Loan Eligibility Prediction

Alessandro Sottile 1873637

July 2022

Contents

Abstract

Today there are many people and businesses that apply for bank loans. The main activity of every bank is the distribution of loans, so its goal must be to give money to people who will pay it back. But for the verification process they take a long time. To predict if a client may be eligible for a loan I applied three models 1) Bayesian Logistic Regression, 2) Bayesian Cloglog and 3) Frequentist Logistic Regression. In this way, using certain characteristics of loan applicants, the models were able to predict an applicant's eligibility quite well, as affirmed by the high values of the calculated metrics.

1 Introduction

Our banking systems have many products to sell, but it's know that the main profit comes directly from the loan's interest. The banks are central in the modern economy, they have to decide if a costumer is a good(non-defaulter) or bad(defaulter) one before giving the loans to the borrowers. This type of prediction is a very difficult and time consuming task for any bank or organization.

Although considering the delicacy in dealing with this topic, in fact, if a bank lends money to many individuals who cannot repay the debt, this will have a powerful economic effect, my goal is to try to automate and speed up the process of verifying the requirements for obtaining a loan.I used Bayesian and Frequentist Logist regression to predict the outcome of the elegibilty for a loan.

In addition, I created interesting new features on which I applied statistical models, evaluated through numerous metrics. Finally, in the conclusion are presented some recommendations and salient points that I found during the analysis

2 Related Works

A lot of researcher have worked on how to build predictive models to automate the process of targeting the right applicants. Professor Amruta Sankh and his students of the Atharva College of Engineering in Mumbai use different machine learning models such as Random Forest, Naive Bayes, Decision Tree and logistic regression. In a similar way, many other data scientist treated of this problem. For example 26 Kaggle users have uploaded very interesting codes on numerous statistical models, using the same data that I have used on these pages. Moreover, an article about this theme has been publishing in the famous site Towards Data Science.

3 Dataset and Benchmark

To conduct the analysis presented before i have used the ”Loan Eligible Dataset” from Kaggle, it contains 614 observation of thirteen variables:

1. **Loan_ID**: the unique loan identifier
2. **Gender**: the gender of the costumer
3. **Married**: it refers if the costumer is married (Y) or not (N)
4. **Dependents**: it refers to the number of dependents of the client
5. **Education**: it refers if the costumer is graduated (Y) or not (N)
6. **Self_Employed**: it refers if the costumer is self employed (Y) or not (N)
7. **ApplicantIncome**: the applicant income in dollars
8. **CoapplicantIncome**: the coapplicant income in dollars
9. **LoanAmount**: the loan amount in thousands of dollars
10. **Loan_Amount_Term**: the term of a loan in months
11. **Credit_History**: it refers if the costumer has repaid his past debt (Y) or not (N)
12. **Property_Area**: it refers to the property area: Urban/ Semi-Urban/ Rural
13. **Loan_Status**: it refers if the costumer has obatined the loan (Y) or not (N)

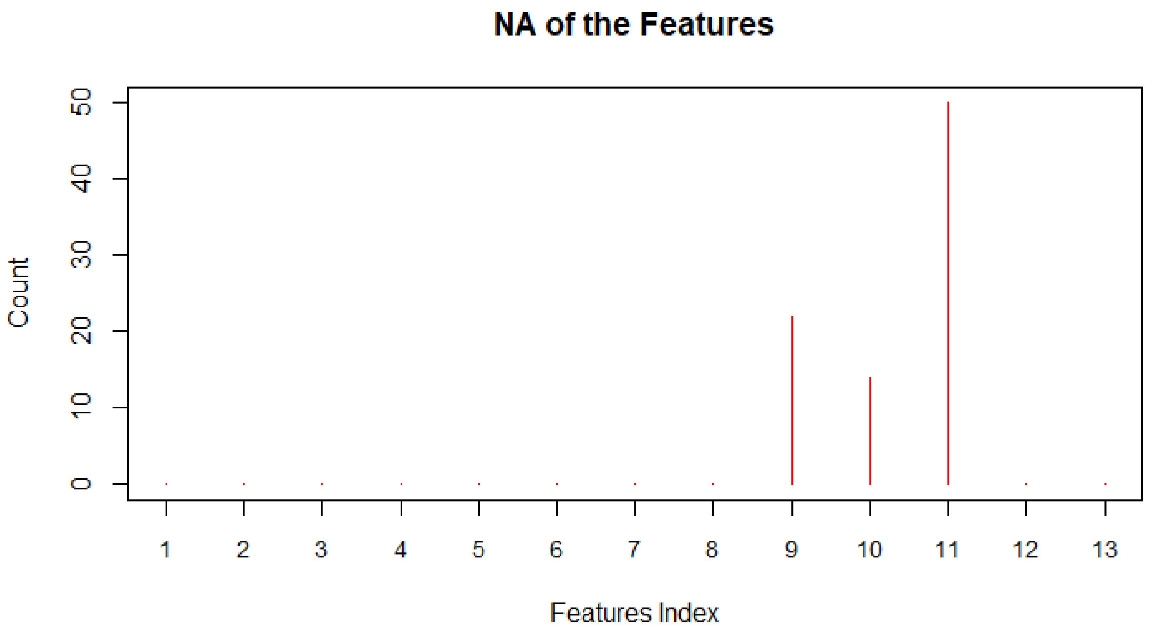


Figure 1

Figure 1 shows the NA count for each feature in the dataset, you can see that only three variables have NAs, and in particular the feature Credit History has 50 missing values, which is more than all. The three varibles with missing values are also the most delicate in the dataset.

About the performance of the models built on this dataset,from the results that i saw on Kaggle and in almost all cited papers I can say that the accuracy of logistic regression is pretty high, in mean that metric is about 80%. in addition, almost all the other models, even the more complex ones, also performed well by always having rather high metric values.

Even in my case, with the models reported in this paper the results were quite satisfactory, on average every metric shown has values higher than 80%.

4 Features engineering

To improve the interpretability of the variables I decided to implement some modifications to the features:

The first thing I did was to transform the variables that had Y/N modes into binary 1/0 features. Then, I have created the feature **Income**: is the combination of *ApplicantIncome* and *CoapplicantIncome*, it can be very useful to have the total income merged versus the two separated. I also have changed the name *Loan_Status* in **Output**.

From the original dataset i have decided to use in the analysis only these variables:

- **X1 = Income**
- **X2 = Credit_History**
- **X3 = Amount**
- **X4 = Education**
- **Y = Output**

I decided to select only the variables listed because, both based on the results read in the literature by other scholars who have built models on this dataset and through some of my own considerations. For example, I did not include the variable Loan_term because in analyzing the distribution 85% of the data assumed the 360 mode, and the other 9 modes distributed the remaining 15%, and I thought it was not very useful in my work.

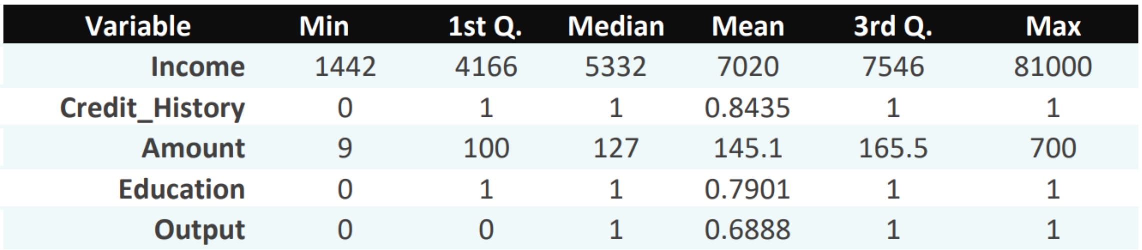
Another example is the variable related to gender, I decided not to select it because in the article reported on towardsdatascience, the scholar found that it was almost irrelevant to the target variable.

In addition, I decided to delete all rows where there was at least one missing value, so the number of observations after this procedure is qeual to 543. I chose to go this route rather than impute missing data, as I wanted only data actually recorded in the questionnaire.

In the end, I divided the dataset into train and test, via the 80-20 split rule. i used the train dataset to fit the model and then used the other to apply the metrics reported at the end of my paper.

5 Exploratory Data Analysis

The first thing i did was to visualize the summary of the dataset



Variable	Min	1st Q.	Median	Mean	3rd Q.	Max
Income	1442	4166	5332	7020	7546	81000
Credit_History	0	1	1	0.8435	1	1
Amount	9	100	127	145.1	165.5	700
Education	0	1	1	0.7901	1	1
Output	0	0	1	0.6888	1	1

Table 1: Summary of the dataset

As shown in Table 1, the binary variables are more concentrated around the value 1 instead of 0, in fact for these variables the mean is greater than 0.5 and the median is 1. Focusing on both numerical features, it is possible to say that the range of values they take on is very wide and there seems to be the presence of outliers, especially in the variable Income.

To further my analysis, I decided to graphically represent my viariables

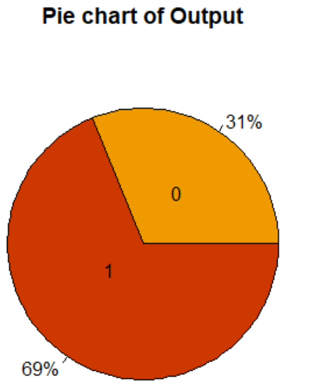


Figure 2

From the Fig.2, it is possible to say that there are more people who got the loan than those who did not get the loan. So, my target variable is a little unbalanced.

