Statistical Insights into Bank Marketing: Bayesian vs. Frequentist Approaches

Andrea Sciortino

2025-01-09

Bank Phone Marketing Campaign Analysis

Introduction and Data Exploration

This document aims to provide a clear and comprehensive implementation of two distinct statistical approaches, both developed from scratch: the frequentist and the bayesian frameworks. The primary objective is to interpret the results to better understand the philosophical and methodological differences between these two statistical paradigms to analyse the regression coefficient, with necessary distinction and comparison, highlighting some key statistical strength of the Bayesian approach.

The data-set is from a Portuguese banking institution, sourced from the UCI Machine Learning Repository, to analyze direct marketing campaigns promoting term deposits. The goal is to build a classification with a multiple logistic regression model, to infer the client subscription decisions. The data-set collected 4,521 observations, their outputs of the call (subscribe a deposit or none) and 16 features for each consumer like:

- duration of the call,
- preexisting house loan? Yes|No
- date of the campaign
- balance of the customer
- number of previous call etc.

`duration` has been immediately discarded as non-informative (*further information at data description on UCI*) and used `bank.cvs` instead of `bank_all.cvs` for computational reasons.

```
# Read data and encoding of the responce
bank_raw = read.csv("bank.csv", sep = ";")
bank_raw$y = ifelse(bank_raw$y == "yes", 1, 0)

# Check unbalanced
table(bank_raw$y)
```

```
##
##    0    1
## 4000  521
```

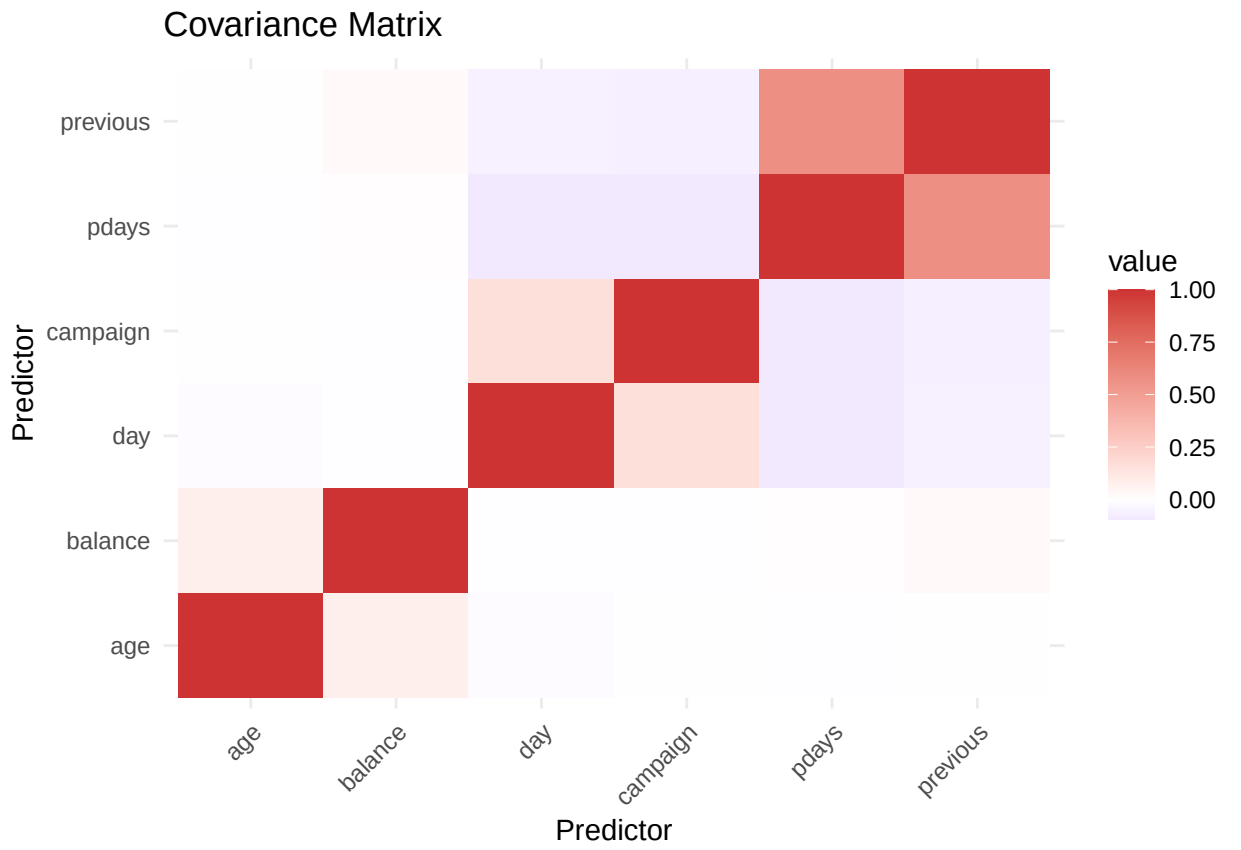
The data may suffer from an issue of unbalanced response variable, causing the model to favor predicting the majority class (i.e., `0`) over the minority class (i.e., positive subscriptions `1`). However, this issue is not investigated, as the aim of this project is not to achieve the best predictions, but rather to focus more on *implementation* and *interpretation* of the approaches.

