COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

BACHELOR'S DEGREE IN
MATHEMATICAL ENGINEERING
AND ARTIFICIAL INTELLIGENCE

BACHELOR'S THESIS

UNCERTAINTY-AWARE CLASSIFICATION FOR ANTHROPOLOGICAL DATA

Author: Ignacio Bayón Jiménez-Ugarte

Director: Simón Rodríguez Santana

Madrid, August 2025

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería ICAI
Bachelor's Degree in Mathematical Engineering and Artificial Intelligence

I hereby declare, under my own responsibility, that the Project presented with the title

Uncertainty-Aware Classification for Anthropological Data

at the High Technical School of Engineering ICAI of Universidad Pontificia Comillas in the academic year **2024/25** is my own work, original and unpublished, and has not been previously submitted for any other purpose.
The Project is not a copy of someone else's work, neither totally nor partially, and any information taken from other documents has been properly referenced.

Signed: **Ignacio Bayón Jiménez-Ugarte** Date: .....

Project submission authorized
PROJECT SUPERVISOR

Signed: **Simón Rodríguez Santana** Date: .....

UNIVERSIDAD PONTIFICIA COMILLAS
Escuela Técnica Superior de Ingeniería ICAI
Bachelor's Degree in Mathematical Engineering and Artificial Intelligence

Uncertainty-Aware Classification for Anthropological Data

Author: Ignacio Bayón Jiménez-Ugarte
Director: Simón Rodríguez Santana
Collaborating Entity: ICAI – Universidad Pontificia Comillas

Abstract

This work applies Probabilistic Machine Learning (PML) to the classification of anthropological data from dental microwear texture analysis (DMTA) in Cercopithecoidea primates. Given the dataset's small size and inherent noise, Bayesian approaches are employed to model uncertainty and improve robustness. Three variable sets (SSFA, ISO, Other) are compared, alongside dimensionality reduction and variable selection, to identify the most informative features. Classical models, neural networks, and Bayesian logistic regression are evaluated using cross-validation. Results highlight the advantages of uncertainty-aware models, providing interpretable predictions and reduced feature subsets useful for anthropological research.

Keywords: Probabilistic Machine Learning (PML), Bayesian Inference, Dental Microwear Texture Analysis (DMTA), Feature Selection, Markov Chain Monte Carlo (MCMC),

1. Introduction

In dental microwear texture analysis (DMTA), used to infer diet and habitat in Cercopithecoidea primates, traditional machine learning often struggles with small, noisy datasets. Probabilistic approaches, by contrast, explicitly model uncertainty, leading to more robust inference and integration of prior knowledge. This project applies such methods to address the challenges of limited, noisy data, and improve interpretability.

2. State of the Art

DMTA is widely used in anthropology to infer diet and ecology from enamel surface roughness, through ISO, SSFA, and texture descriptors. Classical analyses rely on ANOVA, non-parametric tests, and PCA, but these focus on group comparisons and lack predictive power [1]. Probabilistic Machine Learning (PML) addresses these limitations by modeling parameter and predictive uncertainty. Here, a Multinomial Bayesian Logistic Regression with Markov Chain Monte Carlo (MCMC) is implemented to enable robust classification, uncertainty-aware predictions, and the evaluation of variable informativeness.

3. Methodology and Experimental Design

The dataset used in this study is both small (≈ 100 samples) and noisy, reflecting the inherent challenges of anthropological data collection. Such conditions make robust preprocessing and

uncertainty-aware modeling indispensable. The data consist of DMTA variables for Cercopithecoidea primates, organized into three sets: ISO metrics, Scale-Sensitive Fractal Analysis (SSFA), and additional texture descriptors. These sets, as well as their combinations, are systematically evaluated to determine their discriminative power.

Preprocessing begins with normality testing, transformation of non-Gaussian variables, and outlier removal to stabilize distributions. Dimensionality reduction methods—PCA, LDA, UMAP, and t-SNE—are employed both for visualization and to detect hidden structure in the feature space. This step is especially relevant for small datasets, where redundancy among variables can obscure meaningful patterns.

Classification experiments follow a staged design. First, classical machine learning models (LDA and QDA), LogReg, RFs, and NB are applied to establish baseline performance. Neural networks are also explored, though their reliance on large datasets limits their effectiveness in this context.

Final part of the pipeline is the implementation of a Multinomial Bayesian Logistic Regression. Here, prior distributions play a central role in incorporating expert knowledge and regularizing inference under scarce data. Three prior families are tested: Gaussian priors, which assume simple normal variability; Laplace priors, encouraging sparsity and feature selection; and Spike-and-Slab priors, which combine strong shrinkage for irrelevant variables with flexibility for informative ones [2, 3]. Posterior distributions are inferred using Markov Chain Monte Carlo (MCMC), providing probabilistic predictions accompanied by measures of uncertainty.

Evaluation is carried out through rigorous cross-validation, including both Leave-One-Out (LOO) and K-Fold schemes. Metrics such as accuracy, F1-score, and AUC-ROC are used alongside posterior predictive checks. This pipeline ensures that models are not only quantitatively reliable but also interpretable, a key requirement for anthropological application under noisy and data-limited conditions.

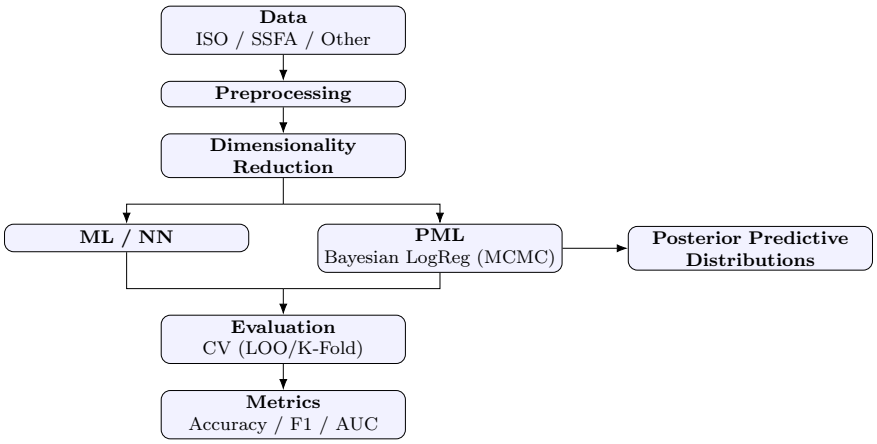


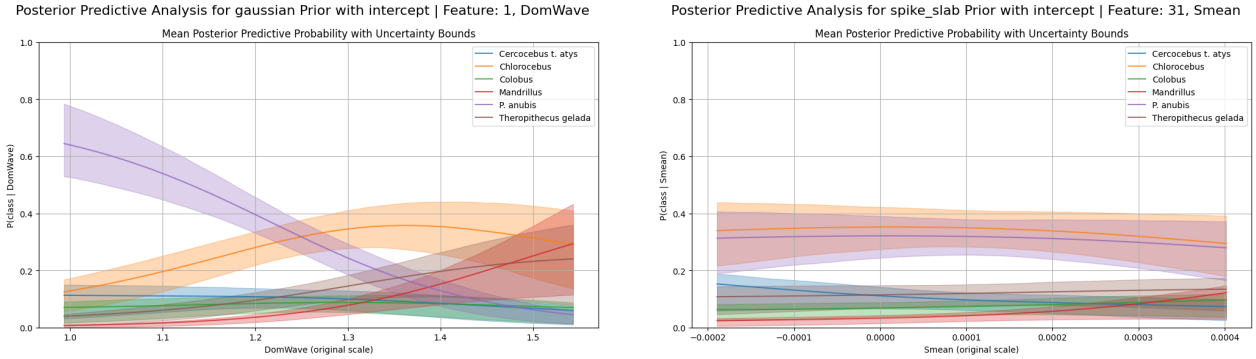
Figure 1: Compact Pipeline

4. Results and Analysis

We evaluated Multinomial Bayesian LogReg with Gaussian, Laplace, and Spike-and-Slab priors across ISO, SSFA, and Other variables. ISO features achieved slightly higher median accuracy, though performance varied considerably across folds due to the small, noisy dataset. The Spike-and-Slab prior consistently performed well, particularly with reduced feature sets.

Feature selection proved beneficial: using the top 10 or 32 variables improved stability and often raised accuracy compared to all features. The intercept term had little effect on accuracy but enhanced interpretability. Without it, predictions collapsed to uniform probabilities when inputs were near zero; with it, the model learned realistic baselines.

Posterior predictive distributions (PPDs) illustrate this effect. With informative variables, a single feature value can strongly indicate a class (e.g., low DomWave $\approx$ P. anubis (2a)). In contrast, variables lower in the ranking produce flatter, less informative distributions (2b), correctly reflecting greater model uncertainty. Overall, Bayesian models matched classical performance while adding uncertainty estimates and interpretability, offering a robust framework for small anthropological datasets.



(a) PPD for DomWave (Gaussian prior).

(b) PPD for Smean (Spike-and-Slab prior).

Figure 2: Comparison of PPDs: informative vs. less informative features.

5. Conclusions and Future Work

This work applied PML to DMTA, demonstrating that Bayesian LogReg provides robust classification under noisy, small-sample conditions. Feature selection improved stability, and posterior predictive distributions offered interpretable insights, showing how single variables can drive confident classifications. Compared with classical approaches, Bayesian methods matched accuracy while adding uncertainty quantification and greater transparency.

Future work could extend these methods to larger datasets, test alternative probabilistic models such as Gaussian processes or Bayesian neural networks, and explore hierarchical structures to incorporate phylogenetic or ecological priors, further enhancing interpretability and anthropological applicability.

References (Summary)

[1] Ghislain Thiery et al. "Introducing trident: a graphical interface for discriminating groups using dental microwear texture analysis". In: *Peer Community Journal* 4.e90 (2024). DOI: 10.24072/pcjournal.467.

[2] Hemant Ishwaran and J. Sunil Rao. "Spike and slab variable selection: Frequentist and Bayesian strategies." In: *Annals of Statistics* 33.2 (2005), pp. 730–773. DOI: 10.1214/009053604000001147.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006. ISBN: 978-0387-31073-2.