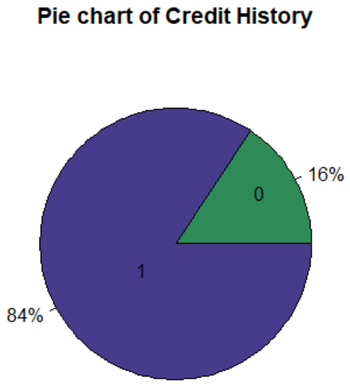


Figure 3

There are many more people in the dataset with a good Credit_History than those with a bad one (Fig.3).This could also hide a baias, as it is likely that individuals who have not repaid their debts in the past knowing the stringent banking rules, will not even go for a new one.

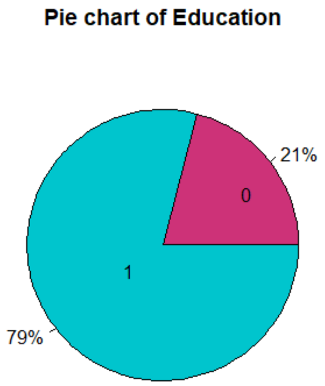


Figure 4

Moreover, there are more highly educated individuals than those with low levels of education, as shown in Figure 3.

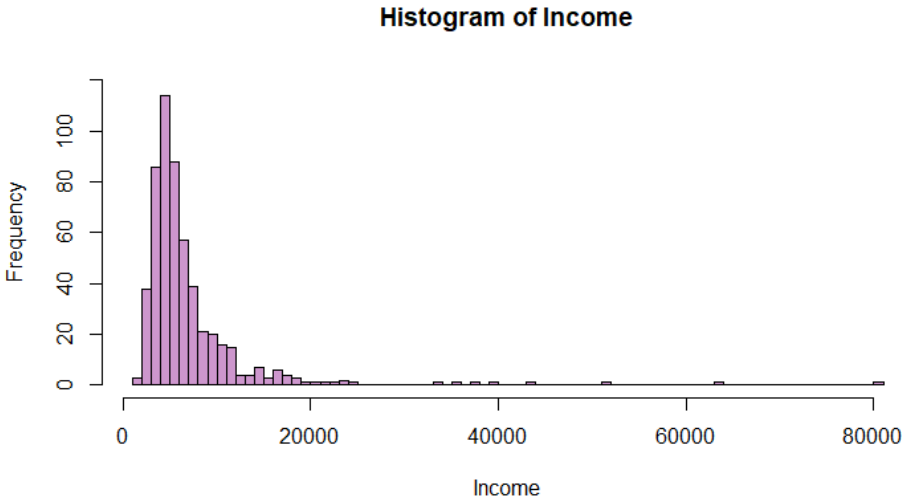


Figure 5

Analyzing the income variable (Fig.5) we can immediately see that the histogram seems to follow the classical income distribution, in which the vast majority of the population is in a very small range, and there are very few individuals who earn much more than others. Another indicator of the non-symmetry of this figure is the fact that in the summary of Table 1 the mean is larger than the median. indeed, it is well known that the mean is an indicator of position that is greatly affected by outliers, as oppose to the median. In the dataset we find values ranging from $1442 percipitated by the lowest income person, to $81000 by the highest income person.

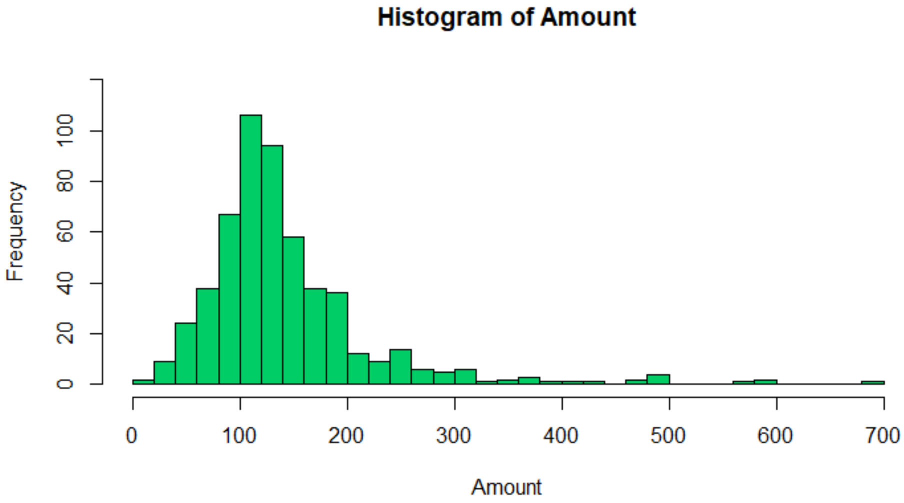


Figure 6

As shown in Figure 6, the same things said above also apply to this variable, although the latter having a smaller range of values, as it is expressed in thousands of dollars, makes the dispersion of values and the presence of outliers smaller and more contained.

If I had analyzed only Fig.2 it would have seemed that it is easy to get the loan, but by analyzing the distributions of the other variables in the dataset as well, the argument changes: there were more people who got the loan, because there are more people in the dataset with positive than negative characteristics (such as the Credit History variable).

Now, I want to analyze the relationship between some features.

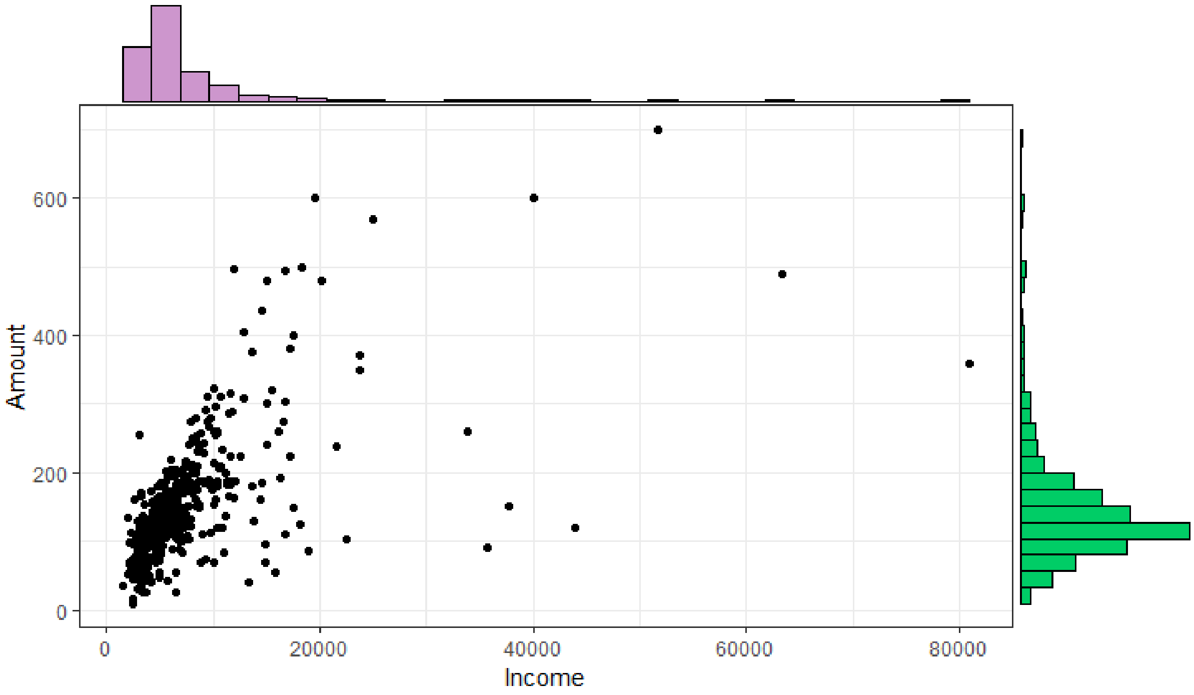


Figure 7: Income distribution versus the amount of loan required

The plot in Figure 7 is tring to capture the relationship between the income and the amount of money required by each people in the dataset. It's possibile to notice that there is a notable value of correlation among them: the higher the income, the larger the amount requested to be borrowed. This also makes sense from a logical point of view, as an individual typically borrows a consideously higher amount than he or she earns.

