Titanic dataset classification

Carsten Stahl

2024-06-24

Using bayesian classification to predict the titanic dataset

```
head(df)
```

```
##   X PassengerId Survived Sex    Age       Fare Pclass_1 Pclass_2 Pclass_3
## 1 0           1        0   1 0.2750 0.01415106        0        0        1
## 2 1           2        1   0 0.4750 0.13913574        1        0        0
## 3 2           3        1   0 0.3250 0.01546857        0        0        1
## 4 3           4        1   0 0.4375 0.10364430        1        0        0
## 5 4           5        0   1 0.4375 0.01571255        0        0        1
## 6 5           6        0   1 0.3500 0.01650950        0        0        1
##   Family_size Title_1 Title_2 Title_3 Title_4 Emb_1 Emb_2 Emb_3
## 1         0.1       1       0       0       0     0     0     1
## 2         0.1       1       0       0       0     1     0     0
## 3         0.0       0       0       0       1     0     0     1
## 4         0.1       1       0       0       0     0     0     1
## 5         0.0       1       0       0       0     0     0     1
## 6         0.0       1       0       0       0     0     1     0
```

building X and y

First removing the id:

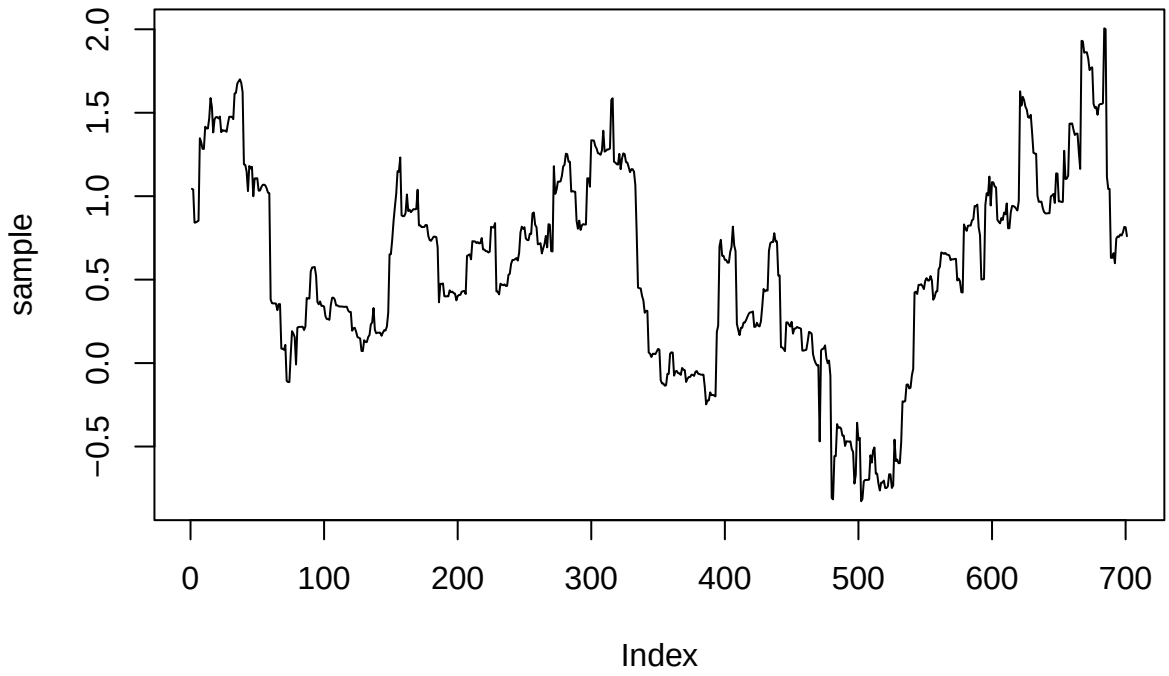
```
df <- df[,-2:-1]
# df$interc <- rep(1, length(df[,1]))

y <- df[,1]


X <- as.matrix(df[,-1])

head(X)
```