Contents

1 Introduction

1.1 Context and Motivation

In anthropological and evolutionary studies, a recurring challenge is that many available specimens lack clear species or dietary classification. Similarly, numerous fossil remains from related species provide little information about their diets, even though this knowledge would help reconstruct the environments in which they lived. To address this gap, robust and reliable classification models are needed that can infer dietary patterns from measurable variables in dental remains. In this context, Dental Microwear Texture Analysis (DMTA) has become a powerful tool, as it analyzes microscopic wear patterns on teeth to reveal connections between diet and ecological behaviors, particularly in African primates of the Cercopithecoidea family

The dataset we work with in this project comprises detailed measurements of dental surface textures collected from 100 primate specimens. These features are grouped into three main sets: *International Standarization Organization (ISO)* features, *Scale-Sensitive Fractal Analysis (SSFA)* features, and a third group of additional texture descriptors.

1.2 Problem Statement

Despite the potential of DMTA for species classification, the available dataset poses significant analytical challenges. It is relatively small in size, exhibits high noise levels, and contains substantial feature redundancy and multicollinearity. These conditions reduce the reliability of standard machine learning (ML) models, which typically rely on large, well-conditioned datasets and often function as black boxes with limited interpretability.

In anthropological applications, where interpretability, robustness, and confidence in predictions are essential, conventional ML approaches are insufficient. There is a clear need for classification frameworks that can not only handle the limitations of the data but also incorporate expert priors and provide uncertainty-aware predictions. This thesis addresses these needs by exploring Probabilistic Machine Learning (PML) methods as a more suitable alternative.

1.3 Objective of the Project

The main objective of this project is to develop a robust and interpretable classification framework capable of handling a complex and limited dataset, such as the DMTA dataset we are working with. To that end, the first step involves designing a comprehensive data processing pipeline that addresses key challenges such as non-normal feature distributions, outliers, and multicollinearity. This preprocessing phase is critical to ensure that subsequent modeling efforts are both statistically sound and biologically meaningful.

Following this, we evaluate the performance of several baseline machine learning classifiers, including both linear and nonlinear models, as a benchmark. We then extend this analysis to neural networks to test whether their increased modeling capacity can provide improvements despite the dataset's small size. The core focus of the study, however, lies in the implementation of Probabilistic Machine Learning (PML) techniques, with a particular emphasis on Multinomial Bayesian Logistic Regression. These

models allow us to quantify uncertainty in predictions and incorporate prior knowledge, which is especially valuable in anthropological contexts.

A central goal of the project is to compare the performance and interpretability of PML approaches against traditional ML methods, highlighting the advantages of uncertainty-aware modeling. In doing so, we aim to (i) determine which of the three feature sets—ISO, SSFA, or Other—offers the most predictive value for species classification, and (ii) identify the most informative individual features, regardless of their group, to support minimal yet effective models for practical use in fieldwork.

1.4 Structure of the Thesis

This thesis is structured into seven chapters, each addressing a key aspect of the research process:

1. **Introduction:** Presents the context and motivation for the study, outlines the main challenges of the dataset, defines the objectives of the project, and provides a roadmap of the thesis.

2. **State of the Art:** Reviews the current approaches in dental microwear texture analysis (DMTA) and highlights the potential of probabilistic machine learning methods for uncertainty-aware classification in anthropology.

3. **Methodology and Experimental Design:** Describes the dataset and preprocessing steps, introduces the classification models considered (traditional ML algorithms, neural networks, and probabilistic models), and explains the experimental setup used for training and evaluation.

4. **Results and Analysis:** Presents the comparative performance of the different modeling approaches, analyzes feature relevance, and discusses the added value of uncertainty quantification in the probabilistic framework.

5. **Conclusions and Future Work:** Summarizes the main contributions of the thesis, reflects on its limitations, and proposes directions for future research on the use of probabilistic methods in anthropological data analysis.

2 State of the Art

2.1 Current Approaches in DMTA

The analysis of dental microwear texture (DMTA) has become a standard tool in anthropology for inferring dietary patterns and ecological adaptations in primates and other mammals. The methodology relies on quantifying surface roughness parameters from 3D scans of tooth enamel, with variable sets such as ISO metrics, Scale-Sensitive Fractal Analysis (SSFA), and other texture descriptors. These parameters are then compared across species or dietary groups to assess ecological or functional differences.

Traditionally, DMTA studies have relied on classical statistical analyses. Analysis of Variance (ANOVA) is commonly applied to test whether mean values of microwear variables differ significantly between

dietary groups. When assumptions of normality or homoscedasticity are not met, nonparametric tests such as Kruskal-Wallis are employed, often followed by post-hoc comparisons (pairwise or average rank comparisons) to identify which groups differ. For exploratory visualization, Principal Component Analysis (PCA) is frequently used to reduce dimensionality and to highlight clustering or separation among species.[1]

While these methods have provided valuable insights, they share important limitations. First, they are primarily designed for group-level comparisons, rather than predictive classification. Second, they typically produce point estimates without explicit uncertainty quantification, which restricts their interpretability in contexts where data are inherently noisy and sample sizes are small. Finally, variable selection is often ad hoc, with limited systematic evaluation of the relative informativeness of ISO, SSFA, and texture parameters.

As a result, although DMTA has proven effective in distinguishing broad dietary categories, current techniques remain anchored in traditional frequentist statistics, and more advanced approaches capable of addressing uncertainty, variable redundancy, and predictive robustness have yet to be systematically applied.

2.2 Probabilistic Machine Learning and Uncertainty-Aware Models

Recent advances in Machine Learning, particularly probabilistic approaches, have not yet been fully leveraged in the context of DMTA, despite their potential to address many of the limitations of traditional statistical techniques. Probabilistic Machine Learning (PML) provides a principled framework for modeling uncertainty in data and in model predictions. This is especially valuable in scientific domains such as anthropology, where datasets are typically small, noisy, and collected under variable conditions. Traditional Machine Learning (ML) models often fail to adequately capture these challenges, offering high-capacity prediction functions but with limited interpretability or robustness in low-data regimes.

2.2.1 Uncertainty in Classification

In standard classification pipelines, predictions are deterministic and model parameters are treated as fixed quantities. This approach provides no notion of confidence or uncertainty in the output. In contrast, PML techniques, grounded in Bayesian inference, represent both model parameters and predictions as probability distributions. This allows for uncertainty quantification at both the epistemic level (model uncertainty due to limited data) and the aleatoric level (inherent noise in the data).

In this project, we primarily implement and analyze a *Multinomial Bayesian Logistic Regression* model. This model is well-suited for multi-class classification tasks and allows us to incorporate expert knowledge through prior distributions. Instead of computing point estimates for the model parameters, we infer a posterior distribution using sampling techniques from the Markov Chain Monte Carlo (MCMC) family. These inference methods enable the estimation of predictive distributions, providing both a class label and a measure of certainty in each pre-

diction.

2.2.2 Summary of Tools and Techniques

The methodology implemented in this thesis draws from the following pillars:

- **Bayesian Inference:** Using MCMC to approximate posterior distributions over model parameters.
- **Model Evaluation:** Employing metrics such as accuracy, F1-score, and AUC-ROC, alongside posterior predictive distributions to assess uncertainty.
- **Feature Comparison:** Evaluating the discriminative power of the ISO, SSFA, and Other variable sets, both jointly and independently.
- **Dimensionality Reduction:** Utilizing PCA, LDA, UMAP, and t-SNE for visualization and structure discovery in the feature space.

This technical foundation positions the thesis at the intersection of probabilistic modeling, feature selection, and real-world anthropological application, advancing both the methodology and its practical utility.

3 Methodology and Experimental Design

The methodological framework of this thesis is designed to address the unique challenges posed by the DMTA dataset: small sample size, high dimensionality, noise, and correlated features. To ensure both robustness and interpretability, the approach integrates data preprocessing, feature selection, model specification, and evaluation into a unified experimental pipeline. This pipeline begins with a detailed characterization and cleaning of the dataset, continues with the implementation of a range of classification models—from traditional machine learning baselines to shallow neural networks and probabilistic approaches—and culminates in rigorous evaluation through cross-validation and uncertainty-aware performance metrics. By structuring the methodology and experiments together, we emphasize that the experimental design is inseparable from the preprocessing and modeling decisions that shape the validity of the results.

3.1 Dataset and Preprocessing

The dataset used in this project is a collection of dental microwear texture measurements from 100 primate specimens, specifically from the Cercopithecoidea family. The data is organized into three main groups of features: *International Standardization Organization (ISO)* features, *Scale-Sensitive Fractal Analysis (SSFA)* features, and a third group of additional texture descriptors. The original dataset includes 152 features, from which 63 are a combination of other features. Therefore, after discarding these redundant features which do not add information to our analysis, we are left with 89 features: 57 ISO, 24 SSFA, and 8 Other features. As we can see, there is a significant differencee in the number of features between sets, which is a key point to consider in our analysis.

3.1.1 Set Colinearity

The first step in our preprocessing pipeline is to analyze the collinearity between the different feature sets. A quick analysis reveals that the ISO and SSFA sets are correlated, with a mean correlation coefficient accross all features of approximately 0.36. This suggests that not all features in these sets are independent and that some may be redundant, which highlights the need for careful feature selection in our modeling process. Below, we can see the Density of Correlations between feature sets. For more detail, please refer to subsubsection A.1.2

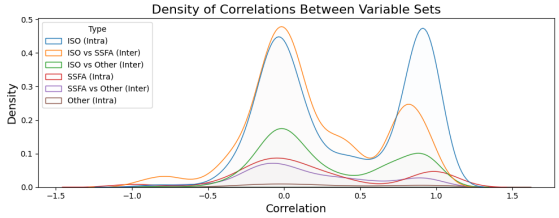


Figure 1: Density of Correlations between sets of variables

3.1.2 Preprocessing

An important factor to consider is the distribution of the features. Running the Anderson-Darling test oon the features reveals that 73% of the ISO features, 90% of the SSFA features, and 75% of the Other features do not follow a Gaussian. To address this, we opted for the Yeo-Johnson transformation [2], which is a generalization of the Box-Cox transformation that can handle zero and negative values. It is defined as follows:

$$y = \begin{cases} \frac{((x+1)^{\lambda}-1)}{\lambda} & \text{if } \lambda \neq 0, x \geq 0 \\ \log(x+1) & \text{if } \lambda = 0, x \geq 0 \\ \frac{-((|x|+1)^{2-\lambda}-1)}{2-\lambda} & \text{if } \lambda \neq 2, x < 0 \\ -\log(|x|+1) & \text{if } \lambda = 2, x < 0 \end{cases} \tag{3.1}$$

However, when applying this transformation, we found that some variables which were tightly clustered around 1 were transformed into disproportionately large values, which could lead to numerical instability in the models. To mitigate this, we later applied standardization to the transformed features, ensuring that their scale does not affect the model training.