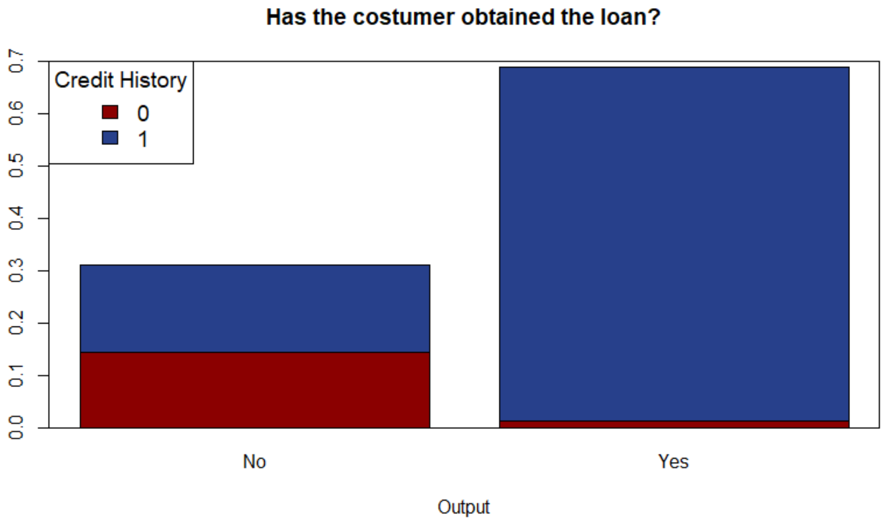
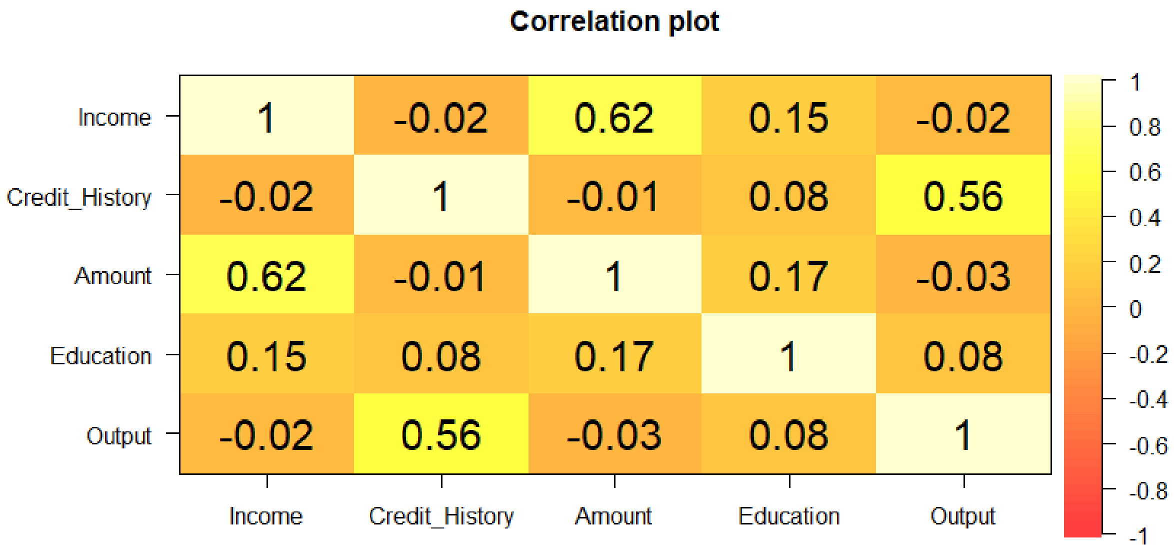


Figure 8: Output in relation to Credit History

The Fig. 8 reports a very important relationship: it compares in individuals who obtained the loan and in those who did not obtain the loan the fraction of those with good and bad credit history. When analyzing those who have obtained the loan, it can be seen that the vast majority are individuals with a good credit history. on the other hand, those who have not obtained the debt are divided almost equally between the two groups.This means that banks have very stringent rules: having a good credit confidence is necessary to get the loan but it is not enough.

Finally, to get a summarizing idea of the relationships among variables, I calculated Pearson's correlation between the features (Fig 9).



	Income	Credit_History	Amount	Education	Output
Income	1	-0.02	0.62	0.15	-0.02
Credit_History	-0.02	1	-0.01	0.08	0.56
Amount	0.62	-0.01	1	0.17	-0.03
Education	0.15	0.08	0.17	1	0.08
Output	-0.02	0.56	-0.03	0.08	1

Figure 9: Correlation among the variables

There are only two couples of features that have a noteworthy relationship: (Amount, Income) and (Output, Credit). The first has a correlation of 0.62, and the second of 0.56, these corresponds of what already seen earlier in the figures. Thus, it is understood that the variable most related to the target feature is Credit History, as it was possible to expect .

6 First Model

In the first model, I decide to use the following Logistic Regression model with the logit as link function:

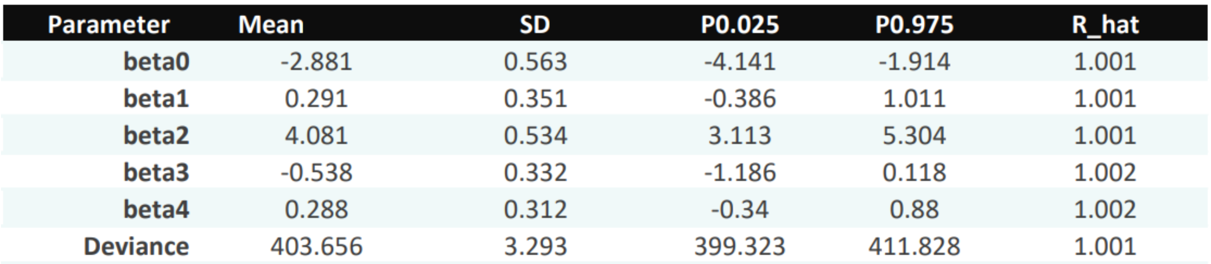
$$Y = y_1 ... y_n \sim Bernoulli(\pi)$$

$$logit(\pi) = log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \beta_4 * x_4$$

$$\beta_i \sim beta(0, 0.000001), i = 1, 2, 3, 4$$

The variable of interest is binary, so it's natural to use a Bernoulli distribution for it. In addition I have scaled the features X_1 and X_3

The main obiective is to estimate the posterior distribution for the parameters $\beta_0, \beta_1, \beta_2, \beta_3, \beta$ through Bayesian methods. I used the JAGS package of R to fit the model defined above, in wich I have implemented two chains, the number of iterations is equal to 1000 and a burn in period of 1000. In addition, I have splitted the dataset in Test and Train, and i have used the latter one to train all the models.

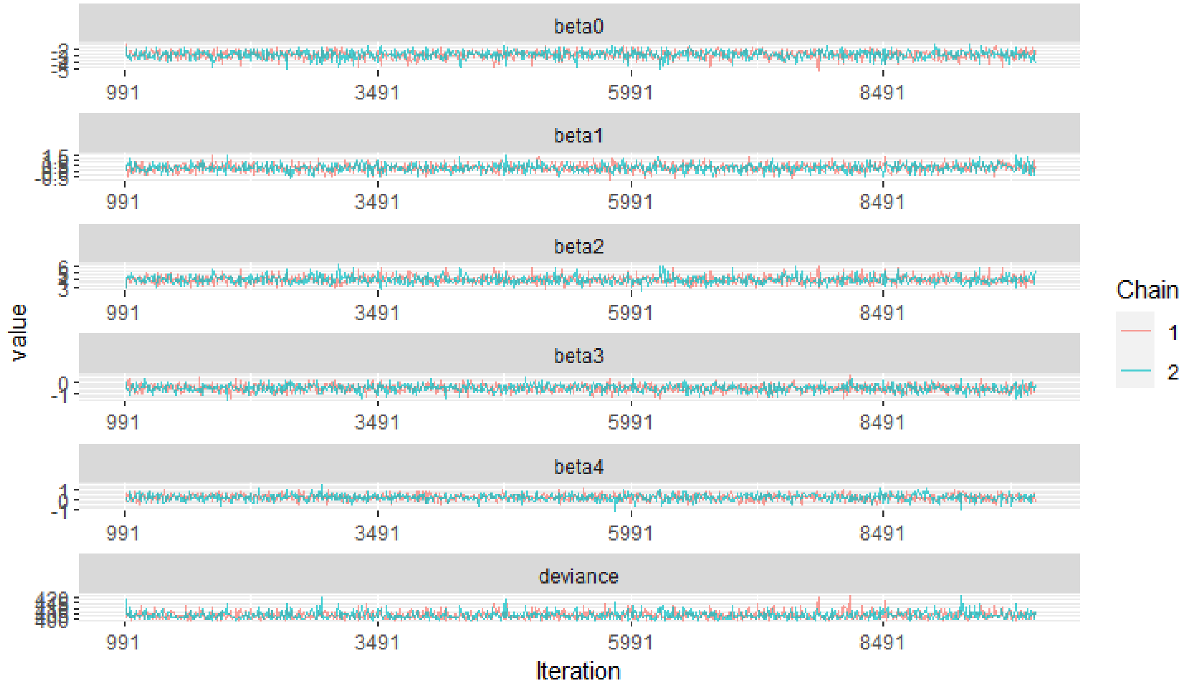


Parameter	Mean	SD	P0.025	P0.975	R_hat
beta0	-2.881	0.563	-4.141	-1.914	1.001
beta1	0.291	0.351	-0.386	1.011	1.001
beta2	4.081	0.534	3.113	5.304	1.001
beta3	-0.538	0.332	-1.186	0.118	1.002
beta4	0.288	0.312	-0.34	0.88	1.002
Deviance	403.656	3.293	399.323	411.828	1.001

