Trends in Computational and Applied Mathematics, **26** (2025), e01865
Sociedade Brasileira de Matemática Aplicada e Computacional
Online version ISSN 2676-0029 www.scielo.br/tcam
ORIGINAL ARTICLE
doi: 10.5540/tcam.2025.026.e01865

Can you beat logistic regression? The performance of four binary classifiers predicting premature birth

M. A. DINIZ[1*] and L. B. SARTORI[2]

Received on May 2, 2025 / Accepted on October 5, 2025

ABSTRACT. This paper presents a comparison of four predictive models for classification tasks: logistic regression, support vector machines (SVM), Gaussian process (GP) classification, and a model based on partial exchangeability. The models were evaluated using a dataset with 7 binary covariates, where 70% of the sample was used for training and the remaining 30% for testing. Predictive performance was assessed through three metrics: the area under the receiver operating characteristic curve (AUC), Brier score, and logarithmic score. Results show that, while the SVM model performed distinctly, the other three models exhibited similar performance, with logistic regression demonstrating the best overall results, though not by a large margin. Theoretical insights from de Finetti's representation theorem for partially exchangeable data suggest that logistic regression can be seen as a specific case of a more general model, which captures the influence of covariates on the outcome probabilities. Furthermore, we explored the Bayesian counterpart of logistic regression, using a flat prior on the model coefficients, and found that its predictive performance was almost identical to the frequentist approach. Although logistic regression was the most effective model for this dataset, the paper discusses the advantages of more flexible models, like SVM and GP classification, which may be better suited for capturing complex nonlinear relationships in data. The results indicate that while logistic regression is a reliable, fast, and interpretable model, alternative models may outperform it in cases where complex interactions between covariates and the target variable exist. However, the results shown in Section 4 indicate that the dataset studied in this article most probably does not reveal complex nonlinear interactions between the covariates and the target variable.

Keywords: classifier, logistic regression, Gaussian processes, partial exchangeability.

1 INTRODUCTION

Logistic regression is widely regarded as one of the first binary classifiers introduced by [1], although Bliss originally proposed what is now known as the probit model. Despite this distinction, logistic regression became popular due to its ease of numerical implementation and

*Corresponding author: M. A. Diniz – E-mail: marciodiniz@ufscar.br
[1]Universidade Federal de São Carlos, Departamento de Estatística, Rod. Washington Luís, km 235, 13565-905, São Carlos, SP, Brazil – E-mail: marciodiniz@ufscar.br https://orcid.org/0000-0002-8239-4263
[2]Universidade Federal de São Carlos, Departamento de Estatística, Rod. Washington Luís, km 235, 13565-905, São Carlos, SP, Brazil – E-mail: leticiasartori@estudante.ufscar.br https://orcid.org/0009-0009-5035-0280

the intuitive interpretation it offers. These qualities facilitated its adoption across various fields where the dependent variable is binary. The natural extension of logistic regression to multi-class classification further expanded its applicability.

Over the past few decades, advances in computational methods and hardware have spurred the development of other classifiers, particularly in the context of machine learning, where predictive accuracy is a primary concern. As a result, the performance of logistic regression has often been compared to that of newer techniques.

The goal of this study is to assess and compare the predictive efficacy of four binary classifiers, including logistic regression. The other classifiers are Gaussian processes for classification, support vector machines (SVM), and an inductive model based on partial exchangeability. To perform this comparison, we use a dataset containing information on 5,060 childbirths recorded in Maringá, Brazil, in 2017. The target variable is whether a birth was premature or preterm (i.e., before 37 weeks of gestation). Notably, the dataset is imbalanced, with only 11.8% of births being preterm. Moreover, all covariates are categorical. Given this setup, it is not immediately clear which model will demonstrate the best predictive performance. We use the area under the receiver operating characteristic curve (AUC), Brier and logarithmic scores as the evaluation metrics for the model comparisons.

Therefore, the main contributions of this work may be summarized as follows. First, it develops a unified perspective on binary classification inspired by de Finetti's ideas on partial exchangeability, showing that common methods such as logistic regression and Gaussian process classification can be understood as particular cases of a broader probabilistic framework. Second, it adapts Gaussian process classification to settings with purely categorical covariates by indexing the latent function on all possible combinations and applying a Laplace approximation, which yields closed-form predictive probabilities. Third, it conducts a systematic comparison of four classifiers—logistic regression, Gaussian processes, support vector machines, and a partially exchangeable model on a real dataset of over five thousand births, using three complementary performance metrics. Finally, it shows that Bayesian logistic regression with weak priors has virtually the same predictive performance as its frequentist counterpart, thereby confirming in practice the theoretical link between these approaches.

This paper is structured as follows: Section 2 describes the dataset, while Section 3 provides an overview of the statistical models employed. Section 4 presents the prediction results for each model, and Section 5 discusses the conclusions of the study.

2 DATASET

The dataset analyzed in this study comes from the Live Birth Information System (SINASC, *Sistema de Informações sobre Nascidos Vivos*), which compiles records from the Live Birth Declaration (DNV, *Declaração de Nascido Vivo*) issued for each live birth. The data were originally described by [6, 7] and were kindly shared by the authors. The dataset contains records of 5,060 live births in the city of Maringá, located in the state of Paraná, Brazil, during 2017. It includes

information on 11 potential explanatory variables related to the mother, prenatal care, childbirth, and the newborn.

The target variable, prematurity, is defined according to the World Health Organization's classification: a birth is considered premature if the child is born alive before completing 37 weeks of gestation; otherwise, the birth is classified as on-term (non-premature). As shown in the last row of Table 1, the dataset is imbalanced: only 11.8% of the births are classified as preterm, while 88.2% are non-premature. This imbalance may pose challenges for some classifiers, as they typically perform better with balanced datasets.

According to [7], the key explanatory variables for preterm births include the mother's age, whether the mother had a partner at the time of birth, parity (whether the pregnancy was primary or multiple), type of pregnancy (singleton or multiple), type of childbirth (vaginal or Cesarean), prenatal care (whether the mother had fewer or more than 7 prenatal visits), the mother's race or skin color (white or non-white), and whether the newborn had a congenital malformation. All variables are dichotomous, with the proportions of each category presented in Table 1 for both preterm and non-premature births.

For the predictive modeling, we exclude variables that are unknown prior to birth, namely, the type of childbirth and congenital malformation. Additionally, unlike [6], we include the mother's level of education (whether she completed 12 or more years of schooling) as a covariate in our models. It is important to highlight that childbirth location was not used as covariate in our model since there is no variation of the target variable when the covariate has label 1 ("other"), meaning that the birth did not take place at a hospital.

3 METHODOLOGY

3.1 Logistic regression

Logistic regression is one of the most well-known and widely used models for classifying binary variables. Although it predicts the probability that a given observation belongs to a specific category—thus functioning like a regression method—it is commonly regarded as a classifier.

To formally describe this model, let us first establish some notation. Let $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n) \in \{0,1\}^n$ represent a random vector of binary target variables, with dimension $n \in \mathbb{N}$. Here, $Y_i = 1$ indicates that a feature of interest is present for the i-th individual or observation, and $Y_i = 0$ otherwise. For each individual $i = 1, \ldots, n$, let $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ic})$ denote the vector of $c \in \mathbb{N}$ explanatory (independent) variables, also known as covariates. Given the observed value of $\mathbf{x}_i$, we assume that the random variables Y_i are independent and follow a Bernoulli distribution with parameter $p_i = P(Y_i = 1 \mid \mathbf{x}_i)$.