Features selection

The aim of this phase is select a subset of the features, possibly the best, to fit the model. Different approaches can be implemented in this phase:

- Covariance Matrix
- Bayesian Feature Selection via Latent Variables Gamma

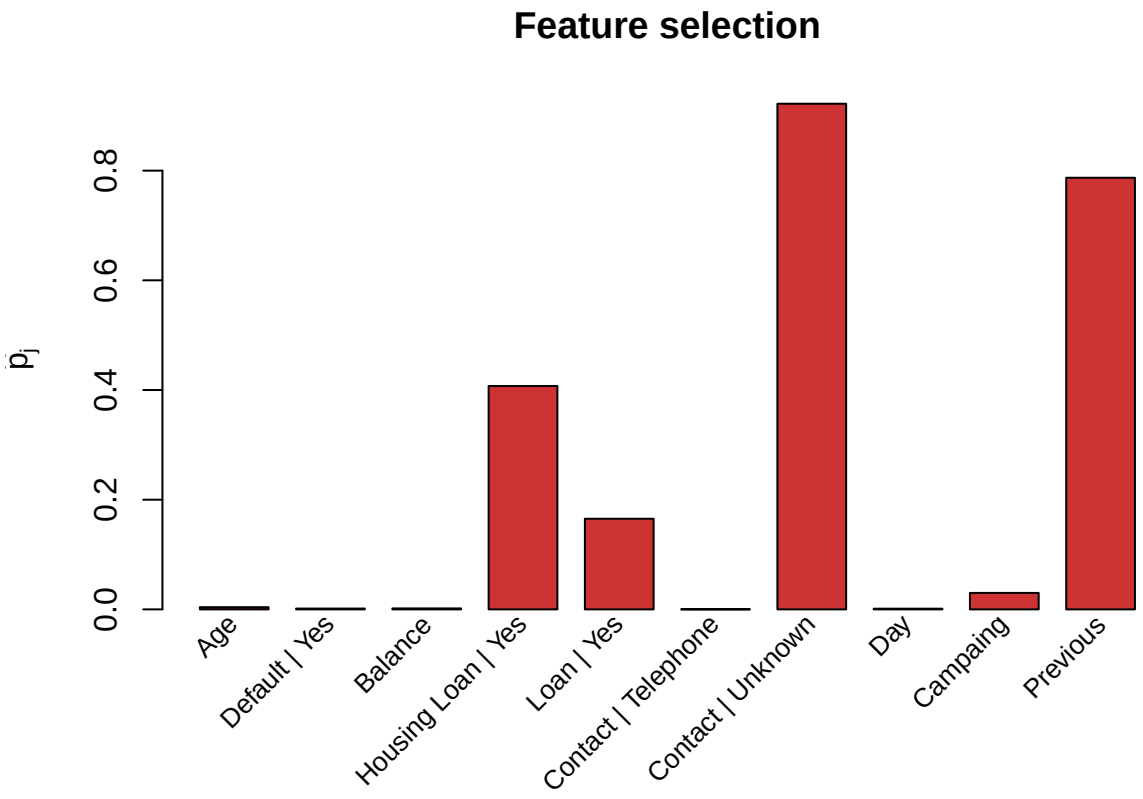
Covariance matrix is the first tool to discard eventual linearly correlated continousfeatures in order to omit them.



Bayesian Feature Selection using Latent Variables, specifically the Gamma procedure, is thoroughly detailed in the *'Bank_Cassola_Sciortino.pdf'* . This approach identifies the features with the highest probabilities of inclusion across the entire model space.

```
## Loading required package: rjags

## Loading required package: coda

## Linked to JAGS 4.3.2

## Loaded modules: basemod,bugs

##
## Attaching package: 'R2jags'
```

```
## The following object is masked from 'package:coda':
##
##     traceplot

## module glm loaded

## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 4521
##    Unobserved stochastic nodes: 21
##    Total graph size: 63319
##
## Initializing model

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    0    0    0    1    0    0    1    0    0     0
## [2,]    0    0    0    1    0    0    1    0    0     1
## [3,]    0    0    0    0    0    0    1    0    0     1
## [4,]    0    0    0    0    0    0    1    0    0     1
## [5,]    0    0    0    0    0    0    1    0    0     0
## [6,]    0    0    0    0    1    0    1    0    0     0

## [1] 4000   10
```



`contact` has been omitted since 'unknown' contact is an irrelevant information at the moment.

Complex models leads to higher time complexity for this algorithms, consider that the bayesian approach take almost 10,000 simulations of each parameter, in more complex scenario some adjustments are needed. Secondly more difficult interpretation, making it challenging for users to understand how predictions are made and which features are influential.