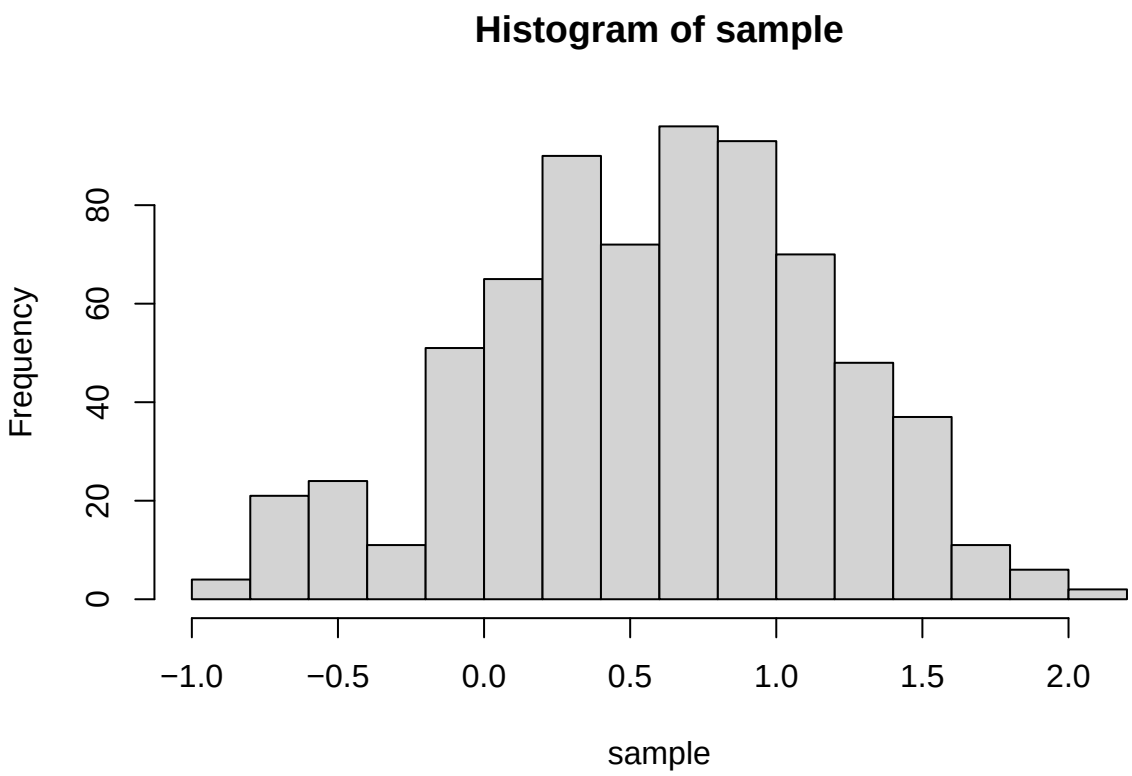
```
##      Sex    Age       Fare Pclass_1 Pclass_2 Pclass_3 Family_size Title_1
## [1,]   1 0.2750 0.01415106        0        0        1         0.1       1
## [2,]   0 0.4750 0.13913574        1        0        0         0.1       1
## [3,]   0 0.3250 0.01546857        0        0        1         0.0       0
## [4,]   0 0.4375 0.10364430        1        0        0         0.1       1
## [5,]   1 0.4375 0.01571255        0        0        1         0.0       1
```