Under these assumptions, we have $p_i = E(Y_i \mid \mathbf{x}_i) = r(\mathbf{x}_i)$, where $r(\mathbf{x}_i)$ is referred to as the regression function. The standard linear regression model assumes that $r(\mathbf{x})$ is a linear function of unknown parameters (coefficients of the covariates or functions of those covariates). However,

Table 1: Proportions according to premature birth in Maringá, PR, 2017.

Covariate	Categories	Premature	
		0: no	1: yes
Age (years)	0: < 35	88.9	11.2
	1: ≥ 35	85.8	14.2
Prenatal care	0: ≥ 7	90.1	9.9
	1: ≤ 7	77.6	22.4
Partner	0: yes	87.8	12.2
	1: no	89.2	10.8
Childbirth location	0: hospital	88.2	11.8
	1: other	100.0	0.0
Education (years)	0: ≥ 12	87.2	12.8
	1: < 12	88.8	11.2
Sex	0: female	88.2	11.8
	1: male	88.2	11.8
Parity	0: multiple	88.9	11.1
	1: primary	87.1	12.9
Race/skin color	0: white	87.4	12.6
	1: non white	90.2	9.8
Pregnancy type	0: singleton	90.2	9.8
	1: multiple	25.0	75.0
Congenital malformation	0: no	88.3	11.7
	1: yes	75.5	24.5
Childbirth type	0: vaginal	91.6	8.4
	1: Cesarean	87.2	12.8
Total		88.2	11.8

since p_i represents a probability, it must lie within the interval $[0,1]$, making the standard linearity assumption unsuitable in this context.

A generalized linear model (GLM) addresses this issue by introducing a *link function* that relates the regression function to the so-called *linear predictor*. More specifically,

$$G(p_i) = G(r(\mathbf{x}_i)) = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic} = \beta_0 + \eta(\mathbf{x}_i),$$

where G is the link function, and $\beta_0 + \eta(\mathbf{x}_i)$ represents the linear predictor for individual i. Rewriting this relationship using the inverse of G, we have

$$p_i = r(\mathbf{x}_i) = G^{-1}(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic}) = G^{-1}(\beta_0 + \eta(\mathbf{x}_i)). \tag{3.1}$$

This formulation shows that distribution functions of continuous real random variables can serve as inverse link functions, as they map any real value to the unit interval $[0,1]$. The most common

choices for G^{-1} are the standard normal distribution, yielding the *probit* model, and the logistic function, which underlies logistic regression.

In logistic regression, we specify

$$p_i = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{i1}+\ldots+\beta_c x_{ic})}}, \tag{3.2}$$

implying that

$$\ln\left(\frac{p_i}{1-p_i}\right) = \beta_0+\beta_1 x_{i1}+\ldots+\beta_c x_{ic}.$$

This expression provides an intuitive interpretation of the coefficients in terms of the log-odds of the characteristic of interest.

Under the frequentist framework, the logistic regression model is estimated by finding the parameter values that maximize the likelihood function—i.e., the joint probability of observing the given sample as a function of the unknown parameters. Given our assumptions about the target variable, the likelihood function is defined as

$$L(\boldsymbol{\beta} \mid \mathbf{X},\mathbf{y}) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}, \tag{3.3}$$

where p_i is given by (3.2), $\mathbf{X} = \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$ represents the observed covariates, $\mathbf{y} = \{y_1,\ldots,y_n\}$ denotes the observed labels for each of the n individuals and $\boldsymbol{\beta} = (\beta_0,\beta_1,\ldots,\beta_c)$. Since this likelihood function is nonlinear in the parameters, maximizing the logarithm of the likelihood is often a simpler approach.

The original sample used to estimate the model parameters is referred to as the *training set*. From this training set, we obtain parameter estimates, denoted by $\widehat{\beta}_j$, which enable probabilistic predictions for individuals not included in the training set. Denoting the covariate values of one of these out-of-sample individuals by $\mathbf{x}^* = (x_1^*,\ldots,x_c^*)$, the probability this individual has the characteristic of interest is simply

$$p^* = r(\mathbf{x}^*) = G^{-1}(\widehat{\beta}_0+\widehat{\beta}_1 x_1^*+\ldots+\widehat{\beta}_c x_c^*) = G^{-1}(\widehat{\beta}_0+\widehat{\eta}^*).$$

The set with values for covariates and target variables not used in the estimation of the model is called *test* set.

3.2 Gaussian Processes

A Gaussian process (GP) is a stochastic process, or collection of random variables, $\{Z(t)\}_{t\in T}$, all defined within the same probability space. The probability space is represented by the triple $(\Omega,\mathscr{A},P)$, where Ω is the sample space, $\mathscr{A}$ is a sigma-algebra of subsets of Ω, and P is a probability measure. Here, T is an index set.

Gaussian processes are characterized by the property that, for any finite subset of indices $t_1, t_2, \ldots, t_n \in T$ with $n \in \mathbb{N}$, the corresponding random vector $(Z(t_1), Z(t_2), \ldots, Z(t_n))$ has a joint distribution that is multivariate Gaussian. This distribution is fully characterized by

$$\begin{bmatrix} Z(t_1) \\ \vdots \\ Z(t_n) \end{bmatrix} \sim N_n \left(\begin{bmatrix} m(t_1) \\ \vdots \\ m(t_n) \end{bmatrix}, \begin{bmatrix} k(t_1,t_1) & \cdots & k(t_1,t_n) \\ \vdots & \ddots & \vdots \\ k(t_n,t_1) & \cdots & k(t_n,t_n) \end{bmatrix} \right)$$

where $m : T \to \mathbb{R}$ denotes the mean function, and $k : T \times T \to \mathbb{R}$ is the covariance function of the process. The covariance function k must be symmetric in its inputs and satisfy $k(t,t) \geq 0$ for every $t \in T$, ensuring that the covariance matrix is symmetric and positive semi-definite. When the process $Z(t)$ is described by a Gaussian process, we denote it as $Z(t) \sim \mathscr{GP}(m(t), k(t,t'))$.

Note that the definition of a Gaussian process accommodates processes with a discrete index set T, as is the case for processes defined over discrete time sets. In particular, T may also be finite, in which case the joint distribution of all random variables in the process will be a multivariate Gaussian distribution with the same dimension as T.

The set of all possible covariate values, $\mathscr{X}$, typically serves as the index set, which can be more general. For instance, $\mathscr{X}$ could be $\mathbb{R}^c$, where $c \in \mathbb{N}$ represents the number of covariates, as described in Section 3.1. This setup allows GPs to describe a distribution over regression functions, such that $E(Y \mid \mathbf{x}) = r(\mathbf{x})$ for $\mathbf{x} \in \mathscr{X}$. Modeling regression functions as GPs offers greater flexibility, as it does not restrict the functions to linear combinations of parameters (as in standard regression analysis) or transformations of these linear combinations (as in GLMs). This approach to regression is known as Gaussian process regression.