To ensure the robustness of the modeling pipeline, we implemented a z-score-based outlier filter. The function computes the standard z-score for each variable (excluding near-constant features), and removes any observation that exceeds a specified threshold (set to 3 by default) in absolute value for at least one feature. We also tried winsorizing the features, but this approach did not yield significant improvements in model performance, so we opted for the z-score filter instead. This choice is justified by the fact that the dataset is relatively small, and the z-score filter effectively removes extreme outliers without distorting the feature distributions.

Dimensionality Reduction Techniques: LDA and PCA

In high-dimensional datasets such as the one used in this study, dimensionality reduction techniques play a crucial role in improving both model performance and interpretability. Two prominent methods applied in this thesis are *Principal Component Analysis (PCA)* and *Linear Discriminant Analysis*

(LDA). The techniques were applied to gain insight into the structure of the feature space and to identify variables with the greatest discriminative or variance-explaining power.

Principal Component Analysis (PCA) revealed that variables such as `SRC`, `Sha`, and `Sda` have the strongest influence on global variance, as they dominate the projection space defined by the first two principal components. These variables consistently rank among the top ten in terms of absolute loadings, suggesting their relevance across multiple directions of variance. However, the PCA projection does not reveal any clear class separation, which is consistent with the unsupervised nature of the method and its focus on variance rather than class structure.

In contrast, Linear Discriminant Analysis (LDA) exposes substantial discriminative structure across the groups. Variables such as `SRCth`, `RLy`, and `Smean` exhibit high coefficients along the first two linear discriminants (LD1 and LD2), marking them as key contributors to intergroup separation. This is visually evident in the corresponding heatmaps and projection plots provided in Appendix A.1.

These results guided subsequent modeling choices by identifying candidate variables with high relevance, both for variance explanation and class separability. They also support the construction of reduced-dimensional visualizations and informed variable selection in the classification pipelines.

Feature Selection

As we discussed earlier, one of the key objectives of the project is to select the most relevant / representative features. To compute the feature importance ranking, we trained a Multinomial Logistic Regression model with Lasso Regularization (3.2.1). Then, we sorted the absolute values of the coefficients, as a higher absolute value indicates a greater importance of the corresponding feature in the model.

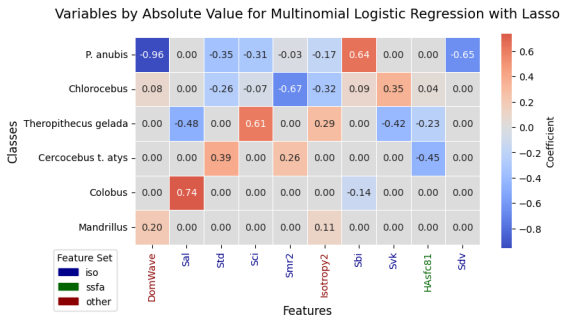


Figure 2: Top 10 Features by Absolute Value for Multinomial Logistic Regression with Lasso

Figure 2 shows the top 10 variables ranked by the absolute sum of their coefficients across all classes. For the complete 32 variable ranking, please refer to Figure A.5 in Appendix A.1.

Table 1: Number of variables from each set in the full dataset, Top 32, and Top 10 rankings.

Variable Set	Total Variables	Top 32	Top 10
ISO	57	19	8
SSFA	24	9	1
Other	8	5	2

The distribution of variable sets across the rankings reveals several important patterns. In the full dataset, the ISO set dominates with 57 variables (64%), followed by SSFA with 24 (27%) and Other with 8 (9%). However, in the top 32 ranking, the ISO and SSFA sets contribute with 19 (59%), and 9 (28%) variables, with Other providing the remaining 5 (16%), which are similar numbers

relative to their overall representation in the dataset. However, when we focus on the top 10 ranking, the disparity becomes more pronounced: ISO features account for 8 of the top 10, while SSFA contributes just 1 and Other contributes 2. This trend highlights the dominance of ISO features. Another important observation is that the most informative feature is `DomWave`, from the extra features, which indicates it should clearly be included in any model despite not being part of the traditional feature sets.

3.2 Classification Models

This section outlines the classification models evaluated in this study, divided into two categories for clarity: traditional machine learning algorithms and shallow neural networks. While both fall under the broader umbrella of supervised machine learning, their implementation and characteristics differ significantly.

3.2.1 Traditional Machine Learning Models

The first group of models consists of "Traditional" ML algorithms, which are well-established in the field and provide a solid baseline for comparison. These models include:

- **Linear Discriminant Analysis (LDA):** A linear classifier that projects data onto a lower-dimensional space to maximize class separability.
- **Quadratic Discriminant Analysis (QDA):** An extension of LDA that allows for non-linear decision boundaries
- **Logistic Regression (LogReg):** A linear model for binary classification, which can be extended to multi-class problems. We also used the Lasso and ElasticNet variants, which add regularization to the basic logistic regression model.
- **Gaussian Naive Bayes (NB):** A probabilistic classifier based on Bayes' theorem, assuming independence betweenfeatures.
- **Random Forest (RF):** An ensemble method that builds multiple decision trees and combines their predictions.
- **XGBoost:** A gradient boosting framework that builds an ensemble of decision trees in a stage-wise fashion. Each new tree is trained to correct the errors of the previous ones, optimizing a differentiable loss function. It is widely used for its speed, regularization capabilities, and strong predictive performance on structured data.
- **NGBoost:** A gradient boosting framework that extends XGBoost by modeling the distribution of the target variable, rather than just point estimates.

3.2.2 Neural Networks

Given the relatively small number of observation and the moderate number of input features, a shallow neural network architecture was chosen for this study. Deeper networks, while more expressive, require substantially more data to generalize well and are prone to overfitting in low-data regimes.

In contrast, shallow networks, with fewer parameters, offer a better bias-variance trade-

off under these conditions. The architecture used consists of two fully connected hidden layers with a low number of neurons (32, 16 or 8). We decided to use the ReLU activation function, which is a common choice for hidden layers in neural networks due to its simplicity and effectiveness. We also implemented an optional dropout layer to mitigate overfitting. As for the loss, we used the cross-entropy loss function, which is standard for multi-class classification tasks. The model was trained using the Adam optimizer with weight decay, which proved to be effective in this context.

Given the limited number of data points, it was not feasible to reserve a separate validation set or rely on a standard train/test/-validation split. Each observation was valuable, and setting aside even a small portion for validation would have significantly reduced the training signal. To address this, we employed a *K-Fold Cross-Validation* strategy during training. Although computationally expensive, this approach allowed us to maximize the use of all available data by repeatedly rotating the test set across folds. This not only produced more robust performance estimates but also reduced the risk of certain classes being excluded from evaluation, which is particularly important in a multi-class classification task with limited samples per class.

The best performing neural network model achieved an accuracy of 0.41 with K=10. However, to get the best approximation of the performance of the model, we performed Leave-One-Out Cross-Validation (LOOCV), which yielded an accuracy of 0.34. To better understand the performance, we plotted the confusion matrix and the classification report:

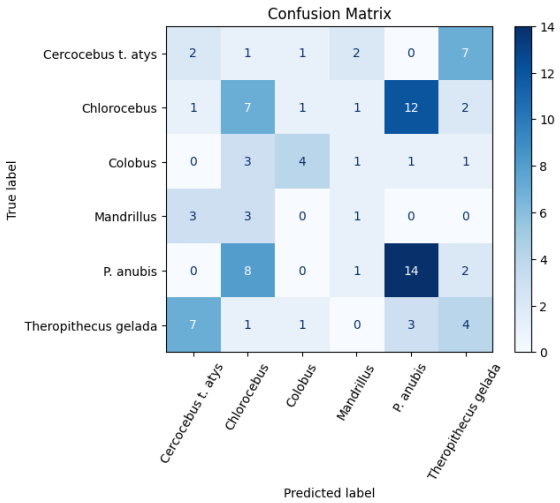


Figure 3: Confusion Matrix for the Neural Network Model

As we can see, the class imbalance is evident, which affects the model's performance. The model struggles to correctly classify the less represented classes and tends to confuse the two most represented classes, *Chlorocebus* and *P. anubis*.

Metric	**Score**
Accuracy	0.3368
Precision	0.3353
Recall	0.3368
F1-score	0.3333
ROC AUC Score	0.6993

Table 2: NN evaluation metrics.

The results indicate that the neural network model, while capable of learning complex patterns, does not outperform the traditional techniques in this specific context. The accuracy of 0.34 is comparable to the traditional models, but as NNs are more complex and less interpretable, they are not the best choice for our usecase.

For further details, including the full classification report (*Table A.1*) and ROC curves (*Figure A.9*), refer to Appendix *A.2*. For the code, refer to the jupyter notebook *classification_nn.ipynb* in the Github repository.

3.3 Probabilistic Machine Learning Models

3.3.1 Introduction to Multinomial Bayesian Logistic Regression

The core of this thesis is the implementation and analysis of a *Multinomial Bayesian Logistic Regression* (MBLR) model (*Bishop 2006, 4.5. Bayesian Logistic Regression* [3]). This model is particularly well-suited for multiclass classification tasks, as it extends the traditional logistic regression framework to handle multiple classes in a probabilistic manner. The Bayesian approach provides a posterior distribution over the model parameters, allowing us to quantify uncertainty in predictions.

3.3.2 Model Formulation

Let $\mathbf{x} \in \mathbb{R}^D$ denote a feature vector and $y \in \{1, \ldots, K\}$ its corresponding class label, where K is the number of possible classes. In the multinomial logistic regression model, the probability that an observation $\mathbf{x}$ belongs to class k is given by the softmax function:

$$P(y = k \mid \mathbf{x}, \Theta) = \frac{\exp(\boldsymbol{\theta}_k^\top \mathbf{x})}{\sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^\top \mathbf{x})}$$

where $\boldsymbol{\theta}_k \in \mathbb{R}^D$ is the weight vector for class k, and $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ is the collection of all weight vectors.

Given a dataset $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$, the likelihood of the data under the model is:

$$p(\mathcal{D} \mid \Theta) = \prod_{i=1}^{N} P(y^{(i)} \mid \mathbf{x}^{(i)}, \Theta)$$

Bayesian Perspective

In the Bayesian formulation, we place a prior distribution over the parameters Θ and use Bayes' theorem to obtain a posterior distribution:

$$p(\Theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \Theta)\, p(\Theta)}{p(\mathcal{D})}$$

Here, $p(\Theta)$ encodes our prior beliefs about the weights, and $p(\mathcal{D})$ is the marginal likelihood (also known as the evidence), which acts as a normalization constant.