Regression Analysis: Model Fitting and Performance Evaluation

Regression analysis examines the relationship between a quantitative response variable, Y , and one or more explanatory variables, $X_1; ...; X_k$, traces the conditional distribution of Y as a function of the X. Model are before trained on data, and tested with unseen data, in order to evaluate some over-fitting or generalization issue.

Our model is defined as:

$$Y_i = \eta(\beta_0 + \beta_1 X_1 + ... + \beta_4 X_4 + \epsilon_i)$$

We start defining the model:

- Y or labels, is a binary vector (random component)
- X or features, is a 4069 x 5 matrix (intercept, 2 categorical and 2 quantitative)
- link function η : logit function

We proceed dividing the data-set in `train` and `test`:

```
##   Dataset Rows Columns
## 1    Test  452       5
## 2   Train 4069       5
```

and fit the frequentist logistic regression model:

```
# Fit the logistic regression model
glm_model = glm(y ~ ., family = binomial(), data = train)

# Summarize the models
summary(glm_model)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial(), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8533  -0.5638  -0.4558  -0.3764   3.2036
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.47463    0.09226 -15.983  < 2e-16 ***
## housingyes  -0.63937    0.10012  -6.386 1.70e-10 ***
## loanyes     -0.73727    0.17450  -4.225 2.39e-05 ***
## campaign    -0.09477    0.02519  -3.762 0.000169 ***
## previous     0.14041    0.02311   6.075 1.24e-09 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2924.6  on 4068  degrees of freedom
## Residual deviance: 2808.2  on 4064  degrees of freedom
## AIC: 2818.2
##
## Number of Fisher Scoring iterations: 5
```

In the context of statistical modeling, the Bayesian approach to regression offers a robust framework for incorporating prior beliefs about model parameters and updating these beliefs with observed data. This method allows for the estimation of posterior distributions, providing a comprehensive understanding of parameter uncertainty.

Mathematically, the Bayesian approach to regression is based on **Bayes' Theorem**, which is expressed as:

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}$$

The posterior distribution of the coefficients β is then derived as:

$$P(\beta|X, y) \propto P(y|X, \beta)P(\beta)$$

Where:

- $P(\beta|X, y)$ is the posterior distribution of the regression parameters given the data

- $P(y|X, \beta)$ is the likelihood function of the data given the parameters

- $P(\beta)$ is the prior distribution of the parameters

In practice, sampling methods like **Markov Chain Monte Carlo (MCMC)** and **Metropolis-Hastings Algorithm** are used to approximate the posterior distribution of the parameters when it's difficult to compute directly. For the specific problem we need to implement a **Multivariate Normal Logistic Regression** model, fit the data and obtain the posterior distribution of the parameters $\beta_i = [\beta_0, ..., \beta_4]^T$ choosing a non-informative prior distribution.

$$P(y = 1|\mathbf{X}) = \frac{1}{1 + \exp(-\mathbf{X}\beta_i)}$$

Note: *all technicalities about feature selection, convergence diagnostics of the MCMC and comparison between different link function, that are better analyzed in my project for the course of Bayesian Modelling at Univeristà Cattolica del Sacro Cuore; developed on the same data and with the precious help of my colleague Vittoria Cassola, are stored in this repository on GitHub as* `Bank_Project_Sciortino_Cassola.pdf`.

```r
## Predictions
# Predict probabilities for the test dataset
test_predictions = predict(glm_model, newdata = test, type = "response")

# Convert probabilities to binary classes (0 or 1)
predicted_classes_freq = ifelse(test_predictions > 0.35, 1, 0)
```

```r
# Bayesian Logistic Regression Model
# we define the algorithm to compute the posterior distribution of the parameters

logistic_regr = function(y, X, beta.0, s2.0, S){
  ## out : (S,p) matrix collecting S draws from the posterior of beta = (beta1,...,betap)
  n = nrow(X)
  p = ncol(X)

  # IMPORTANT:
  beta_post = matrix(NA, S, p) # inizialize beta_posterior vector

  ## Set initial values (MLE estimates)
  beta = glm(y ~ X - 1, family = binomial(link = logit))$coefficients

  # first row is the initial value (B^T)
  beta_post[1,] = beta

  ###
  ### [Metropolis Hastings scheme sampler]
  ###

  for(s in 1:S){

    for(j in 1:p){

      ## Current value of betaj is:

      betaj = beta[j]

      ## 1. Propose betaj ##

      m.j  = mean(beta_post[1:s,j], na.rm = T) # mean of all betaj values up to iteration s
      s2.j = var(beta_post[1:s,j], na.rm = T)  # variance of all betaj values up to iteration s

      if(is.na(s2.j) | s2.j == 0){s2.j = 1}

      # obs: at the beginning, e.g. s = 1, s2.j can be 0 or NA! In that case, I set s2.j = 1

      betaj.star = propose.beta.j(m.j = m.j, s2.j = s2.j)

      # I create an auxiliary vector beta.star.tmp with all components equal to the current value of
      # beta except for betaj.star
      # This is needed for likelihood evaluation
      beta.star.tmp = beta
      beta.star.tmp[j] = betaj.star

      ## 2. Compute rj ##

      rj = log.like.eval(y, X, beta.star.tmp) - log.like.eval(y, X, beta) +
             log.prior.eval(betaj.star, beta.0[j], s2.0[j]) -
        log.prior.eval(betaj, beta.0[j], s2.0[j]) +
               log.proposal.eval(m.j, betaj.star, s2.j) - log.proposal.eval(betaj.star, m.j, s2.j)
```

```
      ## 3. Accept/reject betaj.star ##

      log.u = log(runif(1))

      if(log.u < rj){betaj = betaj.star}

      beta[j] = betaj

    }

    ## Store sampled draws ##

    beta_post[s,] = beta

  }
  return(posterior = list(beta_post = beta_post))
}
```

