.549035	0.608805
4	MORE_score_2020	0.548894	0.608805
5	total_assets_2022	0.546760	0.608805
6	total_assets_2021	0.546760	0.608805
7	total_assets_2020	0.546760	0.608805
8	current_assets_2022	0.546366	0.608805
9	current_assets_2021	0.546507	0.608805
10	current_assets_2020	0.546366	0.608805
11	shareholders_funds_2022	0.545439	0.608805
12	shareholders_funds_2021	0.545186	0.608805
13	shareholders_funds_2020	0.546366	0.608805
14	current_liabilities_2022	0.546366	0.608805
15	current_liabilities_2021	0.546760	0.608805
16	current_liabilities_2020	0.546760	0.608805
17	turnover_2022	0.545580	0.608805
18	turnover_2021	0.547153	0.608805
19	EBIT_2022	0.547125	0.608805
20	EBIT_2021	0.547013	0.608805
21	EBIT_2020	0.546760	0.608805
22	net_income_2022	0.547013	0.608805
23	net_income_2021	0.547125	0.608805
24	net_income_2020	0.547153	0.608805
25	EBITDA_2022	0.546226	0.608805

Continued on next page

Table 5.1 – *Continued from previous page*

Index	Variable	Max RGR	Threshold
26	EBITDA_2021	0.546366	0.608805
27	EBITDA_2020	0.547547	0.608805
28	sectors_Financials	0.547013	0.608805
29	sectors_Health.Util	0.547013	0.608805
30	sectors_Manufacturing	0.546760	0.608805
31	sectors_Tech.Com	0.546760	0.608805
32	regions_Center	0.546760	0.608805
33	regions_Islands	0.547013	0.608805
34	regions_Northeast	0.547153	0.608805
35	regions_Northwest	0.546760	0.608805
36	regions_South	0.547265	0.608805
	Grouped	**0.547240**	**0.609799**

Key Findings:

1. **Moderate Robustness**: All variables show RGR values around 0.54-0.55, indicating moderate robustness. While not perfect, these values suggest the model maintains reasonable stability under perturbations.
2. **Historical Credit Ratings Matter Most**: The 2021 credit rating (MORE_score_2021) shows the highest RGR (0.549), meaning it is most robust to perturbations, while also being identified as the most important feature in our model.
3. **Grouped Effects**: When we perturb all variables simultaneously (grouped perturbation), RGR drops to 0.547, showing that combined perturbations have a cumulative effect on model stability.

Figure 5.2 shows how our ROC curves change when we perturb the three most sensitive variables:

The curves remain remarkably similar, confirming our moderate robustness findings. Small perturbations do not dramatically alter the ROC curve shape, suggesting reliable performance even with data uncertainty.

5.2.2 Comparing with Traditional Feature Importance

To put our robustness findings in context, let's compare RGR with traditional explainability methods:

The Key Difference:

- **Traditional methods** (Figure 5.3 and 5.4) ask: "Which features contribute most to predictions?" [29]

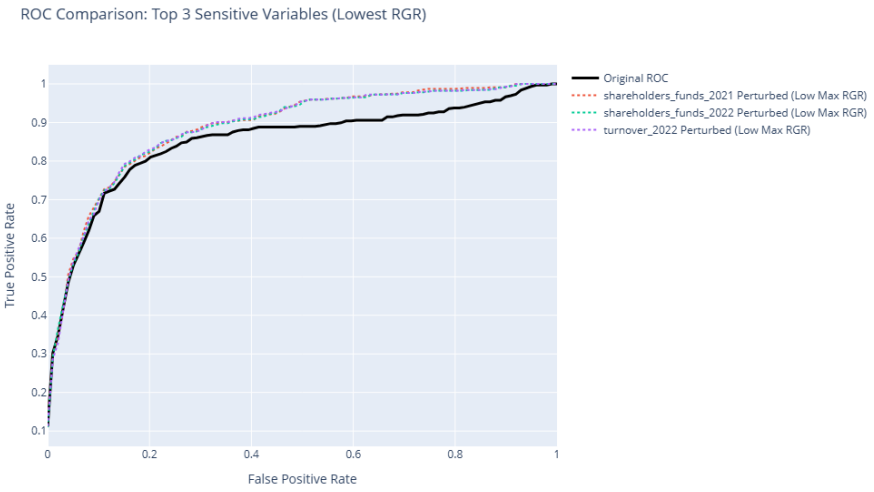

Figure 5.2: ROC curve stability analysis: Original curve (black) vs. perturbed curves for the most sensitive variables