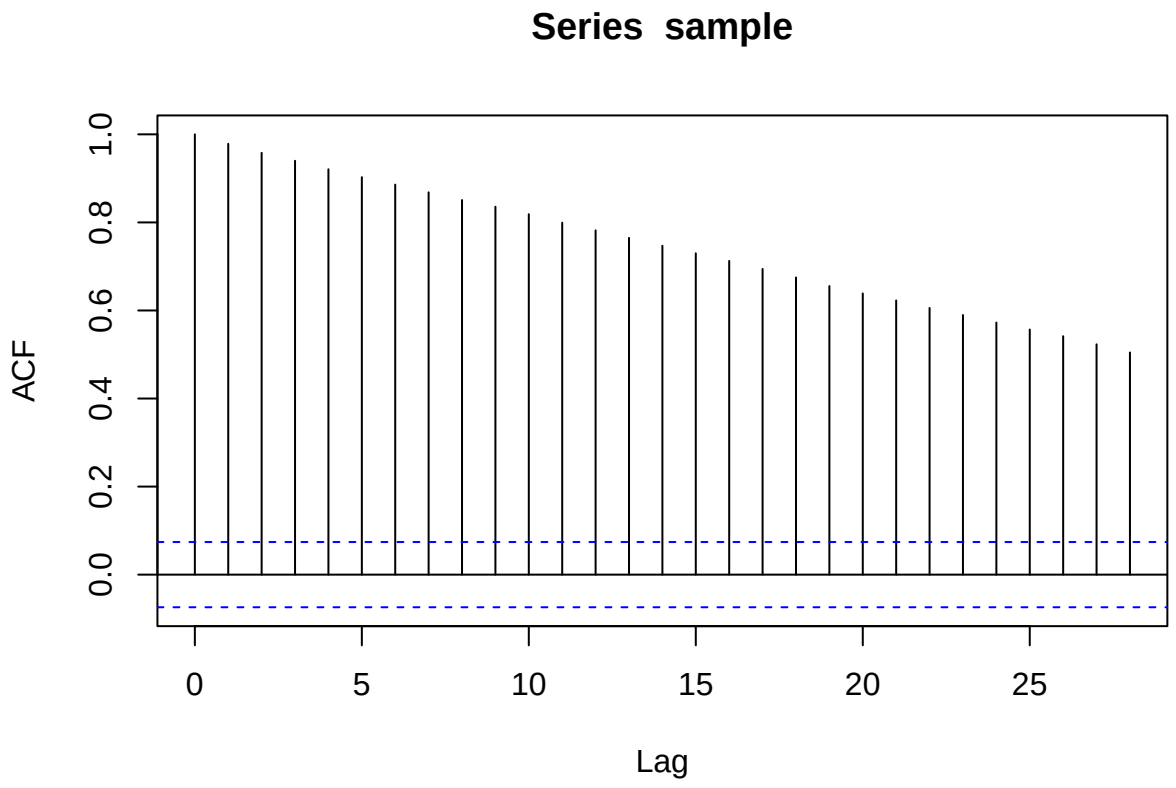
```
## [6,]   1 0.3500 0.01650950        0        0        1         0.0       1
##      Title_2 Title_3 Title_4 Emb_1 Emb_2 Emb_3
## [1,]       0       0       0     0     0     1
## [2,]       0       0       0     1     0     0
## [3,]       0       0       1     0     0     1
## [4,]       0       0       0     0     0     1
## [5,]       0       0       0     0     0     1
## [6,]       0       0       0     0     1     0
```

applying the bayesian logistic regression

```
prior <- function(beta){prod(dnorm(beta))}

beta_samples <- bayes_logit_reg_slice(1000, y, X,
                                      prior = prior,
                                      intveral_width = rep(10, length(X[1,])))

distribution_report <- function(var, burn) {
  sample <- beta_samples[burn:1000,var]
  plot(sample, type="l")
  hist(sample)
  acf(sample)
}

distribution_report(3, 300)
```


sample
2.0
1.5
1.0
0.5
0.0
−0.5
0
100
200
300
400
500
600
700
Index


Histogram of sample
Frequency
0
20
40
60
80
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
sample




```
beta_mean <- apply(beta_samples, 2, mean)


compute_accuracy <- function(beta, X, y) {
  eta <- X %*% beta
  y_hat <- expit(eta)

  y_hat_pred <- as.numeric(y_hat >= 0.5)

  accuracy = sum(y_hat_pred == y) / length(y_hat_pred)

  return(accuracy)
}

compute_accuracy(beta_mean, X, y)
```

```
## [1] 0.8156566
```

```
y_test <- df_test$Survived

X_test <- as.matrix(df_test[,-3:-1])

compute_accuracy(beta_mean, X_test, y_test)
```

```
## [1] 0.86
```