Gaussian processes can also be applied to binary classification problems using a strategy similar to that in GLMs, as the regression function needs to be constrained to the interval $[0,1]$. In this context, GPs model a latent function $f(\mathbf{x})$, which is then restricted to $[0,1]$ by a function that maps real values into this interval. Using the notation from Section 3.1, we define $r(\mathbf{x}) = G^{-1}(f(\mathbf{x}))$. Specifically, we replace the linear predictor $\beta_0 + \eta(\mathbf{x}_i)$ in equation (3.1) with $f(\mathbf{x}_i)$, where f is a Gaussian process.

The latent function f acts as a nuisance function, much like unknown parameters in linear models: we do not observe the values of f directly; instead, we observe only the inputs $\mathbf{x}_i$ and the class labels y_i. Our main interest lies in $r(\mathbf{x}_i)$, particularly for test values $r(\mathbf{x}^*)$.

Due to the stochastic nature of f and r, it is generally not appropriate to estimate $r(\mathbf{x}^*)$ simply by computing $G^{-1}(\widehat{f}(\mathbf{x}^*))$, where $\widehat{f}$ is an estimate of f. The correct approach is to derive the expected value of $r(\mathbf{x}^*)$, with the distribution of the latent variable $f^* = f(\mathbf{x}^*)$ for a test case, given by $\pi_{n+1}(f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*)$ as follows

$$\pi_{n+1}(f_* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \int_{\mathbb{R}^n} \pi(f_* \mid \mathbf{X}, \mathbf{x}^*, \mathbf{f}) \cdot \pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y}) \, d\mathbf{f} \tag{3.4}$$

where $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$, the vector of latent variables, and

$$\pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y}) = \frac{\pi(\mathbf{y} \mid \mathbf{f}, \mathbf{X}) \cdot \pi_0(\mathbf{f} \mid \mathbf{X})}{\int_{\mathbb{R}^n} \pi(\mathbf{y} \mid \mathbf{f}, \mathbf{X}) \cdot \pi_0(\mathbf{f} \mid \mathbf{X}) d\mathbf{f}} \tag{3.5}$$

is the posterior over $\mathbf{f}$, $\pi(\mathbf{y} \mid \mathbf{f}, \mathbf{X})$ is the likelihood and $\pi_0(\mathbf{f} \mid \mathbf{X})$ the prior over $\mathbf{f}$, described by a Gaussian process.

Using the distribution over the latent variable f^*, we can produce a probabilistic prediction

$$P(y^* = 1 \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \int_{\mathbb{R}} G^{-1}(f^*) \cdot \pi_{n+1}(f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) \, df^*.$$

As discussed in Section 2, we will use 7 binary variables from the dataset as covariates, so that $\mathscr{X} = \{0,1\}^7$ with $|\mathscr{X}| = 128$. Following the notation established above, let $\mathbf{y} = (y_1, \ldots, y_n) \in \{0,1\}^n$, where n represents the size of the training set . Therefore, some covariate vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$ may be identical.

To account for this, we introduce modified notation: let $\mathbf{f}$ be the vector of dimension 128, consisting of values of f evaluated at each element of $\mathscr{X}$, denoted by $\mathbf{e}_j$ to distinguish them from observed covariate vectors $\mathbf{x}_i$ for individuals i. Thus, we have $\mathscr{X} = \{\mathbf{e}_1, \ldots, \mathbf{e}_{128}\}$ and $\mathbf{f} = (f(\mathbf{e}_1), \ldots, f(\mathbf{e}_{128}))$. When $\mathscr{X}$ is continuous, the elements of $\mathbf{X}$ will almost surely not coincide, and $\mathbf{f}$ will have the same dimension as the training set.

Following this notation and using the standard normal distribution function Φ as the inverse of the link function, the likelihood function is given by

$$\pi(\mathbf{y} \mid \mathbf{f}, \mathbf{X}) = \prod_{i=1}^{128} [\Phi(f(\mathbf{e}_i))]^{y(\mathbf{e}_i)} [1 - \Phi(f(\mathbf{e}_i))]^{n(\mathbf{e}_i) - y(\mathbf{e}_i)},$$

where

$$y(\mathbf{e}_i) = \sum_{\{j : \mathbf{x}_j = \mathbf{e}_i\}} y_j \quad \text{and} \quad n(\mathbf{e}_i) = \left|\{j : \mathbf{x}_j = \mathbf{e}_i\}\right|.$$

The GP prior for $\mathbf{f}$, with mean function $m(\mathbf{e})$ and covariance function $k(\mathbf{e}_i, \mathbf{e}_j)$, is in this case equivalent to a multivariate normal distribution of dimension 128 with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{V}$. In this work, we use a constant mean function reflecting a prior probability of 0.1 that childbirth will be preterm, along with the radial basis function (RBF) or Gaussian kernel as the covariance function

$$\pi_0(\mathbf{f} \mid \mathbf{X}) \sim N_{128}(\mathbf{m}, \mathbf{V}), \quad \text{where} \quad \mathbf{m} = \Phi^{-1}(0.1)\mathbf{1}_{128}, \quad \text{and}$$

$$\mathbf{V} = k(\mathbf{e}_i, \mathbf{e}_j) = \exp\left(-\frac{1}{2}\left\|\mathbf{e}_i - \mathbf{e}_j\right\|^2\right), i, j = 1, \ldots, 128.$$

Following [9, sec. 3.4], we approximate the posterior $\pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y})$ with a Gaussian distribution $q(\mathbf{f} \mid \mathbf{X}, \mathbf{y})$

$$\mathbf{f} \mid \mathbf{X}, \mathbf{y} \sim q = N_{128}\left(\widehat{\mathbf{f}}, \left(\mathbf{V}^{-1} + \mathbf{W}\right)^{-1}\right),$$

where $\widehat{\mathbf{f}} = \text{argmax}_{\mathbf{f}}\, \pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y})$ is the maximum a posteriori (MAP) estimator of $\mathbf{f}$, and $\mathbf{W} = -\nabla\nabla \log \pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y})\big|_{\mathbf{f}=\widehat{\mathbf{f}}}$ is the Hessian matrix of the log-posterior evaluated at this point. The MAP estimator was obtained using a Newton-Raphson algorithm to maximize the log of the unnormalized posterior, as the denominator of $\pi_n(\mathbf{f} \mid \mathbf{X}, \mathbf{y})$ does not depend on $\mathbf{f}$.

The posterior predictive distribution of f^*, from equation (3.4), is also approximated by a Gaussian distribution. For this, we require only the mean and variance of f^* under the Laplace approximation

$$E_q[f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*] = \mathbf{k}_*^\top V^{-1} \widehat{\mathbf{f}}, \quad \text{and}$$

$$\text{var}_q[f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*] = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}_*^\top \left(V + W^{-1}\right)^{-1} \mathbf{k}_*,$$

where $\mathbf{k}_* = k(\mathbf{x}^*, \mathscr{X})$ is the vector of covariances between the test point $\mathbf{x}^*$ and all covariate vectors in $\mathscr{X}$. Finally, the predictive probability $P(y^* = 1 \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*)$ is approximated by

$$E_q(\pi_* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*) = \int_{\mathbb{R}} \Phi(f^*) \cdot q(f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*)\, df^* = \Phi\left(\frac{E_q[f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*]}{\sqrt{1 + \text{var}_q(f^* \mid \mathbf{X}, \mathbf{y}, \mathbf{x}^*)}}\right). \tag{3.6}$$

This integral has an exact solution because q has a Gaussian density. For details, see [9, sec. 3.9].