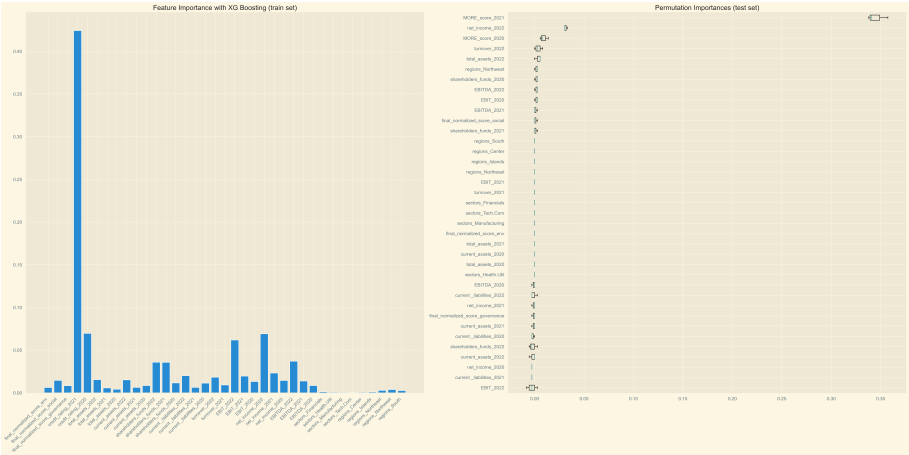

Figure 5.3: Traditional feature importance vs. permutation importance analysis

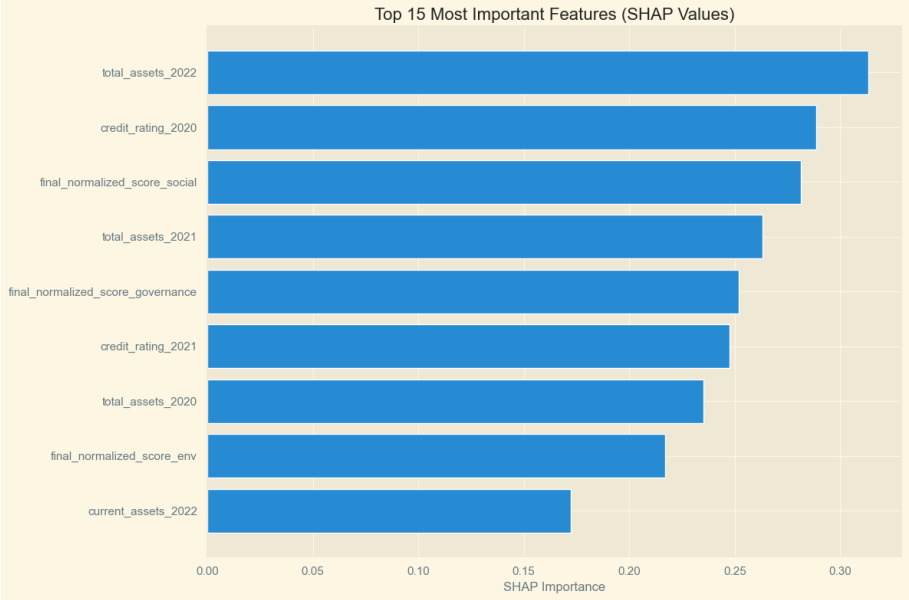

Figure 5.4: SHAP values showing feature contributions to model predictions

- **RGR methodology** asks: ”Which features, when changed, cause the least disruption to decision rankings?”.

Feature importance is measured in decision trees through a feature’s capability to create pure node splits with respect to Gini impurity or entropy, assigns higher importance to variables that effectively partition the data. Similarly, permutation importance evaluates feature significance by measuring the increase in model error when feature values are randomly shuffled, the underlying logic being that if a model relies heavily on a feature for prediction, shuffling its values will substantially degrade performance.

SHAP values extend the concept of Shapley values from cooperative game theory to machine learning, where ”the ’game’ is the prediction task for a single instance, the ’gain’ is the actual prediction minus the average prediction, and the ’players’ are the feature values that collaborate to achieve the prediction” [29].

According to our SHAP analysis, 2022 total assets emerge as the most influential feature, followed by the 2020 credit rating, which consistently ranks as the second most important feature across all traditional approaches, as well as the RGR framework. However, the third most important feature, social ESG rating, presents an interpretational challenge, as the direct relationship between a company’s social responsibility score and its credit rating may not be economically intuitive, potentially questioning the appropriateness of SHAP values for this specific application.

Importantly, the features with the highest predictive importance are not necessarily the most stable ones according to the RGR methodology. Our robustness analysis revealed that shareholders’ funds (for both 2021 and 2022) and 2022 turnover are among the most sensitive features to perturbations. The sensitivity of turnover makes intuitive sense, as company revenues can fluctuate dramatically due to market conditions, making this feature inherently unstable. This distinction between traditional importance measures and stability assessment proves crucial because it allows us to identify both the most and least stable features under uncertainty at specific decision thresholds, exactly what is needed in financial risk models

where reliability under stress conditions is paramount.