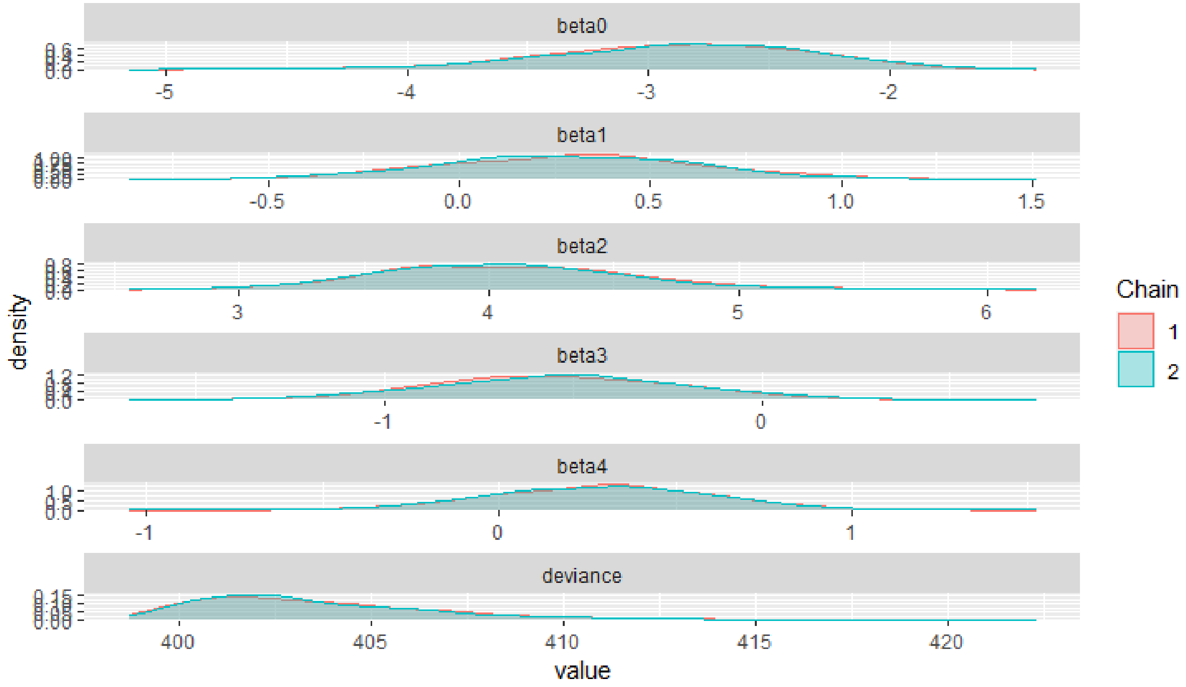
Table 2: Parameters from Bayesian Multiple Logistic Regression

- N: 435
- DIC: 409.1

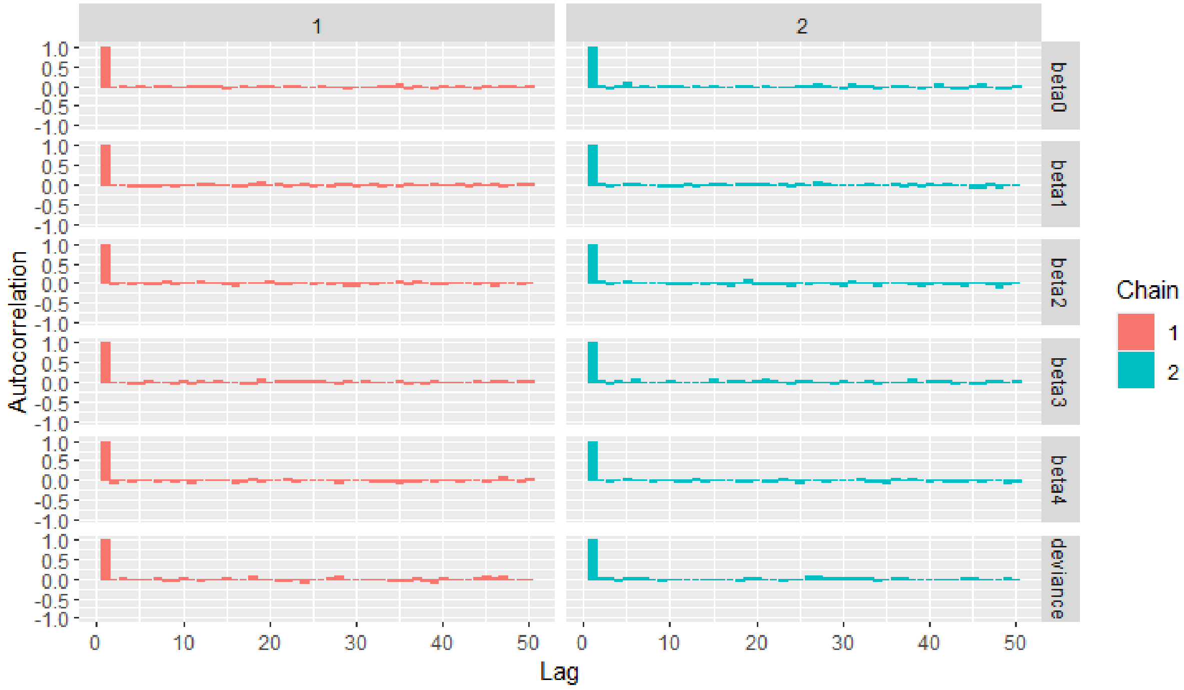
The Table 2 represents the display of the first model proposed in this work. From the table it's possible to view the estimates for all the parameter (the Mean column). In addition, the two quantiles can be seen as the two estremes of the 95% credible set interval. In the end, the Deviance Information Criteria (DIC), that will be usefull to compare the model,is equal to 409.1 .



From the traceeplots it's possible to see that the distribution of the Markov Chain for every parameter is balanced around the mean.



The parameters have a prior distribution that is more or less symmetric in resepct to the mean.

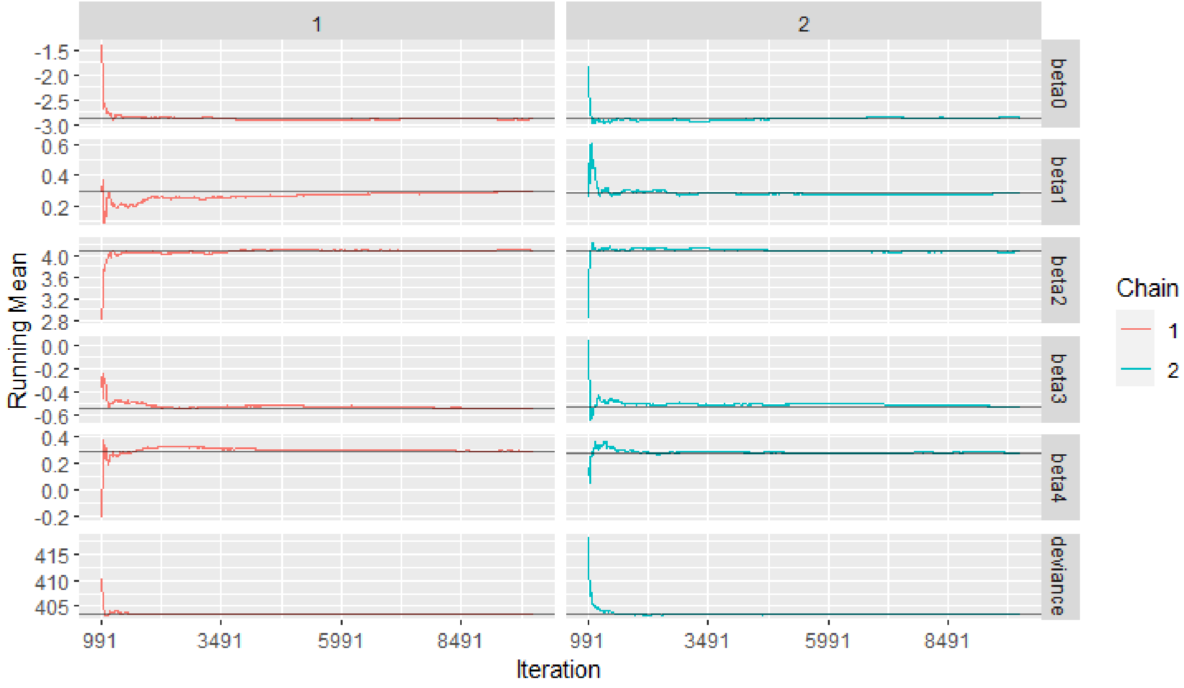


The autocorrelation for the parameters is very small.

In addition, I want to evaluate the empirical averege of t, increasing t=1,...,T

$\hat{I}_t = \frac{1}{T}\sum_{i=1}^{T} h(\theta^{(i)})$

It's know that $I \approx \hat{I}_t$ via the SLLN theorem



It's possible to state that, even if for all parameters the two chains started in different initial points, they achives the same end point.