3.3 Support Vector Machines (SVM)

Support Vector Machines (SVMs) generalize a classifier known as the *maximal margin classifier*. This classifier assumes it is possible to find a hyperplane that perfectly separates the observations in the training set based on their target variable values.

To make this more precise, let us slightly modify the notation: in this section, $y_i = 1$ indicates, as before, that the i-th childbirth was preterm, while $y_i = -1$ now indicates that the birth was at term. If these values are separable, then a hyperplane exists, i.e., a set $\{\mathbf{z} \in \mathbb{R}^c : \beta_0 + \beta_1 z_1 + \ldots + \beta_c z_c = 0\}$, such that for each observation in the training set, $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$, we have

$$\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic} > 0, \text{if} \quad y_i = 1, \text{and} \quad \beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic} < 0, \text{if} \quad y_i = -1.$$

Using the notation from Section 3.1, we define $h(\mathbf{x}_i) = \beta_0 + \eta(\mathbf{x}_i)$. Then a separating hyperplane has the property that $y_i \cdot h(\mathbf{x}_i) > 0$ for all $i = 1, \ldots, n$. The maximal margin classifier then seeks to find β_0 and now defining $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_c)$ defining the hyperplane that maximizes the *margin*—the distance between the closest points of the training classes with labels 1 and -1. This is achieved by solving the following optimization problem

$$\underset{\beta_0,\, \boldsymbol{\beta},\, M}{\text{maximize}}\ M \text{ subject to (3.7.1) } \sum_{i=1}^{c} \beta_i^2 = 1 \text{ and (3.7.2) } y_i \cdot h(\mathbf{x}_i) \geq M \tag{3.7}$$

for $i = 1, \ldots, n$, where M is the margin to be maximized.

The first constraint, $\sum_{i=1}^{c} \beta_i^2 = 1$, requires the vector $\boldsymbol{\beta}$ to have unit norm. This is not a constraint on the hyperplane itself, since if $\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic} = 0$ defines a hyperplane, then any scaling

$\gamma(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic}) = 0$ with $\gamma \neq 0$ will also define a hyperplane. However, including this constraint gives (3.7.2) a specific interpretation: it can be shown (see [5, sec. 4.5]) that the perpendicular distance from the i-th sample point to the hyperplane is precisely $y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic}) = y_i \cdot h(\mathbf{x}_i)$.

Hence, constraints (3.7.1) and (3.7.2) ensure that each observation lies on the correct side of the hyperplane, each at least a distance M from the hyperplane. This distance M is the *margin*, which is maximized by choosing optimal values for β_0 and $\boldsymbol{\beta}$. Given that constraint (3.7.1) primarily serves a secondary purpose, we can reformulate problem (3.7) without the unit norm constraint on β as follows

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \; ||\boldsymbol{\beta}|| \quad \text{subject to} \quad y_i \cdot h(\mathbf{x}_i) \geq 1 \quad \text{for} \quad i = 1, \ldots, n,$$

where $||\boldsymbol{\beta}||$ is the L_2 norm of $\boldsymbol{\beta}$. Note that this formulation implies $M = 1/||\boldsymbol{\beta}||$. The maximal margin classifier provides an intuitive approach for classification when a hyperplane can fully separate the classes, which, however, is not always the case for real-world datasets.

When the classes overlap in covariate space, we may still aim to maximize M, but allow some points to fall on the incorrect side of the margin. To achieve this, we introduce non-negative auxiliary variables $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)$, known as *slack* variables. This modifies problem (3.7) to

$$\underset{\beta_0,\, \boldsymbol{\beta},\, M,\, \boldsymbol{\varepsilon}}{\text{maximize}} \; M \text{ subject to (3.8.1) } \sum_{i=1}^{c} \beta_i^2 = 1, \text{ (3.8.2) } y_i \cdot h(\mathbf{x}_i) \geq M(1 - \varepsilon_i)$$
$$\text{and (3.8.3)} \sum_{i=1}^{n} \varepsilon_i \leq \text{ constant}, \; \varepsilon_i \geq 0, \tag{3.8}$$

for $i = 1, \ldots, n$. The variable ε_i in constraint (3.8.2) has the following interpretation: if $\varepsilon_i = 0$, the i-th sample point is correctly classified and lies on the correct side of the margin; if $0 < \varepsilon_i \leq 1$, the sample point lies within the margin but is correctly classified and if $\varepsilon_i > 1$, the sample point is misclassified. Thus, if $y_i \cdot h(\mathbf{x}_i) < 0$, constraint (3.8.2) implies $M(1 - \varepsilon_i) \leq y_i \cdot h(\mathbf{x}_i) < 0$ since $M > 0$.

Thus, constraint (3.8.3) ensures that both the number of misclassifications and the total extent by which points lie on the incorrect side of the margin are limited by a chosen constant. The unit norm constraint on β can be removed again, leading to a reformulation of problem (3.8) as

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \; ||\boldsymbol{\beta}|| \quad \text{subject to} \quad y_i \cdot h(\mathbf{x}_i) \geq 1 - \varepsilon_i, \quad \sum_{i=1}^{n} \varepsilon_i \leq \text{constant}, \tag{3.9}$$

$\varepsilon_i \geq 0$ for $i = 1, \ldots, n$, known as the *soft margin classifier*, as it allows some points to violate the margin constraint. This can further be rewritten as

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \; \frac{1}{2}||\boldsymbol{\beta}||^2 + Q\sum_{i=1}^{n} \varepsilon_i, \quad \text{subject to} \quad \varepsilon_i \geq 0 \quad \text{and} \quad y_i \cdot h(\mathbf{x}_i) \geq 1 - \varepsilon_i \tag{3.10}$$

for $i = 1, \ldots, n$, where the parameter Q replaces the constant in problem (3.9). This formulation defines the *support vector classifier* or *linear support vector machine*, with the separable case as a limit when $Q \to \infty$.

The support vector classifier performs effectively when the boundary between classes is linear, but real-world boundaries are often non-linear. To handle this, [2] applied the principles of the support vector classifier on a transformation of the original covariate space, achieving (soft) linear separation in this transformed space. Therefore, a suitable transformation function $\phi : \mathscr{X} \to \mathscr{F}$ must be chosen such that class labels become linearly separable in the transformed space $\phi(\mathscr{X})$. The codomain of ϕ, denoted $\mathscr{F}$, is known as the *feature space*, and may have a high dimension.

Given a transformation ϕ, we can define $h(\mathbf{x}_i) = \beta_0 + \phi(\mathbf{x}_i)^\top \boldsymbol{\beta}$, where $\phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_d(\mathbf{x}_i))$, with $\phi_j(\mathbf{x})$ as basis functions. Here, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d)$, and d is the dimension of the feature space. With this new definition of h, problem (3.10) is solved to obtain the classifier.