Both traditional approaches and RGR analysis identify the 2021 credit rating as crucial, but RGR provides the additional insight that this feature is also the most *stable* under uncertainty, precisely the characteristic we seek in a financial risk model. This convergence suggests a potential relationship between feature stability and predictive importance, indicating that the most reliable features may also be those that contribute most consistently to model predictions across varying market conditions.

5.2.3 Practical Implications for Credit Risk Management

Our robustness analysis provides several actionable insights:

- **Model Validation**: RGR values around 0.55 indicate our model has moderate robustness for deployment in credit risk applications.
- **Threshold Selection**: The optimal threshold (around 0.61) represents the point where our model is most robust to perturbations.
- **Feature Monitoring**: Variables with lower RGR scores require more careful monitoring in production, as they may be more susceptible to data quality issues.
- **Regulatory Compliance**: This analysis demonstrates compliance with EU AI Act requirements for robustness testing in high-risk financial applications.

The integration of robustness analysis with multiclass ROC curves provides a comprehensive framework for model validation that goes beyond traditional performance metrics, ensuring our credit risk models remain reliable even under challenging market conditions.

The framework's emphasis on rank-based measurements offers a more detailed understanding of model behavior under stress, moving beyond simple accuracy metrics to capture the stability characteristics that are essential for trustworthy AI deployment in regulated environments. Importantly, this approach not only considers overall model stability but also examines how robustness varies across different decision thresholds, a crucial consideration for real-world deployment in high-stakes domains like medicine and financial services, where different thresholds may provide varying levels of stability under uncertainty.

This framework also enables us to assess model sensitivity to changes in specific variables, allowing practitioners to identify which features contribute most to model instability. In credit risk management, this threshold-specific robustness analysis proves particularly valuable, as it enables institutions to select decision boundaries that maintain reliable performance even when market conditions introduce data perturbations.

5.3 Methodology Summary: Constructing the Multiclass ROC Curve

Step 1: Probability Extraction and Stabilization Extract predicted probabilities from the trained multiclass classifier using the `predict_proba` function, then apply ZCA-correlation whitening with nu-

merical stabilization, through the function `whitening_predicted_proba_stable`[2] to ensure scale invariance and computational stability. This process decorrelates class predictions and normalizes their variances while preserving ranking relationships essential for ROC analysis.

Step 2: Multidimensional Gini Index Calculation Compute the Multidimensional Gini coefficient from the whitened probability matrix using the Equation 3.8, this can be done using the `multidim_gini`.[3] This yields both the aggregate Gini index and individual class weights that reflect each class's contribution to overall discriminative power.

Step 3: Theoretical AUC Derivation Calculate the theoretical multiclass AUC using the relationship in Equation 3.36, using function `aggregate_AUC`[4], providing a single scalar performance metric that maintains the probabilistic interpretation of traditional AUC while incorporating the multiclass structure.

Step 4: Multiclass ROC Metrics Computation Execute the `metrics_multiroc` function[5] to generate aggregated False Positive Rate (FPR) and True Positive Rate (TPR) arrays by:

- Computing individual class ROC curves using One-vs-Rest decomposition.
- Interpolating each curve to a common FPR grid for consistent aggregation.
- Applying Gini-derived weights to create weighted averages, as in Equation 3.2.

Step 5: Visualization and Interpretation Generate comprehensive visual representations including[6]:

- Static multiclass ROC curve with aggregate AUC and confidence bands.
- Interactive Plotly-based visualization enabling performance metric calculation across thresholds.
- Comparative analysis with per-class ROC curves to demonstrate the aggregation behavior.

All steps can be automatically calculated by using the `complete_roc_analysis`[7], which displays key results (theoretical and empirical AUC and Multidimensional Gini values, as well as weights, FPR, TPR, threhsolds, weighted performance metrics and whitened probabilities), all the plots, and also provides the macro and micro AUC values for comparison.

ROC curves enable comprehensive analysis of individual model predictive power across varying decision thresholds, True Positive Rates, and true negative rates, while facilitating performance comparisons between different classifiers. Summary measures such as AUC, specifically designed for multiclass settings, provide straightforward frameworks for cross-model comparison. However, ROC curves and AUC must be utilized alongside other diagnostic tools including confusion matrices, learning curves, and complementary performance metrics for comprehensive model evaluation.