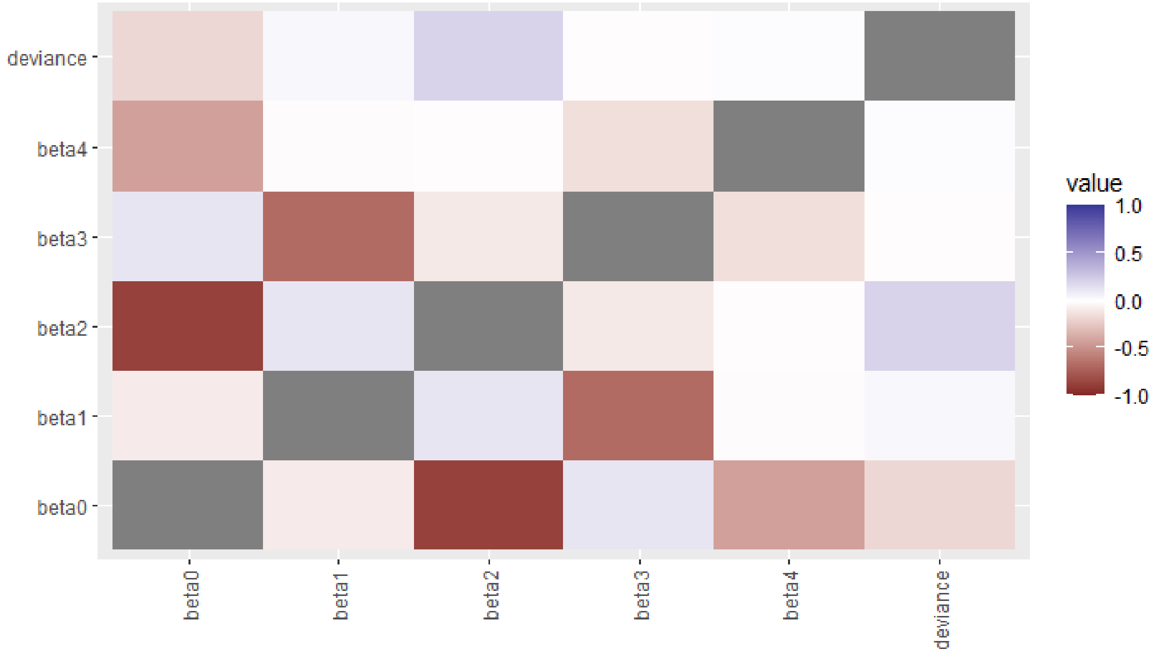


Figure 10

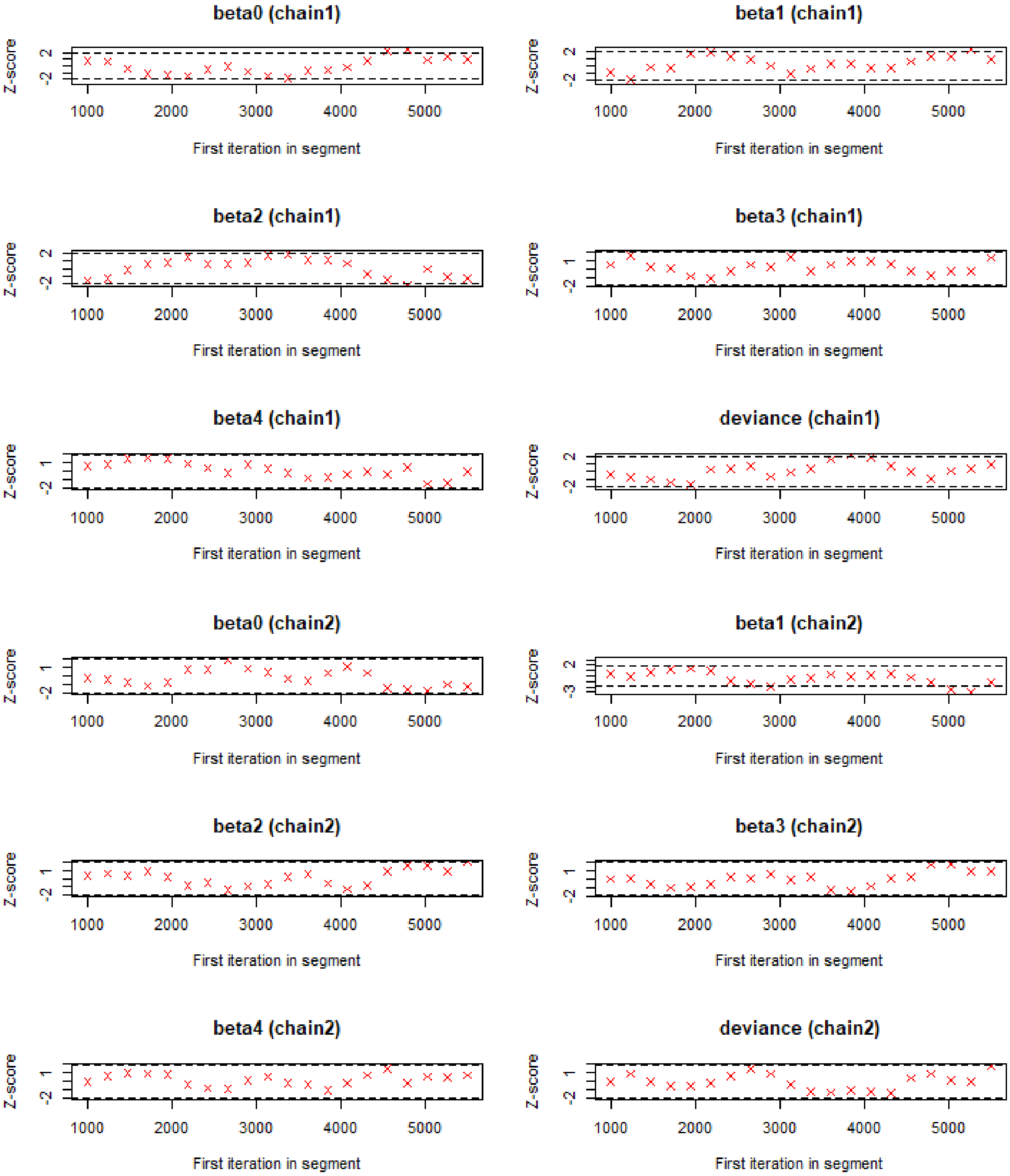
From the Correlation matrix reported in Figure 10, I can say that the couple with the strongest correlation (in absolute sense) are beta0 and beta2. (beta1,beta3) also has a notable degree of correlation.

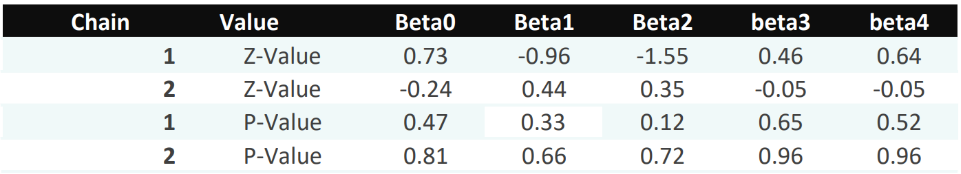
6.1 Diagnostics

Four different diagnoses are presented in this section:

- The Geweke Diagnostic
- The Raftery Lewis Diagnostic
- Gelman Diagnostic
- Heidel Diagnostic

The first tool that i want to use to check is the **Geweke's Diagnostic**.This diagnostic is based on a test for equality of the means of the first 10% and last 50% of a Markov chain. If the samples are extracted from a stationary chain distribution, the two averages are equal and the Geweke statistic has an asymptotically normal distribution. The Geweke's Diagnostic helps to understand if the burning period choosen is enough.





Chain	Value	Beta0	Beta1	Beta2	beta3	beta4
1	Z-Value	0.73	-0.96	-1.55	0.46	0.64
2	Z-Value	-0.24	0.44	0.35	-0.05	-0.05
1	P-Value	0.47	0.33	0.12	0.65	0.52
2	P-Value	0.81	0.66	0.72	0.96	0.96

Table 3: Results of Geweke Test

From the plot, can be seen that most of the values are inside the interval -2 to 2, so the null hypotesis of equality of means is not rejected, that mean that the burning period is enough for all parameters.

Then the **Raftery & Lewis Diagnostic** estimates the number of iterations needed to achive the given level of precision in posterior samples.

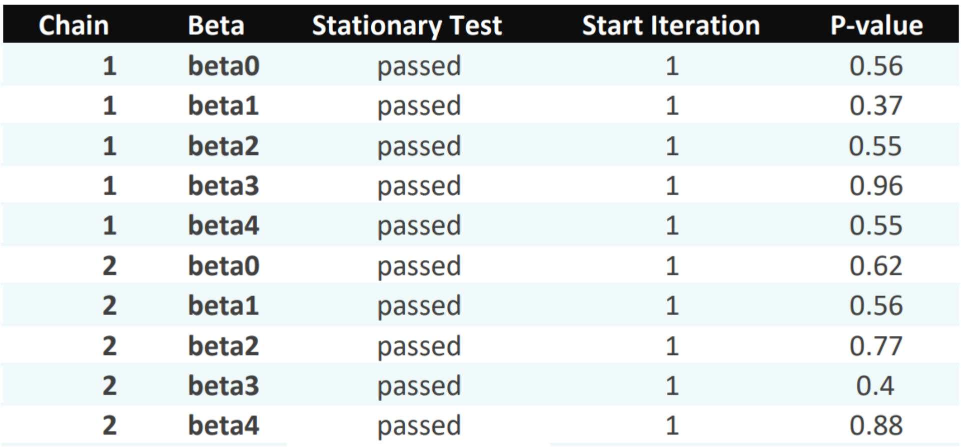
I used those values:

- Quantile Q = 0.025
- Accuracy R = 0.005
- Probability of Accuracy S = 0.95

From the function, i discover that are needed **3746** samples.

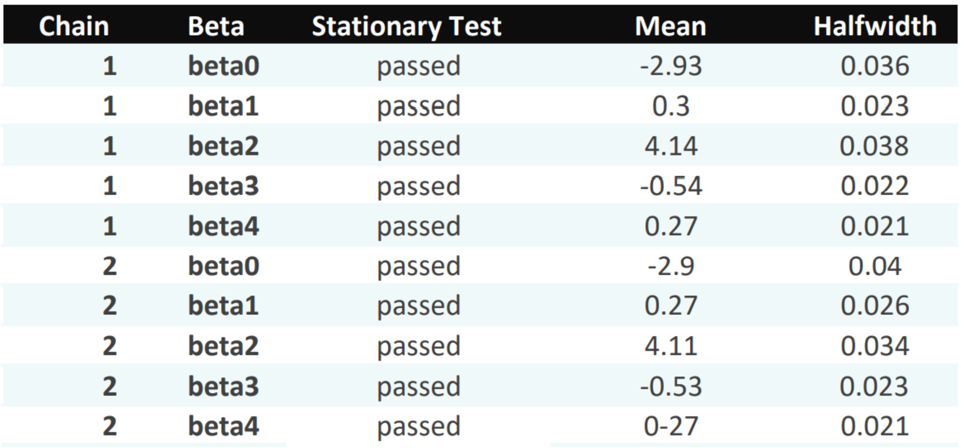
The **Heidel Diagnostic** is used to accept or reject the null hypotesis that the Markov chain is from a stationary distribution. The test is divided in two part:

In the First part (the convergence test in Table 4) it calculates Cramer-von-Mises over the chain in a iteratively way, until the null hypothesis holds stationary or until half of the chain had been discarded. The half-width test (Table 5) calculates half the widith of a (1-α)% credible interval for the mean. If the ratio of the half-width and the mean is lower than ϵ, the chain passes the test. Otherwise the length of the sample is deemed not long enough to estimate the mean with sufficient accuracy.



