Bayesian Logistic Regression for Iris Species Classification

Anastasia Putri Hartanti [2602100120], Jenibelle Wievin [2602105632], Lauren Abigail [2602108426], Matthew Nicholas Alfian [2602105960], Stanley Raditya [2602103671]

Bina Nusasntara University, Kemanggisan, School of Computer Science, Indonesia
Computer Science Department, BINUS Undergraduate Program - Data Science, Bina Nusantara Univerity, Jakarta, Indonesia 11530

Emails: anastasia.hartanti@binus.ac.id, jenibelle.wievin@binus.ac.id, lauren.abigail@binus.ac.id, matthew.alfian@binus.ac.id, stanley.raditya@binus.ac.id

1. Introduction

This mini-project centers around the exploration of the iris dataset, a rich source of information obtained from Kaggle.com. The dataset encapsulates details on 150 iris plants, presenting a diverse array of variables that elucidate their distinctive characteristics. The primary focus of interest lies in discerning the species of the iris plants, which are categorized into three classes: setosa, versicolor, and virginica. However, the analytical lens of this mini-project narrows down to the application of binary logistic regression, concentrating specifically on the determination of whether a given iris plant belongs to the setosa species or not.

Within the dataset, multiple variables contribute significantly to the overall characterization of the iris plants. These variables, often referred to as features, assume a pivotal role in the logistic regression model employed for the analysis. Encompassing a spectrum of botanical attributes, these features provide valuable insights into the unique traits exhibited by each iris specimen. The decision to focus on the binary classification task of identifying setosa species serves to simplify the analysis, diverting attention from the intricacies associated with multi-class classification problems commonly encountered in iris datasets.

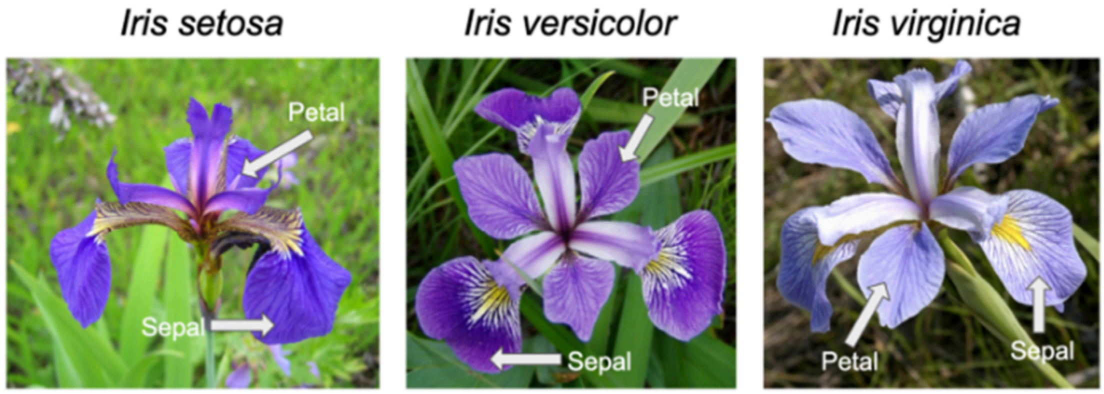
As the mini-project unfolds, the exploration and utilization of these features become the bedrock for constructing a robust logistic regression model. This tailored approach not only facilitates a targeted analysis but also showcases the adaptability of logistic regression techniques in scenarios where the objective is to distinguish between two specific outcomes. The mini-project thus offers a practical and insightful journey into the application of logistic regression within the context of real-world datasets, shedding light on the nuances of the iris dataset and the broader implications of binary classification.

Here is the list of variables in the dataset, along with the description:

-	Id	: plant id
-	SepalLenghtCm	: sepal length in cm
-	SepalWidthCm	: sepal width in cm
-	PetalLenghtCm	: petal length in cm

- PetalWidthCm : petal width in cm
- Species : iris species

The sepal is a crucial part of a flower's anatomy, serving as the outermost protective structure of the blossom. These leaf-like structures encase and shield the developing bud during its early stages. Sepals also play a role in regulating the opening of the flower and protecting its reproductive organs, contributing to the overall health and development of the floral structure. The details on sepals, specifically how they look, can be seen in the picture below.