Now we run the algorithm for the bayesian approach to compute the posterior distribution of the parameters:

```
# Initializate inputs
n = nrow(X)
p = ncol(X)
beta.0 = rep(0, p)
s2.0   = rep(100, p)

## Run the MCMC
set.seed(24)

S = 7000
out_logistic = logistic_regr(y, X, beta.0, s2.0, S = S)
```

```
## Summary Statistics of Posterior Samples:

##              Mean   Q2.5  Q97.5
## (Intercept) -2.16 -2.178 -2.132
## housing|yes -0.32 -0.320 -0.315
## loan|yes    -0.27 -0.269 -0.263
## campaign    -0.30 -0.320 -0.276
## previous     0.23  0.223  0.233
```

Once fitted both models on the training data-set, observe above the estimated parameter linked to the different independent variables.

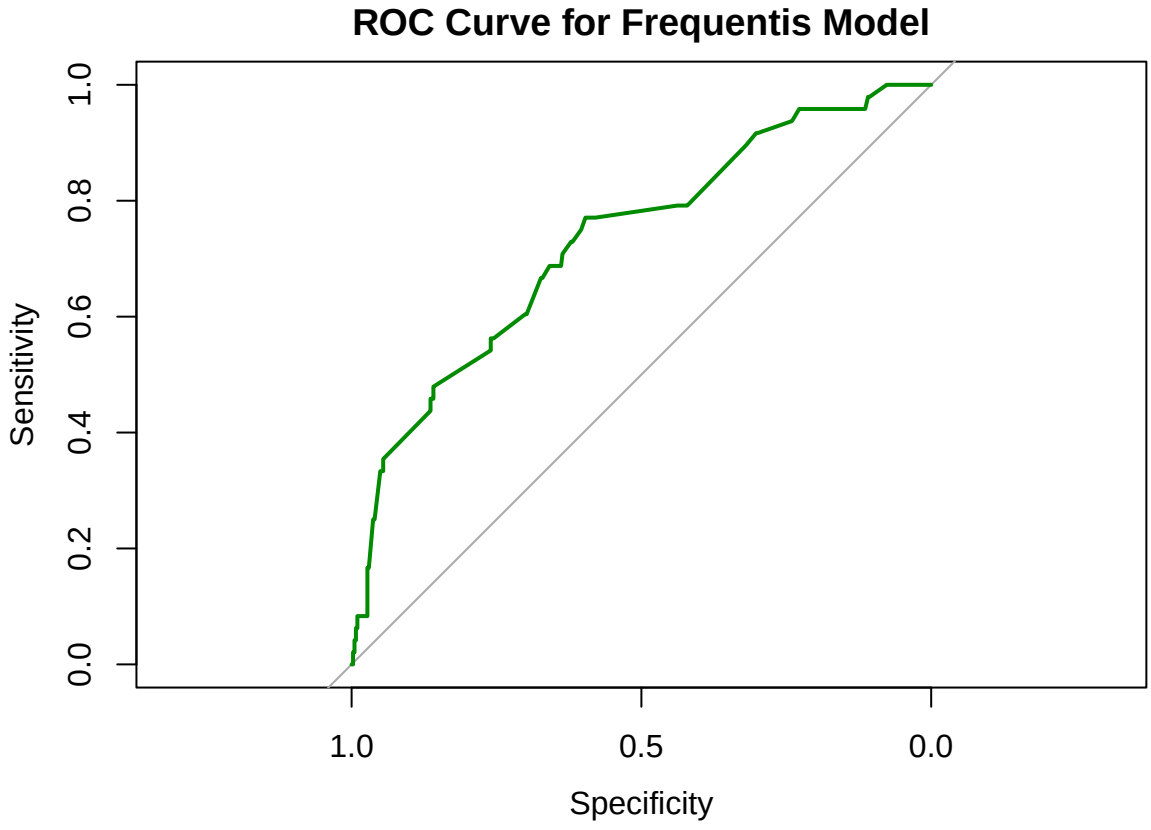
When using the scale() function in regression analysis, the data is standardized at mean equal to 0 and standard deviation to 1. This transformation affects the interpretation of the parameters of both numeric and binary categorical variables. For example 'loan' and 'housing' as binary, refer to the change in the intercept when corresponding variable are affirmative (note that the model suggest a negative effect to the intercept, and so the probability of deposit, for both predictors).