Chain	Beta	Stationary Test	Start Iteration	P-value
1	beta0	passed	1	0.56
1	beta1	passed	1	0.37
1	beta2	passed	1	0.55
1	beta3	passed	1	0.96
1	beta4	passed	1	0.55
2	beta0	passed	1	0.62
2	beta1	passed	1	0.56
2	beta2	passed	1	0.77
2	beta3	passed	1	0.4
2	beta4	passed	1	0.88

Table 4: Part 1 of the Heidel Test



Chain	Beta	Stationary Test	Mean	Halfwidth
1	**beta0**	passed	-2.93	0.036
1	**beta1**	passed	0.3	0.023
1	**beta2**	passed	4.14	0.038
1	**beta3**	passed	-0.54	0.022
1	**beta4**	passed	0.27	0.021
2	**beta0**	passed	-2.9	0.04
2	**beta1**	passed	0.27	0.026
2	**beta2**	passed	4.11	0.034
2	**beta3**	passed	-0.53	0.023
2	**beta4**	passed	0-27	0.021

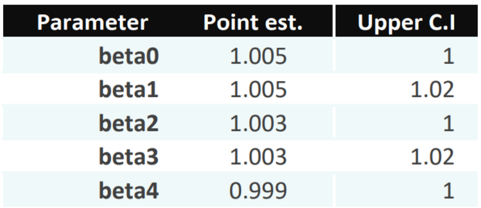
Table 5: Part 2 of the Heidel Test

All the stationarity tests are passed for both chains.

The last tool I have used in this section is the **Gelman Test**, it calculates the potential scale reduction factor for each variable, together with upper and lower confidence limits.

The test diagnoses Approximate convergence when the upper limit is close to 1.

The confidence limits are based on the assumption that the stationary distribution of the variable under examination is normal. Hence the 'transform' parameter may be used to improve the normal approximation.



Parameter	Point est.	Upper C.I
beta0	1.005	1
beta1	1.005	1.02
beta2	1.003	1
beta3	1.003	1.02
beta4	0.999	1

Table 6: Results of Gelman Test

In this case, most of the values are equal to 1, and all are extremly close to that value.

It's possible to say that π is a stationary distribution if $\pi * A = \pi$ (in the discrete case, where A is the transition probability matrix). The $\sum_{i:1}^{N}(\pi_i)$ must sum to 1. In the continuos case instead: π is a stationary distribution if $\pi(y) = \int(q(x,y)\pi(x)dx$ (q(x,y) is the transition density)

6.2 Parameters Recovery

To check the ability of my Bayesian model to correctly recover the model parameters, I decided to compute Parameter Recovey. I performed a simulation with the data simulated from the Bayesian Logistic Regression. I defined True Parameters the ones estimated by the model:

- beta0 = -2.916
- beta1 = 0.284
- beta2 = 4.124
- beta3 = -0.531
- beta4 = 0.270

By means of the runjaggs packet, i was able to calculate the output of the model. I have used the same model as before:

$$Y = y_1 ... y_n \sim Bernoulli(\pi)$$

$$logit(\pi) = log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \beta_4 * x_4$$

$$\beta_i \sim beta(0, 0.000001), i = 1, 2, 3, 4$$

Then, I have simulated the response variable **Output**

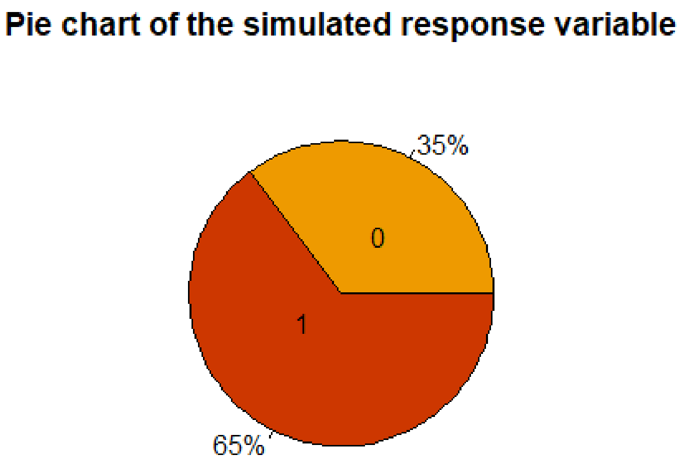
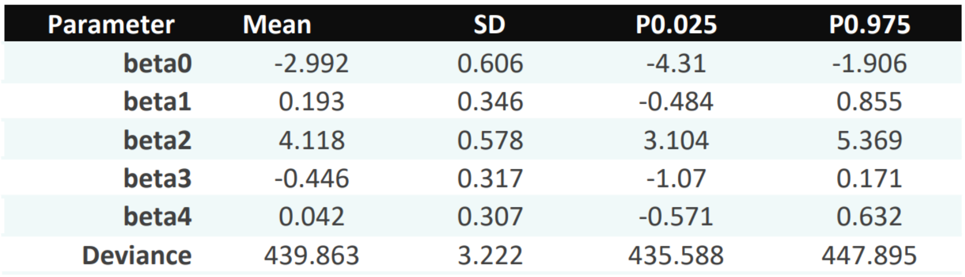


Figure 11

The Pie chart (Fig. 11) of the simulated values shows that the Output variable is as unbalanced as the original ones were.

In the end, I fitted the model by jaggs.



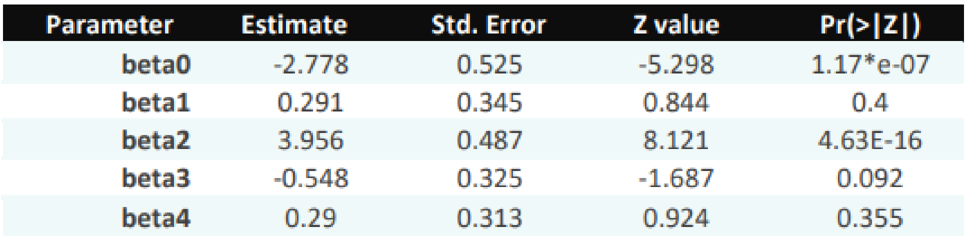
Parameter	Mean	SD	P0.025	P0.975
beta0	-2.992	0.606	-4.31	-1.906
beta1	0.193	0.346	-0.484	0.855
beta2	4.118	0.578	3.104	5.369
beta3	-0.446	0.317	-1.07	0.171
beta4	0.042	0.307	-0.571	0.632
Deviance	439.863	3.222	435.588	447.895

Table 7: Parameters from simulated Bayesian Multiple Logistic Regression

From the Table, it is possible to notice that estimated parameters are very close to the ”real” betas, they are all inside the credible intervals. Thus, the proposed Bayesian model can correctly recover the model parameters.

7 Frequentist Model

I fitted a Frequentist Logistic Regression in R to compare the performance of the Bayesian Logistic Regression. I used the glm() function on the same data of before. In addition I have scaled the features X_1 and X_3.



Parameter	Estimate	Std. Error	Z value	Pr(>|Z|)
beta0	-2.778	0.525	-5.298	1.17*e-07
beta1	0.291	0.345	0.844	0.4
beta2	3.956	0.487	8.121	4.63E-16
beta3	-0.548	0.325	-1.687	0.092
beta4	0.29	0.313	0.924	0.355

Table 8: Parameters from Frequentist Multiple Logistic Regression

- AIC: 408.5

As shown in Table 8, the coefficient estimated in this approach are very similiar to those obtained in the Bayesian model. Also the value of the AIC is almost the same of the DIC of the Bayesian approach (409). The coefficients *beta0* and *beta2* are very significant in the model.