This posterior distribution reflects the updated belief about the model parameters after observing the data and forms the basis for both prediction and uncertainty quantification. In practice, since the denominator $p(\mathcal{D})$ is intractable, we resort to approximate inference methods such as Markov Chain Monte Carlo (MCMC) or Variational Inference (VI).

Motivating the Intercept Term

The model described so far does not include any intercept term. This implies that the decision boundaries are constrained to pass through the origin in feature space. However, in many practical situations—especially when features are standardized to have zero mean—it is beneficial to allow the model to learn a class-specific baseline log-odds.

To achieve this, we introduce an intercept (or bias) term $b_k \in \mathbb{R}$ for each class. The predictive distribution then becomes:

$$P(y = k \mid \mathbf{x}, \Theta, \mathbf{b}) = \frac{\exp(\boldsymbol{\theta}_k^\top \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\boldsymbol{\theta}_j^\top \mathbf{x} + b_j)}$$

where $\mathbf{b} = [b_1, \ldots, b_K]$ is the vector of class-specific intercepts. These intercepts are treated as additional parameters in the Bayesian framework, and priors are placed on them as well. In our implementation, we have chosen a Gaussian prior for the intercepts, which allows them to adapt flexibly to the data while preventing the possibility of zeroing out the intercept, which could be the case if we used a Laplace prior, for instance.

3.3.3 Sampling Methods

To better understand the Bayesian inference process, we implemented various sampling methods from the MCMC family, including the Metropolis-Hastings and Gibbs sampling algorithms (*Bishop 2006, 11. Sampling Methods* [3]). These methods allow us to approximate the posterior distribution of the model parameters Θ and $\mathbf{b}$ by generating samples from the posterior distribution. The choice of sampling method can significantly affect the convergence speed and quality of the posterior approximation.

Metropolis-Hastings Sampling

The *Metropolis-Hastings* algorithm is an MCMC method used to generate samples from a target probability distribution when direct sampling is intractable. Given a target distribution $p(x)$ and a proposal distribution $q(x'|x)$, the algorithm iteratively generates a new sample x' from the proposal distribution and accepts or rejects it based on the acceptance probability:

$$\alpha = \min\left(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right) \tag{3.2}$$

If the sample is accepted, it becomes the next sample in the chain; otherwise, the current sample is retained. This process continues for a specified number of iterations, allowing the Markov Chain to approximate the target distribution.

Gibbs Sampling

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method used to draw samples from a joint distribution $p(x_1, x_2, \ldots, x_d)$ when direct sampling is infeasible. It is especially useful when the full conditional distributions $p(x_i \mid x_{-i})$, where x_{-i} denotes all variables except x_i, are available in closed form.

The joint distribution encodes the full probabilistic structure of the model, while the conditional distributions represent how each variable depends on the others. Gibbs sampling iteratively samples from these conditionals to generate samples from the joint.

The algorithm proceeds as follows:

1. **Define the model:** Specify the joint distribution $p(x_1, x_2, \ldots, x_d)$ and derive the full conditionals $p(x_i \mid x_{-i})$ for each variable.

2. **Initialization:** Choose an initial value $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_d^{(0)})$. This may be randomly sampled or based on prior information.

3. **Iterative sampling:** At each iteration t, generate a new sample $x^{(t+1)}$ by sequentially sampling each variable from

its conditional distribution:

$$x_1^{(t+1)} \sim p(x_1 \mid x_2^{(t)}, \ldots, x_d^{(t)})$$

$$x_2^{(t+1)} \sim p(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \ldots, x_d^{(t)})$$

$$\vdots$$

$$x_d^{(t+1)} \sim p(x_d \mid x_1^{(t+1)}, \ldots, x_{d-1}^{(t+1)})$$

4. **Repeat:** Iterate this process for a fixed number of steps. To ensure convergence to the target distribution, an initial portion of samples (the *burn-in*) is typically discarded.

Because each sample is drawn from the exact conditional distribution, all proposals are accepted by construction. As the number of iterations increases, the samples produced by the algorithm converge in distribution to the true joint distribution $p(x_1, x_2, \ldots, x_d)$.

3.3.4 Prior Distribution Selection

A crucial component of Bayesian modeling is the choice of prior distributions over the parameters. The prior encodes our beliefs about the parameter values before observing the data and has a direct influence on the posterior, particularly in low-data or high-dimensional settings. In this work, we explore and compare three types of priors with distinct characteristics: Gaussian, Laplace, and Spike-and-Slab (*Murphy 2012, 8. Logistic Regression [4]*).

Gaussian Prior

The Gaussian prior is the most commonly used due to its mathematical convenience and conjugacy properties. It assumes that each parameter θ_i follows a normal distribution:

$$\theta_i \sim \mathcal{N}(0, \sigma^2)$$

This prior expresses a belief that the parameters are likely to be near zero, but allows for moderate deviations. The variance σ^2 controls the strength of regularization: smaller values induce stronger shrinkage. Gaussian priors are appropriate when we expect the parameters to be small but not necessarily sparse.

Laplace Prior

The Laplace prior, also known as the double exponential prior, is defined as:

$$\theta_i \sim \text{Laplace}(0, b) = \frac{1}{2b} \exp\left(-\frac{|\theta_i|}{b}\right)$$

This prior has heavier tails and a sharp peak at zero compared to the Gaussian, promoting sparsity in the posterior. As it tends to drive many parameter estimates exactly to zero, the Laplace prior is therefore suitable when we expect that only a small subset of features are relevant.

Spike-and-Slab Prior

The Spike-and-Slab prior is a mixture distribution designed to explicitly model sparsity [5]. It combines a point mass at zero (the spike) with a continuous distribution (the slab), typically Gaussian:

$$\theta_i \sim \pi \cdot \mathcal{N}(0, \tau^2) + (1 - \pi) \cdot \delta_0$$

where δ_0 denotes the Dirac delta function at zero, and $\pi \in [0, 1]$ controls the prior probability of inclusion. This prior models uncertainty about whether a parameter is active

(non-zero) or not, and is particularly effective for variable selection in high-dimensional settings. While it provides strong interpretability and sparsity, inference is more complex due to the non-conjugate and discrete nature of the spike component.

Each of these priors imposes different inductive biases on the model. In practice, we evaluate their impact by comparing predictive performance, sparsity levels, and uncertainty calibration in the resulting posterior distributions.

To better understand the behavior of each prior, we plot the probability density functions of the Gaussian, Laplace, and Spike-and-Slab distributions centered at zero.

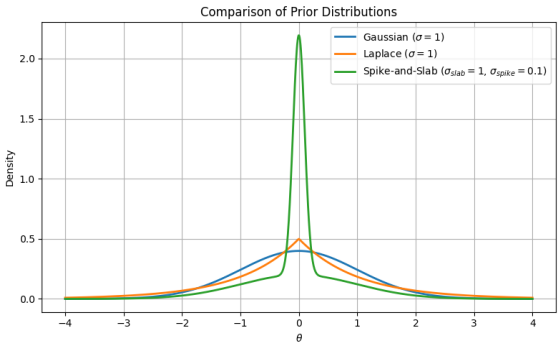


Figure 4: Comparison of prior distributions.

3.3.5 Connection Between Priors and Regularization

In Bayesian models, the choice of prior distribution over the parameters plays a role analogous to regularization in frequentist frameworks. Specifically, imposing a prior can be interpreted as introducing a penalty term in the optimization objective, and different priors correspond to different types of regularization [6].

Gaussian Prior and Ridge Regularization

1. Bayesian Inference

In Bayesian inference, we're interested in the posterior distribution over the parameters given the data:

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$$

Where $p(\mathcal{D} \mid \theta)$ is the likelihood of the data given the parameters, and $p(\theta)$ is the prior distribution.

2. MAP estimation

To find the maximum a posteriori (MAP) estimate, we maximize the posterior:

$$\theta^* = \arg\max_\theta p(\theta \mid \mathcal{D}) = \arg\max_\theta p(\mathcal{D} \mid \theta)p(\theta)$$

Taking the logarithm, we get the log-posterior:

$$\theta^* = \arg\max_\theta \left(\log p(\mathcal{D} \mid \theta) + \log p(\theta)\right)$$

Therefore, the log-prior can be interpreted as a penalty / regularization term in the optimization problem. Let's now consider the case of a Gaussian Prior

3. Gaussian Prior

Placing a zero-mean Gaussian prior on each parameter,

$$\theta_i \sim \mathcal{N}(0, \sigma^2)$$

The probability density is:

$$p(\theta_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\theta_i^2}{2\sigma^2}\right)$$

Taking the logarithm gives us the log-prior:

$$\log p(\theta_i) = -\frac{1}{2\sigma^2}\theta_i^2 + \text{const.}$$

Summing over all parameters gives:

$$\log p(\theta) = -\frac{1}{2\sigma^2}\sum_i \theta_i^2 + \text{const.}$$

4. Interpreting it as Regularization

If we plug it into the log-posterior, we get:

$$\begin{aligned}\theta^* &= \arg\max\left(\log p(\mathcal{D} \mid \theta) + \log p(\theta)\right) \\ &= \arg\max(\text{log likelihood} \\ &\quad -\frac{1}{2\sigma^2}\sum_i \theta_i^2 + \text{const.})\end{aligned}$$

Maximizing the log-posterior is equivalent to minimizing the negative log-posterior, which leads to the optimization problem:

$$\theta^* = \arg\min_\theta \left(-\text{log likelihood} - \lambda \sum_i \theta_i^2\right)$$

with $\lambda = \frac{1}{\sigma^2}$. This is exactly Ridge regression: minimizing the loss plus an ℓ_2-norm penalty on the weights.

Laplace Prior and Lasso Regularization

Just as a Gaussian prior leads to an ℓ_2-penalty (Ridge regularization), placing a Laplace prior on the parameters leads to an ℓ_1-penalty. Specifically, assuming

$$\theta_i \sim \text{Laplace}(0, b)$$

and applying the same MAP framework, the resulting optimization problem becomes:

$$\theta^* = \arg\min_\theta \left(-\log p(\mathcal{D} \mid \theta) + \lambda \sum_i |\theta_i|\right)$$

with $\lambda = \frac{1}{b}$. This is equivalent to Lasso regularization and promotes sparsity in the parameter estimates.