As the details in the appendix show, the solution $\widehat{\beta}_0$ and $\widehat{\boldsymbol{\beta}}$ provides an optimal classifier based on

$$\widehat{h}(\mathbf{x}) = \widehat{\beta}_0 + \phi(\mathbf{x})^\top \widehat{\boldsymbol{\beta}} = \widehat{\beta}_0 + \sum_{i=1}^{n} \widehat{\alpha}_i \cdot y_i \cdot \langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle, \tag{3.11}$$

where $\langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle$ represents the inner product $\phi(\mathbf{x})^\top \cdot \phi(\mathbf{x}_i)$, and $\widehat{\alpha}_i$ are derived from the dual formulation of problem (3.10). More details are provided in the appendix.

At first glance, this approach appears to require explicit knowledge of ϕ, but this is avoided through a technique known as the *kernel trick*. To understand this, we first introduce kernels.

Definition 1 (Mercer kernels). *A real-valued function $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is called a kernel if it satisfies the following properties:*

1. *$k(\mathbf{u}, \mathbf{v}) = k(\mathbf{v}, \mathbf{u})$ for every $\mathbf{u}, \mathbf{v} \in \mathscr{X}$ (i.e., it is symmetric), and*
2. *matrix $[k(\mathbf{v}_i, \mathbf{v}_j)]_{i,j=1}^{\ell}$ is positive semidefinite for every $\mathbf{v}_1, \ldots, \mathbf{v}_\ell \in \mathscr{X}$ and $\ell \in \mathbb{N}$.*

The kernel trick relies on the following theorem.

Theorem 1 (Kernel trick). *If k is a kernel, then there exists an inner product space $\mathscr{F}$ and a transformation function ϕ such that $k(\mathbf{u}, \mathbf{v}) = \langle \phi(\mathbf{u}), \phi(\mathbf{v}) \rangle$ for any $\mathbf{u}, \mathbf{v} \in \mathscr{X}$.*

Since equation (3.11) shows that the classifier depends on ϕ only through its inner product, the kernel trick theorem guarantees that specifying a kernel function is equivalent to implicitly defining the mapping function ϕ. For our work, we will employ the Gaussian (or RBF) kernel, previously introduced in Section 3.2 in relation to Gaussian processes. Other kernel functions may also be used, with common choices being the polynomial kernel and the neural network or sigmoid kernel. The d-th degree polynomial kernel is defined as $k(\mathbf{u}, \mathbf{v}) = (\gamma_0 \langle \mathbf{u}, \mathbf{v} \rangle + \gamma_1)^d$, while the sigmoid kernel is defined as $k(\mathbf{u}, \mathbf{v}) = \tanh(\gamma_0 \langle \mathbf{u}, \mathbf{v} \rangle + \gamma_1)$, where d, γ_0, and γ_1 are parameters to be specified.

The resulting classifier is thus based on (3.11): for a test sample $\mathbf{x}^*$, it is classified as class 1 if $\widehat{h}(\mathbf{x}^*) > 0$ and as class -1 if $\widehat{h}(\mathbf{x}^*) < 0$. Using the kernel trick, this can be expressed as

$$\widehat{C}(\mathbf{x}^*) = \text{sign}\left[\widehat{\beta}_0 + \sum_{i=1}^{n} \widehat{\alpha}_i \cdot y_i \cdot k(\mathbf{x}^*, \mathbf{x}_i)\right]. \tag{3.12}$$

As shown in equation (3.12), SVMs yield predicted *labels* but do not directly provide probabilistic outputs. To estimate class probabilities, we use Platt's scaling [8], which applies a logistic transformation to the classifier scores. This approach involves computing parameters A and B such that

$$P(y^* = 1 \mid \mathbf{x}^*) = \frac{1}{1 + \exp\left(A\widehat{h}(\mathbf{x}^*) + B\right)}, \tag{3.13}$$

where $\mathbf{x}^*$ is the covariate vector of a test observation. In practice, A and B are estimated by fitting a logistic regression model on the classifier scores, $\widehat{h}(\cdot)$, from the training set, using maximum likelihood estimation as outlined in Section 3.1.

3.4 Induction by partial exchangeability

An exchangeable sequence of random variables is a (potentially infinite) sequence in which any finite permutation of indices results in a joint probability distribution identical to that of the original sequence. To capture the type of symmetry that characterizes inductive inference in practical applications, Bruno de Finetti [3] generalized this idea by introducing the concept of partial exchangeability.

In de Finetti's words, partial exchangeability assumes that "'*except for unforeseen circumstances that induce us to change our attitude,' we shall always use the outcomes of the observations symmetrically and shall apply them symmetrically to all future trials. [...] [This means], for instance, that the observation of an appreciable difference among the outcomes of the trials made under different circumstances will induce us to attribute different probabilities varying with these circumstances. For instance, in a medical experiment, whether on humans or guinea pigs, it might initially be held presumptive that the effect of a given treatment would not be influenced by circumstances varying among individuals of the same species.*" [4, p. 217]

In this framework, a sequence of random variables is partially exchangeable if it can be divided into different classes, within which exchangeability holds. Thus, the joint probability distribution remains unchanged under permutations within each class, but not necessarily between classes.

This definition is particularly applicable to binary random variables, such as the target variable in this work, which indicates whether a birth is preterm or not. In the context of partial exchangeability, the sequence of binary random variables is such that the order of preterm births within each class is irrelevant.

A representation theorem exists for partially exchangeable sequences, with a notable special case for binary random variables grouped into only two classes.

Theorem 2 (de Finetti representation theorem [3], p. 13). *Let $\{Y_i(g_1)\}_{i\in\mathbb{N}}$ and $\{Y_i(g_2)\}_{i\in\mathbb{N}}$ be random sequences representing binary outcomes (e.g., success or failure) for individuals in group 1 (g_1) and group 2 (g_2), respectively. Assuming these sequences are partially exchangeable, there exists a unique probability measure μ on the unit square $[0,1]^2$ such that*

$$P(Y_1(g_1)=z_1,\ldots,Y_c(g_1)=z_c,Y_1(g_2)=w_1,\ldots,Y_\ell(g_2)=w_\ell)= \\ =\int_0^1\int_0^1 \theta_1^a(1-\theta_1)^{c-a}\,\theta_2^b(1-\theta_2)^{\ell-b}\,d\mu(\theta_1,\theta_2)$$

for any binary sequences $\{z_i\}_{1\le i\le c}$ and $\{w_i\}_{1\le i\le \ell}$, with $a=\sum_{i=1}^c z_i$ and $b=\sum_{i=1}^\ell w_i$.

Applying this theorem, we can build a classifier for our study on classifying childbirths as preterm or not. Consider the sequence $(y_1,\mathbf{x}_1),(y_2,\mathbf{x}_2),\ldots,(y_n,\mathbf{x}_n)$, where each $y_i\in\{0,1\}$ represents the outcome for individual i, and $\mathbf{x}_i\in\mathscr{X}$ represents the vector of covariates (features) observed for individual i, where $i=1,\ldots,n$.