Predictive Performance Comparison of Approaches

Test data-set are used as unseen observations to make predictions from the two models to have a basis for a comparison. Tool like confusion matrix and related measures, inform as about the performance of a model

for prediction.

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

##           Actual
## Predicted   0   1
##         0 401  45
##         1   3   3

## Accuracy: 0.89

## Precision: 0.50

## Recall: 0.06

## F1-Score: 0.11

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



```
##           Actual
## Predicted   0   1
##         0 403  45
##         1   1   3
```

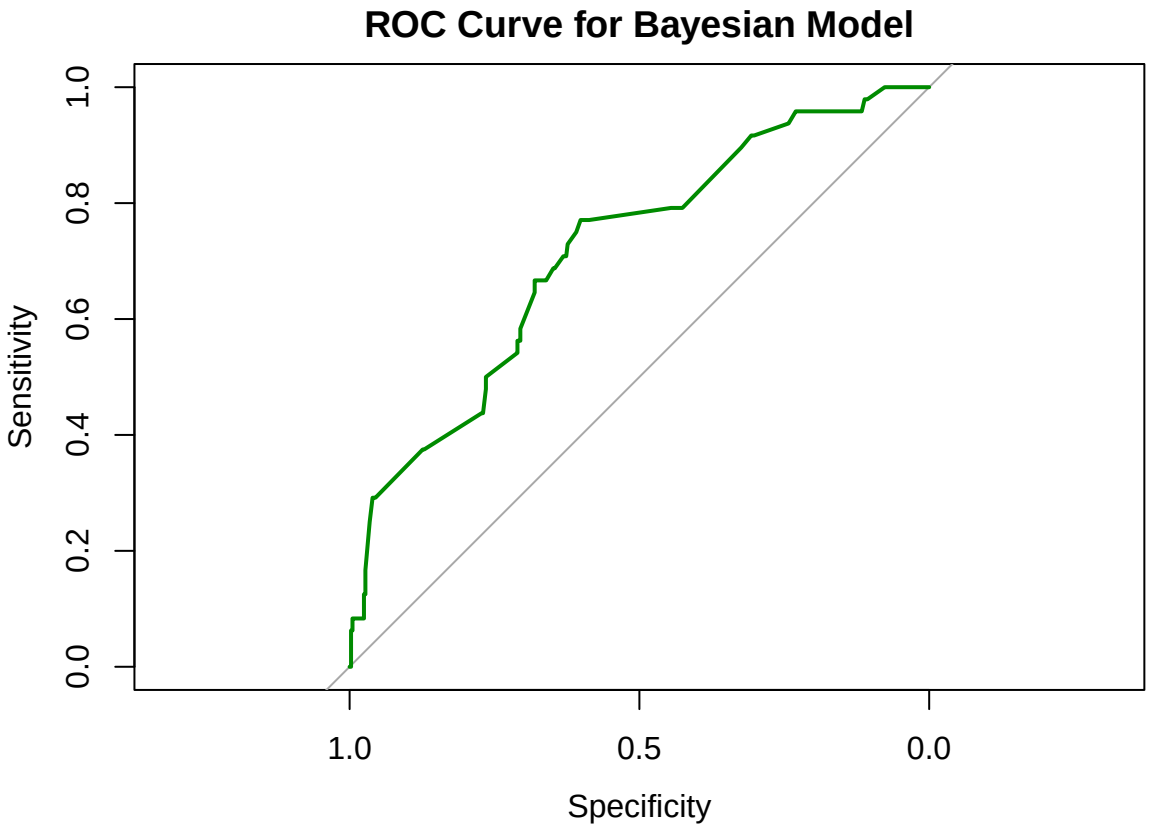
```
## Accuracy: 0.89
```

```
## Precision: 0.75
```

```
## Recall: 0.06
```

```
## F1-Score: 0.12
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```



Due to unbalanced data, the model has difficulties to predict the minority class.

When a non-informative priors in Bayesian framework has been choosed, the resulting confusion matrix and performance metrics of the model reflect only information coming from the data, like in the frequentist model. This similarity often arises when both methodologies are applied to the same data-set, but keep in mind when data is highly imbalanced or exhibits complex interactions, the models may diverge in their predictions.

Precisation:

- the posterior distribution of the random variable for the parameters β_i is derived using Maximum Likelihood Estimation (MLE) like the Frequentist framework, but with a critical distinction: it models parameters as probability distributions rather than fixed values.

This new posterior distribution of the parameters, can serve as prior information for next experiments, as it incorporates information obtained from trained model (Before-After-Control-Impact design techniques could be implemented). Bayesian inference offers a more flexible framework for statistical modeling when prior knowledge or information from previous experiments is available. This probabilistic treatment facilitates a deeper understanding of parameter uncertainty.

Quantifying and Evaluating Uncertainty of Parameter Estimation

In statistical inference, the ***frequentist approach*** is used to quantify and evaluate uncertainty around parameter estimates with two regression analysis tools: ***hypothesis testing*** witch provide a formal decision on whether the predictor is statistically significant, and ***confidence intervals*** that quantify the uncertainty in terms of precision/magnitude around the estimated coefficients, offering a range of plausible values for the true fixed and unknown population parameters.

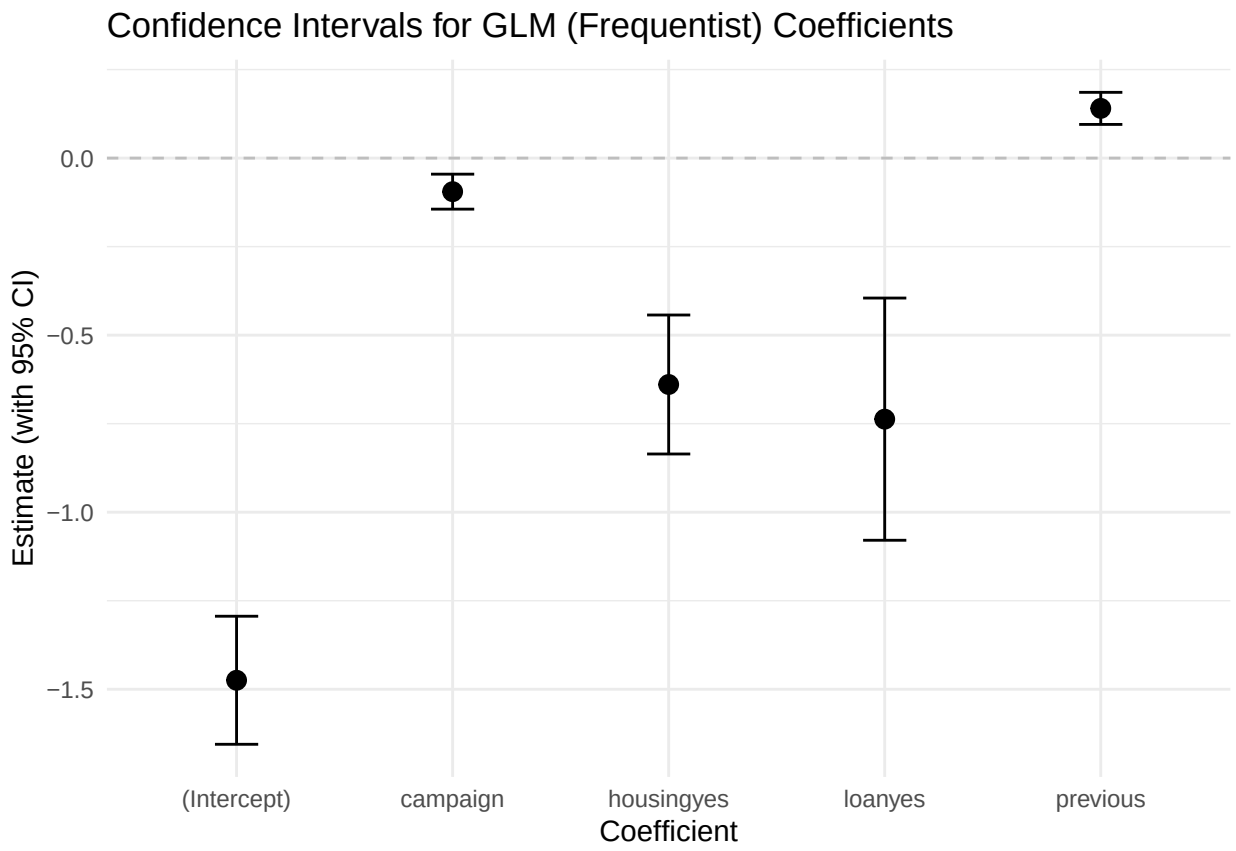
```
## Confidence Interval
```