[2] Can be found in the file `proba_whitening` in my GitHub profile.
[3] Can be found in the file `gini_whitening`.
[4] Can be found in the file `metrics_multi_roc`.
[5] Can be found in the file `metrics_multi_roc`.
[6] All functions available in the file `multi_roc_plotting`.
[7] Can be found in the file `multi_roc_analysis`.

These curves offer particular advantages in applications where each class holds equal relevance, enabling interpretation of decision thresholds to calibrate models according to specific requirements. This calibration capability allows practitioners to tailor models based on whether true positives carry greater weight than true negatives, analyzing the optimal balance point between False Positive Rate (FPR) and True Positive Rate (TPR). In financial applications, this balance becomes particularly critical for risk assessment and regulatory compliance.

Through implementing these visualization techniques and novel diagnostic measures for multiclass tasks, we demonstrate that classical performance curves can be reweighted to reflect true class-level uncertainty. This approach provides both richer numerical summaries and more nuanced graphical diagnostics, enabling practitioners to understand not merely how well a model separates classes, but how its uncertainty is distributed across them.

The Multiclass Gini Multidimensional ROC curve offers a unified and interpretable measure of classification quality within this framework. The weight vector derived during the whitening and Gini computation process provides direct insight into which classes the model distinguishes most clearly and which it confuses. Classes exhibiting the highest confusion are those with the lowest weights.

For ESG and credit analysts, this framework proves particularly valuable by enabling practitioners to ask: ”Which ESG tiers are most reliably predicted, and where does the model struggle?” For instance, a high Gini weight on the ’High Risk’ ESG category may indicate that the model demonstrates particular proficiency in identifying risky entities, a crucial capability in impact investing applications. This framework could serve as the foundation for measuring the marginal impact of ESG ratings on credit ratings.

For credit analysts, risk managers, and regulators, the interpretability of Gini-weighted class contributions enhances model validation and auditability processes. Consider a classifier that exhibits high aggregate accuracy but allocates minimal Gini weight to low-rated credit classes. This would signal potential bias that might systematically overlook high-risk borrowers. Such insights support more informed threshold setting, internal rating validation, and comprehensive model risk management.

The implementation of ZCA-whitening ensures removal of correlations between predicted class probabilities, resulting in a cleaner decomposition of model performance sources. This proves particularly valuable in credit risk management, where multicollinearity presents a common challenge. The approach allows practitioners to compare models not only by overall scores but also by their structural behavior and the discriminatory power each class contributes to overall predictive capability. This enhanced interpretability makes model outputs more explainable and defensible in high-stakes financial settings, where stringent regulation exists to safeguard financial system resilience and stability as well as individuals’ life savings.

Our decomposition of aggregate inequality considers the contributions of each class to the total separability of the model. Utilizing the Gini index allows classes with higher discriminative power to exert greater influence on the ROC curve, with the Gini index indicating the magnitude of separation between classes.

Conclusions

This thesis introduces a novel methodology for constructing unified multiclass ROC curves using the multidimensional Gini index, addressing key limitations in existing multiclass evaluation approaches. The methodology delivers a single, interpretable ROC curve that preserves binary analysis properties while capturing multiclass discrimination complexity, providing financial institutions and regulators with a unified model validation framework that meets EU AI Act explainability requirements.

5.3.1 Core Methodological Innovation

Traditional multiclass metrics fail because they ignore the fundamental question: "Can this model actually distinguish between classes?" Macro-averaging treats classes with 5 samples identically to those with 500 samples, ignoring statistical reliability differences. Micro-averaging allows majority class performance to dominate, masking critical failures in minority classes, precisely where financial models fail most catastrophically.

Our Gini-weighted approach prioritizes discriminative capability over frequency considerations, ensuring performance assessments reflect genuine classification ability rather than dataset artifacts. The multidimensional Gini index provides scale invariance and risk-aware weighting that emphasizes classes where models demonstrate genuine discriminative power.

5.3.2 Key Empirical Findings

Our analysis yielded several key insights:

- **Empirical AUC of 0.85** achieved by Random Forest, representing a five-fold improvement over random chance (0.111 baseline for 9-class problem).
- **Mahalanobis distance analysis** confirmed genuine class separability (distances ranging 1.45-5.99), validating that performance metrics reflect true discriminative ability rather than statistical artifacts.
- **Convergence pattern**: Well-performing classifiers show alignment between empirical Gini-AUC and traditional metrics, while divergence indicates poor separability or model inadequacy.

5.3.3 Methodological Components

The framework provides four key diagnostic tools along with the Multiclass ROC curve and its AUC:

1. **Multidimensional Gini Coefficient**: Single interpretable metric capturing overall discriminative power across all classes simultaneously.
2. **Gini Weights**: Reveal which classes drive model discrimination, enabling identification of model strengths and weaknesses.