Here, $\mathscr{X}$ is a finite set with cardinality 128, so each $\mathbf{x}$ can be viewed as a categorical variable with a finite set of possible levels. Suppose that all birth outcomes Y_i with the same covariate vector value (reflecting mother and child characteristics) are exchangeable (within each fixed value of the covariate vector), and the data is part of an infinite sequence with this property. Then, de Finetti's theorem guarantees the existence of a joint probability measure μ on the unit hypercube $\mathscr{U}=[0,1]^{128}$ such that

$$P(s_i \text{ preterm births out of } n_i \text{ births of type } i, 1\le i\le 128)= \\ =\prod_{i=1}^{128}\binom{n_i}{s_i}\int_{\mathscr{U}}\prod_{i=1}^{128}\theta_i^{s_i}(1-\theta_i)^{n_i-s_i}\,d\mu(\theta_1,\ldots,\theta_{128}), \tag{3.14}$$

where $s_i=y(\mathbf{e}_i)$ and $n_i=n(\mathbf{e}_i)$ are as defined above, and $\theta_i=P(Y=1\mid\mathbf{x}=\mathbf{e}_i)$ represents the probability of a preterm birth for covariate level $\mathbf{e}_i$. Note that θ_i is similar, though not equivalent, to p_i defined in Section 3.1 as $p_i=P(Y_i=1\mid\mathbf{x}_i)$.

Using pre-established notation, we can express the probability of the observed outcomes as

$$P(\mathbf{y}\mid\mathbf{X})=\int_{\mathscr{U}}\prod_{i=1}^{128}\theta_i^{s_i}(1-\theta_i)^{n_i-s_i}\,d\mu(\theta_1,\ldots,\theta_{128}). \tag{3.15}$$

This result enables us to obtain predictive probabilities as follows

$$P(y^*=1\mid\mathbf{X},\mathbf{y},\mathbf{x}^*)=\frac{P(y^*=1,\mathbf{y}\mid\mathbf{X},\mathbf{x}^*)}{P(\mathbf{y}\mid\mathbf{X})},$$

where $\mathbf{x}^*\in\mathscr{X}$ is a new observation's covariate vector. Note that the numerator also takes the form of an integral like (3.15). For example, if $\mathbf{x}^*=\mathbf{e}_j$, we have

$$P(y^*=1\mid\mathbf{X},\mathbf{y},\mathbf{x}^*=\mathbf{e}_j)=\frac{\displaystyle\int_{\mathscr{U}}\theta_j\prod_{i=1}^{128}\theta_i^{s_i}(1-\theta_i)^{n_i-s_i}\,d\mu(\theta_1,\ldots,\theta_{128})}{\displaystyle\int_{\mathscr{U}}\prod_{i=1}^{128}\theta_i^{s_i}(1-\theta_i)^{n_i-s_i}\,d\mu(\theta_1,\ldots,\theta_{128})}=E_{\pi_n}(\theta_j),$$

where π_n represents the posterior of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{128})$ given the training data. Specifically, the posterior is

$$\pi_n(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) = \frac{\mu(\boldsymbol{\theta}) \prod_{i=1}^{128} \theta_i^{s_i} (1-\theta_i)^{n_i - s_i}}{P(\mathbf{y} \mid \mathbf{X})},$$

where $\mu(\boldsymbol{\theta})$ is the prior on $\boldsymbol{\theta}$. Thus, the predictive probability for $\mathbf{x}^* = \mathbf{e}_j$ is simply the posterior expected value of θ_j.

4 RESULTS

In this section, we present the predictive performance of the four models discussed in Section 3, detailing the computational methods used for each model.

Each model was estimated 10 times, with each iteration using a unique training sample composed of 70% of the data points (3542 observations) randomly drawn from the full dataset. Performance metrics were then computed based on probabilistic predictions for the corresponding test set, which consisted of the remaining 30% of the sample (1518 observations).

All models were implemented in Python. Logistic regression and SVM were estimated using the `scikit-learn` library. We used the default SVM hyperparameter settings—particularly $Q = 1$—because values selected via five-fold cross-validation delivered essentially identical predictive performance. The partially exchangeable model was implemented in Stan via the `Pystan` library. As described in Section 3.2, obtaining the Laplace approximation to the posterior density in the Gaussian Process model required maximizing the posterior density, which we accomplished using a Newton-Raphson algorithm. This algorithm, along with the other routines, is available at `https://github.com/Dinizmarz/Classifiers-for-preterm-birth`.

The performance metrics used in this work are the AUC (area under the curve) of the receiver operating characteristic (ROC) curve, the Brier score, and the logarithmic score.

The ROC curve plots the true positive rate (TPR) on the vertical axis against the false positive rate (FPR) on the horizontal axis at each threshold level of a binary classifier. Each threshold sets the level at which an observation is classified as positive (label 1) or negative (label 0). A model that perfectly predicts all instances for each threshold setting has a TPR of 1 and an FPR of 0 at all thresholds. In contrast, a completely random classifier, which predicts the positive and negative classes with equal probability, produces a diagonal ROC curve from (0,0) to (1,1). For a random classifier, the TPR equals the FPR because it produces an equal number of true and false positive predictions at any threshold.

The closer the ROC curve is to the top-left corner, the better the model's ability to distinguish between labels. The AUC ranges from 0 to 1, with a score of 1 indicating perfect predictive performance.

The Brier score (BS) is given by

$$BS = \frac{1}{n^*} \sum_{i=1}^{n^*} (p_i^* - y_i^*)^2,$$

where p_i^* is the probabilistic prediction, y_i^* is the actual outcome for observation i (0 if the event does not occur, and 1 if it does), and n^* is the size of the test sample. The Brier score can be interpreted as the mean squared error of the forecast, offering a straightforward way to assess the accuracy of probabilistic predictions.

The logarithmic, or cross-entropy, score (*LS*) is given by

$$LS = -\frac{1}{n^*}\sum_{i=1}^{n^*}\left[y_i^*\ln(p_i^*) + (1-y_i^*)\ln(1-p_i^*)\right],$$

which is calculated as the log of the probability assigned to the observed outcome. The interpretation of this score is based on information theory: given that $p_i^* = P(y_i^* = 1 \mid \mathbf{x}^*)$, the expected value of the *LS* can be written as $-\frac{1}{n^*}\sum_{i=1}^{n^*} p_i^*\ln(p_i^*)$, closely related to the entropy of the predicted probabilities. One limitation of this metric is that it is undefined when p_i^* is exactly 0 or 1.

We now provide a rationale for using the RBF kernel as the covariance matrix for the GP prior on $\mathbf{f}$, the vector of latent functions defined in Section 3.2. As explained in Section 3.3, kernels quantify the similarity between vectors, which is essential for defining the covariance matrix of $\mathbf{f}$ in terms of the covariance between vectors $\mathbf{e}_1,\ldots,\mathbf{e}_{128}$. A kernel such as the RBF creates a positive semidefinite matrix, satisfying the conditions for a valid covariance matrix.

Given the structure of our dataset, it is reasonable to assume, for instance, that $f(\mathbf{e}_1)$ and $f(\mathbf{e}_2)$ are correlated, with this correlation being greater than that between $f(\mathbf{e}_1)$ and $f(\mathbf{e}_4)$ if the Euclidean distance between $\mathbf{e}_1$ and $\mathbf{e}_2$ is smaller than between $\mathbf{e}_1$ and $\mathbf{e}_4$.

The same RBF kernel was used for the SVM model in Section 3.3 because the data are not linearly separable, as shown by the nature of the seven binary covariates, which place data points on the 128 vertices of a hypercube. With sample points on certain vertices having both labels ($y_i = 1$ and $y_i = 0$), the data are clearly not linearly separable. Although other kernels could be employed in this scenario, the RBF kernel offers flexibility and provided strong predictive performance compared to other options.