In this way, Bayesian priors and frequentist regularizers reflect the same underlying modeling assumptions. A Gaussian prior (Ridge) assumes all features may contribute with small weights, while a Laplace prior (Lasso) encodes the belief that only a subset of features are relevant. This duality bridges Bayesian and frequentist perspectives and provides intuition for the effect of different prior choices.

3.4 Model Training

The training of the Multinomial Bayesian Logistic Regression model was implemented using the PyMC library [7], which provides a powerful framework for probabilistic programming and Bayesian Inference. The sampling method used for posterior inference was the No-U-Turn Sampler (NUTS) [8], an adaptive variant of Hamiltonian Monte Carlo which is the industry standard nowadays for performance and convergence.

In theory, the spike-and-slab prior is defined as a mixture between a Dirac delta function at zero (the "spike") and a wider, typically Gaussian distribution (the "slab"), allowing for exact sparsity by setting some coefficients strictly to zero. However, in practice, the Dirac delta is not used directly be-

cause it is not a proper probability density and leads to discontinuities that make inference computationally intractable with standard Bayesian methods such as MCMC. To overcome this, the spike is approximated using a narrow Gaussian distribution with very small variance. This allows for efficient computation while still strongly encouraging coefficients to shrink toward zero, effectively mimicking the behavior of a true Dirac spike. This approximation maintains the interpretability and sparsity benefits of the original prior while enabling practical implementation.

The model was trained using K-fold cross-validation to assess its generalization performance and to account for the limited sample size in the dataset. We chose $K = 10$, as in the NN training, in order to ensure a robust evaluation. We compare the performance of each model with the CV and for our best model, we perform LOOCV to have the most accurate estimate of its performance. For each fold, we constructed a PyMC probabilistic model including a prior over the weight matrix (of shape $C \times D$, with C the number of classes and D the number of features), computed the class logits using a dot product between features and weights, and defined a categorical likelihood over the observed class labels.

Depending on the chosen prior type (Gaussian, Laplace, or Spike-and-slab), the weight distribution was adapted accordingly. In the case of Spike-and-slab, each coefficient was associated with a binary inclusion variable sampled from a Bernoulli distribution. The inclusion controlled whether the corresponding coefficient would be drawn from the "spike" (a narrow $\mathcal{N}(0, 0.1^2)$) or from the "slab" (a wider $\mathcal{N}(0, 1^2)$), as implemented via the `pm.math.switch` operation in PyMC.

Posterior inference was carried out using the NUTS sampler with 2000 tuning steps and 2000 draws per chain. After inference, the mean of the posterior samples for the weights was used to compute the logits on the test fold, and predictions were obtained via the softmax function. Classification accuracy was then evaluated by comparing the predicted class labels to the ground truth.

A reduced version of the PyMC training loop used for each fold is included in A.3. For the full code, please refer to the jupyter notebook *multinomial_bayesian_logreg.ipynb* in the Github repository.

4 Results and Analysis

In this section, we present the results of our experiments with the Multinomial Bayesian Logistic Regression model, comparing the performance of the different prior distributions (Gaussian, Laplace, and Spike-and-slab) and feature sets (ISO, SSFA, and Other), as well as the impact of feature selection. Exact accuracy values (means and standard deviations) for the following figures are provided in the tables referenced in each figure caption in Appendix A.4.

4.1 Performance Metrics

We evaluate the performance of the models using several metrics, including accuracy, precision, recall, and F1-score. However, for the model selection, we focus primarily on accuracy, as it is the most relevant metric for our multi-class classification task. The following box plot (*Figure 5*) summarizes the accuracy distribution across the different prior types

and feature sets.

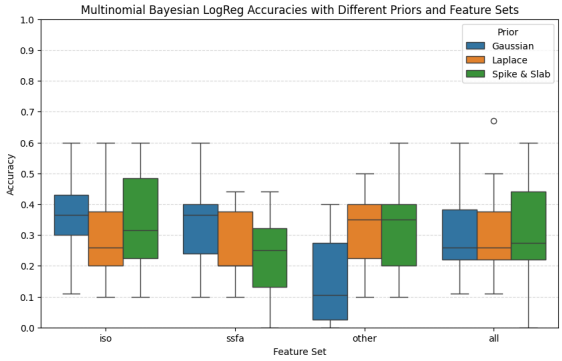


Figure 5: Box plot of accuracy distribution across prior types and feature sets. [Table A.2]

The plot shows the ISO dataset slightly outperforms the other sets. As we can see, the Other feature set is more irregular, which is to be expected. We also observe a large IQR for all feature sets, indicating that the performance varies significantly across folds due to its strong dependence on the chosen training and test samples, as we discussed earlier.

Now we will test the performance of the models trained with and without feature selection. We decided to train the model with three different feature sets: all features, the top 32, and the top 10 most relevant features according to the Multinomial Logistic Regression ranking carried out earlier (see Equation 3.1.2). We also trained the model with and without the intercept term, as we discussed earlier. The results are summarized in the following box plots:

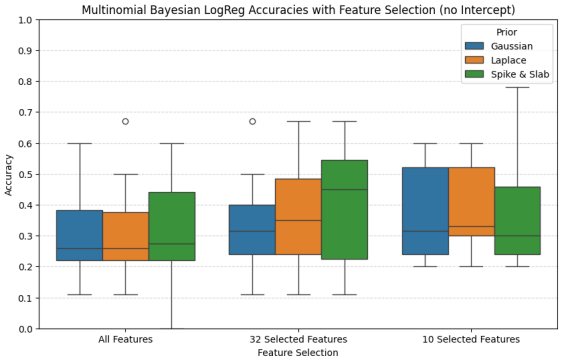


Figure 6: Box plot of accuracy distribution across feature sets and priors (no intercept). [Table A.3]

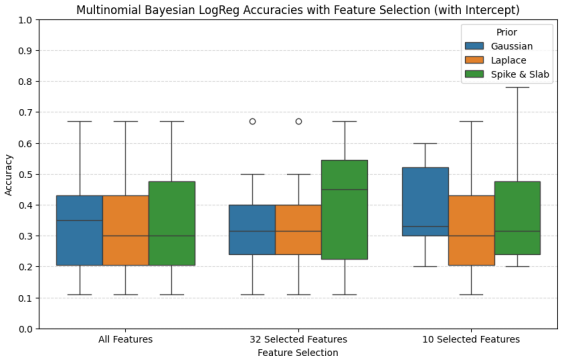


Figure 7: Box plot of accuracy distribution across feature sets and priors (with intercept). [Table A.4]

Across all priors (Gaussian, Laplace, and Spike & Slab), the overall performance remains relatively consistent between the two setups. The boxplots show comparable medians and variances, indicating that including an intercept does not significantly affect model accuracy in this classification task.

In both configurations, using feature selection tends to produce slightly higher accuracy and less variance compared to using all features, particularly for the Laplace and Spike & Slab priors. This suggests that reducing

dimensionality can help stabilize model performance, regardless of the inclusion of an intercept.

Notably, the Spike & Slab prior consistently yields competitive or best median performance, particularly on the reduced feature sets, reflecting its strength in handling sparsity. However, the impact of the intercept remains statistically minor, likely because the features were standardized and centered, reducing the need for a separate bias term.

It is encouraging that including an intercept does not degrade performance, as it allows for a more interpretable and statistically appropriate model. From a probabilistic standpoint, forcing the model to predict class probabilities based solely on feature values (i.e., without a bias term) implies that when all features are zero, the class distribution is fixed. This assumption is often unrealistic, especially when feature values are standardized or zero does not represent a meaningful baseline. By including an intercept, the model can learn a baseline distribution over classes, improving interpretability without sacrificing accuracy. This observation will be clearer in the next subsection, where *spaghetti plots* of the posterior predictive distributions show how including an intercept allows the model to shift class probabilities meaningfully, even when input features are near zero.

4.2 Posterior Predictive Distributions

This section focuses on the posterior predictive distributions of the Multinomial Bayesian Logistic Regression model as a function of individual features. We will analyse how the model's predictions change with respect to each feature, both with and without the intercept term. The following plots are often referred to as *spaghetti plots*, because each line represents a sample from the posterior distribution. However, since plotting all samples for every feature would result in overly cluttered visualizations, we display the full spaghetti plot only once for illustrative purposes. For the remaining features, we present the posterior **mean** class probabilities along with a shaded region representing the IQR across posterior samples. This approach preserves the interpretability of the plots while effectively communicating the model's uncertainty.

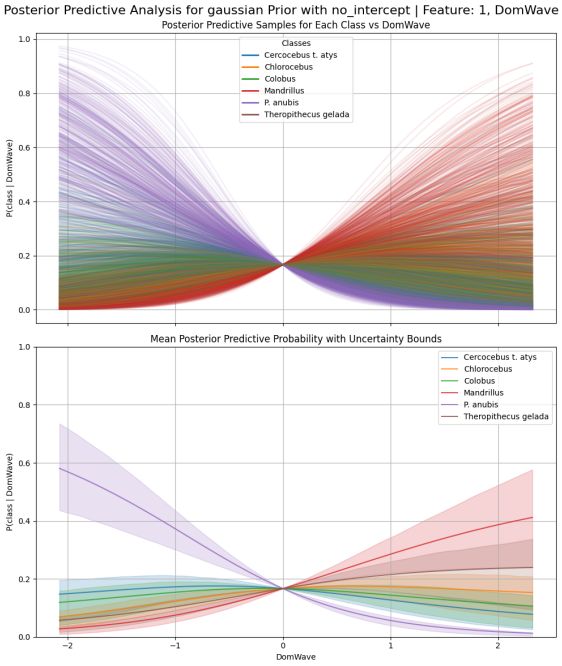


Figure 8: Spaghetti plot of posterior predictive distributions for 'DomWave' with Gaussian Prior (no intercept).

This model was trained only with the top 10 features selected by the absolute sum of the coefficients of the Multinomial Logistic Regression with Lasso mentioned earlier on.

We are showing the most relevant feature, `DomWave`, as an example.

A key pattern visible in both panels of Figure 8 is that the predicted class probabilities converge at the point where the input feature (`DomWave`) equals zero. That is, for all classes, the predicted probability curves intersect at approximately the same point on the vertical axis when $x = 0$. This behavior is a direct consequence of the model being trained *without an intercept term*.