```
##              2.5 % 97.5 %
## (Intercept) -1.655 -1.294
## housingyes  -0.836 -0.443
## loanyes     -1.079 -0.395
## campaign    -0.144 -0.045
## previous     0.095  0.186
```

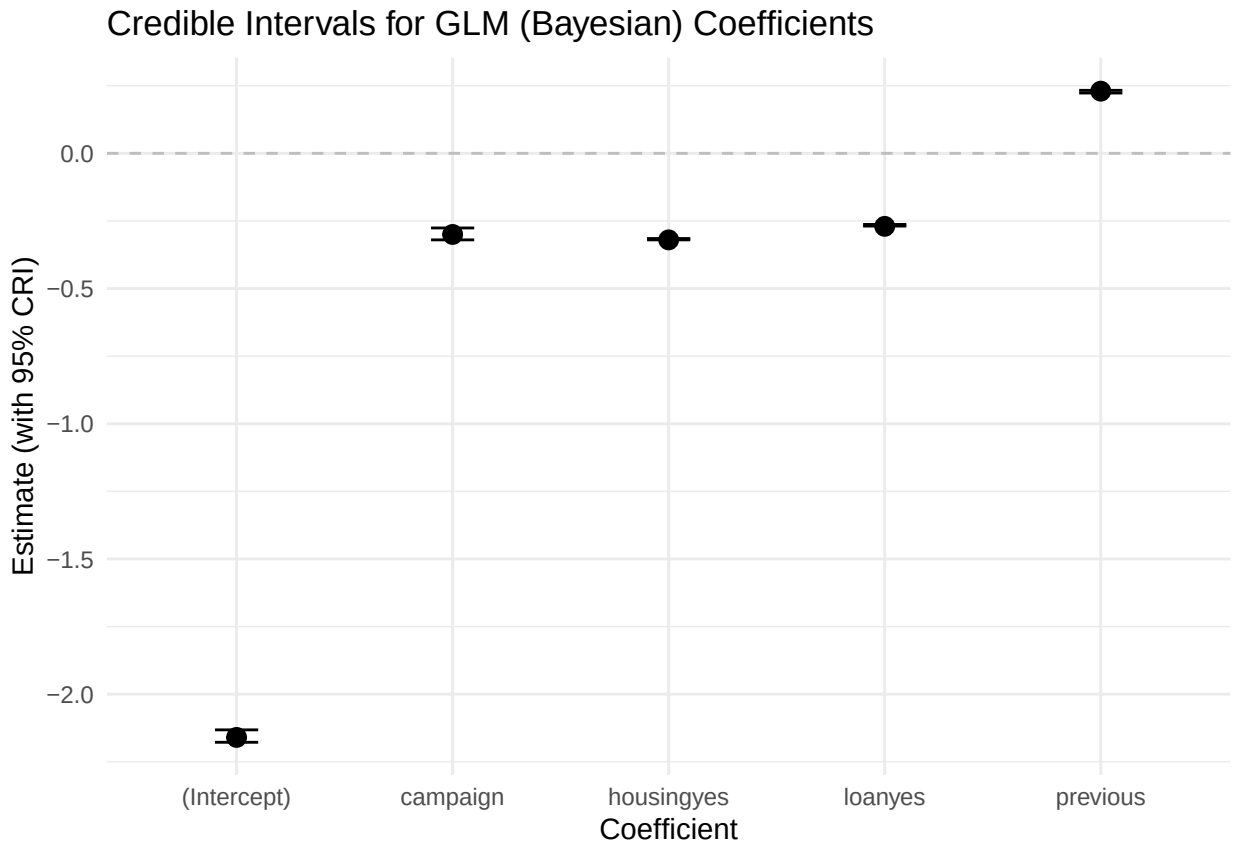
This result can already give some insight:

- `housingyes` (witch is a dummy variable encoded from `housing` representing if the contacted person has already signed a house loan) is between `(-0.925, -0.529)` within a 95% confidence interval, that can be interpretative as negative effect on the **interceprt** for the linear component linked to the probability to subscribe a deposit when the other variable are fixed at their mean value.
- `previous` CI are `(0.112, 0.199)` indicate a positive association with number of previous and the probability of subscription, keeping all other variable constant and at their baseline level.