The RBF kernel is also relevant for the prior used in the partially exchangeable model. Here, the prior is a truncated multivariate normal of dimension 128 with a mean vector of $0.1-$which is consistent with $\boldsymbol{m} = \Phi^{-1}(0.1)\mathbf{1}_{128}$, the mean function for the GP prior discussed in Section 3.2$-$and an RBF covariance matrix defined on $\boldsymbol{\theta}$

$$k(\theta_i,\theta_j) = \exp\left(-\frac{1}{2}\left\|\theta_i-\theta_j\right\|^2\right), \quad i,j = 1,\ldots,128,$$

with the distribution constrained to the hypercube $[0,1]^{128}$, the parameter space in this model. This choice for the prior covariance matrix is appropriate because kernels like the RBF quantify similarity between vectors and ensure a valid covariance structure.

Table 2 shows that, with the exception of the kernel SVM, the other three models exhibited similar predictive performance in terms of AUC. However, when considering the logarithmic and Brier scores, Tables 3 and 4 demonstrate that the predictive performance of all four models

was comparable. Notably, logistic regression outperformed the others on average, though the difference was small.

Table 2: Models' AUC for 10 different test samples.

	Models			
Round	Logistic	Kern. SVM	Part. exch.	Gauss. proc.
1	0.6447	0.5757	0.6299	0.6289
2	0.6880	0.5740	0.6830	0.6833
3	0.6831	0.6067	0.6787	0.6775
4	0.6573	0.5912	0.6531	0.6510
5	0.6955	0.6251	0.6683	0.6740
6	0.7047	0.5878	0.6776	0.6947
7	0.6955	0.5813	0.6726	0.6805
8	0.7116	0.5737	0.6743	0.6903
9	0.7028	0.6206	0.6798	0.6899
10	0.6533	0.5982	0.6564	0.6631
Mean	0.6837	0.5934	0.6674	0.6733
St. Dev.	0.0236	0.0189	0.0164	0.0205

Table 3: Models' log scores for 10 different test samples.

	Models			
Round	Logistic	Kern. SVM	Part. exch.	Gauss. proc.
1	0.3261	0.3326	0.3306	0.3295
2	0.3420	0.3535	0.3449	0.3430
3	0.3229	0.3320	0.3298	0.3269
4	0.3364	0.3383	0.3420	0.3406
5	0.3121	0.3208	0.3266	0.3215
6	0.3040	0.3132	0.3144	0.3084
7	0.3199	0.3286	0.3270	0.3232
8	0.3222	0.3360	0.3302	0.3267
9	0.3226	0.3317	0.3289	0.3247
10	0.3298	0.3382	0.3327	0.3299
Mean	0.3238	0.3325	0.3307	0.3274
St. Dev.	0.0109	0.0108	0.0084	0.0097

Table 4: Models' Brier scores for 10 different test samples.

	Models			
Round	Logistic	Kern. SVM	Part. exch.	Gauss. proc.
1	0.0917	0.0935	0.0926	0.0922
2	0.0981	0.1008	0.0990	0.0982
3	0.0902	0.0929	0.0927	0.0917
4	0.0946	0.0953	0.0970	0.0965
5	0.0866	0.0890	0.0915	0.0899
6	0.0845	0.0861	0.0880	0.0859
7	0.0897	0.0916	0.0915	0.0902
8	0.0914	0.0945	0.0933	0.0922
9	0.0909	0.0928	0.0928	0.0914
10	0.0929	0.0952	0.0941	0.0932
Mean	0.0911	0.0932	0.0932	0.0921
St. Dev.	0.0038	0.0039	0.0030	0.0034

5 DISCUSSION AND FINAL REMARKS

Except for the SVM models, the similar performance of the other three methods can be expected, as they may be considered particular cases of a more general model. The key to understanding this relationship is provided by de Finetti's Theorem, discussed in Section 3.4. Recalling equation (3.15), if the childbirth outcomes with the same value of the covariate vector are considered exchangeable, the data are partially exchangeable. Thus, the probability of the outcomes, $P(\mathbf{y} \mid \mathbf{X})$, can be written as

$$P(\mathbf{y} \mid \mathbf{X}) = \int_0^1 \cdots \int_0^1 \prod_{i=1}^{128} \theta_i^{s_i} (1-\theta_i)^{n_i - s_i} d\mu\,(\theta_1, \ldots, \theta_{128}),$$

where $\theta_i = P(Y = 1 \mid \mathbf{x} = \mathbf{e}_i)$.

De Finetti's theorem remains valid even when the covariate space $\mathscr{X}$ has infinite dimension. In this case, expressing $\theta_i = \theta(\mathbf{x}_i) = P(Y = 1 \mid \mathbf{x}_i)$, the representation of $P(\mathbf{y} \mid \mathbf{X})$ becomes

$$P(\mathbf{y} \mid \mathbf{X}) = \int_{\Theta(\mathbf{x})} \prod_{i=1}^{n} \theta(\mathbf{x}_i)^{y_i} (1-\theta(\mathbf{x}_i))^{1-y_i} d\mu\,(\theta(\mathbf{x}_1), \ldots, \theta(\mathbf{x}_n)),$$

where the covariate values $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are distinct (i.e., there are no ties), and $\Theta(\mathbf{x})$ represents the function space of $\theta(\mathbf{x})$.

The connection with GPs is direct. Assuming $\theta(\mathbf{x}_i) = \Phi(f(\mathbf{x}_i))$, the probability becomes

$$P(\mathbf{y} \mid \mathbf{X}) = \int_{\mathbb{R}^n} \prod_{i=1}^{n} [\Phi(f(\mathbf{x}_i))]^{y_i} [1 - \Phi(f(\mathbf{x}_i))]^{1-y_i} \pi_0(\mathbf{f} \mid \mathbf{X}) d\mathbf{f},$$

where the integral is taken over the space of the latent function $\mathbf{f}$.

Alternatively, by specifying $\theta(\mathbf{x}_i)$, for example, by setting $f(\mathbf{x}_i)$ as a linear function of the covariates and using a generic (inverse of) link function instead of Φ, we get

$$\theta(\mathbf{x}_i) = p_i = G^{-1}(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_c x_{ic}) = G^{-1}(\beta_0 + \eta(\mathbf{x}_i)),$$

where G is the link function, as described in Section 3.1. The prior measure μ is then rewritten as a measure μ_0 relating to the parameters $\beta_0, \ldots, \beta_c$, and we have

$$P(\mathbf{y} \mid \mathbf{X}) = \int_{\mathbb{R}^{c+1}} \prod_{i=1}^{n} p_i^{y_i} (1-p_i)^{1-y_i} d\mu_0(\beta_0, \ldots, \beta_c).$$

This integral may be recognized as the prior predictive distribution derived from a Bayesian logistic regression model. In fact, the integrand is identical to the likelihood function of frequentist logistic regression, as given by equation (3.3), and μ_0 represents the prior over the parameters $\beta_0, \ldots, \beta_c$.