In a multinomial logistic regression model, the class probabilities are computed via a softmax over linear logits of the form $\mathbf{w}_c^\top \mathbf{x}$, where $\mathbf{w}_c$ is the weight vector for class c and $\mathbf{x}$ is the input feature vector. When the model lacks an intercept and the input $\mathbf{x}$ is zero (which occurs when `DomWave` equals zero), all logits become zero: $\mathbf{w}_c^\top \mathbf{0} = 0$. The softmax of a zero vector yields a *uniform distribution* over classes. Therefore, the model predicts equal class probabilities at $x = 0$, regardless of the actual class distribution in the data.

This behavior is clearly visible in the bottom panel of the figure, where all class probability curves cross near $P = \frac{1}{6}$, the uniform probability for six classes. While this symmetry is mathematically consistent with the model structure, it may not be desirable in practice—particularly if zero is not a meaningful baseline after feature standardization. Including an intercept term (as shown in later figures) breaks this constraint and allows the model to learn a more realistic baseline class distribution when features are near zero.

Figure 9 shows the posterior predictive distribution for the same feature, but in this case, the model includes an intercept term.

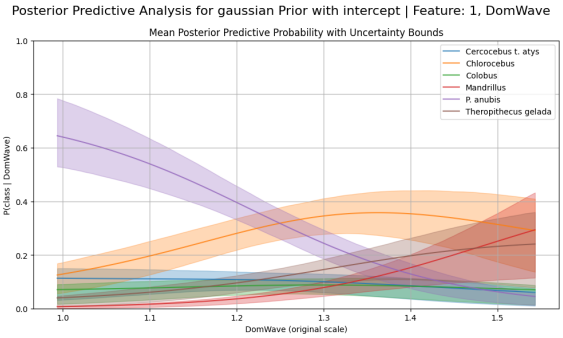


Figure 9: Posterior predictive distribution for 'DomWave' with Gaussian Prior.

Unlike in the no-intercept case, the predicted class probabilities no longer converge to a common value when the feature value is low. Instead, the model is able to assign different baseline probabilities to each class even when the feature value is close to its lower range. This behavior results from the inclusion of the intercept term, which allows each class to have a different fixed logit regardless of the input. In other words, the intercept introduces a *class-specific bias* that determines the predicted probability when the input feature has little or no influence.

Statistically, this means that the model now expresses:

$$\text{logit}_c(x) = \mathbf{w}_c^\top x + b_c,$$

where b_c is the intercept for class c. The softmax of these adjusted logits yields more flexible, and often more realistic, baseline predictions.

In this plot, for example, *P. anubis* has the highest probability for low values of `DomWave`, while *Chlorocebus* and *Mandrillus* become more probable as the feature value increases. This kind of behavior is more consistent with real-world class separation, where probabilities are not necessarily symmetric or uniform

at any particular input value. The inclusion of the intercept thus improves both interpretability and predictive flexibility.

Figures 9, 10, and 11 show the posterior predictive distributions for the `DomWave` feature under Gaussian, Laplace, and spike-and-slab priors. The three priors produce broadly similar results, with consistent class trends and smooth probability transitions. This suggests that the predictions are largely driven by the data, and that the choice of prior has limited influence in this setting.

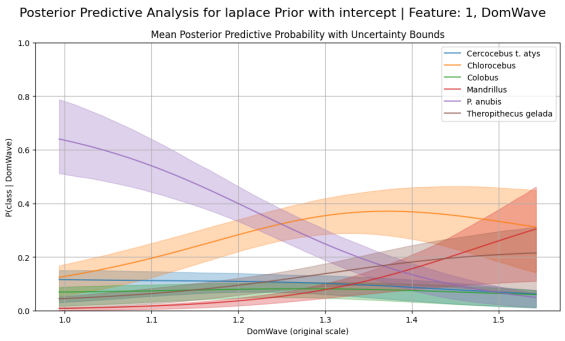


Figure 10: Posterior predictive distribution for 'DomWave' with Laplace Prior.

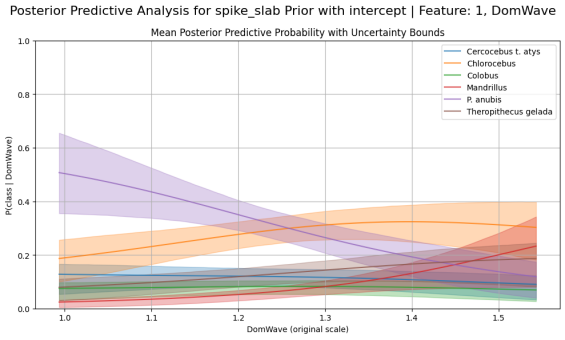


Figure 11: Posterior predictive distribution for 'DomWave' with Spike-and-Slab Prior.

The spike-and-slab prior yields slightly more uniform predictions, with lower peak probabilities across classes. This may seem surprising, as spike-and-slab is designed to zero out irrelevant features. However, with only one feature and limited data, the model may remain uncertain about which weights to suppress, leading to more diffuse class probabilities.

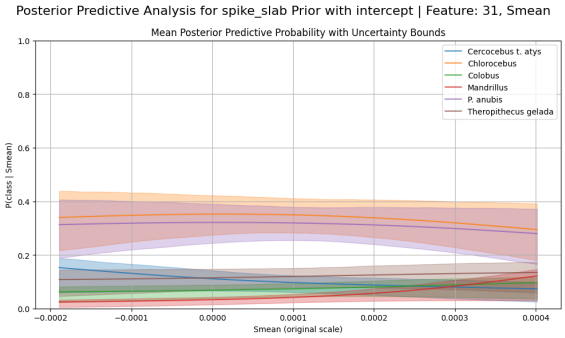


Figure 12: Posterior predictive distribution for 'Smean' with Spike-and-Slab Prior.

As shown in Figure 12, when the selected feature is not among the most informative ones (e.g., `Smean`, top 31 in the ranking), the posterior predictive plot reflects this lack of discriminative power. Class probability curves are nearly flat and heavily overlap, indicating that the model does not adjust its predictions substantially across the range of the feature.

In summary, the posterior predictive analysis confirms that the intercept term plays a key role in allowing the model to represent realistic baseline probabilities, avoiding the uniformity constraint observed when it is absent. Across priors, predictive trends remain consistent for the most informative features, indicating that the data signal dominates the prior influence. Conversely, less informative features yield flatter and more uncertain predictions, reflecting the model's ability to express limited confidence when ev-

idence is weak. This is extremely helpful as it enables us, in some cases, to classify a sample with reasonable confidence from a single feature value: for example, a low `DomWave` value strongly indicates *P. anubis* (Figure 9). This coincides with the LogReg coefficients used for the ranking, as if we refer to Figure 2, we can see that `DomWave` has a particularly high coefficient for `P. anubis`. It is noteworthy that different modelling approaches — such as Bayesian posterior predictive analysis and Lasso-based Logistic Regression — converge on similar conclusions regarding the most informative features. This agreement increases confidence in the robustness and reliability of the results.

5 Conclusions and Future Work

5.1 Conclusions

This project has demonstrated the applicability and advantages of Probabilistic Machine Learning (PML) techniques for the classification of noisy, small-scale datasets, specifically in anthropology, in the context of dental microwear texture analysis of African Cercopithecoidea. Through a systematic pipeline of preprocessing, variable selection, and model comparison, we evaluated both classical and probabilistic approaches, highlighting the capacity of Bayesian inference to explicitly model uncertainty and provide richer interpretability.

In practice, the choice of prior distribution (Gaussian, Laplace, Spike & Slab, etc.) did not substantially alter the classification results. Since the features were standardized, setting a prior variance of 1 was straightforward and did not require deep domain-specific prior knowledge. With such noisy and limited data, the results were driven more by the variability and scarcity of the observations than by the precise form of the prior. This suggests that, under these data conditions, the benefits of Bayesian methods arise primarily from their capacity to quantify uncertainty rather than from the fine-tuning of prior assumptions.

By quantifying predictive uncertainty, the proposed models not only improve robustness to noise but also offer anthropologists a tool to assess the confidence of species or group identifications. This approach contributes to more cautious and informed interpretations, aligning with the scientific goals of reproducibility, transparency, and applicability in real-world fieldwork conditions.

5.2 Future Work

A natural continuation of this work is the implementation of Variational Inference (VI) as an alternative to the MCMC-based approaches used here. While MCMC provides accurate posterior estimates, it is computationally expensive and scales poorly with larger datasets or more complex hierarchical models. VI offers a faster, optimization-based approximation to the posterior by projecting it onto a simpler family of distributions, enabling more efficient experimentation with different model specifications and hyperparameters.

Applying VI to the present problem would make it feasible to explore richer probabilistic models, such as hierarchical structures capturing inter-species and intra-species variabil-

ity, or Bayesian neural networks incorporating uncertainty in a more flexible manner. This could allow broader comparative studies over a larger number of runs, offering a clearer picture of model robustness and the stability of variable importance rankings.

References

[1] Ghislain Thiery et al. "Introducing trident: a graphical interface for discriminating groups using dental microwear texture analysis". In: *Peer Community Journal* 4.e90 (2024). DOI: 10.24072/pcjournal.467.

[2] In-Kwon Yeo and Richard A. Johnson. "A New Family of Power Transformations to Improve Normality or Symmetry". In: *Biometrika* 87.4 (2000), pp. 954–959. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2673623.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006. ISBN: 978-0387-31073-2.

[4] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012. ISBN: 978-0-262-01802-9.

[5] Hemant Ishwaran and J. Sunil Rao. "Spike and slab variable selection: Frequentist and Bayesian strategies." In: *Annals of Statistics* 33.2 (2005), pp. 730–773. DOI: 10.1214/009053604000001147.

[6] Youngwon. *How Ridge/Lasso regression and Gaussian/Laplace prior are connected?* 2023. URL: https://medium.com/@pora05/how-ridge-regression-and-gaussian-prior-are-connected-36eeb8125253.

[7] *PyMC Documentation*. 2025. URL: https://www.pymc.io.

[8] Matthew D. Hoffman and Andrew Gelman. "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research (JMLR)* 15.1 (2014), pp. 1593–1623. URL: https://www.jmlr.org/papers/volume15/hoffman14a/hoffman14a.pdf.