3. **Unidimensional Gini Indices**: Class-specific discrimination assessment for detailed regulatory compliance analysis where authorities require demonstration of discriminative capability across all segments.

4. **Gini-Weighted Performance Metrics**: Conservative assessments weighting precision, recall, and F1-scores by actual discriminative power rather than class frequency, preventing overconfident deployment of inadequate models.

The multidimensional Gini index offers scale invariance, a property completely absent in existing approaches, ensuring consistent performance assessment across diverse datasets and measurement scales.

Traditional ROC approaches treat all misclassifications equally, which is inappropriate for applications with varying error costs. Our framework addresses this through discriminative weighting that emphasizes critical classification boundaries based on their actual separability rather than frequency.

Beyond credit risk, this methodology applies to any domain where:

- Class imbalance biases traditional metrics
- Misclassification costs vary significantly across class pairs
- Regulatory oversight demands interpretable performance assessment

The discriminative focus ensures performance metrics reflect genuine model capability rather than artifacts of class distribution, making it particularly valuable for high-stakes applications requiring robust, explainable AI systems.

Technical innovations include ZCA-correlation whitening with numerical stabilization for computational robustness, rank-based rescaling maintaining threshold interpretability, and interactive visualization enabling real-time threshold optimization.

5.3.4 Robustness Analysis Application

The robustness analysis demonstrates a practical application of our multiclass ROC framework using the SAFE AI methodology. The key finding revealed an interesting divergence: features with the highest predictive importance are not necessarily the most stable under perturbations. While traditional feature importance methods identify contributors to predictions, our stability analysis shows that the 2021 credit rating emerged as both highly important and most robust (RGR = 0.549). Future research should investigate this relationship between feature stability and predictive importance more systematically.

5.3.5 Practical Applications

The Gini coefficient's established role in credit scoring makes this approach particularly compelling for financial institutions. The methodology enables:

- **Regulatory Compliance**: Meeting EU AI Act explainability requirements through interpretable visualization.

- **Threshold-Independent Evaluation**: Assessing performance across all operating conditions.
- **Model Validation**: Demonstrating discriminative capability across all risk segments for regulatory oversight.

Beyond finance, the framework applies to any domain with class imbalance, varying misclassification costs, or regulatory oversight requirements.

5.3.6 Limitations and Future Directions

The substantial difference between the theoretical (0.0187) and empirical (0.70) Gini indices reflects stabilization process impacts. While the empirical measure proves more practically relevant, this divergence requires careful interpretation. Moreover, discriminatory power is still indirectly influenced by the number of samples in a class. The whitening process introduces computational overhead compared to simple averaging approaches.

Future research should address:

- **Dataset Enhancement**: Future research should evaluate how the methodology performs across higher class numbers to fully demonstrate scalability, larger samples enabling robust minority class assessment, reduced multicollinearity, and high-dimensional data.
- **Deep Learning Extensions**: Adapting the framework for neural architectures with complex probability calibration.
- **Improved Numerical Integration**: Replacing trapezoidal approximation with smoother interpolation methods.
- **Broader Compliance**: Integration with Basel III/IV, IFRS 9, and ESG reporting standards.

This research demonstrates that discriminative power should take precedence over frequency considerations when evaluating multiclass classification systems. By providing a unified framework that preserves decision-theoretic properties while capturing multiclass complexity, our methodology offers a foundation for responsible AI deployment in high-stakes environments that increasingly rely on AI for critical decisions. Frameworks like ours become essential for maintaining regulatory compliance while enabling continued advancement of quantitative risk management and improved model decision-making through enhanced model transparency.

Bibliography

[1] Jesus S. Aguilar-Ruiz and Marcin Michalak. “Multiclass Classification Performance Curve”. In: *IEEE Access* 10 (2022), pp. 68915–68921.

[2] Gennaro Auricchio, Paolo G. Giudici, and Giuseppe Toscani. “Extending the Gini Index to Higher Dimensions via Whitening Processes”. In: *arXiv preprint arXiv:2409.10119* (2024). Submitted on 16 Sep 2024. URL: https://arxiv.org/abs/2409.10119.

[3] Golnoosh Babaei, Paolo Giudici, and Emanuela Raffinetti. “A Rank Graduation Box for SAFE AI”. In: *Expert Systems with Applications* 259 (2025), p. 125239. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.125239. URL: https://www.sciencedirect.com/science/article/pii/S0957417424021067.

[4] Basel Committee on Banking Supervision. *Basel III: Finalising post-crisis reforms*. Bank for International Settlements. 2017. URL: https://www.bis.org/bcbs/publ/d424.pdf.