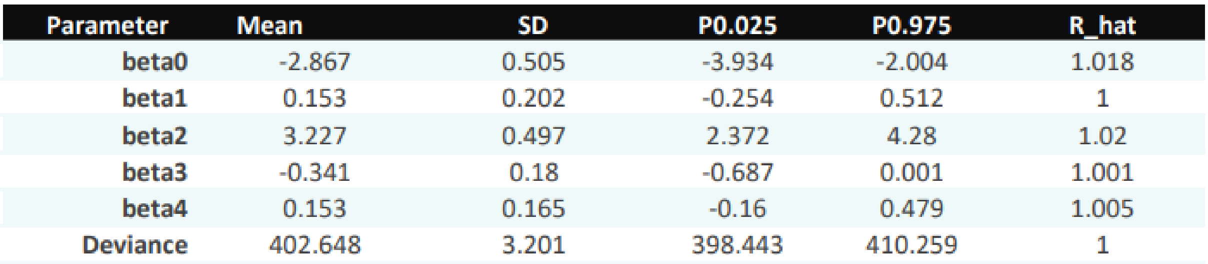
8 Second Model

Now, I want to compare the model in the Section 6 with another one, the cloglog. As before I have scaled the features X_1 and X_3:

$$Y = y_1 ... y_n \sim Bernoulli(\pi)$$

$$cloglog(\pi) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \beta_4 * x_4$$

$$\beta_i \sim beta(0, 0.000001), i = 1, 2, 3, 4$$

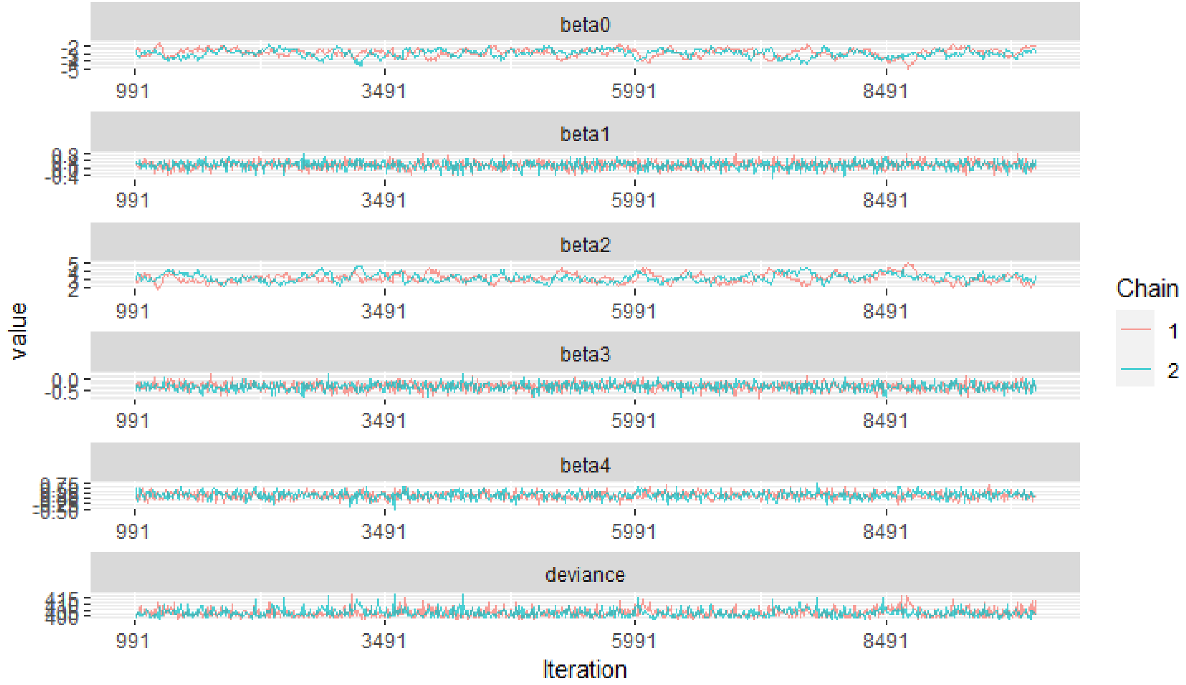


Parameter	Mean	SD	P0.025	P0.975	R_hat
beta0	-2.867	0.505	-3.934	-2.004	1.018
beta1	0.153	0.202	-0.254	0.512	1
beta2	3.227	0.497	2.372	4.28	1.02
beta3	-0.341	0.18	-0.687	0.001	1.001
beta4	0.153	0.165	-0.16	0.479	1.005
Deviance	402.648	3.201	398.443	410.259	1

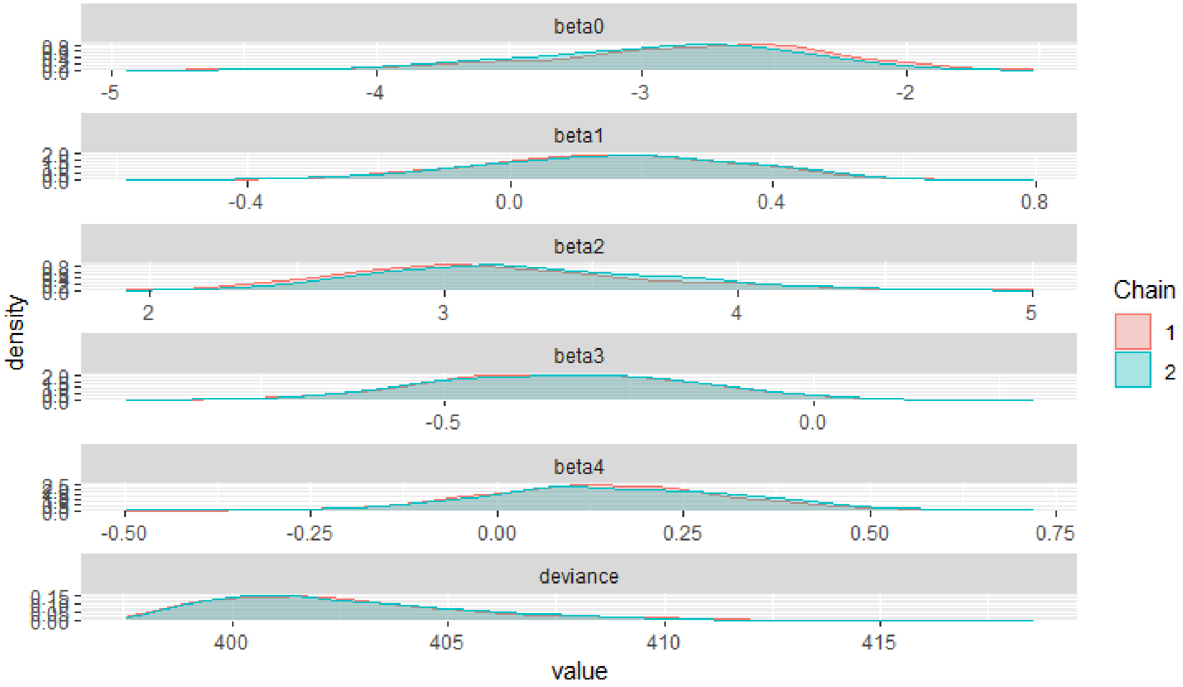
Table 9: Parameters from Bayesian Cloglog Regression

- DIC: 407.8

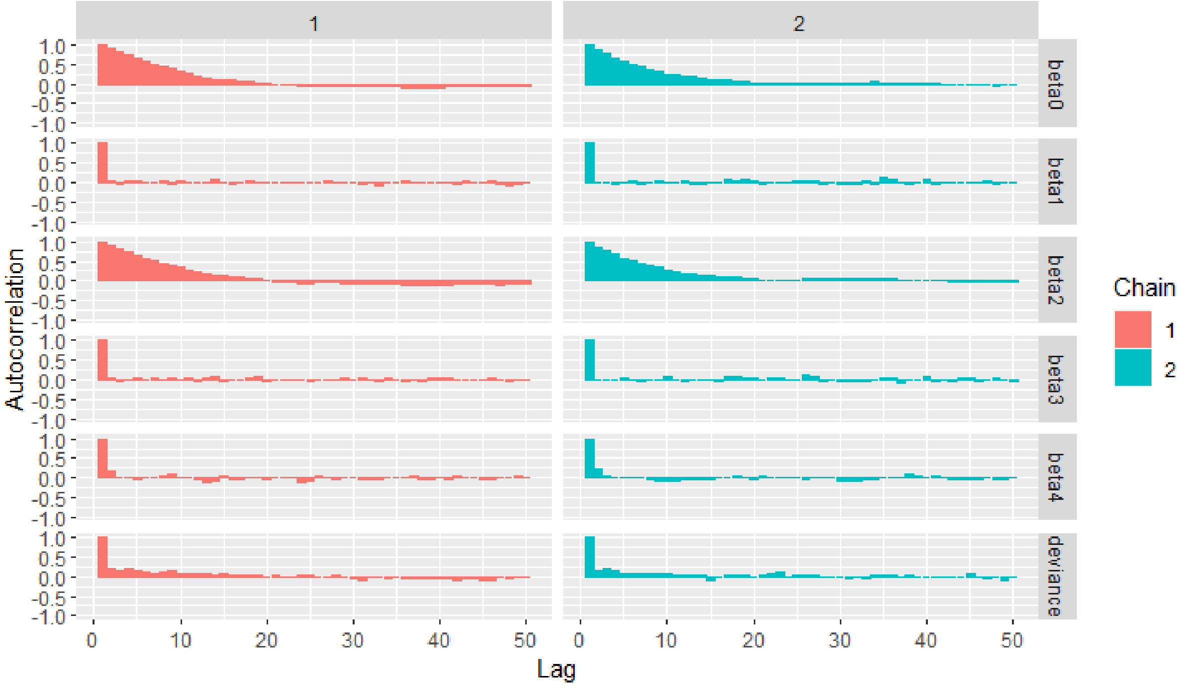
In Table 9 are reported the results of the cloglog model, and even in this case the estimations are close to the ones of the first model. The deviance information criterion is a bit lower than the logistic one.



From the traceeplots it's possible to see that The Markov Chains have steady behavior around the estimated values of the parameters.



The parameters have a prior distribution that is more or less symmetric in resepct to the mean.



The autocorrelation is high at first and then just bounces around zero for all parameters.

9 Comparing Results

In this section my aim is to compare the three models fitted in the previous pages.The first comparison that it is possible to carried out regards the Values of DIC and AIC.

Akaike's Information Criterion (AIC) is an estimator of the prediction error and thus the relative quality of statistical models for a given data set.

$AIC = 2d - 2 * Loglikelihood$

In the formula above, d is equal to the number of parameters of the model and the likelihood is the likelihood that the model could have produced my observed y-values. In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit the simplicity of the model. In other words, AIC deals with both the risk of overfitting and underfitting.