Appendix

A.1 Exploratory Analysis

A.1.1 Dimensionality Reduction

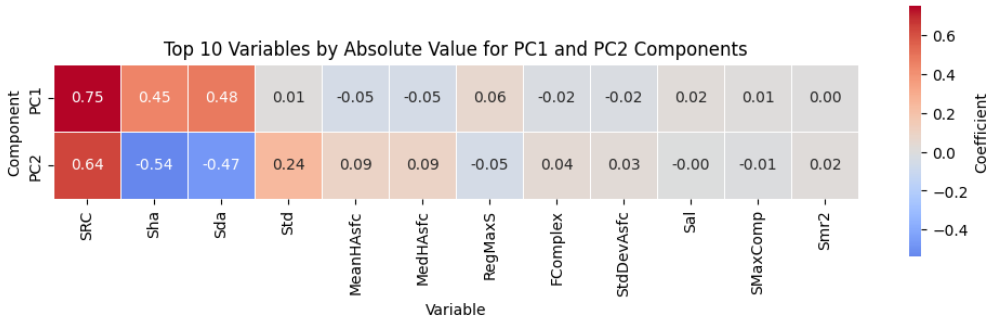


Figure A.1: Top 10 variables by absolute coefficient value in PC1 and PC2.

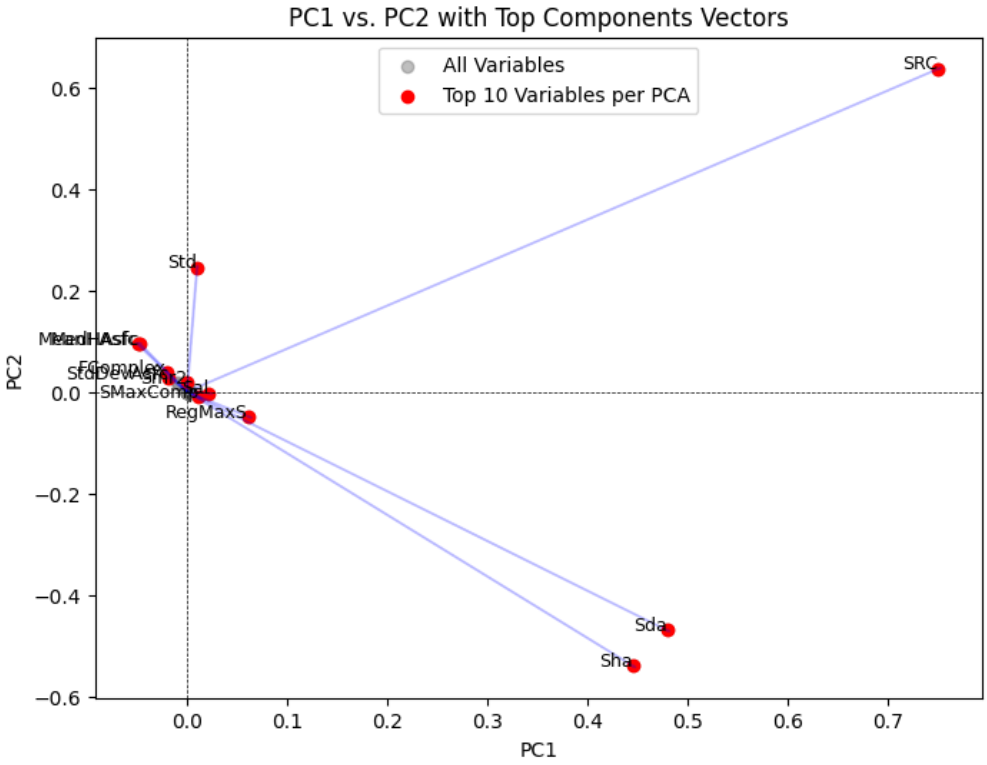


Figure A.2: Biplot of PC1 vs. PC2 with top contributing variables.

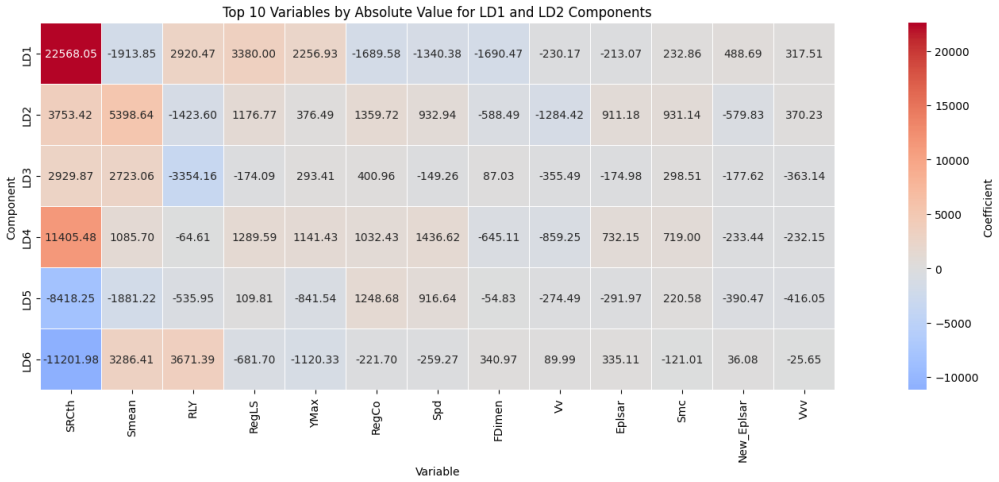


Figure A.3: Top 10 variables by absolute coefficient value across LD1 to LD6.

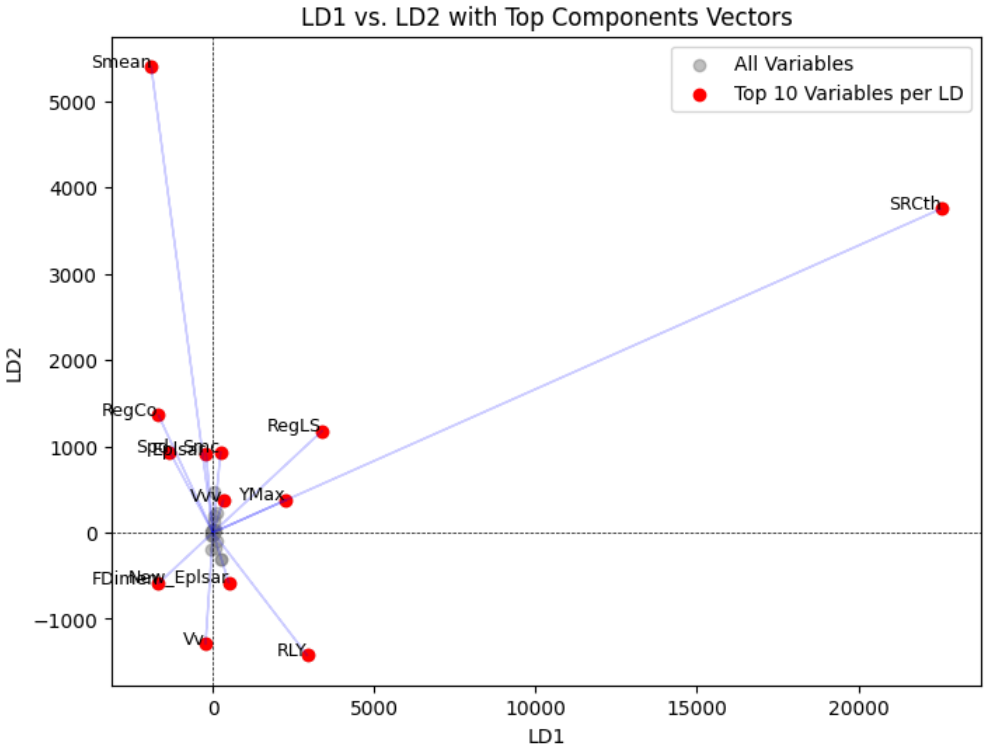


Figure A.4: Biplot of LD1 vs. LD2 with top contributing variables.

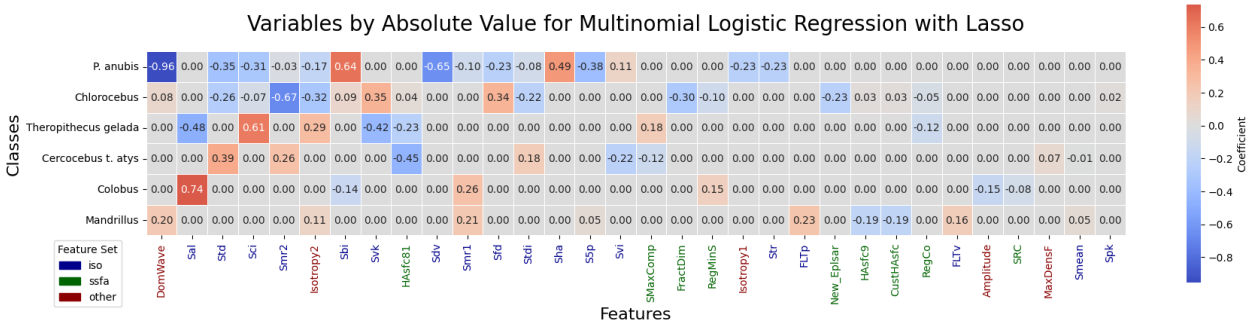


Figure A.5: Top 32 Features by Absolute Value for Multinomial Logistic Regression with Lasso

A.1.2 Correlation Matrices

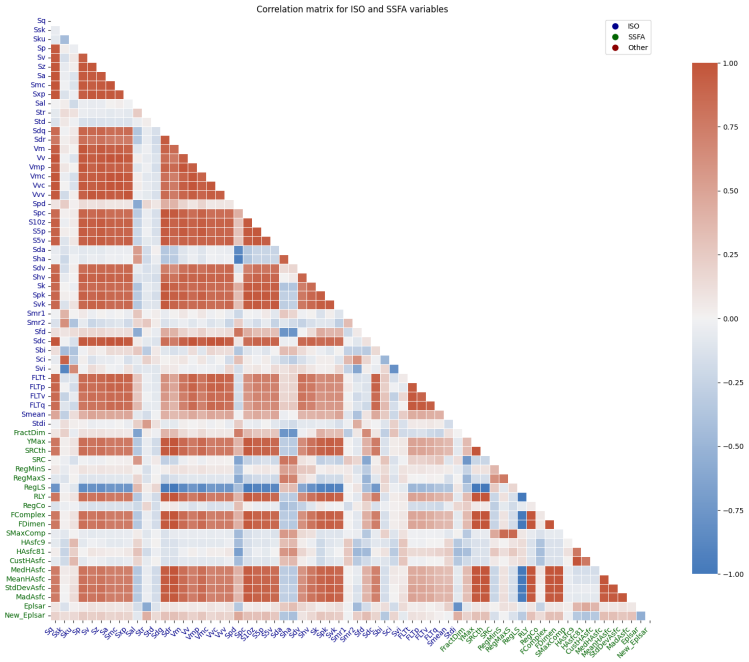


Figure A.6: Correlation matrix for ISO and SSFA variables.