Therefore, assuming a very flat prior on these parameters, it is reasonable to expect that the predictive performance of this model would be similar to that of the frequentist logistic regression. To validate this conjecture, we adopted the same prior for the coefficients $\beta_0, \ldots, \beta_7$ used by [6], where the parameters are considered independent and each follows a normal distribution with mean zero and variance 10^6.

Table 5: Bayesian Logistic regression metrics for 10 different test samples.

	Metrics		
Round	AUC	Log. score	Brier score
1	0.6447	0.3270	0.0920
2	0.6880	0.3425	0.0982
3	0.6831	0.3228	0.0901
4	0.6573	0.3366	0.0946
5	0.6954	0.3117	0.0864
6	0.7052	0.3040	0.0845
7	0.6956	0.3195	0.0895
8	0.7132	0.3218	0.0912
9	0.7028	0.3224	0.0907
10	0.6533	0.3302	0.0930
Mean	0.6839	0.3238	0.0910
St. Dev.	0.0239	0.0112	0.0039

Table 5 confirms that the conjecture is indeed correct: using the same training samples as in Tables 2−4, the predictive performance of Bayesian logistic regression was, on average, essentially the same as that of frequentist logistic regression. The Bayesian logistic regression was also implemented using `Pystan`.

Therefore, de Finetti's theorem demonstrates that logistic regression is a particular case that can be highly effective for predictive classification. Moreover, it is numerically simple and fast to implement, works well with categorical covariates, and provides estimates of the coefficients that directly indicate the influence of variables on the outcome probabilities.

However, despite achieving the best predictive performance in the dataset analyzed in this study, there may be cases where the assumptions of the logistic model are not suitable to describe the data. In such instances, models like SVM and Gaussian process classification may outperform logistic regression, especially when there are highly complex nonlinear relationships between the covariates and the target variable. This is evident when comparing how $\theta(\mathbf{x}_i)$ is modeled by Gaussian processes versus logistic regression. By assuming $\theta(\mathbf{x}_i) = \Phi(f(\mathbf{x}_i))$, the model gains more flexibility to capture potential nonlinear relationships.

Conflict of interest

Authors have no conflict of interest to declare.

Data availability

Datasets related to this article are available upon request to the corresponding author.

Associate editor: Mário de Castro

REFERENCES

[1] C.I. Bliss. The Calculation of the Dosage-Mortality Curve. *Annals of Applied Biology*, **22** (1935), 134–167.

[2] B.E. Boser, I.M. Guyon & V.N. Vapnik. A training algorithm for optimal margin classifiers. In "Proceedings of the fifth annual workshop on Computational learning theory – COLT '92" (1992), p. 144–152.

[3] B. de Finetti. Sur la condition d'equivalence partielle. In "Actualités Scientifiques et Industrielles", 739. Hermann, Paris (1938), p. 5–18.

[4] B. de Finetti. "Probability, Induction and Statistics: The art of guessing". Wiley (1972).

[5] J. Friedman, T. Hastie & R. Tibshirani. "The Elements of Statistical Learning: Data Mining, Inference and Prediction". Springer (2009).

[6] R. Galo, R.M. Rossi, D.C. Alves & R.R. Oliveira. Bayesian binary regression using power and power reverse link functions: an application to premature birth data. *Brazilian Journal of Biometrics*, **41** (2023), 131–143.

[7] R.R. Oliveira. "Nascimento prematuro no Estado do Paraná e no município de Maringá". Phd thesis, Universidade Estadual de Maringá, Maringá (2015).

[8] J.C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In “Advances in Large Margin Classifiers” (1999), p. 61–74.

[9] C.E. Rasmussen & C.K.I. Williams. “Gaussian Processes for Machine Learning”. MIT Press (2006).

[10] V.N. Vapnik. “Statistical Learning Theory”. Wiley (1998).

How to cite

M. A. Diniz & L. B. Sartori. Can you beat logistic regression? The performance of four binary classifiers predicting premature birth. *Trends in Computational and Applied Mathematics*, **26**(2025), e01865. doi: 10.5540/tcam.2025.026.e01865.

APPENDIX

The objective of this appendix is to explain how equation (3.11) is obtained as a solution to problem (3.10). Considering the hyperplane in the transformed covariate space, $h(\mathbf{x}_i) = \beta_0 + \phi(\mathbf{x}_i)^\top \boldsymbol{\beta}$, the primal Lagrangian corresponding to problem (3.10) is

$$\mathscr{L}(\beta_0, \boldsymbol{\beta}, \varepsilon) = \frac{1}{2}||\boldsymbol{\beta}||^2 + Q\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n}\alpha_i\left[y_i \cdot h(\mathbf{x}_i) - (1-\varepsilon_i)\right] - \sum_{i=1}^{n}\gamma_i\varepsilon_i \tag{A.1}$$

where α_i and γ_i are Lagrange multipliers. The first order conditions to minimize (A.1) are

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{A.2}$$

$$\boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i \cdot y_i \cdot \phi(\mathbf{x}_i) \tag{A.3}$$

$$\alpha_i = Q - \gamma_i \tag{A.4}$$

$$\alpha_i, \gamma_i, \varepsilon_i \geq 0, \tag{A.5}$$

for $i = 1, \ldots, n$. Replacing these into (A.1), one obtain the Lagrangian dual function

$$\mathscr{L}_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j \cdot y_i y_j \cdot \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

and if a given kernel is given, it can be written as

$$\mathscr{L}_D(\boldsymbol{\alpha}) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j \cdot y_i y_j \cdot k(\mathbf{x}_i, \mathbf{x}_j). \tag{A.6}$$

Equation (A.6) gives a lower bound to the objective function of problem (3.10) for any feasible point. Maximizing $\mathscr{L}_D$ subject to $0 \leq \alpha_i \leq Q$ and $\sum_{i=1}^{n}\alpha_i y_i = 0$ the Karush-Kuhn-Tucker conditions are

$$\alpha_i\left[y_i \cdot h(\mathbf{x}_i) - (1-\varepsilon_i)\right] = 0 \tag{A.7}$$

$$\gamma_i\varepsilon_i = 0 \tag{A.8}$$

$$y_i \cdot h(\mathbf{x}_i) - (1-\varepsilon_i) \geq 0, \tag{A.9}$$

for $i = 1, \ldots, n$, in addition to $(5.2) - (5.5)$. Equations $(5.2) - (5.9)$ characterize the solution of the primal and dual forms. From (5.3) one may see that the solution for $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}} = \sum_{i=1}^{n}\widehat{\alpha}_i y_i \phi(\mathbf{x}_i)$$

and, consequently

$$\widehat{h}(\mathbf{x}) = \widehat{\beta}_0 + \phi(\mathbf{x})^\top\widehat{\boldsymbol{\beta}} = \widehat{\beta}_0 + \sum_{i=1}^{n}\widehat{\alpha}_i \cdot y_i \cdot \langle\phi(\mathbf{x}), \phi(\mathbf{x}_i)\rangle = \widehat{\beta}_0 + \sum_{i=1}^{n}\widehat{\alpha}_i \cdot y_i \cdot k(\mathbf{x}, \mathbf{x}_i)$$

where $\widehat{\alpha}_i$ are non-null only for observations i for which (A.9) are satisfied. These observations compose the support vectors since $\widehat{\beta}$ is written in terms of them. More technical details about SVM's can be found in [5, ch. 12] and [10, ch. 10].