2. Models

   a. Model Formulation

   For this binary classification, we will be using Binary Logistic Regression. Binary Logistic Regression is a statistical method used for modeling the relationship between a binary dependent variable and one or more independent variables. In contrast to linear regression, which is applied to predict continuous outcomes, binary logistic regression is specifically designed for situations where the dependent variable represents two distinct categories or classes. The logistic regression model employs the logistic function to transform a linear combination of input features into probabilities, ensuring predictions fall within the range of 0 to 1. This makes it particularly well-suited for tasks such as classification, where the goal is to assign observations to one of the two predefined categories. By estimating the probability of an event occurring, binary logistic regression provides valuable insights into the factors influencing the outcome, making it a widely used and interpretable tool in fields such as healthcare, finance, and social sciences. The model is shown in the equation below:

$$log(\frac{P(Y=1)}{P(Y=0)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ \dots \ + \beta_p X_p$$

   b. Likelihood

   The likelyhood distribution for this classification task is the Bernoulli Distribution. The Bernoulli distribution is a fundamental concept in probability theory and statistics, named after the Swiss mathematician Jacob Bernoulli. It represents a discrete probability distribution of a random variable that can take on one of two possible outcomes, typically labeled as success and failure. The distribution is characterized by a single parameter, usually denoted as "p," which represents the probability of success. In a Bernoulli trial, the outcomes are mutually exclusive, and the probability of success remains constant across each trial. This distribution serves as the building block for more complex probability models and is widely employed in various fields, including binary classification problems in machine learning, where events can be dichotomously categorized. The Bernoulli distribution provides a fundamental framework for understanding and modeling scenarios with binary outcomes, offering insights into the probabilistic nature of such systems.

$Y \mid \theta \sim Bernouli(\theta)$

c. Prior

The prior distribution for this classification task is the Normal Distribution, using all input variables. Normal distribution, often referred to as the Gaussian distribution or the bell curve, is a fundamental concept in statistics and probability theory. This symmetrical probability distribution is characterized by a bell-shaped curve, where the majority of the data points cluster around the mean, creating a smooth, continuous pattern. In a normal distribution, the mean, median, and mode are all equal, and specific proportions of data fall within standard deviations from the mean. This distribution is prevalent in various natural phenomena and human-made systems, making it a cornerstone in statistical analysis. The Central Limit Theorem further underscores its significance, stating that the sum or average of a large number of independent, identically distributed random variables will be approximately normally distributed, irrespective of the original distribution of the variables. The normal distribution's mathematical elegance and widespread applicability make it a powerful tool for statistical modeling and inference in diverse fields, ranging from economics and biology to physics and social sciences.

$\theta \sim Normal(\mu, \sigma^2)$

Where mean (miw) equals to 0 and the precision (1/var) is 0,01.

3. Computation
   a. Number of burn-in sample : 1000
   b. Number of post burn-in sample : 5000
   c. Number of chains : 3

4. Codes for binary logistic regression
   with priors consisting of every variables given in the dataset.
   a. Data encoding and feature engineering.

```
df <- iris
# target variable = species - iris setosa

colnames(df)
df$Species <- ifelse(df$Species == "setosa", 1, 0)
df
```

The dataset under consideration exhibits a lack of null values, and it exclusively comprises numerical data across all input variables. Consequently, in order to ready the data for the specific classification task at hand, a preprocessing step is required. This involves encoding the 'Species' column, wherein all instances of 'setosa' are transformed into the numerical value 1, while the remaining categories are represented by the numerical value 0. This encoding process ensures that the machine learning algorithm can effectively interpret and process the categorical information within the 'Species' column, facilitating seamless integration into the classification model.

b. Model fitting

```
library(geoR)
Y <- df$Species  ## label // output
X <- scale(df[,1:4]) # features scaling // standardise
n <- lengte3wh(Y)

library(rjags)
burn <- 1000 # burning first 1000 samples to maintain convergence
iters <- 6000 # 5000 iterations
chains <- 3 # 3 chains (markov chains)

mod <- textConnection("model{
        for(i in 1:n){
                Y[i] ~ dbern(pi[i])
                logit(pi[i]) <- beta[1] + X[i,1]*beta[2] + X[i,2]*beta[3] + X[i,3]*beta[4]
        + X[i,4]*beta[5]
                }
        for(j in 1:6){beta[j] ~ dnorm(0,0.01)}
}")

data <- list(Y=Y,X=X,n=n)
model <- jags.model(mod,data=data, n.chains = chains,quiet=TRUE)
update(model,burn, progress.bar="none")
samps <- coda.samples(model,variable.names=c("beta"),n.iter =
iters,n.thin=5,progress.bar="none")
summary(samps)
plot(samps)
```