[5] Davide Chicco and Giuseppe Jurman. “The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification”. In: *BioData Mining* 16.1 (2023), p. 4. DOI: 10.1186/s13040-023-00322-4.

[6] Jesse Davis and Mark Goadrich. “The Relationship Between Precision-Recall and ROC Curves”. In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML ’06. 2006, pp. 233–240. DOI: 10.1145/1143844.1143874.

[7] Alex Dmitriev and Edgar Dobriban. “A geometric proof of the equivalence between AUC ROC and Gini index area metrics for binary classifier performance assessment”. In: *arXiv preprint arXiv:2212.14541* (2022). URL: https://arxiv.org/abs/2212.14541.

[9] European Central Bank. *Report on financial structures*. ECB Statistical Data Warehouse. Statistics on fintech growth and SME lending in Europe. 2023.

[10] European Commission. *Regulation (EU) 2024/1689 of the European Parliament and of the Council on laying down harmonised rules on artificial intelligence*. Official Journal of the European Union. AI Act, entered into force 1 August 2024. 2024. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1689.

[11] Tom Fawcett. “Introduction to ROC analysis”. In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[12] C. Ferri, J. Hernandez-Orallo, and M.A. Salido. “Volume under the ROC Surface for Multi-class Problems”. In: *Machine Learning: ECML 2003*. Springer, 2003, pp. 108–120. DOI: 10.1007/978-3-540-39857-8_12.

[13] Peter Flach. “ROC Analysis”. In: *Encyclopedia of Machine Learning*. Springer, 2016, pp. 869–875.

[14] George Forman and Martin Scholz. “Apples-to-Apples in Cross-Validation Studies: Pitfalls in Classifier Performance Measurement”. In: *SIGKDD Explorations Newsletter* 12.1 (2010), pp. 49–57. DOI: 10.1145/1882471.1882479.

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

[16] Corrado Gini. "Measurement of inequality of incomes". In: *The Economic Journal* 31.121 (1921), pp. 124–125. DOI: 10.2307/2223319.

[17] Corrado Gini. "Sulla misura della concentrazione e della variabilità dei caratteri". In: *Atti del Reale Istituto Veneto di Scienze, Lettere ed Arti* 73 (1914), pp. 1203–1248.

[19] Steve Halligan, Douglas G. Altman, and Susan Mallett. "Disadvantages of using the area under the receiver operating characteristic curve to assess imaging tests: A discussion and proposal for an alternative approach". In: *British Journal of Radiology* 88.1041 (2015), p. 20150433.

[20] David J. Hand. "Measuring classifier performance: a coherent alternative to the area under the ROC curve". In: *Machine Learning* 77.1 (2009), pp. 103–123. DOI: 10.1007/s10994-009-5119-5.

[21] David J. Hand and Robert J. Till. "A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems". In: *Machine Learning* 45 (2001), pp. 171–186. DOI: 10.1023/A:1010920819831.

[22] Richard A Johnson and Dean W Wichern. *Applied Multivariate Statistical Analysis*. 6th. Upper Saddle River, NJ: Pearson Prentice Hall, 2007.

[23] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. "Optimal whitening and decorrelation". In: *The American Statistician* 72.4 (2018), pp. 309–314. DOI: 10.1080/00031305.2016.1277159.

[24] Thomas C.W. Landgrebe and Robert P.W. Duin. "Approximating the multiclass ROC by pairwise analysis". In: *Pattern Recognition Letters* 28.13 (2007), pp. 1747–1758. DOI: 10.1016/j.patrec.2007.03.010.

[25] Olivier Ledoit and Michael Wolf. "A Well-Conditioned Estimator for Large-Dimensional Covariance Matrices". In: *Journal of Multivariate Analysis* 88.2 (2004), pp. 365–411. DOI: 10.1016/S0047-259X(03)00096-4.

[26] Guoying Li and Jian Zhang. "Sphering and its properties". In: *Sankhyā: The Indian Journal of Statistics, Series A* 60.1 (1998), pp. 119–133.

[27] Max O. Lorenz. "Methods of measuring the concentration of wealth". In: *Publications of the American Statistical Association* 9.70 (1905), pp. 209–219. DOI: 10.2307/2276207.

[28] P.C. Mahalanobis. "On the Generalised Distance in Statistics". In: *Proceedings of the National Institute of Sciences of India* 2 (1936), pp. 49–55.

[30] Douglas Mossman. "Three-way ROCs". In: *Medical Decision Making* 19.1 (1999), pp. 78–89. DOI: 10.1177/0272989x9901900110.

[32] Juri Opitz and Sebastian Burst. "Macro F1 and Macro F1". In: *arXiv preprint arXiv:1911.03347* (2019).

[34] David M. W. Powers. "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation". In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63.