```
##                   2.5 %      97.5 %
## (Intercept) -1.65546606 -1.29379694
## housingyes  -0.83560958 -0.44313304
## loanyes     -1.07927697 -0.39525673
## campaign    -0.14414266 -0.04538842
## previous     0.09510807  0.18570477
```



With CI we can infer that statistically, if the same campaign were repeated many time, after computing the estimate and the confidence interval, 95% of those intervals contain the true unknown parameter on the long-run, so no information about the parameter uncertainty is given as it's considered a fixed value.




```
## Credible Interval

##                2.5 % 97.5 %

## (Intercept)   -2.18  -2.13
## housing|yes   -0.32  -0.32
## loan|yes      -0.27  -0.26
## campaign      -0.32  -0.28
## previous       0.22   0.23
```

From a ***Bayesian perspective,*** parameters are represented as random variables. Thus parameters $\beta = [\beta_0, ..., \beta_4]^T$ are random variable $\beta_i \sim p(\beta_i)$ with probability functions prior information, combined with the data, $p(y|\beta)$ likelihood function of our data.

With the Bayes theorem we get:

$$p(\beta_i) \propto p(data|\beta_i)p(\beta_i)$$

Now posterior density function of the different parameters can be sampled or obtained from the MCMC and the Metropolis-Hastings Algorithm, so we can directly compute and visualize the distribution of all parameters of the regression. *Credible intervals* (CRI) rely on the inference on those posterior distributions, i.e. identifying the 95% region intervals (conceptually similar to CI) but with the relevant and important difference that CRI can be interpreted implying a quantification of the uncertainty around β_i.

The logistic regression model can be expressed as:

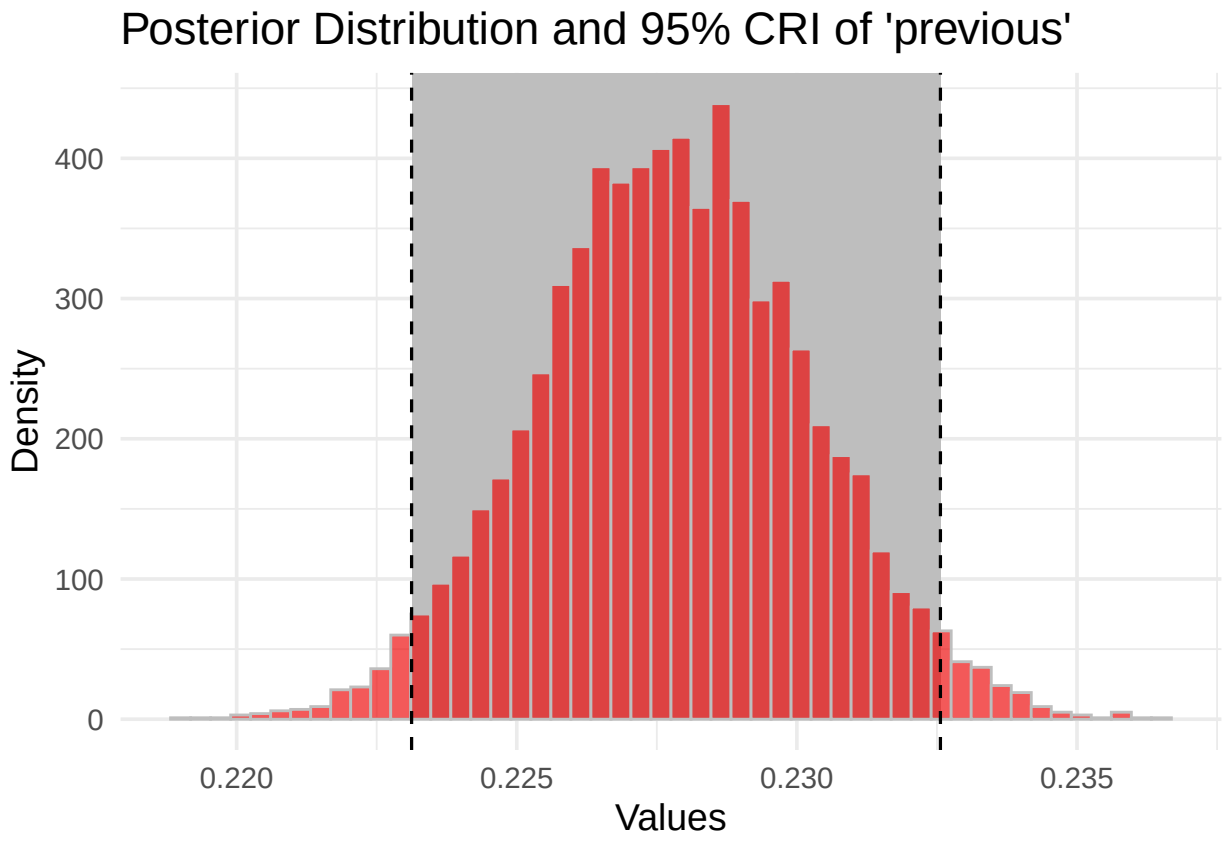
$$P(\hat{Y} = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k)}}$$

where $(\beta_0, \beta_1, \ldots, \beta_k)$ are the coefficients estimated.

- `housingyes` for the bayesian approach we can assert that the true parameter for the change in intercept, linked to the probability to subscribe the deposit $p(Y = 1)$, with 95% probability lies in the CRI.

- `previous` 95% of the times the true parameter lies between (0.23, 0.24) meaning there is a high probable positive effect to probability of subscription, when increasing previous calls and all others variable are considered at their mean value.

The figure below is a visualization of the posterior distribution of the parameter with its CRI.



Conclusion

This project implemented and compared two statistical paradigms, Frequentist and Bayesian, in the context of predicting client subscription decisions for a direct marketing campaign. Below are the key takeaways and conclusions from the analysis:

Frequentist approach

Confidence intervals and hypothesis testing offered tools for assessing the reliability and precision of estimates,

> Limitations: Confidence intervals rely on repeated sampling and do not quantify the probability of the true parameter lying within a single interval.

Bayesian approach

> Strengths : Bayesian inference quantify parameter uncertainty in terms of probabilities and allow the incorporation of prior knowledge. The posterior distributions of parameters also provide a foundation for iterative modeling, where prior knowledge from past experiments can inform future analyses.

> Limitations: The computational cost of Bayesian modeling, particularly with custom MCMC algorithms, is significantly higher than frequentist approaches. - Model performance was not markedly superior to the frequentist model in this case, given the simplicity of the data-set and the absence of strong prior information.

Performance of the model implemented are bad, the use of other model or further implementation in the case is required. The philosophical differences between the Frequentist and Bayesian paradigms reflect broader debates about the nature of knowledge and inference, because thinking to unknown parameters of a population as fixed ... The decision to choose one approach over the other often depends on the specific context of the analysis and its complexity.