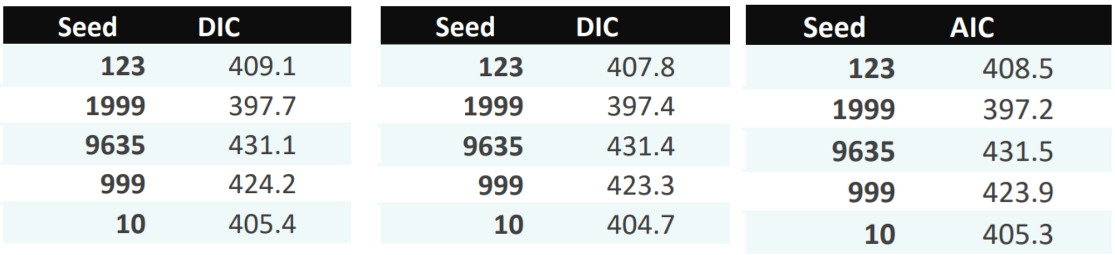
Deviance Information Criterion (DIC) is another way to compare models

$DIC = p_d + \bar{D}(\theta)$, where $p_d = \bar{D}(\theta) - D(\theta)$

$\bar{D}(\theta)$ is the posterior mean of the deviance, the latter one is equal to $-2 * Loglikelihood$. p_d, instead, regards the number of parameters in the model.

The lower these measurements are, the better the model.

Knowing that the split of the dataset into training and testing is random and governed by the chosen seed, I decided to perform the split 5 times each with a different seed. In the Table 10 are reported Values. The values, vary as the generating seed varies, even if the deviation is minimal. Moreover, the difference between the various models, holding the seed constant, is low. Therefore, the three models can be expected to act similarly.



Seed	DIC
123	409.1
1999	397.7
9635	431.1
999	424.2
10	405.4

(a) Bayesian Logistic

Seed	DIC
123	407.8
1999	397.4
9635	431.4
999	423.3
10	404.7

(b) Cloglog

Seed	AIC
123	408.5
1999	397.2
9635	431.5
999	423.9
10	405.3

(c) Frequentist Logistic

Table 10: Comparison of DIC and AIC values

I also decided to calculate four metrics to evaluate my three models:

- Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Recall: $\frac{TP}{TP+FN}$
- F1 score: $2 * \frac{Precision*Recall}{Precision+Recall}$

In my problem set the most important metric trought it evaluating the models is the Recall. This metric is very usefull when there is a high cost associated with False Negative, In fact, if a model classifies many people who will not pay the loan as eligible to get it, this will create serious risks for the bank. This metric is always higher than 90%

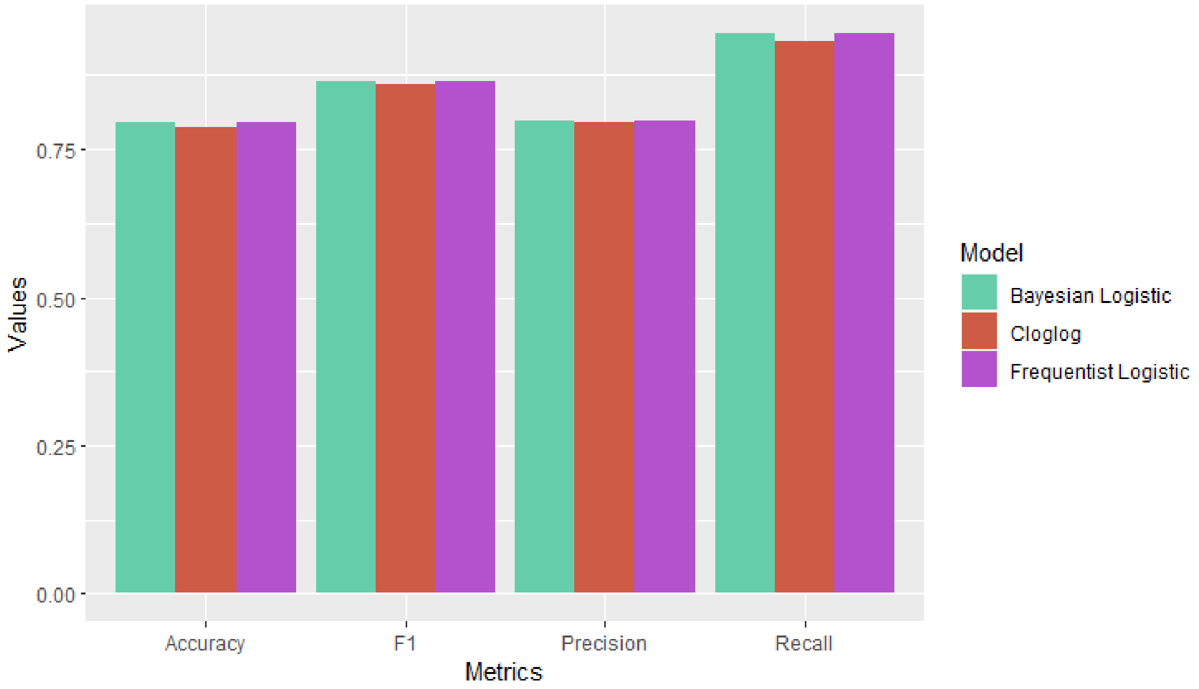


Figure 12: Camparison of metrics

As i already did for the DIC and AIC values, I want to aveluate the Recall Metric with other seeds, because as sad before this metric is the most important in my work.

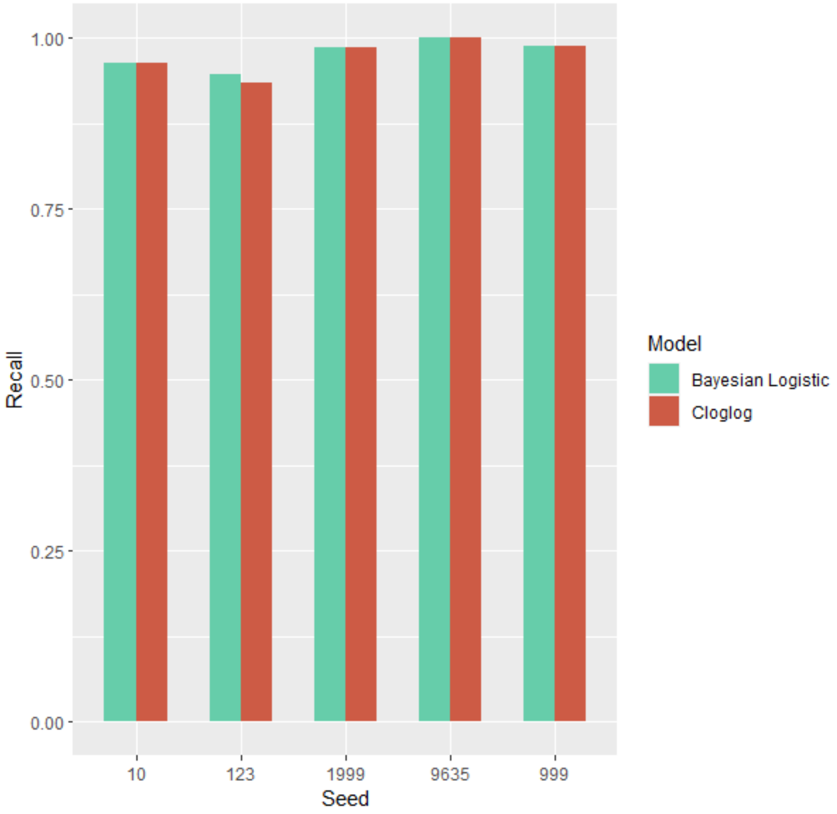


Figure 13: Recall for some seeds

From Figure 13, you can see that the value of this metric is always high even when changing the seed. Another thing that can be noticed is that, except for seed 123, the two models always have the same value: this is because recall is based on the True positives (TP) and the false negatives (FN), in my experiments these two values in the two models were always concordant, what varied were instead the False positives (FP) and the True negatives (TN).

10 Conclusion

In conclusion, I can say that the three models lead to almost the same results in predicting loan approval status, with particularly high values of the metrics, especially recall. Moreover, the Bayesian logistic model seems to recover correctly the "real" values of the parameters.

11 Further Work

I can recommend some future analyses that may be helpful:

- Create and implement a more well-stocked dataset , both with a larger number of observations

(N) and with more features and information on loan applicants.

- Include some interaction among the features.
- Make the dataset more balanced through the smote procedure and redo the analysis done in this paper.

12 References

1. https://www.kaggle.com/datasets/vikasukani/loan-eligible-dataset **(Dataset)**
2. https://towardsdatascience.com/predict-loan-eligibility-using-machine-learning-models-7a14ef904057 **(Article 1)**
3. https://www.analyticsvidhya.com/blog/2022/02/loan-approval-prediction-machine-learning/ **(Article 2)**
4. https://www.irjet.net/archives/V9/i4/IRJET-V9I4118.pdf **(Article 3)**
5. https://www.kaggle.com/code/caesarmario/loan-prediction-w-various-ml-models **(Codes from a kaggle user)**
6. http://patricklam.org/teaching/convergence_print.pdf **(Convergence Diagnostics)**
7. https://oliviergimenez.github.io/blog/sim_with_jags/ **(Parameter Recovery)**
8. https://r-charts.com/ **(R plots)**
9. https://www.rdocumentation.org **(R documentation)**