[35] Foster Provost and Pedro Domingos. “Tree Induction for Probability-Based Ranking”. In: *Machine Learning* 52.3 (2003), pp. 199–215. DOI: 10.1023/A:1024099825458.

[36] Takaya Saito and Marc Rehmsmeier. “The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets”. In: *PLOS ONE* 10.3 (2015), e0118432. DOI: 10.1371/journal.pone.0118432.

[37] Edna Schechtman and Gideon Schechtman. “The relationship between Gini terminology and the ROC curve”. In: *METRON* 77.6 (2019). DOI: 10.1007/s40300-019-00160-7.

[38] Si Shi et al. “Machine learning-driven credit risk: a systemic review”. In: *Neural Computing and Applications* 34.17 (2022), pp. 14327–14339. DOI: 10.1007/s00521-022-07472-2.

[39] Marina Sokolova and Guy Lapalme. “A systematic analysis of performance measures for classification tasks”. In: *Information Processing & Management* 45.4 (2009), pp. 427–437. DOI: 10.1016/j.ipm.2009.03.002.

Sitography

[8] Wikipedia The Fre Encyclopedia. *Rotation of axes in two dimensions*. Accessed: 2025-04-30. 2025. URL: https://en.wikipedia.org/wiki/Rotation_of_axes_in_two_dimensions.

[18] Google. *Classification: ROC and AUC*. Accessed: 2025-04-29. 2023. URL: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[29] Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Accessed: August 24, 2025. 2022. URL: https://christophm.github.io/interpretable-ml-book/.

[31] S. Narkhede. *Understanding Confusion Matrix*. Accessed: 2025-04-29. 2018. URL: https://medium.com/data-science/understanding-confusion-matrix-a9ad42dcfd62.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. *scikit-learn Decision Trees User Guide: Gini impurity*. Accessed: 2025-08-20. 2024. URL: https://scikit-learn.org/stable/modules/tree.html#gini-impurity.

List of Figures

List of Tables

Appendices

Multiclass ROC curve for the other models considered in the study

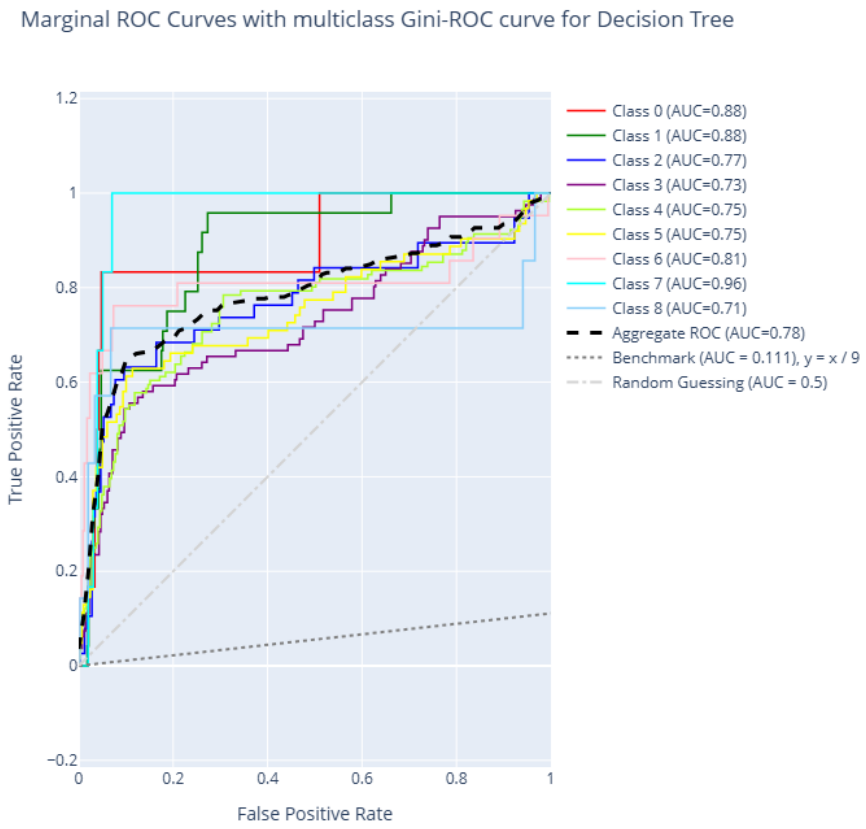

Figure 5: Per-class ROC curves and multiclass ROC curve comparison for the Decision Tree