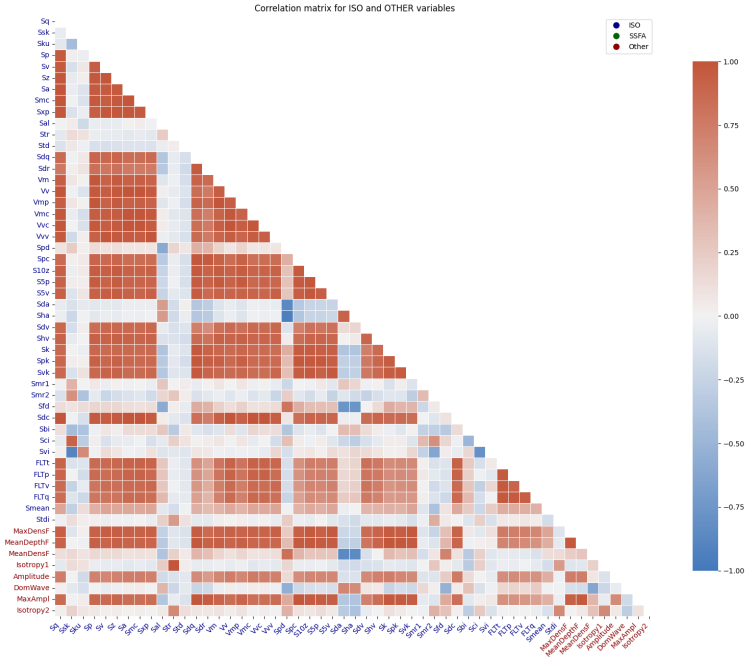


Figure A.7: Correlation matrix for ISO and OTHER variables.

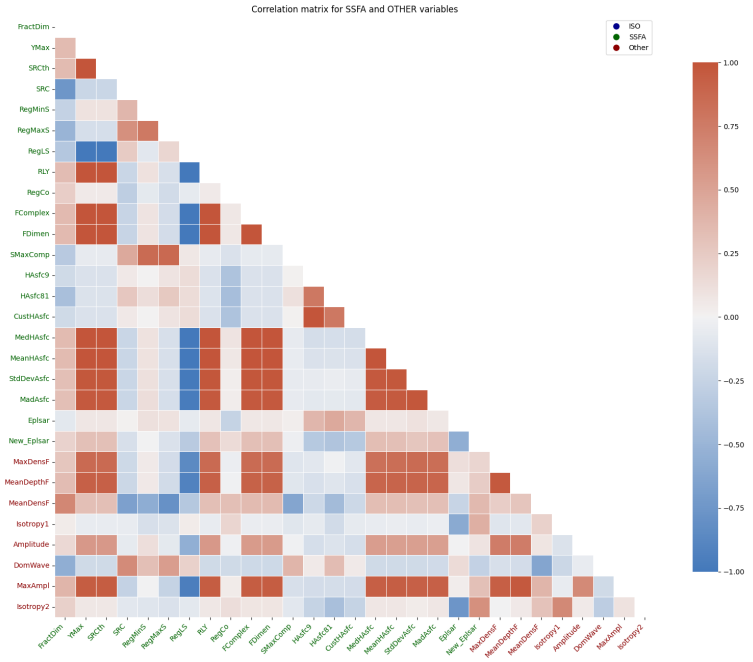


Figure A.8: Correlation matrix for SSFA and OTHER variables.

A.2 Neural Network

Table A.1: Classification report for species-level prediction

Class	Precision	Recall	F1-score	Support
Cercocebus t. atys	0.15	0.15	0.15	13
Chlorocebus	0.30	0.29	0.30	24
Colobus	0.57	0.40	0.47	10
Mandrillus	0.17	0.14	0.15	7
P. anubis	0.47	0.56	0.51	25
Theropithecus gelada	0.25	0.25	0.25	16
Accuracy			**0.34**	95
Macro avg	0.32	0.30	0.31	95
Weighted avg	0.34	0.34	0.33	95

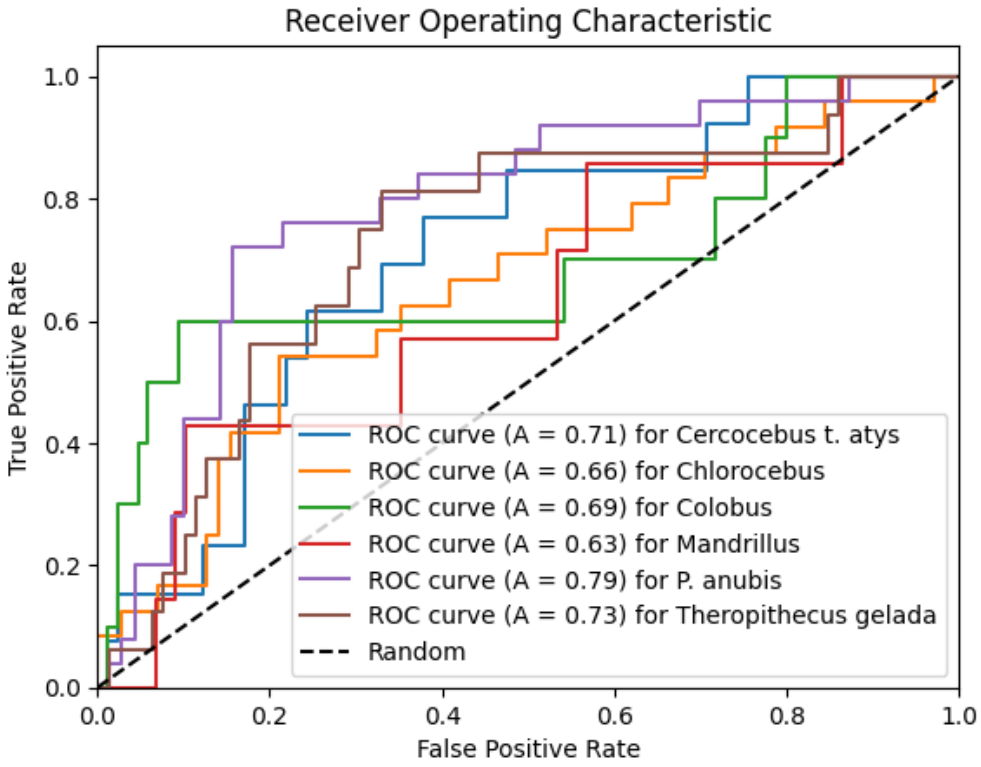


Figure A.9: ROC Curves for each group in the Neural Network Model

A.3 PyMC Training Loop

```python
for fold, (train_idx, test_idx) in enumerate(cv.split(features)):

    X_train, y_train = features.iloc[train_idx], targets.iloc[train_idx]
    X_test, y_test = features.iloc[test_idx], targets.iloc[test_idx]

    with pm.Model() as logistic_model:
        if prior_dist == "gaussian":
            weights = pm.Normal(
                'weights', mu=prior_mu, sigma=prior_sigma, shape=(n_classes, n_features)
            )
        elif prior_dist == "laplace":
            weights = pm.Laplace(
                'weights', mu=prior_mu, b=prior_sigma, shape=(n_classes, n_features)
            )
        elif prior_dist == "spike_slab":
            inclusion = pm.Bernoulli('inclusion', p=0.5, shape=(n_classes, n_features))
            sigma = pm.math.switch(inclusion, 1.0, 0.1)
            weights = pm.Normal(
                'weights', mu=prior_mu, sigma=sigma, shape=(n_classes, n_features)
            )
        logits = pm.math.dot(fold_inputs_pt, weights.T)
        if intercept:
            logits += pm.Normal(
                'intercept', mu=prior_mu, sigma=prior_sigma, shape=n_classes
            )
        # Multinomial Likelihood - Softmax used under the hood
        pm.Categorical('y_obs', logit_p=logits, observed=fold_targets_pt.astype('int64'))

        # Inference
        trace = pm.sample(draws=2000, tune=2000, target_accept=0.95, random_seed=SEED)

    # Collapse chains/draws to get all weights:
    sampled_weights = trace.posterior['weights'].values
    # shape: (chains, draws, n_classes, n_features)
    sampled_weights_flat = sampled_weights.reshape(-1, n_classes, n_features)
    # shape: (chains * draws, n_classes, n_features)

    mean_weights = sampled_weights_flat.mean(axis=0)  # shape: (n_classes, n_features)
    logits_test = np.dot(fold_inputs_test_np, mean_weights.T)  # shape: (n_samples,
        n_classes)

    # Compute probabilities using softmax
    probs = softmax(logits_test, axis=1)  # shape: (n_samples, n_classes)
    preds = np.argmax(probs, axis=1)  # shape: (n_samples,)

    # Accuracy
    acc = np.mean(preds == y_test.astype('int64'))
    accuracies_folds.append(acc)
```

Code 1: PyMC training loop for MBLR model

Table A.2: Accuracies across priors and feature sets.

Prior	**iso**	**ssfa**	**other**	**all**
Gaussian	0.37 ± 0.04	0.33 ± 0.05	0.15 ± 0.04	0.30 ± 0.05
Laplace	0.29 ± 0.05	0.25 ± 0.04	0.31 ± 0.04	0.31 ± 0.05
Spike & Slab	0.34 ± 0.05	0.24 ± 0.04	0.31 ± 0.05	0.31 ± 0.06

A.4 PyMC Performance Results

Table A.3: Accuracies across priors and feature selection (no intercept).

Prior	**All Features**	**32 Selected Features**	**10 Selected Features**
Gaussian	0.30 ± 0.05	0.35 ± 0.05	0.37 ± 0.05
Laplace	0.31 ± 0.05	0.37 ± 0.05	0.38 ± 0.04
Spike & Slab	0.31 ± 0.06	0.39 ± 0.06	0.37 ± 0.06

Table A.4: Accuracies across priors and feature selection (with intercept).

Prior	**All Features**	**32 Selected Features**	**10 Selected Features**
Gaussian	0.34 ± 0.05	0.35 ± 0.05	0.38 ± 0.04
Laplace	0.33 ± 0.05	0.35 ± 0.05	0.33 ± 0.05
Spike & Slab	0.34 ± 0.06	0.39 ± 0.06	0.38 ± 0.06