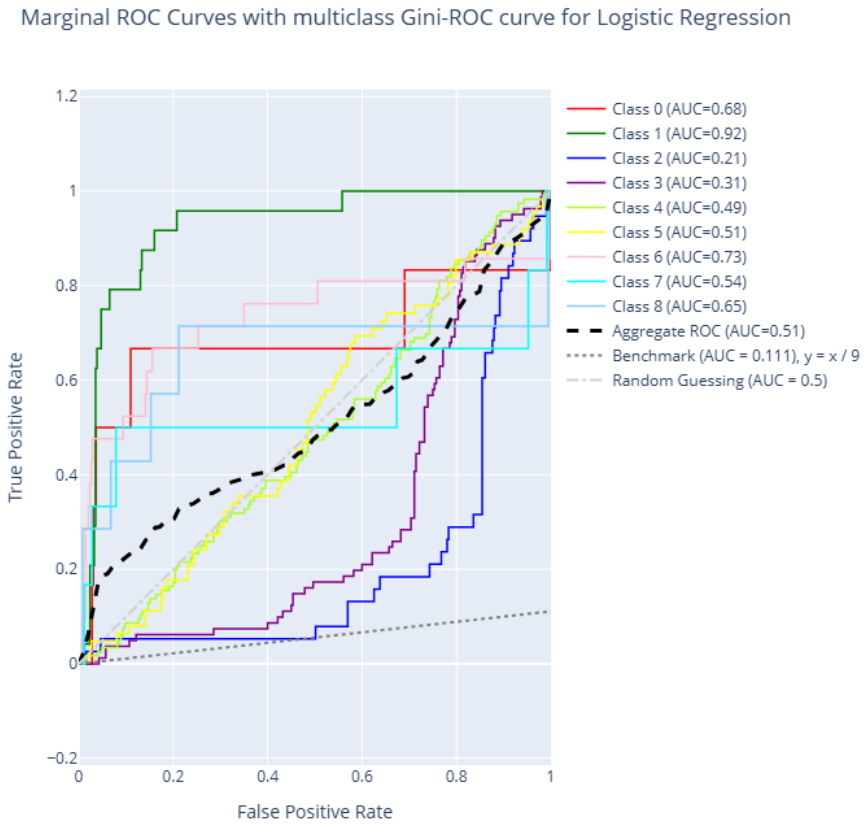

Figure 6: Per-class ROC curves and multiclass ROC curve comparison for the Logistic Regression

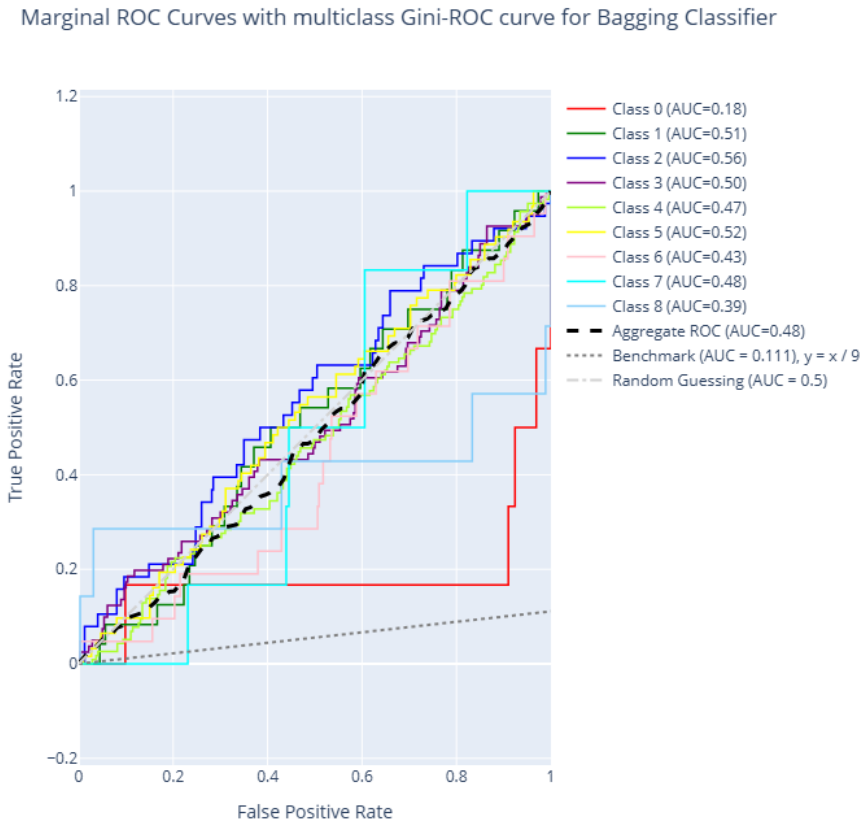

Figure 7: Per-class ROC curves and multiclass ROC curve comparison for the Bagging Classifier